Compaq AlphaServer ES40sv Rackmount System

Installation/User/Service Guide

Part Number: EK-FCLPR-IN. A01

May 1999

Compaq Computer Corporation
Houston, Texas

May 1999

Table of Contents

Preface

1 Introduction

2 Installation

3 Operation

4 Remote System Management

5 Removal and Replacement

6 Troubleshooting and Diagnostics

A Hardware Specifications

B Field Replaceable Units (FRUs)

C Jumpers and Switches

D Connectors

Examples

Figures

Tables

Preface

Overview

This guide provides the information necessary to install the *Compaq AlphaServer* ES40sv rackmount system in a 48.26-cm (19-in.) EIA cabinet. It also provides information on operating and troubleshooting the *Compaq AlphaServer* ES40sv rackmount system, and removing and replacing field replaceable units (FRUs).

Intended Audience

This guide is intended for Compaq service personnel or qualified Self-maintenance customers who are familiar with installing computer systems, and for system managers and others who perform system management tasks.

How to Use This Guide

Read this guide *before* installing the *AlphaServer* ES40sv rackmount system.

Before installation review the warranty. The terms of the warranty agreement with Compaq may require that a qualified Compaq Customer Services representative install the system. Contact your local Compaq representative if you have any questions.

Organization

This guide is organized as follows:

Chapter 1, Introduction – Provides an overview of the *AlphaServer* ES40sv rackmount system and a description of the basic components and controls.

Chapter 2, Installation – Provides site preparation, unpacking, system installation, and option installation information.

Chapter 3, Operation – Provides information for powering up the system and verifying system operation, booting and installing the operating system, system configuration and setup, and updating firmware.

Chapter 4, Remote System Management – Provides information for using the remote management console (RMC) to monitor and operate the system remotely.

Chapter 5, Removal and Replacement – Provides removal and replacement procedures for the major system components.

Chapter 6, Troubleshooting and Diagnostics – Provides information for identifying and resolving system problems that prevent proper operation.

Appendix A, Hardware Specifications – Provides the physical, environmental, and electrical specifications for the *AlphaServer* ES40sv rackmount system.

Appendix B, Field Replaceable Units (FRUs) – Lists all of the FRUs and their part numbers for the *AlphaServer* ES40sv rackmount system chassis.

Appendix C, Jumpers and Switches – Lists all of the system motherboard jumpers and switches along with the jumper name and a description of the settings and the normal positions of the switches. Also lists the PCI backplane jumpers along with the jumper name and a description of the settings.

Appendix D, Connectors – Contains a diagram showing the location and name of the rear chassis connectors and components and a diagram showing the location and name of the PCI backplane connectors.

Related Documents

Other documents related to the *AlphaServer* ES40sv rackmount system include the following:

- *AlphaServer ES40 Owner's Guide* (EK-ES240-UG)
- *AlphaServer ES40 Service Guide* (EK-ES240-SV)
- *AlphaServer ES40 User Interface Guide* (EK-ES240-UI)
- *Data Express DE200I-SW User's Guide* (D89-0000-0041)

Conventions

This guide uses the following conventions:

Convention	Meaning
Note	A note calls the reader's attention to any item of information that may be of special importance.
Caution	A caution contains information essential to avoid damage to the equipment.
Warning	A warning contains information essential to the safety of personnel.
❶	Circled numbers provide a link between figures or examples and text.
`MONOSPACE`	Text displayed on the screen is shown in monospace type.
bold type	Bold type denotes user input or software commands.
Italic type	Italic type emphasizes important information, indicates variables, and indicates complete titles of manuals.

The following symbols appear on the chassis. Please review their definitions below:

This Dangerous Voltage warning symbol indicates risk of electrical shock and indicates hazards from dangerous voltage.

This Attention symbol is used to alert the reader about specific safety conditions, and to instruct the reader to read separate instructional material.

Reader's Comments

Compaq welcomes your comments on this or any other manual. You can send your comments to Compaq in the following ways:

- Internet electronic mail: *reader-comments@digital.com*
- Mail:

  Compaq Computer Corporation

  Information Design

  PKO3-2/21J

  129 Parker Street

  Maynard, MA 01754-2199

For additional information call 1-800-344-4825.

1
Introduction

1.1 General

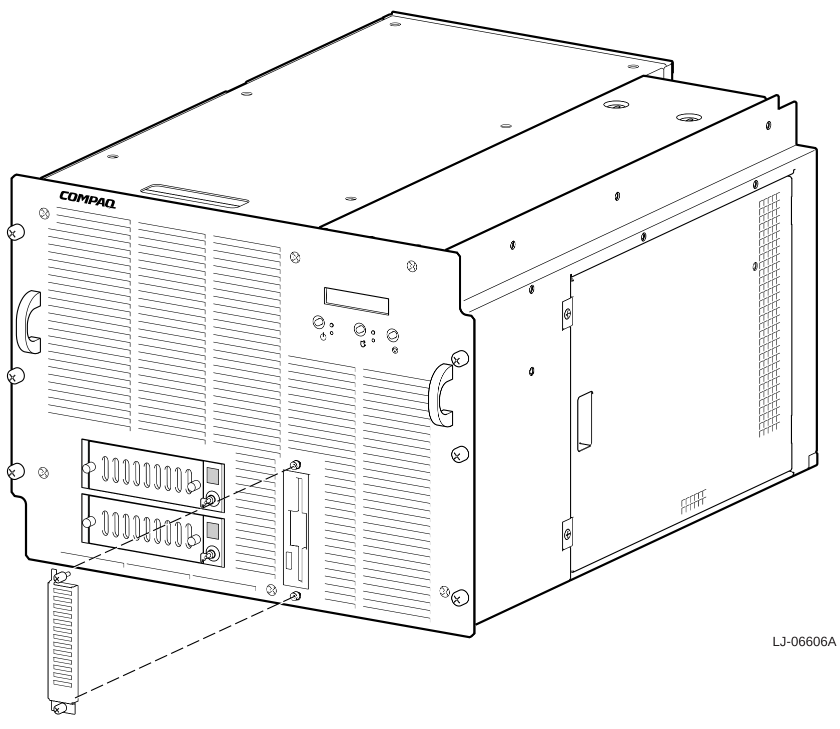
This chapter introduces the *AlphaServer* ES40sv rackmount system and describes the system components and controls.

The *AlphaServer* ES40sv rackmount system (see Figure 1-1) is a high-performance server that is contained in a slide-mounted chassis that fits into a 48.26-cm (19-in.) EIA cabinet.

The *AlphaServer* ES40sv rackmount system is part of the *AlphaServer* ES40 product line. This product line is a family of Alpha, symmetric multiprocessor, server systems that are supported by multiple operating systems (*Tru64 UNIX* and *OpenVMS*). These server systems are suitable for several computing environments: general-purpose commercial, high-performance application and database, and PC LAN server.

The server's CPU is based on the *DECchip* 21264 (EV6) processor chip.

Figure 1-1 *AlphaServer* ES40sv Rackmount System



1.2 Characteristics

Table 1-1 highlights the specific characteristics of the *AlphaServer* ES40sv rackmount system.

Table 1-1 System Characteristics

Characteristic	Description
Processor capability	Contains up to four CPU modules. Each CPU module contains a *DECchip* 21264 (EV6) processor, a two-way set associative virtual 64 KB on-chip instruction cache, and a two-way set associative virtually addressed, physically tagged, write back 64 KB on-chip data cache. Each CPU module contains a 4 MB secondary B-cache consisting of late-write synchronous SRAMs that provide low latency and high bandwidth. Each CPU module also has a 5 ->2 volt power regulator that supplies up to 100 watts at 2.2 volts to the CPU
High-performance PCI I/O subsystem	The server has two 64-bit, 33 MHz PCI busses that support ten 64-bit PCI slots. PCI bus 0 supports four PCI slots and a PCI to ISA bridge. If a graphics adapter is installed, it must be installed in one of these four PCI slots. PCI bus 1 supports six PCI slots.
PCI to ISA bridge	The ISA I/O bus is used to interface to three LM78 system hardware monitors and a Real Time Clock. There are no ISA option slots. Integrated into the PCI to ISA bridge are the: • Dual USB port • Dual-channel IDE controller to drive the CD-ROM drive • Keyboard and mouse controller • Floppy controller • Two serial port controllers • Parallel port controller
Removable mass storage	Contains two receiver frames for installing two hard disk drive carriers with 3.5 in. half-height SCSI disk drives.
Floppy drive	Contains one 1.44 MB floppy drive.
External storage devices	Supports external *StorageWorks* compatible storage devices for low-cost, high-capacity, flexible configurations. This provides the ability to have many independent disks that may be configured in stripe sets, shadow sets, or RAID sets to optimize performance according to customer requirements.
High availability	Supports disk hot swap in external *StorageWorks* shelves as well as clustering with proper software and controllers.
Main memory	Contains four memory motherboards. A memory motherboard has eight slots for installing sets of 200 pin synchronous memory DIMMs. The DIMMs are installed in sets of four. Two sets of DIMMs (eight DIMMs) constitute an array. Each array must contain a minimum set of four of the same DIMMs in order to function.

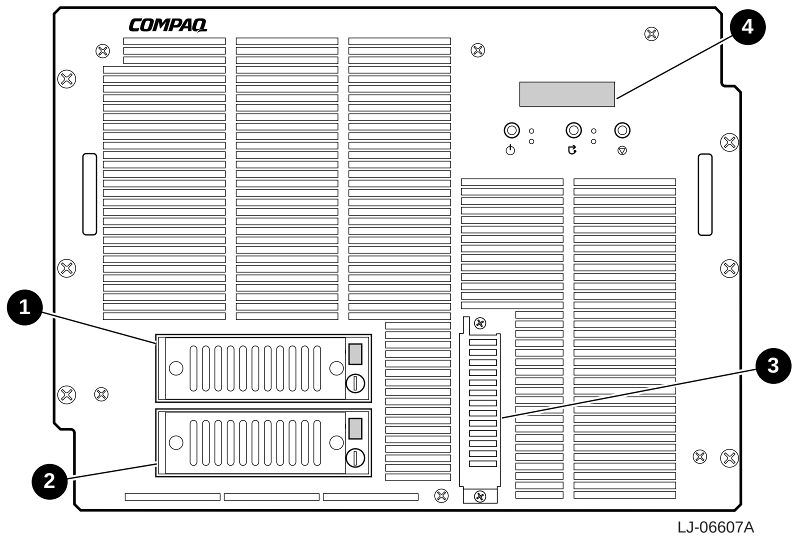
1.3 Components and Controls

System components and controls are located at the front, internally, and at the rear of the system chassis.

1.3.1 Front Components

Figure 1-2 shows the components located on the front of the system chassis.

Figure 1-2 Front Components



❶ Receiver frame for hard disk drive carrier

❷ Receiver frame for hard disk drive carrier

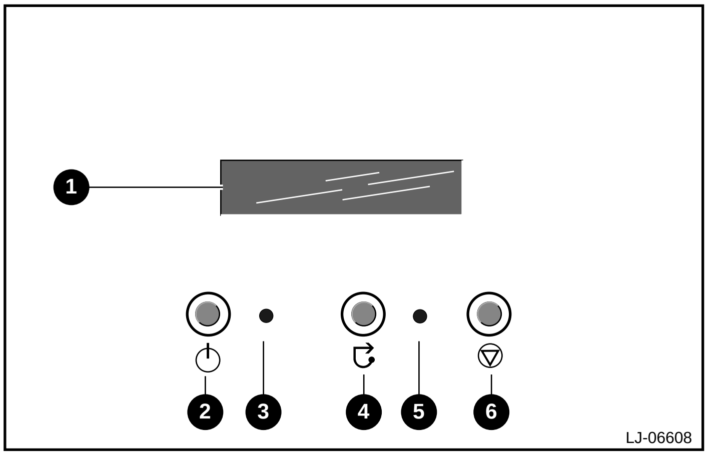
❸ Floppy drive

❹ Operator control panel

1.3.2 Operator Control Panel

The operator control panel (OCP), shown in Figure 1-3, is located on the front of the system chassis and consists of three buttons, two LEDs, and a 16-character alphanumeric display.

Figure 1-3 Operator Control Panel



❶ **Display** — A one-line, 16-character alphanumeric display that indicates status during power-up and testing. See Table 6-7 and Table 6-8 for the messages that are displayed and their meanings. While the operating system is running, the console revision is displayed as the default. This message can be changed by using the SRM console **set ocp_text** command.

❷ **Power button** — Powers the system ac power on or off. If a failure occurs that causes the system to shut down, pressing the power button off and then on clears the shutdown condition and attempts to power the system back on. Conditions that prevent the system from powering on can be determined by entering the **status** command from the remote management console (RMC). The RMC is powered separately from the rest of the system and can operate as long as ac power is present.

❸ **Power LED (green)** — Lights when the power button is depressed.

❹ **Reset button** — A momentary contact switch that restarts the system and reinitializes the console firmware. Power-up messages are displayed, and then the console prompt is displayed or the operating system boot messages are displayed, depending on how the startup sequence has been defined.

❺ **Halt LED (amber)** — Lights when the halt button is depressed.

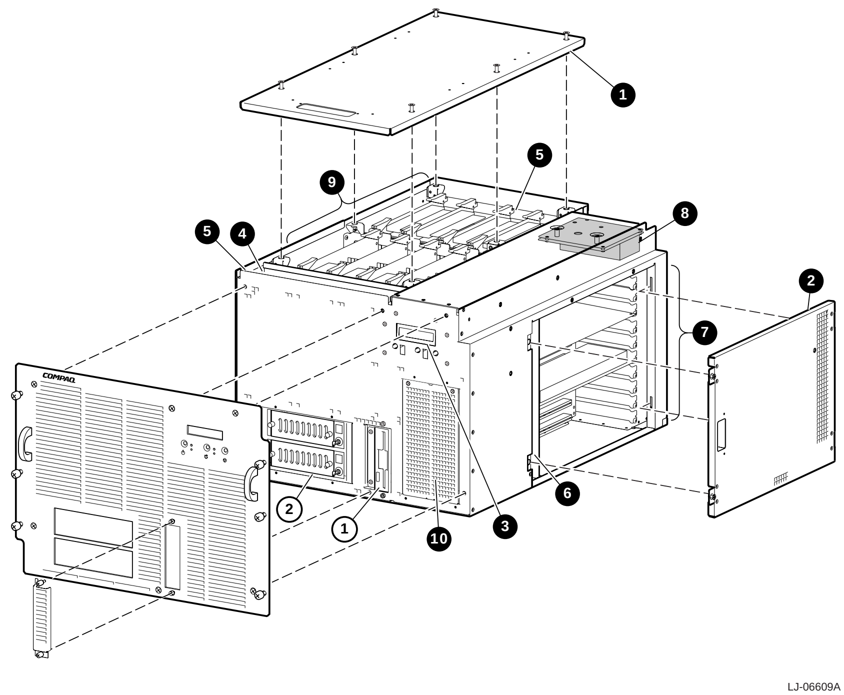
❻ **Halt button** — Pressing this button in does the following:

- If *Tru64 UNIX* or *OpenVMS* is running, it halts the operating system and returns to the SRM console.
- If the Halt button is in when the system is reset or powered up, the system halts in the SRM console, regardless of the operating system. *Tru64 UNIX* and *OpenVMS* systems that are configured for autoboot will not boot if the Halt button is in.

1.3.3 Internal Components

Figure 1-4 shows the internal components of the system chassis.

Figure 1-4 Internal Components

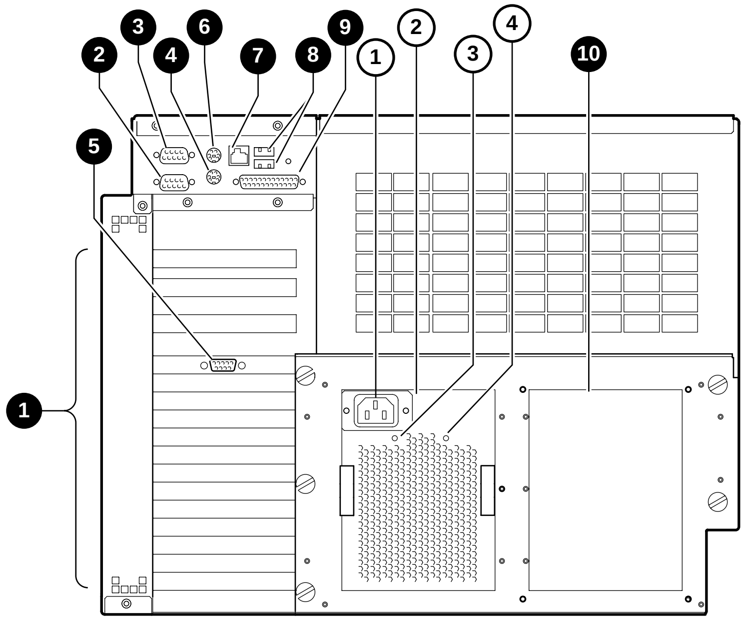


❶	Top cover	❼	PCI card cage
❷	PCI access cover	❽	I/O port module
❸	Operator control panel	❾	System card cage
❹	CPU fan assembly	❿	Hard disk cover
❺	Front and rear EMI/RFI honeycomb filter	①	Floppy drive
❻	PCI card-cage fan assembly	②	Removable media cage

1.3.4 Rear Components

Figure 1-5 shows the components located on the rear of the system chassis.

Figure 1-5 Rear Components



❶	PCI board slots	❽	USB ports
❷	Modem port	❾	Parallel port
❸	Serial port (COM2)	❿	Optional power supply cover
❹	Keyboard port	①	AC input receptacle
❺	PCI slot for VGA controller	②	Power supply
❻	Mouse port	③	Power OK LED
❼	Serial port (COM1)	④	+5V Aux LED

2
Installation

2.1 General

This chapter covers the installation of the *AlphaServer* ES40sv rackmount system in a 48.26-cm (19-in.) EIA cabinet. The major topics covered in this chapter include:

- Site Preparation
- Unpacking the Shipment
- Installing the *AlphaServer* ES40sv Rackmount System
  - Determining the Installation Area
  - Attaching the Slide Assemblies to the Cabinet Rails
  - Installing Rail Nuts on the Front Rails
  - Attaching the Inner Races to the Chassis
  - Mounting the System on the Slides
  - Connecting the Power Cord(s)
- Connecting a VGA Monitor or Serial Terminal
- Removing the Top Cover
- Removing the PCI Access Cover
- Installing Optional CPU Modules
- Installing Memory DIMMs
- Installing PCI Option Modules
- Connecting Serial and Parallel Devices
- Connecting to Networks
- Installing an Optional Power Supply
- Installing a Hard Disk Drive Carrier

2.2 Site Preparation

The installation instructions that follow assume that:

- All cables that you plan to connect to your system are in place and clearly labeled. These cables are:

  Terminal data cables

  Telephone cables

  Network cables

- The specifications and conditions listed in Appendix A have been met.
- The system is located in an area that provides sufficient clearance for ventilation and servicing. A clearance of 61 cm (24 in.) at the rear and 129.6 cm (51 in.) at the front of the cabinet is required for service.

Caution

Do not impede airflow by obstructing the front and rear of the cabinet. Exceeding internal thermal limits can affect system reliability.

WARNING

The *AlphaServer* ES40sv rackmount system can weigh up to 61.24 kg (135 lb). To prevent personal injury and equipment damage, ensure that the system is contained in an enclosure that can be stabilized when the system is pulled out on its slides.

2.3 Unpacking the Shipment

The *AlphaServer* ES40sv rackmount system shipment may include several cartons. Check the packing list to ensure that all items listed have been received.

If the equipment is damaged or if any items are missing, notify the delivery agent and contact the Compaq sales representative.

Save all shipping cartons in case the equipment needs to be moved to a new location, or needs to be returned for repair.

WARNING

The *AlphaServer* ES40sv rackmount system can weigh up to 61.24 kg (135 lb). Use sufficient personnel or the proper lifting equipment when lifting or moving the system.

2.4 Installing the *AlphaServer* ES40sv Rackmount System

The following sections contain the procedures for installing the *AlphaServer* ES40sv rackmount system in a 48.26-cm (19-in.) EIA cabinet.

Tools Required

- No. 1 and no. 2 Phillips-head screwdriver
- No. 1 short (stubby) Phillips-head screwdriver
- Flat-blade screwdriver
- Adjustable wrench

2.4.1 Determining the Installation Area

The *AlphaServer* ES40sv rackmount system requires 35.56 cm (14.00 in.) of vertical space or 24 contiguous holes in a 48.26-cm (19-in.) EIA cabinet.

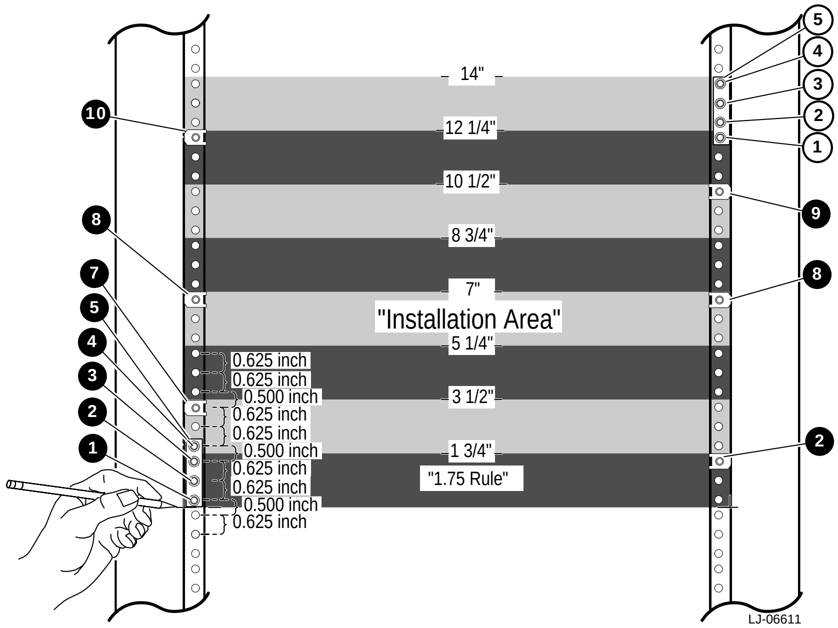
The holes in an EIA cabinet rail follow a pattern of 1.27 cm (0.50 in.), 1.59 cm (0.625 in.), and 1.59 cm (0.625 in.). This pattern is called a *set* and is repeated for the length of the cabinet rails.

To determine the installation area, perform the following steps at the front and the rear cabinet rails (refer to Figure 2-1).

1. Select a section of the cabinet rail where there is a 1.27 cm (0.50 in.) space between two holes and make a mark between these two holes. This is the *starting point* of the installation area.
2. Count up or down 24 holes from the starting point and make a mark above or below the 24th hole. The area between these two marks is the *installation area*. The bottom hole within the installation area is identified as hole 1 of the installation area.
3. Repeat steps 1 and 2 for the same holes on the front of both front rails and the rear of both rear rails.

The total installation area for the *AlphaServer* ES40sv rackmount system is or 35.56 cm (14.00 in.) of vertical space.

Figure 2-1 Installation Area and Rail Hole Pattern



- ❶ Hole 1 for attaching left slide bracket.
- ❷ Hole 2 for attaching left slide bracket.
- ❸ Hole 3 for attaching left slide bracket.
- ❹ Hole 4 for attaching left slide bracket.
- ❺ Bar Nut for attaching left slide to the front and rear rails.
- ❻ Hole 3 for rail nut used to secure system to right front rail.
- ❼ Hole 6 for rail nut used to secure system to left front rail.
- ❽ Hole 12 for rail nuts used to secure system to left and right front rails.
- ❾ Hole 18 for rail nut used to secure system to right front rail.
- ❿ Hole 21 for rail nut used to secure system to left front rail.
- ① Hole 21 for attaching right slide bracket.
- ② Hole 22 for attaching right slide bracket.
- ③ Hole 23 for attaching right slide bracket.
- ④ Hole 24 for attaching right slide bracket.
- ⑤ Bar Nut for attaching right slide to the front and rear rails.

Note

If the installation area is not started between two .5-inch holes, there is the possibility that the left and right slide assemblies can only be secured to the left and right front and rear rails with two or three screws instead of the described four screws.

2.4.2 Attaching the Slide Assemblies to the Cabinet Rails

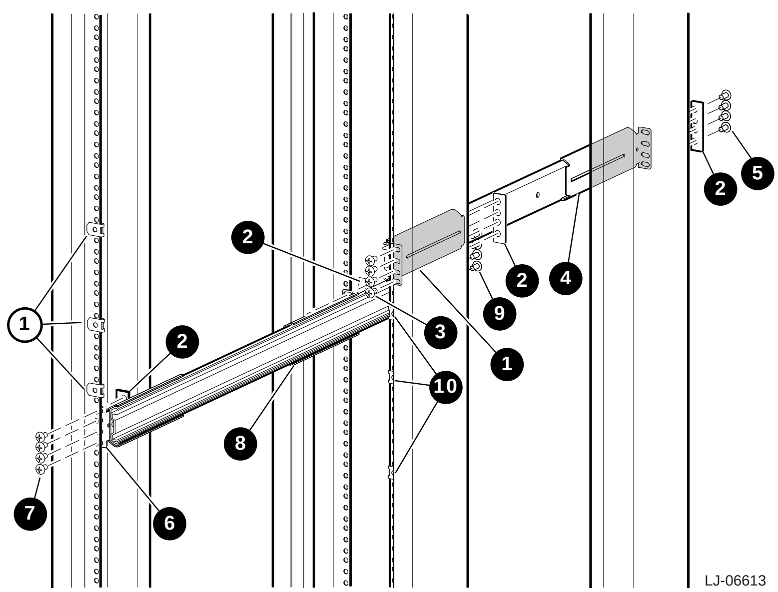
The following sections contain the procedures for attaching the right and left slide assembles to the cabinet rails.

2.4.2.1 Attaching the Right Slide Assembly to the Cabinet Rails

To attach the right slide assembly to the cabinet rails, refer to Figure 2-2 and proceed as follows:

1. Determine the proper mounting holes for the right slide brackets. The proper mounting holes are the 21st, 22nd, 23rd and 24th holes of the installation area (see Figure 2-1).
2. Locate the right slide assembly and remove the inner race by extending the inner race to the fully extended position and then pushing up on the locking lever and pulling the inner race out of the slide assembly.
3. Place the right front slide bracket ❶ on the inside of the right front rail and align four holes of the bar nut ❷ with the four slide bracket holes and with the 21st, 22nd, 23rd and 24th holes of the installation area on the right front rail.
4. Install four 10-32 screws ❸ in the 21st, 22nd, 23rd and 24th holes of the installation area to secure the right front slide bracket to the right front rail, but *do not* tighten.
5. Place the right rear slide bracket ❹ on the inside of the right rear rail and align the four holes of the bar nut ❷ with the four slide bracket holes and with the 21st, 22nd, 23rd and 24th holes of the installation area on the right rear rail.
6. Install four 10-32 screws ❺ in the 21st, 22nd, 23rd and 24th holes of the installation area to secure the right rear slide bracket to the right rear rail, but *do not* tighten.
7. Tighten the two 8-32 screws and nuts that secure the right rear slide bracket to the right slide assembly.
8. Securely tighten all eight 10-32 screws that secure the right slide assembly to the right front and rear cabinet rails.

Figure 2-2 Attaching the Slide Assemblies to the Cabinet Rails



2.4.2.2 Attaching the Left Slide Assembly to the Cabinet Rails

To attach the left slide assembly to the cabinet rails, refer to Figure 2-2 and proceed as follows:

1. Determine the proper mounting holes for the left slide brackets. The proper mounting holes are the 1st, 2nd, 3rd and 4th holes of the installation area (see Figure 2-1).
2. Locate the left slide assembly and remove the inner race by extending the inner race to the fully extended position and then pushing up on the locking lever and pulling the inner race out of the slide assembly.
3. Place the left front slide bracket ❻ on the inside of the left front rail and align four holes of the bar nut ❷ with the four slide bracket holes and with the 1st, 2nd, 3rd and 4th holes of the installation area on the left front rail.
4. Install four 10-32 screws ❼ in the 1st, 2nd, 3rd and 4th holes of the installation area to secure the left front slide bracket to the left front rail, but *do not* tighten.
5. Place the left rear slide bracket ❽ on the inside of the left rear rail and align four holes of the bar nut ❷ with the four slide bracket holes and with the 1st, 2nd, 3rd and 4th holes of the installation area on the left rear rail.
6. Install four 10-32 screws ❾ in the 1st, 2nd, 3rd and 4th holes of the installation area to secure the left rear slide bracket to the left rear rail, but *do not* tighten.
7. Tighten the two 8-32 screws and nuts that secure the left rear slide bracket to the left slide assembly.
8. Securely tighten all eight 10-32 screws that secure the left slide assembly to the left front and rear cabinet rails.

2.4.3 Installing Rail Nuts on the Front Rails

Six rail nuts must be installed on the front cabinet rails to receive the front panel captive screws that secure the system to the cabinet. Refer to Figure 2-2 and use the following procedure to install the rail nuts:

1. Install a rail nut ❿ over the 3rd, 12th and 18th holes of the installation area on the **right front** rail by sliding the rail nut over the edge of the cabinet rail and aligning it with the hole. Ensure that the threaded half of the rail nuts are toward the inside of the cabinet.
2. Install a rail nut ① over the 6th, 12th and 21st holes of the installation area on the **left front** rail by sliding the rail nut over the edge of the cabinet rail and aligning it with the hole. Ensure that the threaded half of the rail nuts are toward the inside of the cabinet.

2.4.4 Attaching the Inner Races to the Chassis

To attach the inner slide races to the chassis, refer to Figure 2-3 and proceed as follows:

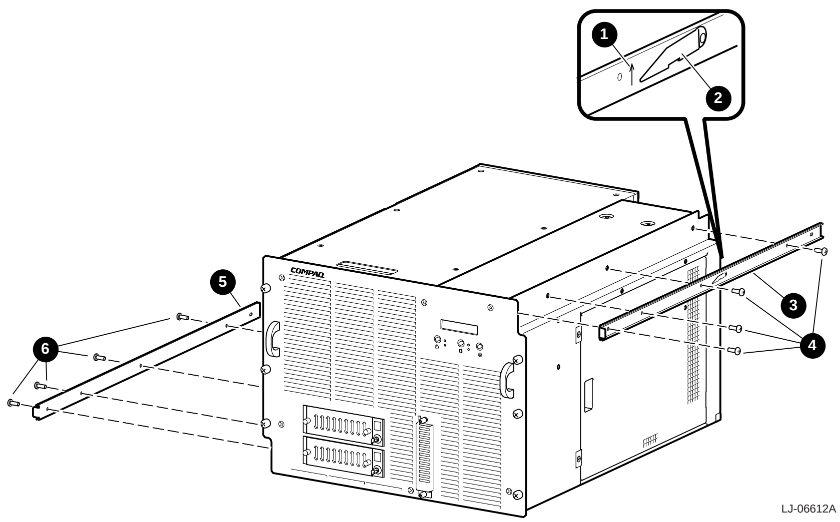
Caution

When performing this procedure, ensure that the arrow ❶ (shown in Figure 2-3) points upward and the locking lever ❷ points toward the front of the system. Otherwise, the slide will be damaged when the system is installed on the slide assemblies.

Attach the right inner slide race ❸ to the upper right side of the system chassis (as viewed from the front) using four 8-32 pan-head screws ❹.

Attach the left inner slide race ❺ to the lower left side of the system chassis (as viewed from the front) using four 8-32 pan-head screws ❻.

Figure 2-3 Attaching the Inner Slide Races



2.4.5 Mounting the System on the Slides

To mount the system on the slides, refer to Figure 2-4 and proceed as follows:

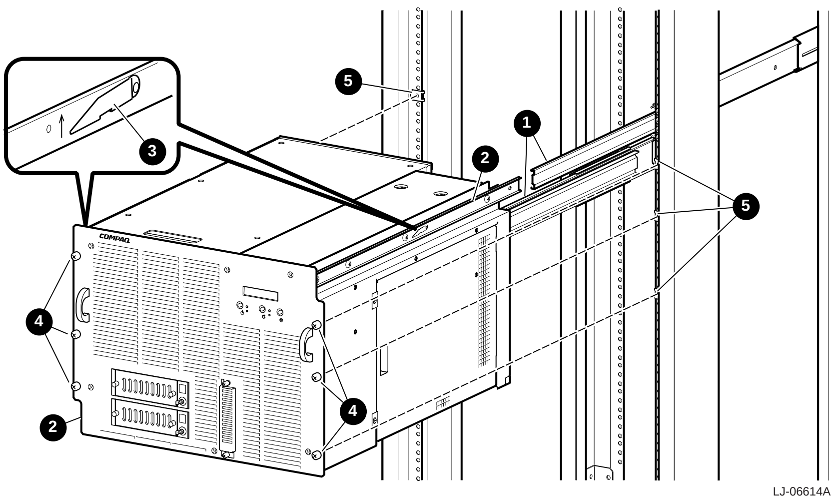
WARNING

The *AlphaServer* ES40sv rackmount system can weigh up to 61.24 kg (135 lb). Use sufficient personnel or the proper lifting equipment when lifting or moving the system.

Stabilize the cabinet before installing the system into the cabinet.

1. Pull both equipment slides ❶ out fully to their locked positions.
2. Lift the chassis and position it so that the slide races ❷ fit into the front end of the slides.
3. Push the system into the slides until it stops. Push up on the two locking levers ❸ and then push the system into the cabinet.
4. Secure the system to the front cabinet rails using the six 10-32 captive screws ❹ on the front panel. The captive screws go into the rail nuts ❺ previously installed on the cabinet rails.

Figure 2-4 Mounting the System on the Slides



2.4.6 Connecting the Power Cord(s)

To install the power cord(s), refer to Figure 2-5 and proceed as follows:

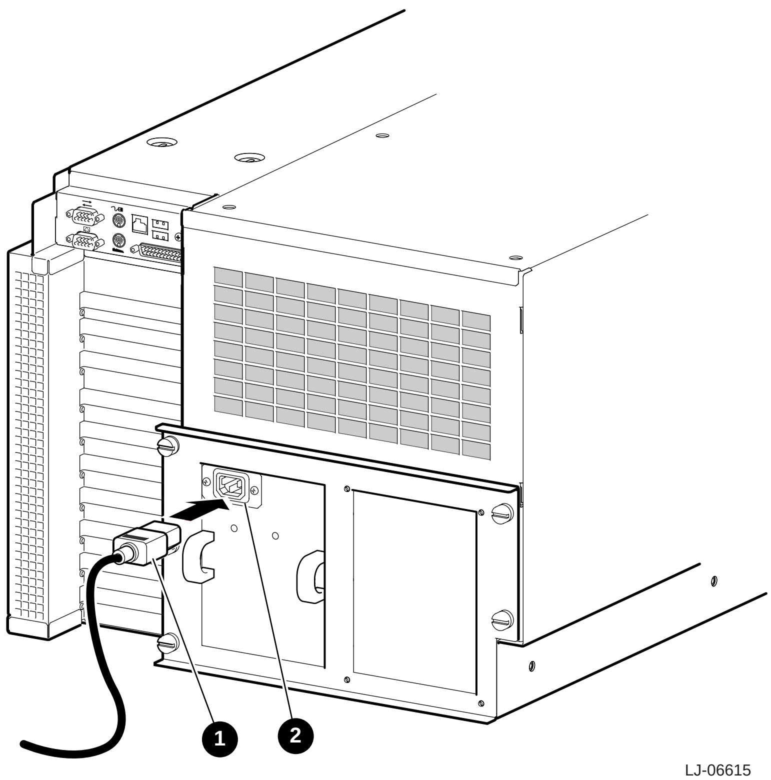
1. Attach the power cord ❶ to the power supply ac input receptacle ❷ located on the rear of the chassis.

Note

At this time, leave the other end of the power cord disconnected from the cabinet power distribution unit.

2. If the system has a second power supply installed, repeat step 1 to connect the power cord for the second power supply.

Figure 2-5 Connecting the Power Cord(s)



2.5 Connecting a VGA Monitor or Serial Terminal

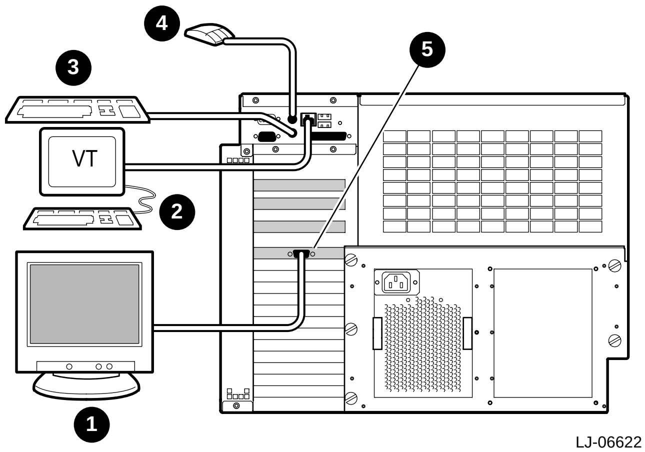
Either an optional VGA monitor or a Compaq VT-series terminal (VT*xxx*) can be connected to the system as shown in Figure 2-6. If a VGA monitor is connected, a VGA adapter must first be installed in a PCI bus 0 slot (slots 1 through 4).

Caution

Before connecting a terminal to the system, turn off the ac power to the system with the power On/Off button on the OCP.

For information about connecting a specific terminal to your system, refer to the documentation for that terminal.

Figure 2-6 Terminal Connections



❶ VGA monitor
❷ Serial terminal
❸ Keyboard
❹ Mouse
❺ VGA adapter

2.6 Removing the Top Cover

Caution

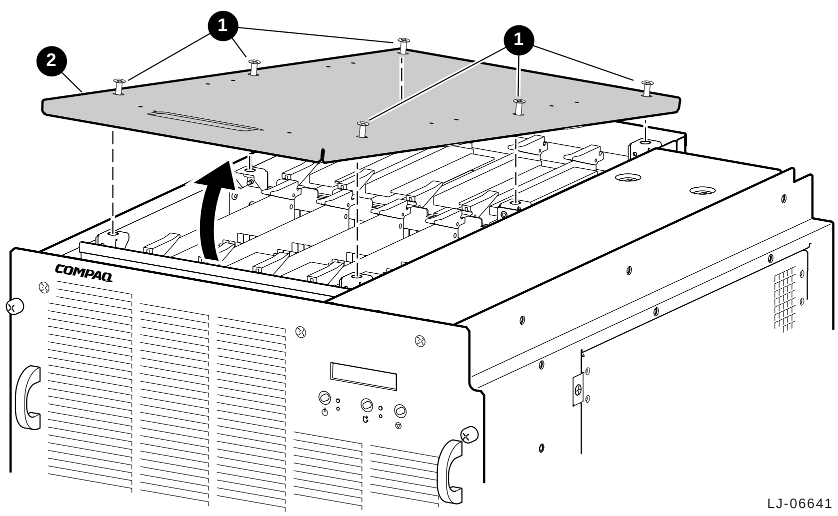
Use care when removing and replacing the cover to prevent damage to the RFI gaskets that are located around the cover and opening.

Perform the following procedure to remove the top cover:

1. Extend the chassis for service (see Section 5.2).
2. Loosen the six captive screws ❶ on the top of the top cover ❷ (see Figure 2-7).
3. Grasp the recessed hand hold along the front edge of the cover and lift the top cover off of the chassis.

To replace the top cover, refer to Figure 2-7 and reverse steps 1 through 3 of the removal procedure.

Figure 2-7 Removing and Replacing the Top Cover



2.7 Removing the PCI Access Cover

Caution

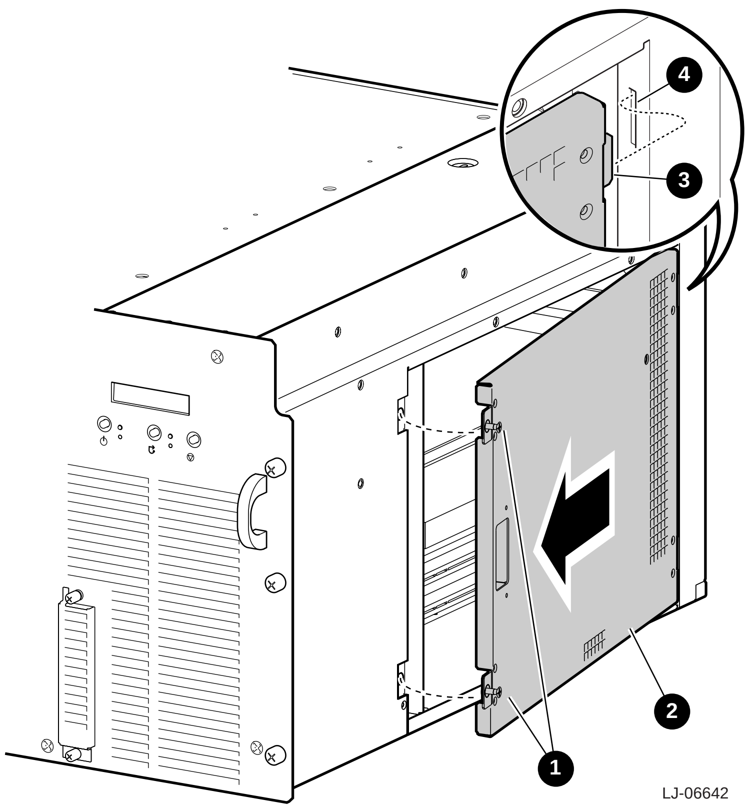
Use care when removing and replacing the cover to prevent damage to the RFI gaskets that are located around the four sides of the cover.

Perform the following procedure to remove the PCI access cover:

1. Extend the chassis for service (see Section 5.2).
2. Loosen the two captive screws ❶ along the front edge of the PCI access cover ❷ (see Figure 2-8).
3. Swing the front of the PCI access cover out from the chassis (see Figure 2-8).
4. Pull the PCI access cover forward until the tabs ❸ along the rear edge come out of the slots ❹ in the chassis, and then pull the PCI access cover out and away from the chassis.

To replace the PCI access cover, refer to Figure 2-8and reverse steps 1 through 4 of the removal procedure.

Figure 2-8 Removing and Replacing the PCI Access Cover

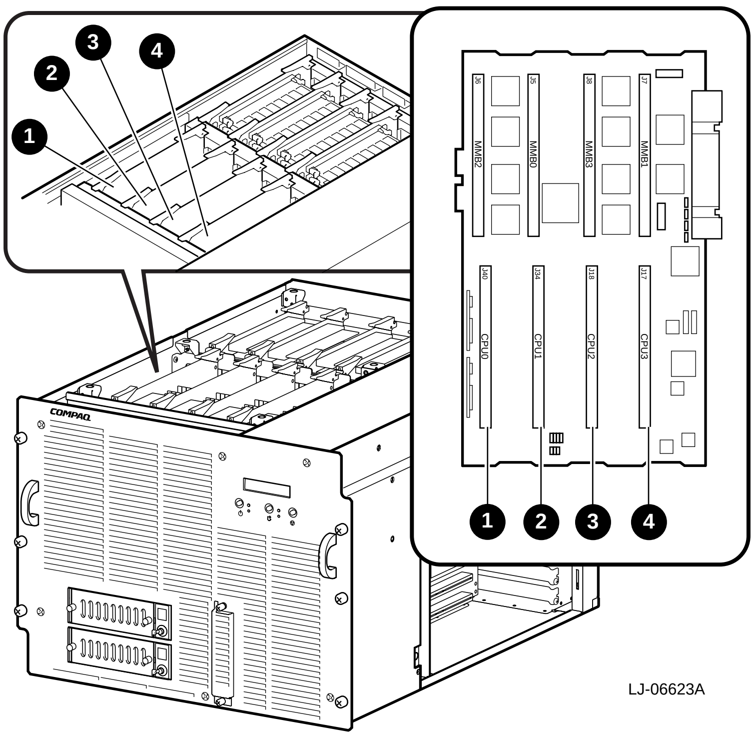


2.8 Installing Optional CPU Modules

The *AlphaServer* ES40sv rackmount system can contain up to four CPU modules. Figure 2-9 shows the CPU slot locations on the system motherboard. The following configuration rules apply to the CPU modules:

- The first CPU module must be in CPU slot 0. CPU slot 0 is designated as the primary CPU.
- Additional CPU modules should be installed in ascending order by slot number.
- All CPU modules must have the same clock frequency.

Figure 2-9 CPU Slot Layout on the System Motherboard



❶ CPU slot 0
❷ CPU slot 1
❸ CPU slot 2
❹ CPU slot 3

Perform the following procedure to install an optional CPU module:

1. Perform an orderly shutdown of the operating system.
2. Ensure that the power On/Off button on the OCP is in the Off position.
3. Extend the chassis for service (see Section 5.2).
4. Disconnect the power cord(s) from the power supply ac input receptacle(s) on the rear of the chassis.
5. Remove the top cover (see Section 2.6).

6. Locate the next available slot for installing a CPU module. CPU modules should be installed in ascending order by slot number (see Figure 2-9 for CPU slot locations).
7. Put on an antistatic wriststrap.

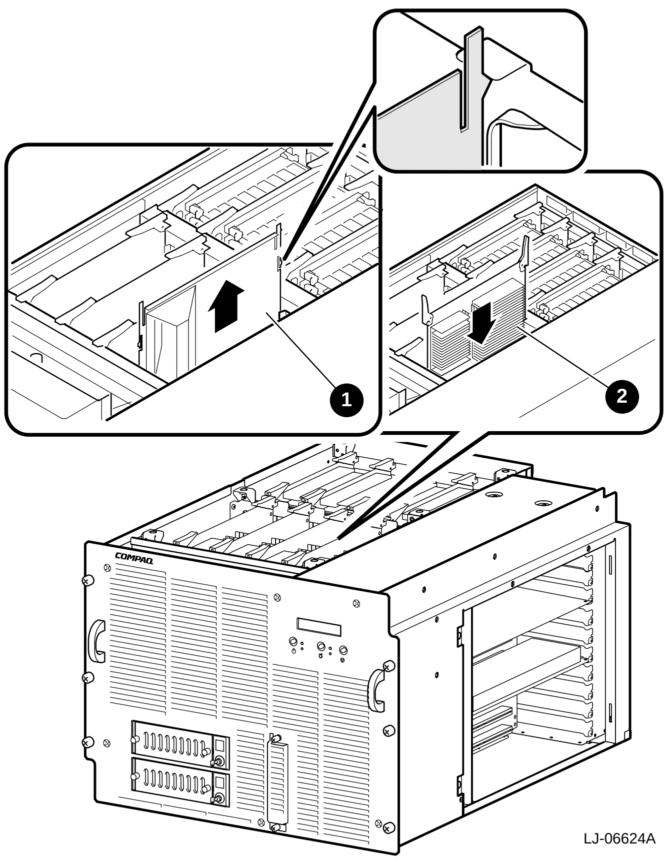
Caution

An antistatic wriststrap *must* be worn when handling any module to prevent damage to the module.

8. Remove and discard the airflow deflector plate ❶ from the selected CPU slot (see Figure 2-10).
9. Hold the CPU module to be installed by the module levers at each end of the module and align the CPU module with the chosen slot (see Figure 2-10).
10. Slide the CPU ❷ module down into the slot until it stops.
11. Push down on the two module levers to lock the CPU module into the slot.

After installing the CPU module, reverse steps 1 through 5 and reboot the system. To verify the installation, observe the screen display during power-up to see if the newly installed CPU module appears in the display. Issue the **show config** command from the SRM console to display the status of the new CPU module.

Figure 2-10 Installing Optional CPU Modules



2.9 Installing Memory DIMMs

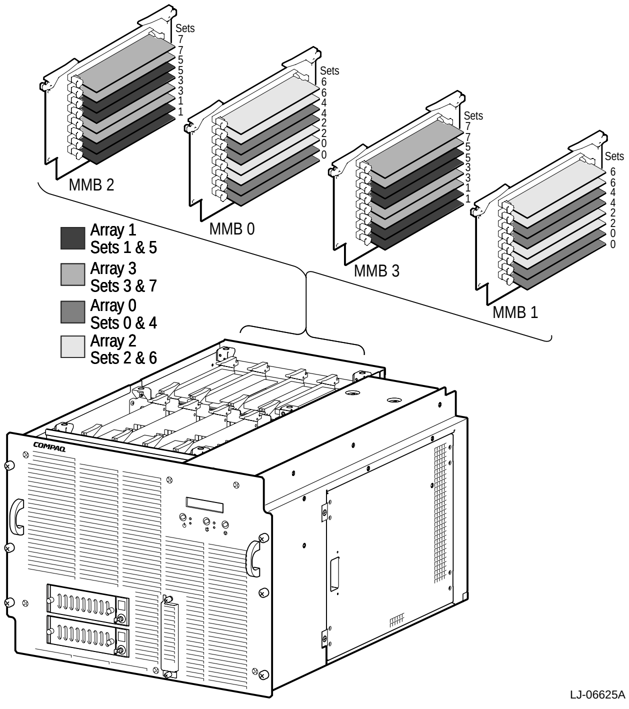
The *AlphaServer* ES40sv rackmount system contains four memory motherboards for installing memory DIMMs. Each memory motherboard has eight slots for installing 200-pin DIMMs. A memory set consists of four DIMMs. The system can be populated with a maximum of eight memory sets, numbered 0 through 7. Figure 2-11 shows the memory motherboard slot locations on the system motherboard and the memory set locations on the memory motherboards. The following configuration rules apply to the memory sets:

- DIMMs are installed in sets of four.
- All DIMMs in a set must be the same size.
- Two sets constitute an array. All DIMMs within an array must be the same size. For example, if Set 0 consists of 64 MB DIMMs, Set 4 must also consist of 64 MB DIMMs. The arrays are as follows:

Array 0	Set 0 and Set 4
Array 1	Set 1 and Set 5
Array 2	Set 2 and Set 6
Array 3	Set 3 and Set 7

- DIMM sets must be installed in numerical order. Set 0 is installed first, then Set 1, Set 2, Set 3, and so forth.

Figure 2-11 Memory Configuration



Perform the following procedure to install memory DIMMs:

1. Perform an orderly shutdown of the operating system.
2. Ensure that the power On/Off button on the OCP is in the Off position.

3. Extend the chassis for service (see Section 5.2).
4. Disconnect the power cord(s) from the power supply ac input receptacle(s) on the rear of the chassis.
5. Remove the top cover (see Section 2.6).
6. Determine where the set of memory DIMMs should be installed (see Figure 2-11).
7. Put on an antistatic wriststrap.

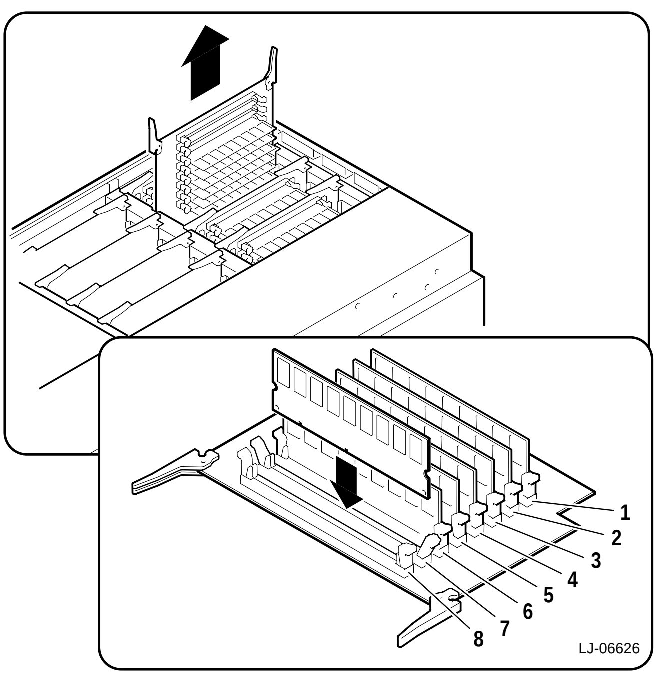
Caution

An antistatic wriststrap *must* be worn when handling any module to prevent damage to the module.

8. Remove the two memory motherboards on which the set of memory DIMMs are to be installed (see Figure 2-12).
9. Open the locking levers on the two appropriate DIMM slots on each of the memory motherboards (see Figure 2-12).
10. Align each DIMM with the appropriate slot and push firmly down until the DIMM is properly seated and the locking levers have closed.
11. After installing the four DIMMs in the memory set, reinstall the two memory motherboards (see Section 5.15).

After installing both memory motherboards, reverse steps 1 through 5 and reboot the system. To verify the installation, observe the screen display during power-up to see if the amount of memory in each array correctly appears in the display. Issue the **show memory** command from the SRM console to display the total amount of memory in the system.

Figure 2-12 Installing Memory DIMMs



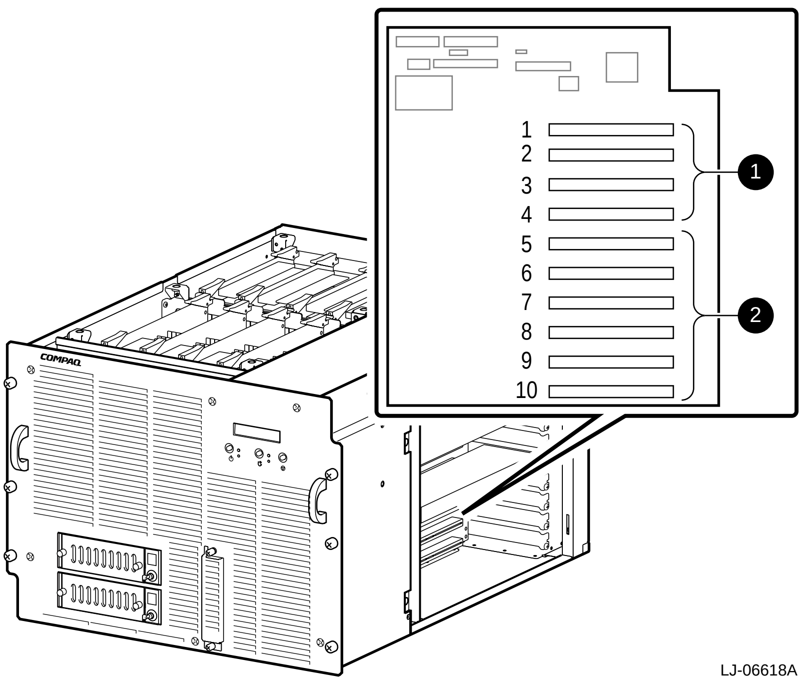
2.10 Installing PCI Option Modules

Refer to Figure 2-13 for the layout of the PCI slots on the PCI backplane. The PCI backplane has ten slots that are dedicated to PCI options.

WARNING

For protection against fire, only modules with current limited output should be used.

Figure 2-13 PCI Backplane Layout



❶ PCI bus 0 ❷ PCI bus 1

Perform the following procedure to install a PCI option module:

1. Perform an orderly shutdown of the operating system.
2. Ensure that the power On/Off button on the OCP is in the Off position.
3. Extend the chassis for service (see Section 5.2).
4. Disconnect the power cord(s) from the ac input receptacle(s) on the rear of the chassis.
5. Remove the PCI access cover (see Section 2.7).
6. Remove the filler panel screw and the blank filler panel ❶ from the slot selected for installing the PCI option module (see Figure 2-14). Save the filler panel screw for securing the option module in place.
7. Put on an antistatic wriststrap.

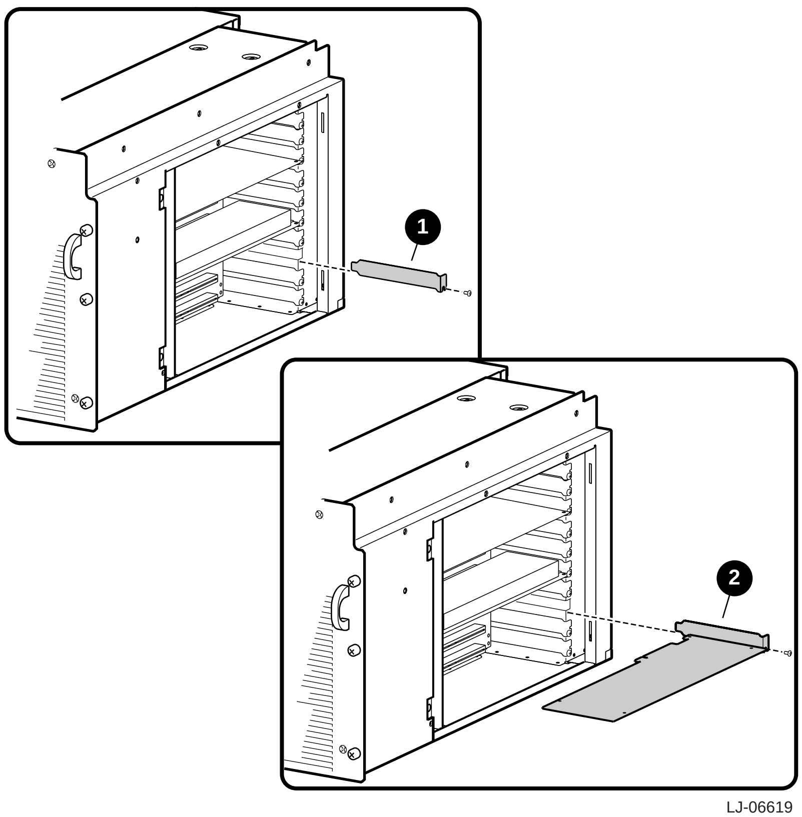
Caution

An antistatic wriststrap *must* be worn when handling any module to prevent damage to the module.

8. Slide the option module ❷ into the selected slot and apply firm pressure until the module is firmly seated (see Figure 2-14). Secure the module in place with the filler panel screw that was removed in step 6.

After installing the option module, reverse steps 1 through 5 and reboot the system. To verify the installation, issue the **show config** and **show device** commands from the SRM console.

Figure 2-14 Installing PCI Option Modules



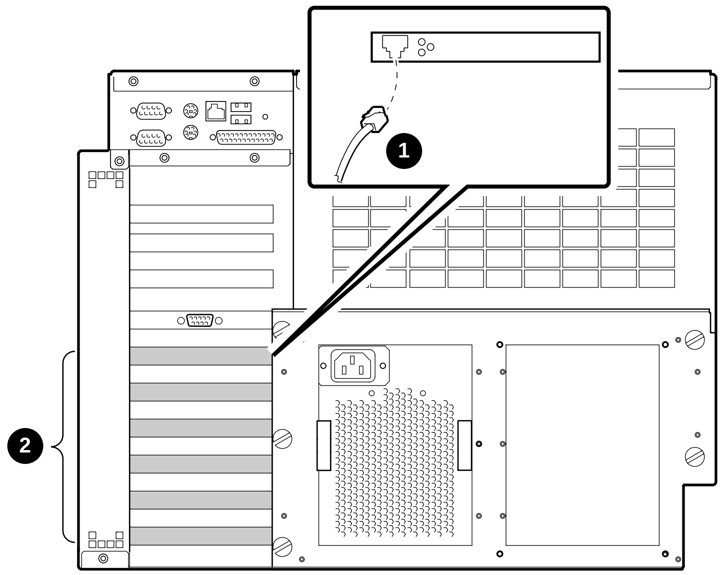
2.11 Connecting to Networks

The system can support Ethernet or other network options by using network adapters that can be connected to PCI bus 1 (slots 5 through 10).

For information about connecting your system to networks other than Ethernet, refer to the documentation that came with the network adapter.

The system can be connected to a 10BASE-T Ethernet network as shown in Figure 2-15.

Figure 2-15 Ethernet Network Connections



❶ 10BASE-T Ethernet connector ❷ Slots 5 through 10

2.12 Connecting Serial and Parallel Devices

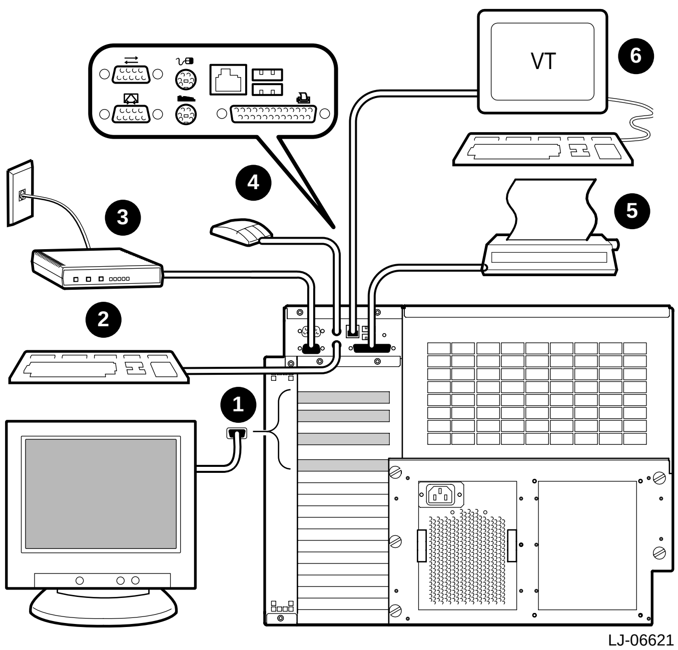
Connect a serial or parallel printer, modem, or console terminal to your system through the serial and parallel ports at the rear of the system (see Figure 2-16).

Caution

Before connecting serial or parallel devices to the system, turn off the ac power to the system with the power On/Off button on the OCP.

For information about connecting a specific device to your system, refer to the documentation for that device.

Figure 2-16 Connecting Serial and Parallel Devices



❶ VGA monitor [the VGA adapter must be installed on PCI bus 0 (slots1 through 4)]

❷ Keyboard

❸ Modem

❹ Mouse

❺ Printer connected to parallel port

❻ Console terminal connected to serial port (COM1)

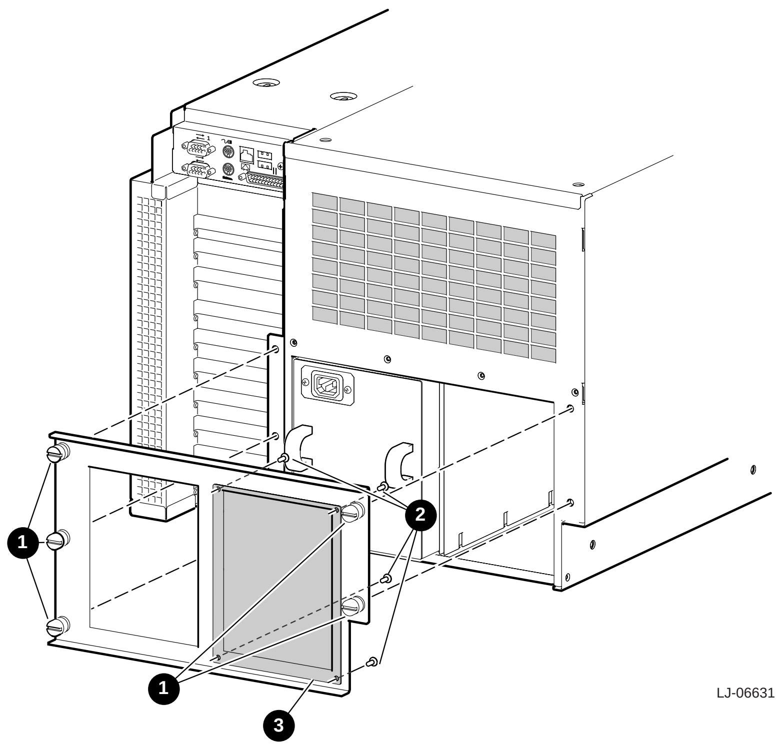
2.13 Installing an Optional Power Supply

If the *AlphaServer* ES40sv rackmount contains one power supply, a second optional power supply can be installed in the system to provide redundant operation. The system does not need to be powered down to install the second power supply.

Perform the following procedure to install an optional power supply:

1. Extend the chassis for service (see Section 5.2).
2. Loosen the five captive screws ❶ that secure the power supply bracket to the rear of the chassis and remove the bracket (see Figure 2-17).
3. Remove the four screws ❷ that secure the blank power supply cover plate ❸ to the inside of the power supply bracket and remove the cover plate (see Figure 2-17).

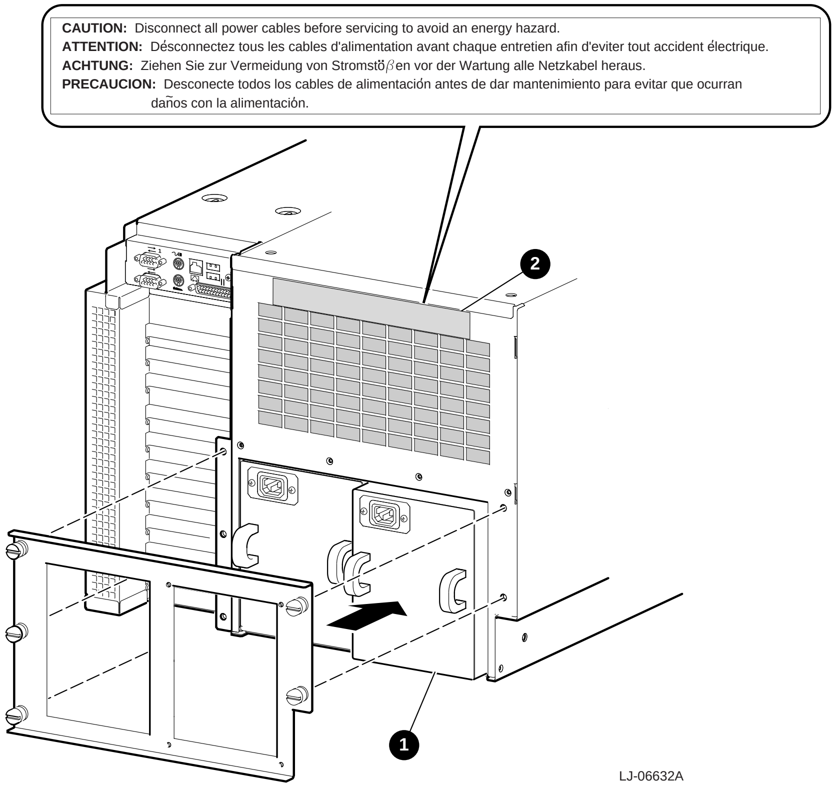
Figure 2-17 Removing Power Supply Bracket and Cover Plate



4. Slide the optional power supply ❶ into the chassis and apply firm pressure until it is seated properly (see Figure 2-18).
5. Install the multi power cable disconnect warning label ❷ on the chassis above the rear EMI/RFI honeycomb filter grill (see Figure 2-18).
6. Connect one end of the power cord to the ac input receptacle on the new power supply and the other end to the cabinet power distribution unit.

After installing the new power supply, reverse steps 1 and 2. To verify the installation, issue the **show power** command from the SRM console to display the status of the new power supply. Check the two LEDs on the new power supply to ensure that both are lit when the system is powered on.

Figure 2-18 Installing the New Power Supply



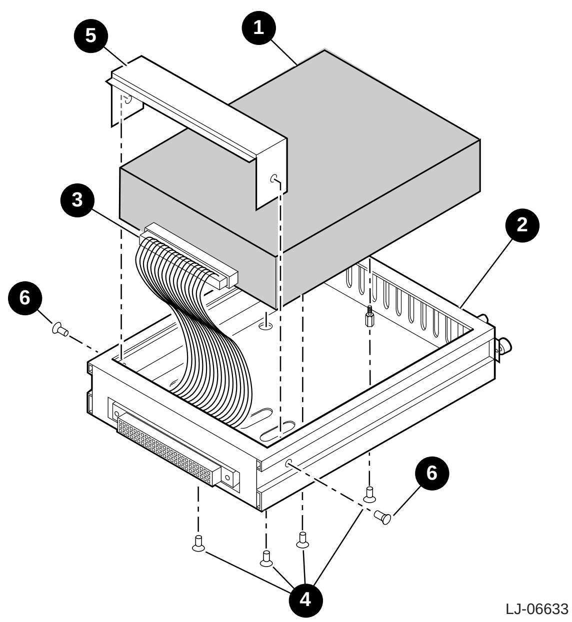
2.14 Installing a Hard Disk Drive Carrier

Receiver frames for installing carrier trays containing hard drives are installed in the *AlphaServer* ES40sv rackmount system removable media cage. The bottom receiver frame has the ID set to 0. The top receiver frame has the ID set to 1 and termination enabled.

Refer to the *Data Express DE200I-SW User's Guide* that is shipped with the carrier and perform the following procedure to install a hard disk drive carrier in the receiver frames:

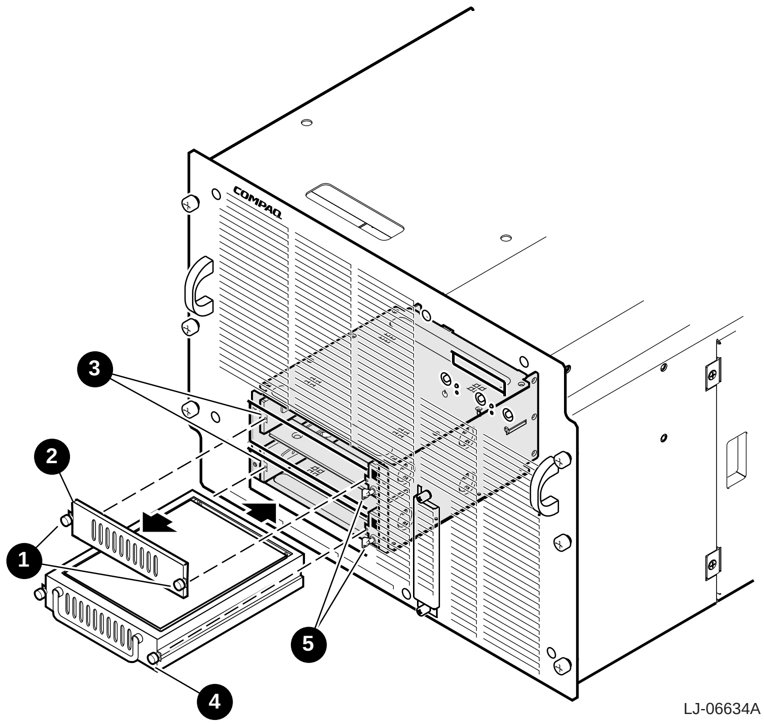
1. Perform an orderly shutdown of the operating system.
2. Ensure that the power On/Off button on the OCP is in the Off position.
3. Install a hard disk drive ❶ in the hard disk drive carrier ❷ by connecting the 80-pin ribbon cable ❸ at the rear of the carrier to the rear of the hard disk drive (see Figure 2-19).
4. Secure the hard disk drive to the carrier with four screws ❹ through the bottom of the carrier (see Figure 2-19).
5. Secure the cable cover ❺ to the carrier with two screws ❻ (see Figure 2-19).

Figure 2-19 Installing a Hard Disk Drive in a Carrier



6. Loosen the two screws ❶ that secure the cover plate ❷ to the front of the receiver frame ❸ and remove the cover plate (see Figure 2-20).
7. Slide the hard disk drive carrier ❹ into the receiver frame ❸ and push it in until it is properly seated (see Figure 2-20).
8. Secure the carrier to the receiver frame with the two captive screws (see Figure 2-20).
9. Turn the key ❺ on the front of the receiver frame counterclockwise to lock the carrier to the receiver frame (see Figure 2-20). This creates a secure hard disk drive and prevents unauthorized personnel from removing and accessing the data on the drive.
10. Reverse steps 1 and 2 and reboot the system.

Figure 2-20 Installing a Hard Disk Drive Carrier in a Receiver Frame



3
Operation

3.1 Introduction

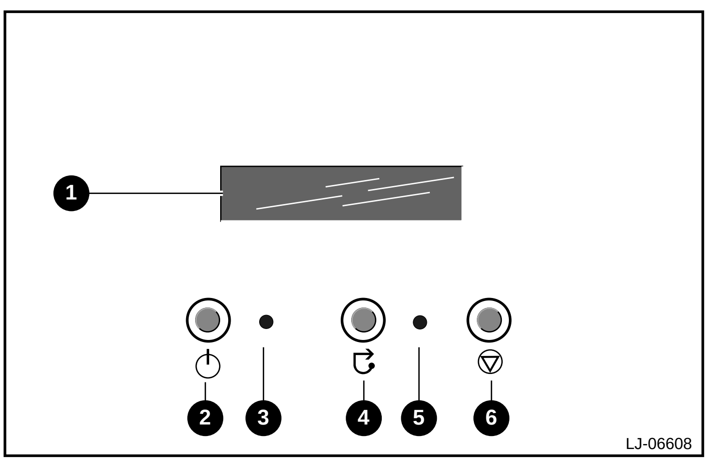
This chapter provides information for powering up the system and verifying system operation, system configuration and setup, booting and installing the operating system, and updating firmware.

3.2 Powering Up the System

Power up the system by pressing the On/Off power button on the operator control panel (see Figure 3-1). Testing begins, and status messages are displayed on the console terminal screen and on the OCP display.

Errors encountered during power-up are displayed on the operator control panel. For information about error messages on the operator control panel, see Chapter 6.

Figure 3-1 Operator Control Panel



❶	Display	❹	Reset button
❷	Power button	❺	Halt LED (amber)
❸	Power LED (green)	❻	Halt button

3.3 Power-Up Displays

Power-up information is displayed on the OCP and the system's console terminal. If the SRM **console** environment variable is set to **serial**, the entire power-up display, consisting of the SROM and SRM power-up messages, is printed on the VT terminal screen. If **console** is set to **graphics**, no SROM messages are displayed, and the SRM messages are delayed until VGA initialization has been completed.

3.3.1 SROM Power-Up Display

SROM code is executed first. Example 3-1 shows the SROM power-up messages and corresponding operator control panel messages. Example 3-2 shows the messages that are displayed once the SROM has transferred control to the SRM console. For a list of SROM power-up status , see Table 3-1.

Example 3-1 Sample SROM Power-Up Display

SROM Power-Up Display		OCP Message
	❶	PCI Test
SROM V1.00 CPU #00 @ 0500 MHz		
SROM program starting	❷	Power on
Reloading SROM		
SROM V1.00-F CPU # 00 @ 0500 MHz		
SROM program starting		
Starting secondary on CPU #1	❸	RelCPU
Starting secondary on CPU #2		
Starting secondary on CPU #3		
Bcache data tests in progress	❹	BC Data
Bcache address test in progress		
CPU parity and ECC detection in progress		
Bcache ECC data tests in progress		
Bcache TAG lines tests in progress		
Memory sizing in progress	❺	Size Mem
Memory configuration in progress		
Memory data test in progress		
Memory address test in progress		
Memory pattern test in progress		
Memory thrashing test in progress		
Memory initialization		
Loading console		Load ROM
Code execution complete (transfer control)	❻	Jump to Console

❶ When the system is powered up, the serial SROM code is loaded into the I-cache on the first available CPU, which becomes the primary CPU. The order of precedence is CPU0, CPU1, and so on. The primary CPU attempts to access the PCI bus. If it cannot, either a hang or a failure occurs and this is the only message displayed.

❷ The primary CPU interrogates the I^2C EEROM on the system motherboard and CPU modules through shared RAM. The primary CPU determines the CPU and system configuration to jump to.

The primary CPU next checks the SROM checksum to determine the validity of the Flash SROM sectors.

If Flash SROM is invalid, the primary CPU reports the error and continues the execution of the serial SROM code. The invalid Flash SROM must be reprogrammed.

If Flash SROM is good, the primary CPU programs the appropriate registers with the values from the Flash data and selects itself as a target CPU to be loaded.

❸ The primary CPU (usually CPU0) initializes and tests the B-cache and memory, then loads the flash SROM code to the next CPU. That CPU then initializes the EV6 (21264 chip) and marks itself as the secondary CPU. Once the primary CPU sees the secondary, it loads the flash SROM code to the next CPU until all remaining CPUs are loaded.

❹ The Flash SROM performs various B-cache tests. For example, the ECC data test verifies the detection logic for single- and double-bit errors.

❺ The primary CPU initiates all memory tests. The memory is tested for address and data errors for the first 32 MB of memory. It also initializes all of the "sized" memory in the system.

If a memory failure occurs, an error is reported. An untested memory array is assigned to address 0 and the failed memory array is de-assigned. The memory tests are re-run on the first 32 MB of memory. If all memory fails, the "No Memory Available" message is reported and the system halts.

❻ If all memory passes, the primary CPU loads the SRM console and transfers control to it.

Table 3-1 SROM Power-Up Status Messages

Status Code	Full SROM Message	Corresponding OCP Message
DF	SROM program starting	Power on
DE	Init and test PCI bus	PCI Test
DD	Bcache data tests in progress	BC Data
DC	Bcache address test in progress	BC Addr
DB	CPU parity and ECC error detection in progress	Par/ECC
DA	Bcache ECC data tests in progress	BC ECC
D8	Bcache TAG lines tests in progress	BC Tag
D7	Reloading SROM	Reload
D5	Memory sizing in progress	Size Mem
D4	Memory configuration in progress	Cfg Mem
D3	Memory data test in progress	Mem Data
D2	Memory address test in progress	Mem Addr
D1	Memory pattern test in progress	Mem Patt
D0	Memory thrashing test in progress	Mem thra
CF	Memory initialization	Mem Init
CD	Loading console	Load ROM
CB	Code execution complete (transfer control)	Jump to Console.
C5	Loading program from floppy	Load Flp
93	Starting secondary on CPU #3	RelCPU 3
92	Starting secondary on CPU #2	RelCPU 2
91	Starting secondary on CPU #1	RelCPU 1
90	Starting secondary on CPU #0	RelCPU 0

3.3.2 SRM Console Power-Up Display

At the completion of SROM power-up, the primary CPU transfers control to the SRM console program. The console program continues the system initialization. Failures are reported to the console terminal through the power-up screen and a console event log.

Example 3-2 SRM Power-Up Display

```
OpenVMS PALcode V1.50-0, Tru64 UNIX PALcode V1.45-5

starting console on CPU 0                                 ❶
initialized idle PCB
initializing semaphores
initializing heap
initial heap 200c0
memory low limit = 144000
heap = 200c0, 17fc0
initializing driver structures
initializing idle process PID
initializing file system
initializing hardware
initializing timer data structures
lowering IPL
CPU 0 speed is 2.00 ns (500MHz)
create dead_eater
create poll
create timer
create powerup
access NVRAM
Memory size 2048 MB
testing memory                                            ❷
...
probe I/O subsystem                                       ❸
probing hose 1, PCI
bus 0, slot 2, function 0 -- pka -- NCR 53C896
bus 0, slot 2, function 1 -- pkb -- NCR 53C896
bus 0, slot 4 -- ewa -- DE500-AA Network Controller
probing hose 0, PCI
probing PCI-to-ISA bridge, bus 1
bus 0, slot 2 -- vga -- DEC PowerStorm
bus 0, slot 15 -- dqa -- Acer Labs M1543C IDE
bus 0, slot 15 -- dqb -- Acer Labs M1543C IDE
starting drivers                                          ❹
entering idle loop
initializing keyboard

starting console on CPU 1                                 ❺
initialized idle PCB
initializing idle process PID
lowering IPL
CPU 1 speed is 2.00 ns (500MHz)
create powerup
entering idle loop
starting console on CPU 2
initialized idle PCB
initializing idle process PID
lowering IPL
CPU 2 speed is 2.00 ns (500MHz)
```

```
create powerup
entering idle loop
starting console on CPU 3
initialized idle PCB
initializing idle process PID
lowering IPL
CPU 3 speed is 2.00 ns (500MHz)
create powerup
Memory Testing and Configuration Status                ❻
  Array       Size       Base Address
---------  ----------  ----------------
    0        256Mb     0000000060000000
    1        512Mb     0000000040000000
    2        256Mb     0000000070000000
    3       1024Mb     0000000000000000

     2048 MB of System Memory
Testing the System
Testing the Disks (read only)
Testing the Network
initializing GCT/FRU at offset 192000
AlphaServer ES40 Console V5.4-5528, built on Mar 5 1999 at 01:42:13❼
P00>>>
```

❶ The primary CPU prints a message indicating that it is running the console. Starting with this message, the power-up display is printed to any console terminal, regardless of the state of the **console** environment variable.

If console is set to **graphics**, the display from this point to the end is saved in a memory buffer and printed to the VGA monitor after the PCI buses are sized and the VGA device is initialized.

❷ The memory size is determined and memory is tested.

❸ The I/O subsystem is probed and I/O devices are reported. I/O adapters are configured.

❹ Device drivers are started.

❺ The console is started on the secondary CPUs. The example shows a four-processor system.

❻ Various diagnostics are performed.

❼ Systems running *Tru64 UNIX* or *OpenVMS* display the SRM console banner and the prompt, P*nn*>>>. The number *n* indicates the primary processor. In a multiprocessor system, the prompt could be P00>>>, P01>>>, P02>>>, or P03>>>. From the SRM prompt, you can boot the *Tru64 UNIX* or *OpenVMS* operating system.

3.4 Configuring the Hardware

Possible configuration tasks include selecting the console and display device; setting environment variables; verifying the system configuration; and changing power-up or boot values.

3.4.1 Selecting the Console and Display Device

The SRM **os_type** environment variable determines which user interface (SRM or AlphaBIOS) is the final console loaded on a power-up or reset. The SRM **console** environment variable determines to which display device (VT-type terminal or VGA monitor) the console display is sent.

Selecting the Console

The **os_type** variable selects the console. **Os_type** is factory configured as follows:

- For Windows NT, **os_type** is set to **nt**.
- For *Tru64 UNIX* or *OpenVMS*, **os_type** is set to **unix** or **vms**, respectively.

If **os_type** is set to **unix** or **vms**, the SRM console is loaded on a power-up or reset. If **os_type** is set to **nt**, the SRM console is loaded and then SRM starts the AlphaBIOS console from system flash ROM.

Selecting the Display Device

The console terminal that displays the SRM user interface can be either a serial terminal (VT320 or higher, or equivalent) or a VGA monitor.

The SRM **console** environment variable determines the display device.

- If **console** is set to **serial**, and a VT-type device is connected, the SRM console powers on in serial mode and prints power-up information to the VT device. The VT device can be connected to the MMJ port or to COM2.
- If **console** is set to **graphics**, the SRM console expects to find a VGA card connected to PCI 0 and, if so, displays power-up information on the VGA monitor after VGA initialization has been completed.

You can verify the display device with the SRM **show console** command and change the display device with the SRM **set console** command. If you change the display device setting, you must reset the system either with the Reset button or the **init** command to put the new setting into effect.

In the following example, the user displays the current console device and then resets it to a serial device:

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

3.4.2 Setting the Control Panel Message

If you are running *Tru64 UNIX* or *OpenVMS*, you can create a customized message to be displayed on the operator control panel after startup self-tests and diagnostics have been completed.

When the operating system is running, the control panel displays the console revision. It is useful to create a customized message if you have a number of systems and you want to identify each system by a node name.

Use the SRM **set ocp_text** command to change this message. The message can be up to 16 characters and must be entered in quotation marks.

Example 3-3 Set ocp_text Command

```
P00>>> set ocp_text “Node Alpha1”
```

3.4.3 Displaying the Hardware Configuration

View the system hardware configuration for *Tru64 UNIX* and *OpenVMS* systems from the SRM console. It is useful to view the hardware configuration to ensure that the system recognizes all devices, memory configuration, and network connections.

Displaying a *Tru64 UNIX* or *OpenVMS* Configuration

Use the following SRM console commands to view system configuration for *Tru64 UNIX* or *OpenVMS* systems. Additional commands to view the system configuration are described in the Compaq *AlphaServer ES40 User Interface Guide*.

show boot*	Displays the boot environment variables.
show config	Displays the logical configuration of interconnects and buses on the system and the devices found on them.
show device	Displays the bootable devices and controllers in the system.
show fru	Displays the physical configuration of FRUs (field-replaceable units). See Chapter 6 for information on this command.
show memory	Displays configuration of main memory.

Example 3-4 show boot *

```
P00>>>show boot*
boot_dev          dka0.0.0.1.1
boot_file
boot_osflags      a
boot_reset        OFF
bootdef_dev       dka0.0.0.1.1
booted_dev
booted_file
booted_osflags
```

Example 3-5 show config

```
P00>>>show config
                        Compaq Computer Corporation
                          Compaq AlphaServer ES40

Firmware                                                                  ❶
SRM Console:  V5.4-5528
ARC Console:  5.68
PALcode:      OpenVMS PALcode V1.50-0, Tru64 UNIX PALcode V1.47-5
Serial Rom:   V1.5-F
RMC Rom:      V1.0
RMC Flash Rom:V1.2

Processors                                                                ❷
CPU 0         Alpha 21264-4 500 MHz  4MB Bcache
CPU 1         Alpha 21264-4 500 MHz  4MB Bcache
CPU 2         Alpha 21264-4 500 MHz  4MB Bcache
CPU 3         Alpha 21264-4 500 MHz  4MB Bcache

Core Logic                                                                ❸
Cchip         DECchip 21272-CA Rev 9(C4)
Dchip         DECchip 21272-DA Rev 2
Pchip 0       DECchip 21272-EA Rev 2
Pchip 1       DECchip 21272-EA Rev 2
TIG           Rev 10

Memory                                                                    ❹
  Array       Size       Base Address
---------  ----------  ----------------
    0        256Mb     0000000060000000
    1        512Mb     0000000040000000
    2        256Mb     0000000070000000
    3       1024Mb     0000000000000000

     2048 MB of System Memory

 Slot  Option              Hose 0, Bus 0, PCI                            ❺
   2/0 NCR 53C896          pke0.7.0.2.0          SCSI Bus ID 7
   2/1 NCR 53C896          pkf0.7.0.102.0        SCSI Bus ID 7
   4   DEC PowerStorm
   7   Acer Labs M1543C                    Bridge to Bus 1, ISA
  15   Acer Labs M1543C IDE   dqa.0.0.15.0
                              dqb.0.1.15.0
                              dqa0.0.0.15.0    TOSHIBA CD-ROM XM-6302B
  19   Acer Labs M1543C USB

       Option              Hose 0, Bus 1, ISA
       Floppy              dva0.0.0.1000.0

 Slot  Option              Hose 1, Bus 0, PCI
   1   NCR 53C895          pka0.7.0.1.1        SCSI Bus ID 7
                           dka0.0.0.1.1        RZ2DD-LS
                           dka100.1.0.1.1      RZ2DD-LS
                           dka200.2.0.1.1      RZ1CB-CS
   3   NCR 53C810          pkb0.7.0.3.1        SCSI Bus ID 7

                           dkb0.0.0.3.1        RZ25
   4   DE500-BA Network Con ewa0.0.0.4.1       00-00-F8-09-90-FF
   6   DECchip 21152-AA                           Bridge to Bus 2, PCI

Slot   Option              Hose 1, Bus 2, PCI
   0   NCR 53C875          pkc0.7.0.2000.1          SCSI Bus ID 7
```

```
   1   NCR 53C875          pkd0.7.0.2001.1            SCSI Bus ID 7
   2   DE500-AA Network Con ewb0.0.0.2002.1           00-06-2B-00-25-5B
P00>>>
```

The elements of the display are as follows.

❶ **Firmware**. Version numbers of the SRM console, AlphaBIOS (ARC) console, PALcode, serial ROM, RMC ROM and RMC flash ROM

❷ **Processors**. Processors present and processor version and clock speed and amount of backup cache

❸ **Core logic**. Version numbers of the chips that form the interconnect on the system motherboard.

❹ **Memory**. Memory arrays and memory size.

❺ PCI bus information.

The "Slots" column lists logical slots, not the physical slots into which devices are installed. See Table 3-2 for the correspondence between the logical slots and the physical slots.

The NCR 53C896 on Hose 0, Bus 0 is a dual-channel Ultra2 SCSI multifunction controller. Two controllers reside on the same chip. They are shown as 2/0 and 2/1. The first number is the logical slot, and the second is the function.

The Acer Labs bridge chip, which is located in PCI logical slot 7, has two built-in IDE controllers. The CD-ROM is on the first controller.

NOTE: *The* naming of devices (for example,dqa.0.0.15.0) follows the conventions described in Table 3-3.

In Example 3-5, the following devices are present:

Hose 0, Bus 0, PCI

Slot 2/0	SCSI controller
Slot 2/1	SCSI controller
Slot 4	VGA controller
Slot 7	PCI to ISA bridge chip
Slot 15	IDE controller and CD-ROM drive
Slot 19	Universal serial bus (USB) controller

Hose 0, Bus 1, ISA

Diskette drive

Hose 1, Bus 0, PCI

Slot 1	SCSI controller and drives
Slot 3	SCSI controller and drives
Slot 4	Ethernet controller
Slot 6	PCI to PCI bridge chip to Bus 2

Hose 1, Bus 2, PCI

Slot 0	SCSI controller
Slot 1	SCSI controller
Slot 2	Ethernet controller

Table 3-2 PCI Slot Mapping

Physical Slot	Logical Slot	PCI 0
1	1	Device
2	2	Device
3	3	Device
4	4	Device

Physical Slot	Logical Slot	PCI 1
5	1	Device
6	2	Device
7	3	Device
8	4	Device
9	5	Device
10	6	Device

Example 3-6 show device

```
P00>>>show device
dka0.0.0.1.1          DKA0                         RZ2DD-LS  0306
dka100.1.0.1.1        DKA100                       RZ2DD-LS  0306
dka200.2.0.1.1        DKA200                       RZ1CB-CS  0844
dkb0.0.0.3.1          DKB0                             RZ25  0900
dqa0.0.0.15.0         DQA0      TOSHIBA CD-ROM XM-6302B  1012
dva0.0.0.1000.0       DVA0
ewa0.0.0.4.1          EWA0            00-00-F8-09-90-FF
ewb0.0.0.2002.1       EWB0            00-06-2B-00-25-5B
pka0.7.0.1.1          PKA0                SCSI Bus ID 7
pkb0.7.0.3.1          PKB0                SCSI Bus ID 7
pkc0.7.0.2000.1       PKC0                SCSI Bus ID 7
pkd0.7.0.2001.1       PKD0                SCSI Bus ID 7
```

Table 3-3 Device Naming Conventions

	Category	Description			
dq	Driver ID	Two-letter designator of port or class driver			
		dk	SCSI disk or CD	ew	Ethernet port
		dq	IDE CD-ROM	fw	FDDI device
		dr	RAID set device	mk	SCSI tape
		du	DSSI disk	mu	DSSI tape
		dv	Diskette drive	pk	SCSI port
		ei	Ethernet port	pu	DSSI port
a	Storage adapter ID	One-letter designator of storage adapter (a, b, c…).			
0	Device unit number	Unique number (MSCP unit number). SCSI unit numbers are forced to 100 X node ID.			
0	Bus node number	Bus node ID.			
0	Channel number	Used for multi-channel devices.			
15	Logical slot number	Corresponds to PCI slot number, as shown in Table 3-2.			
0	Hose number	0 — PCI 0 1 — PCI 1			

Example 3-7 show memory

```
P00>>>show memory
  Array       Size       Base Address
---------  ----------  ----------------
    0        256Mb     0000000060000000
    1        512Mb     0000000040000000
    2        256Mb     0000000070000000
    3       1024Mb     0000000000000000


     2048 MB of System Memory
```

The **show memory** display corresponds to the memory array configuration. The display does not indicate the number of DIMMs or their size. In this example, Array 3 could consist of two sets of 128-MB DIMMs (eight DIMMs) or one set of 256-MB DIMMs (four DIMMs). Either combination provides 1024 MB of memory.

Use the **show fru** command to display the DIMMs in the system and their location.

3.4.4 Setting SRM Environment Variables

You may need to set several SRM console environment variables and built-in utilities to configure systems running the *Tru64 UNIX* or *OpenVMS* operating systems.

Set environment variables at the P00>>> prompt.

- To check the setting for a specific environment variable, enter the **show *envar*** command, where the name of the environment variable is substituted for *envar*.
- To reset an environment variable, use the **set *envar*** command, where the name of the environment variable is substituted for *envar*.

The boot-related environment variables are described in Section 3.5. For other environment variables you may need to set, refer to the Compaq *AlphaServer ES40 User Interface Guide*.

Table 3-4 summarizes the SRM environment variables.

Table 3-4 SRM Environment Variables

Variable	Attributes	Description
auto_action	NV,W[1]	The action the console should take following an error halt or power failure. Defined values are: **boot**—Attempt bootstrap. **halt**—Halt, enter console I/O mode. **restart**—Attempt restart. If restart fails, try boot.
bootdef_dev	NV,W	The device or device list from which booting is to be attempted when no path is specified. Set at factory to disk with factory-installed software; otherwise **NULL**.
boot_file	NV,W	The default file name used for the primary bootstrap when no file name is specified by the **boot** command. The default value is **NULL**.

---

[1] NV—Nonvolatile. The last value saved by system software or set by console commands is preserved across system initializations, cold bootstraps, and long power outages.

W—Warm nonvolatile. The last value set by system software is preserved across warm bootstraps and restarts.

Table 3-4 SRM Environment Variables (Cont.)

Variable	Attributes	Description
boot_osflags	NV,W	Default parameters to be passed to system software during booting if none are specified by the **boot** command. ***OpenVMS***: Additional parameters are the *root_number* and *boot flags*. The default value is NULL. *root_number:* Directory number of the system disk on which *OpenVMS* files are located. 0 (default)—[SYS0.SYSEXE] 1—[SYS1.SYSEXE] 2—[SYS2.SYSEXE] 3—[SYS3.SYSEXE]
boot_osflags (continued)		*boot_flags:* The hexadecimal value of the bit number or numbers to set. To specify multiple boot flags, add the flag values (logical OR). 1—Bootstrap conversationally (enables you to modify SYSGEN parameters in SYSBOOT). 2—Map XDELTA to running system. 4—Stop at initial system breakpoint. 8—Perform a diagnostic bootstrap. 10—Stop at the bootstrap breakpoints. 20—Omit header from secondary bootstrap file. 80—Prompt for the name of the secondary bootstrap file. 100—Halt before secondary bootstrap. 10000—Display debug messages during booting. 20000—Display user messages during booting. ***Tru64 UNIX***: The following parameters are used with this operating system: **a**—Autoboot. Boots /vmunix from bootdef_dev, goes to multi-user mode. Use this for a system that should come up automatically after a power failure. **s**—Stop in single-user mode. Boots /vmunix to single-user mode and stops at the # (root) prompt. **i**—Interactive boot. Requests the name of the image to boot from the specified boot device. Other flags, such as -kdebug (to enable the kernel debugger), may be entered using this option. **D**—Full dump; implies s as well. By default, if *Tru64 UNIX* crashes, it completes a partial memory dump. Specifying **D** forces a full dump at system crash. Common settings are a, autoboot, and Da, autoboot and create full dumps if the system crashes.
com1_baud	NV,W	Sets the baud rate of the COM1 (MMJ) port. The default baud rate is 9600. Baud rate values are 1800, 2000, 2400, 3600, 4800, 7200, 9600, 19200, 38400.
com2_baud	NV,W	Sets the baud rate of the COM2 port. The default baud rate is 9600. Baud rate values are 1800, 2000, 2400, 3600, 4800, 7200, 9600, 19200, 38400.

Table 3-4 SRM Environment Variables (Cont.)

Variable	Attributes	Description
com1_flow, com2_flow	NV,W	The **com1_flow** and **com2_flow** environment variables indicate the flow control on the serial ports. Defined values are: **none**—No data flows in or out of the serial ports. Use this setting for devices that do not recognize XON/XOFF or that would be confused by these signals. **software**—Use XON/XOFF(default). This is the setting for a standard serial terminal. **hardware**—Use modem signals CTS/RTS. Use this setting if you are connecting a modem to a serial port.
com1_modem, com2_modem	NV,W	Used to tell the operating system whether a modem is present on the COM1 or COM2 ports, respectively **On**— Modem is present. **Off**—Modem is not present (default value).
console	NV	Sets the device on which power-up output is displayed. **Graphics**—Sets the power-up output to be displayed at a VGA monitor or device connected to the VGA module. **Serial**—Sets the power-up output to be displayed on the device that is connected to the COM1 (MMJ) port.
cpu_enabled	NV	Enables or disables a specific secondary CPU. All CPUs are enabled by default. The primary CPU cannot be disabled. The primary CPU is the lowest numbered working CPU.
ew*0_inet_init	NV	Determines whether the interface's internal Internet database is initialized from nvram or from a network server (via the bootp protocol).
ew*0_mode	NV	Sets the Ethernet controller to the default Ethernet device type. **aui**—Sets the default device to AUI. **bnc**—Sets the default device to ThinWire. **fast**—Sets the default device to fast 100BaseT. **fastfd**—Sets the default device to fast full duplex 100BaseT. **full**—Set the default device to full duplex twisted pair. **twisted-pair**— Sets the default device to 10BaseT (twisted-pair).
ew*0_protocols	NV	Determines which network protocols are enabled for booting and other functions. **mop**—Sets the network protocol to MOP for systems using the *OpenVMS* operating system. **bootp**—Sets the network protocol to bootp for systems using the *Tru64 UNIX* operating system. **bootp,mop**—When the settings are used in a list, the mop protocol is attempted first, followed by bootp.
kbd_hardware_ type	NV	Used only on systems with the language variant 3C (Français), sets the keyboard hardware type as either PCXAL (102-type keyboard) or LK411 (French keyboard).
kzpsa_host_id	W	Specifies the default value for the KZPSA host SCSI bus node ID.
language	NV	Specifies the console keyboard layout. The default is English (American).

Table 3-4 SRM Environment Variables (Cont.)

Variable	Attributes	Description
memory_test	NV	Specifies the extent to which memory will be tested on *Tru64 UNIX*. The options are: **Full**—Full memory test will be run. Required for *OpenVMS*. **Partial**—First 256 MB of memory will be tested. **None**—Only first 32 MB will be tested.
ocp_text	NV	Overrides the default control panel display text with specified text.
os_type	NV	Sets the default operating system. **vms or unix**—Sets system to boot the SRM firmware. **nt**—Sets system to boot the AlphaBIOS firmware.
pci_parity	NV	Disable or enable parity checking on the PCI bus. **On**—PCI parity enabled (default value) **Off**—PCI parity disabled Some PCI devices do not implement PCI parity checking, and some have a parity-generating scheme in which the parity is sometimes incorrect or is not fully compliant with the PCI specification. In such cases, the device functions properly so long as parity is not checked.
pk*0_fast	NV	Enables fast SCSI devices on a SCSI controller to perform in standard or fast mode. **0**—Sets the default speed for devices on the controller to standard SCSI. If a controller is set to standard SCSI mode, both standard and fast SCSI devices will perform in standard mode. **1**—Sets the default speed for devices on the controller to fast SCSI mode. Devices on a controller that connects to both standard and Fast SCSI devices will automatically perform at the appropriate rate for the device, either fast or standard mode.
pk*0_host_id	NV	Sets the controller host bus node ID to a value between 0 and 7. 0 to 7—Assigns bus node ID for specified host adapter.
pk*0_soft_term	NV	Enables or disables SCSI terminators for optional SCSI controllers. This environment variable applies to systems using the QLogic SCSI controller, though it does not affect the onboard controller. The QLogic SCSI controller implements the 16-bit wide SCSI bus. The QLogic module has two terminators, one for the 8 low bits and one for the high 8 bits. There are five possible values: **off**—Turns off both low 8 bits and high 8 bits. **low**—Turns on low 8 bits and turns off high 8 bits. **high**—Turns on high 8 bits and turns off low 8 bits. **on**—Turns on both low 8 bits and high 8 bits. **diff**—Places the bus in differential mode.

Table 3-4 SRM Environment Variables (Cont.)

Variable	Attributes	Description
tt_allow_login	NV	Enables or disables login to the SRM console firmware on alternative console ports. **0**—Disables login on alternative console ports. **1**—Enables login on alternative console ports (default setting). If the console output device is set to **serial**, **set tt_allow_login 1** allows you to log in on the primary COM1(MMJ) port, or alternate COM2 port, or the VGA monitor. If the console output device is set to **graphics**, **set tt_allow_login 1** allows you to log in through either the COM1(MMJ) or COM2 console port.

3.4.5 Setting SRM Console Security

You can set the SRM console to secure mode to prevent unauthorized personnel from modifying the system parameters or otherwise tampering with the system from the console.

When the SRM is set to secure mode, you can use only two console commands:

- The **boot** command, to boot the operating system
- The **continue** command, to resume running the operating system if you have inadvertently halted the system

The console security commands are as follows:

set password **set secure**	These commands put the console into secure mode.
clear password	Exits secure mode.
login	Turns off console security for the current session.

See the Compaq *AlphaServer ES40 User Interface Guide* for details on setting SRM console security.

3.4.6 Setting Automatic Booting

Tru64 UNIX and *OpenVMS* operating systems are factory set to halt in the SRM console. You can change these defaults, if desired.

Compaq AlphaServer ES40sv systems can boot automatically (if set to autoboot) from the default boot device under the following conditions:

- When you first turn on system power
- When you power cycle or reset the system
- When system power comes on after a power failure
- After a bugcheck (*OpenVMS*) or panic (*Tru64 UNIX*)

Setting *Tru64 UNIX* or *OpenVMS* Systems to Auto Start

The SRM **auto_action** environment variable determines the default action the system takes when the system is power cycled, reset, or experiences a failure. On systems that are factory configured for *Tru64 UNIX* or *OpenVMS*, the factory setting for **auto_action** is **halt**. The **halt** setting causes the system to stop in the SRM console. You must then boot the operating system manually.

For maximum system availability, **auto_action** can be set to **boot** or **restart**.

- With the **boot** setting, the operating system boots automatically after the SRM **init** command is issued or the Reset button is pressed.
- With the **restart** setting, the operating system boots automatically after the SRM **init** command is issued or the Reset button is pressed , and it also reboots after an operating system crash.

To set the default action to **boot**, enter the following SRM commands:

```
P00>>> set auto_action boot
P00>>> init
```

For more information on **auto_action**, see the Compaq *AlphaServer ES40 User Interface Guide*.

3.4.7 Changing the Default Boot Device

Tru64 UNIX* or *OpenVMS

When you install the *Tru64 UNIX* or *OpenVMS* operating system, you designate a default boot device. You can change the default boot device by using the **set bootdef_dev** SRM console command. For example, to set the boot device to the IDE CD-ROM, enter commands similar to the following:

```
P00>>> show bootdef_dev
bootdef_dev    dka400.4.0.1.1
P00>>> set bootdef_dev dqa500.5.0.1.1
P00>>> show bootdef_dev
bootdef_dev    dqa500.5.0.1.1
```

See the Compaq *AlphaServer ES40 User Interface Guide* for more information.

3.5 Setting Boot Options for *Tru64 UNIX* or *OpenVMS*

You can set a default boot device, boot flags, and network boot protocols for *Tru64 UNIX* or *OpenVMS* using the SRM **set** command with environment variables. Once these environment variables are set, the **boot** command defaults to the stored values. You can override the stored values for the current boot session by entering parameters on the **boot** command line.

The SRM boot-related environment variables are listed below and described in the following sections:

bootdef_dev	Defines a default boot device
boot_file	Specifies a default file name to be used for booting when no file name is specified by the boot command
boot_osflags	Defines parameters to enable specific functions during the boot process
ei*0_inet_init or **ew*0_inet_init**	Determines whether the interface's internal Internet database is initialized from nvram or from a network server (through the bootp protocol). Set this environment variable if you are booting *Tru64 UNIX* from a RIS server.
ei*0_protocols or **ew*0_protocols**	Defines a default network boot protocol (bootp or mop).

3.5.1 bootdef_dev

The **bootdef_dev** environment variable specifies one or more devices from which to boot the operating system. When more than one device is specified, the system searches in the order listed and boots from the first device.

Enter the **show bootdef_dev** command to display the current default boot device. Enter the **show device** command for a list of all devices in the system.

The syntax is:

set bootdef_dev *boot_device*

boot_device	The name of the device on which the system software has been loaded. To specify more than one device, separate the names with commas.

Example:

In this example, two boot devices are specified. The system will try booting from dkb0 and, if unsuccessful, will boot from dka0.

```
P00>>> set bootdef_dev dkb0, dka0
```

Note

When you set the ***bootdef_dev*** environment variable, it is recommended that you set the operating system boot parameters as well, using the ***set boot_osflags*** command.

3.5.2 boot_file

The **boot_file** environment variable specifies the default file name to be used for booting when no file name is specified by the **boot** command.

The syntax is:

set boot_file *filename*

Example:

In this example, a boot file is specified for booting *OpenVMS* from the InfoServer. APB_0712 is the file name of the APB program used for the initial system load (ISL) boot program.

```
P00>>> set boot_file apb_0712
```

3.5.3 boot_osflags

The **boot_osflags** environment variable sets the default boot flags and, for *OpenVMS*, a root number.

Boot flags contain information used by the operating system to determine some aspects of a system bootstrap. Under normal circumstances, you can use the default boot flag settings.

To change the boot flags for the current boot only, use the *flags_value* argument with the **boot** command.

The syntax is:

set boot_osflags *flags_value*

The *flags_value* argument is specific to the operating system.

***Tru64 UNIX* Systems**

Tru64 UNIX operating systems take a single ASCII character as the *flags_value* argument.

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

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

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

D Full dump; implies “s” as well. By default, if *Tru64 UNIX* crashes, it completes a partial memory dump. Specifying “D” forces a full dump at system crash.

OpenVMS Systems

OpenVMS systems require an ordered pair as the *flags_value* argument: *root_number* and *boot_flags*.

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

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

boot_flags — The hexadecimal value of the bit number or numbers set. To specify multiple boot flags, add the flag values (logical OR). For example, the flag value 10080 executes both the 80 and 10000 flag settings. See Table 3-5.

Table 3-5 *OpenVMS* Boot Flag Settings

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

Examples:

In the following *Tru64 UNIX* example, the boot flags are set to autoboot the system to multiuser mode when you enter the **boot** command.

```
P00>>> set boot_osflags a
```

In the following *OpenVMS* example, *root_number* is set to 2 and *boot_flags* is set to 1. With this setting, the system will boot from root directory SYS2.SYSEXE to the SYSBOOT prompt when you enter the **boot** command.

```
P00>>> set boot_osflags 2,1
```

In the following *OpenVMS* example, root_number is set to 0 and boot_flags is set to 80. With this setting, you are prompted for the name of the secondary bootstrap file when you enter the **boot** command.

```
P00>>> set boot_osflags 0,80
```

3.5.4 ei*0_inet_init or ew*0_inet_init

The **ei*0_inet_init** or **ew*0_inet_init** environment variable determines whether the interface's internal Internet database is initialized from nvram or from a network server (through the bootp protocol). Legal values are **nvram** and **bootp**. The default value is **bootp**. Set this environment variable if you are booting *Tru64 UNIX* from a RIS server.

To list the network devices on your system, enter the **show device** command. The Ethernet controllers start with the letters "ei" or "ew," for example, ewa0. The third letter is the adapter ID for the specific Ethernet controller. Replace the asterisk (*) with the adapter ID letter when entering the command.

The syntax is:

set ei*0_inet_init *value* or
set ew*0_inet_init *value*

Example:

```
P00>>> set ewa0_inet_init bootp
```

3.5.5 ei*0_protocols or ew*0_protocols

The **ei*0_protocols** or **ew*0_protocols** environment variable sets network protocols for booting and other functions.

To list the network devices on your system, enter the **show device** command. The Ethernet controllers start with the letters "ei" or "ew," for example, ewa0. The third letter is the adapter ID for the specific Ethernet controller. Replace the asterisk (*) with the adapter ID letter when entering the command.

The syntax is:

set ei*0_protocols *protocol_value* or
set ew*0_protocols *protocol_value*

The options for *protocol_value* are:

mop (default)	Sets the network protocol to mop (Maintenance Operations Protocol), the setting typically used with the *OpenVMS* operating system.
bootp	Sets the network protocol to bootp, the setting typically used with the *Tru64 UNIX* operating system.
bootp,mop	When both are listed, the system attempts to use the mop protocol first, regardless of which is listed first. If not successful, it then attempts the bootp protocol.

Example

```
P00>>> show device
.
.
.
ewa0.0.0.1001.0     EWA0          08-00-2B-3E-BC-B5
ewb0.0.0.12.0       EWB0          00-00-C0-33-E0-0D
ewc0.0.0.13.0       EWC0          08-00-2B-E6-4B-F3
.
.
.
P00>>> set ewa0_protocols bootp
P00>>> show ewa0_protocols
ewa0_protocols        bootp
```

3.6 Booting *Tru64 UNIX*

Tru64 UNIX can be booted from a CD-ROM on a local drive (a CD-ROM drive connected to the system); from a local SCSI disk, or from a *Tru64 UNIX* RIS server.

3.6.1 Booting *Tru64 UNIX* from a Local SCSI Disk

Example 3-8 shows a boot from a local SCSI drive. The example is abbreviated. For complete instructions on booting *Tru64 UNIX*, see the *Tru64 UNIX Installation Guide*.

Perform the following tasks to boot a *Tru64 UNIX* system:

1. Power up the system. The system will stop at the SRM console prompt, P00>>>.
2. Set boot environment variables, as described in Section 3.5.
3. Install the boot medium. For a network boot, see Section 3.6.2.
4. Enter the **show device** command ❶ to determine the unit number of the drive for your device.
5. Enter the **boot** command ❷ and command line parameters (if you have not set the associated environment variables). In Example 3-8, boot flags ❸ have already been set.

Example 3-8 Booting *Tru64 UNIX* from a Local SCSI Disk

```
P00>>> sho dev                                                   ❶
dka0.0.0.1.1           DKA0                        RZ2ED-LS  0306
dka100.1.0.1.1         DKA100                      RZ2ED-LS  0306
dka200.2.0.1.1         DKA200                      RZ2DD-LS  0306
dka300.3.0.1.1         DKA300                      RZ2DD-LS  0306
dkc0.0.0.1.0           DKC0                        RZ2DD-LS  0306
dkc100.1.0.1.0         DKC100                      RZ2DD-LS  0306
dkc200.2.0.1.0         DKC200                      RZ2DD-LS  0306
dkc300.3.0.1.0         DKC300                      RZ2DD-LS  0306
dqa0.0.0.15.0          DQA0       TOSHIBA CD-ROM XM-6202B  1110
dva0.0.0.1000.0        DVA0
ewa0.0.0.4.1           EWA0             00-00-F8-10-67-97
pka0.7.0.1.1           PKA0                 SCSI Bus ID 7

P00>>> boot                                                      ❷
(boot dka0.0.0.1.1 -flags a)                                     ❸
block 0 of dka0.0.0.1.1 is a valid boot block
reading 13 blocks from dka0.0.0.1.1
bootstrap code read in
base = 200000, image_start = 0, image_bytes = 1a00
initializing HWRPB at 2000
initializing page table at 1fff0000
initializing machine state
setting affinity to the primary CPU
jumping to bootstrap code

Tru64 UNIX boot - Thu Jan 14 15:03:19 EST 1999

Loading vmunix ...
Loading at 0xfffffc0000230000
Current PAL Revision <0x4000500010130>
Switching to OSF PALcode Succeeded
New PAL Revision <0x400050002012d>
```

```
Sizes:
text = 4836176
data = 1045600
bss = 1603520
Starting at 0xfffffc00005671e0
Loading vmunix symbol table ... [1333528 bytes]
sysconfigtab: attribute Per-proc-address-space not in subsystem proc
Alpha boot: available memory from 0x134c000 to 0x1ffee000
Tru64 UNIX V4.0F-4 (Rev. 1180); Tue Feb 2 13:00:04 EST 1999
physical memory = 512.00 megabytes.
available memory = 492.64 megabytes.
using 1958 buffers containing 15.29 megabytes of memory
Master cpu at slot 0.
Firmware revision: 5.4-5160
PALcode: Tru64 UNIX version 1.45-5
Compaq AlphaServer ES40
.
.
.
Tru64 UNIX Version V4.0F

Login:
```

3.6.2 Booting *Tru64 UNIX* Over the Network

Note

To boot your *Tru64 UNIX* system over the network, make sure the system is registered on a Remote Installation Services (RIS) server. See the *Tru64 UNIX* document entitled *Sharing Software on a Local Area Network* for registration information.

Systems running *Tru64 UNIX* support network adapters, designated ew*0 or ei*0. The asterisk stands for the adapter ID (a, b, c, and so on).

1. Power up the system. The system stops at the SRM P00>>> console prompt.
2. Set boot environment variables, if desired. See Section 3.5.
3. Enter the **show device** command ❶ to determine the unit number of the drive for your device.
4. Enter the following commands. Example 3-9 assumes you are booting from ewa0. If you are booting from another drive, enter that device instead.

   ```
   P00>>> set ewa0_protocols bootp
   P00>>> set ewa0_inet_init bootp
   ```

   The first command ❷ enables the bootp network protocol for booting over the Ethernet controller. The second command ❸ sets the internal Internet database to initialize from the network server through the bootp protocol.
5. Enter the **boot** command ❹ and command line parameters (if you have not set the associated environment variables). In Example 3-9, the **boot** command sets the system to boot automatically from ewa0 and specifies a full memory dump in case of a system shutdown.

For complete instructions on booting *Tru64 UNIX* over the network, see the *Tru64 UNIX Installation Guide*

Example 3-9 Booting *Tru64 UNIX* Over the Network

```
P00>>> show device                                                   ❶
dka0.0.0.1.1           DKA0                          RZ2DD-LS  0306
dka100.1.0.1.1         DKA100                        RZ2DD-LS  0306
dka200.2.0.1.1         DKA200                        RZ1CB-CS  0844
dkb0.0.0.3.1           DKB0                              RZ25  0900
dqa0.0.0.15.0          DQA0           TOSHIBA CD-ROM XM-6302B  1012
dva0.0.0.1000.0        DVA0
ewa0.0.0.4.1           EWA0                 00-00-F8-09-90-FF
ewb0.0.0.2002.1        EWB0                 00-06-2B-00-25-5B
pka0.7.0.1.1           PKA0                     SCSI Bus ID 7
pkb0.7.0.3.1           PKB0                     SCSI Bus ID 7
P00>>> set ewa0_protocols bootp                                      ❷
P00>>> set ewa0_inet_init bootp                                      ❸
P00>>> boot ewa0 Da                                                  ❹
.
.
.
```

3.7 Starting a *Tru64 UNIX* Installation

Tru64 UNIX is installed from the CD-ROM drive connected to the system. The display that you see after you boot the CD depends on whether your system console is a VGA monitor or a serial terminal.

Installation Procedure

Install *Tru64 UNIX* as follows:

1. Boot the operating system from the CD-ROM drive connected to the system.
2. Follow the installation instructions that are displayed after the installation process is loaded.

   If your system console is a VGA monitor, the X Server is started and an Installation Setup window is displayed. Click on the fields in the Installation Setup window to enter your responses to the installation procedure.

   If your system console is a serial terminal, a text-based installation procedure is displayed, as shown in Example 3-10. Enter the choices appropriate for your system.

See the *Tru64 UNIX Installation Guide* for complete installation instructions.

Example 3-10 Text-Based Installation Display

```
P00>>> boot dqa0
(boot dqa0.0.0.15.0 -flags a
block 0 of dqa0.0.0.15.0 is a valid boot block
reading 16 blocks from dqa0.0.0.15.0
bootstrap code read in
base = 200000, image_start = 0, image_bytes = 2000
initializing HWRPB at 2000
initializing page table at 1fff0000
initializing machine state
setting affinity to the primary CPU
jumping to bootstrap code

Tru64 UNIX boot - Thu Jan 14 15:03:19 EST 1999

Loading vmunix ...
.
.
Initializing system for Tru64 UNIX installation. Please
wait...

*** Performing CDROM Installation

Loading installation process and scanning system hardware.

Welcome to the Tru64 UNIX Installation Procedure

This procedure installs Tru64 UNIX onto your system. You will be asked
a series of system configuration questions. Until you answer all
questions, your system is not changed in any way.

During the question and answer session, you can go back to any previous
question and change your answer by entering:  history
You can get more information about a question by entering:  help
```

```
There are two types of installations:

o  The Default Installation installs a mandatory set of
   software subsets on a predetermined file system layout.

o  The Custom Installation installs a mandatory set of
   software subsets plus optional software subsets that you
   select.  You can customize the file system layout.

The UNIX Shell option puts your system in single-user mode with
superuser privileges.  This option is provided for experienced Tru64
UNIX system administrators who want to perform file system or disk
maintenance tasks before the installation.

The Installation Guide contains more information about installing Tru64
UNIX.

1) Default Installation
2) Custom Installation
3) UNIX Shell

Enter your choice:
```

3.8 Booting *OpenVMS*

OpenVMS can be booted from a CD-ROM on a local drive (the CD-ROM drive connected to the system) or from a CD-ROM drive on the InfoServer.

3.8.1 Booting *OpenVMS* from the Local CD-ROM Drive

Example 3-11 shows a boot from a CD-ROM on a local drive. The example is abbreviated. For complete instructions on booting *OpenVMS*, see the *OpenVMS* installation document.

Perform the following tasks before booting an *OpenVMS* system:

1. Power up the system. The system stops at the SRM console prompt P00>>>.
2. Set boot environment variables, as described in Section 3.5.
3. Install the boot medium. For a network boot, see Section 3.8.2.
4. Enter the **show device** command ➊ to determine the unit number of the drive for your device.
5. Enter the **boot** command and command line parameters (if you have not set the associated environment variables). In Example 3-11, the **boot** command with the **–flags** option ➋ causes the system to boot from [SYS0.EXE] on device DKA0.

Example 3-11 Booting *OpenVMS* from the Local CD-ROM Drive

```
P00>>> show device                                                    ➊
dka0.0.0.1.1            DKA0                          RZ2CA-LA  N1H0
dka100.1.0.1.1          DKA100                        RZ2CA-LA  N1H0
dqa0.0.0.15.0           DQA0       TOSHIBA CD-ROM XM-6302B  1012
dva0.0.0.1000.0         DVA0
ewa0.0.0.6.1            EWA0             00-00-F8-10-D6-03
pka0.7.0.1.1            PKA0                  SCSI Bus ID 7
P00>>>
.
.
.
P00>>> boot -flags 0,0 dka0                                           ➋
(boot dka0.0.0.1.1 -flags 0,0)
block 0 of dka0.0.0.1.1 is a valid boot block
reading 898 blocks from dka0.0.0.1.1
bootstrap code read in
base = 200000, image_start = 0, image_bytes = 70400
initializing HWRPB at 2000
initializing page table at 3ffee000
initializing machine state
setting affinity to the primary CPU
jumping to bootstrap code

OpenVMS (TM) Alpha Operating System, Version V7.1-2
```

3.8.2 Booting *OpenVMS* from the InfoServer

You can boot *OpenVMS* from a LAN device on the InfoServer. The devices are designated EW*0 or EI*0. The asterisk stands for the adapter ID (a, b, c, and so on).

1. Power up the system. The system stops at the P00>>> console prompt.
2. Insert the operating system CD-ROM into the CD-ROM drive connected to the InfoServer.
3. Enter the **show device** command ❶ to determine the unit number of the drive for your device.
4. Enter the **boot** command and any command line parameters ❷. In Example 3-12 the device is EWA0. APB_0712 is the file name of the APB program used for the initial system load (ISL) boot program.

   The InfoServer ISL program displays a menu ❸.
5. Respond to the menu prompts ❹, using the selections shown in this example.

For complete instructions on booting *OpenVMS* from the InfoServer, see the *OpenVMS* installation document.

Example 3-12 Booting *OpenVMS* from the InfoServer

```
P00>>> show device                                                      ❶
dka0.0.0.1.1             DKA0                          RZ2CA-LA  N1H0
dka100.1.0.1.1           DKA100                        RZ2CA-LA  N1H0
dqa0.0.0.15.0            DQA0       TOSHIBA CD-ROM XM-6302B  1012
dva0.0.0.1000.0          DVA0
ewa0.0.0.6.1             EWA0             00-00-F8-10-D6-03
pka0.7.0.1.1             PKA0                  SCSI Bus ID 7
P00>>>
.
.
.
P00>>> boot -flags 0,0 -file apb_0712 ewa0                              ❷
(boot ewa0.0.0.6.1 -file APB_0712 -flags 0,0)
Trying MOP boot.
.............
Network load complete.
Host name: CALSUN
Host address: aa-00-04-00-a4-4e
bootstrap code read in
base = 200000, image_start = 0, image_bytes = 70400
initializing HWRPB at 2000
initializing page table at 3ffee000
initializing machine state
setting affinity to the primary CPU
jumping to bootstrap code
Network Initial System Load Function                                    ❸
Version 1.2
```

```
FUNCTION             FUNCTION
ID
1       -            Display Men
2       –            Help
3       -            Choose Service
4       -            Select Options
5       –            Stop
Enter a function ID value:

Enter a function ID Value: 3                                          ❹
OPTION               OPTION
ID
1       -            Find Services
2       -            Enter known Service Name

Enter an Option ID value: 2
Enter a Known Service Name: ALPHA_V71-2_SSB

      OpenVMS (TM) Alpha Operating System, Version V7.1-2
```

3.9 Starting an *OpenVMS* Installation

After you boot the operating system CD-ROM, an installation menu is displayed on the screen. Choose item 1 (Install or upgrade *OpenVMS* Alpha). Then refer to the *OpenVMS* installation document for information on creating the system disk.

1. Boot the *OpenVMS* operating system CD-ROM as described in the previous sections.
2. Choose option 1 (Install or upgrade *OpenVMS* Alpha). To create the system disk, see the *OpenVMS* installation document, *Installing the OpenVMS Alpha Operating System*.

Example 3-13 *OpenVMS* Installation Menu

```
OpenVMS (TM) Alpha Operating System, Version V7.1-2
Copyright © 1999 Digital Equipment Corporation.  All rights reserved.

Installing required known files...

Configuring devices...

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

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

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

Please choose one of the following:

    1)  Install or upgrade OpenVMS Alpha Version V7.1-2
    2)  Display products that this procedure can install
    3)  Install or upgrade layered products
    4)  Show installed products
    5)  Reconfigure installed products
    6)  Remove installed products
    7)  Execute DCL commands and procedures
    8)  Shut down this system

Enter CHOICE or ? for help: (1/2/3/4/5/6/7/8/?) 1
```

3.10 Updating Firmware

The following sections describe how to update to a later version of system firmware. In general, system firmware must be updated whenever the operating system is updated. You might also need to update firmware:

- If you add I/O device controllers and adapters
- If enhancements are made to the firmware
- If the serial ROM or RMC firmware should ever become corrupted

The SRM firmware for *Tru64 UNIX* and *OpenVMS* reside in the flash ROM located on the system motherboard. You can update the system firmware from the following sources:

- Manually
- CD-ROM
- System disk (*OpenVMS*)
- Network protocol (bootp for *Tru64 UNIX* or MOP for *OpenVMS*)

3.10.1 Sources of Firmware Updates

The Alpha Systems Firmware Update Kit comes on a CD-ROM, which is updated quarterly. You can also obtain Alpha firmware updates from the Internet.

Quarterly Update Service

The Alpha Systems Firmware Update Kit CD-ROM is available by subscription from Compaq.

Alpha Firmware Internet Access

- You can also obtain Alpha firmware updates from the Internet:

  http://www.compaq.com/alphaserver/

  Click on the name of the system. On the page for the system, click on Firmware Updates.

- If you do not have a Web browser, you can access files using anonymous ftp:

  ftp://ftp.digital.com/pub/DEC/

  Click down the following directories: Alpha/firmware/readme.html

  The README file explains how to download firmware updates.

3.10.2 Firmware Update Utility

The system firmware is updated from a Loadable Firmware Update Utility. When you boot the media containing the update image, the Loadable Firmware Update Utility banner is displayed.

Before updating the firmware, enter the **list** command to list the current revision of the firmware. Enter the **update** command to update the SRM firmware automatically.

Example 3-14 Update Utility Display

```
          ***** 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.
 Readme      Lists important release information.
 Update      Replaces current firmware with loadable data
             image.
 Verify      Compares loadable and hardware images.
 ? or Help   Scrolls this function table.
------------------------------------------------------------_
UPD> list

Device       Current Revision      Filename     Update Revision

Abios         5.68                 abios_fw      5.69

SRM           5.4                  srm_fw        5.5


UPD> update
```

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

                          DO NOT ABORT!

Abios    Updating to V5.6-9...  Verifying V5.6-9...  PASSED.
srm      Updating to V5.4-7...  Verifying V5.4-7...  PASSED.

UPD> exit
```

3.10.3 Manual Updates

If RMC firmware or serial ROM (SROM) ever become corrupted, you can perform a manual update as follows:

1. Boot the update medium.
2. At the UPD> prompt, enter the **exit** command and answer **y** at the prompt:

   ```
   UPD> exit

   Do you want to do a manual update [y/(n)] y

   AlphaServer ES40 Console V5.4-5528, built on April 6, 1999 at
   05:02:30
   ```

3. To update RMC firmware, enter **update rmc**. To update serial ROM (SROM), enter **update srom**. For example:

   ```
   UPD> update srom
   ```

The remainder of the display is similar to that shown in Example 3-14.

3.10.4 Updating from the CD-ROM

You can update the system firmware from CD-ROM.

1. At the SRM console prompt, enter the **show device** command to determine the drive name of the CD-ROM drive.
2. Load the Alpha Systems Firmware Update CD into the drive.
3. Boot the system from the CD, using the drive name determined in step 1 (in this case, dqa0).

   ```
   P00>>> boot dqa0
   ```

4. Enter the **update** command at the UPD> prompt:
5. When the update is complete, exit from the Firmware Update Utility.

   ```
   UPD> exit
   ```

3.10.5 Updating from an *OpenVMS* System Disk

You can update *OpenVMS* from a system disk.

1. Download the firmware update image from the Firmware Updates Web site.
2. Rename the downloaded file to fwupdate.exe.
3. Enter the following commands on the *OpenVMS* Alpha system:

```
$ set file/attr=(rfm:fix,lrl:512,mrs:512,rat:none) fwupdate.exe
$ copy/contiguous fwupdate.exe "system_disk":[sys0.sysexe]
```

Note

Insert the name of your system disk in place of "system_disk," for example, dka100:.

4. Shut down the operating system to get to the SRM console prompt.
5. Boot the update utility from the SRM console as follows:

```
P00>>> boot dka100 -flags 0,a0
```

Note

Replace dka100 with the name of the system disk, if different.

6. After some messages are displayed, you will be prompted for the bootfile. Enter the directory and file name as follows :

```
Bootfile: [sys0.sysexe]fwupdate.exe
```

7. Enter the **update** command at the UPD> prompt.

3.10.6 *OpenVMS* and *Tru64 UNIX* Network Boots

You can update *OpenVMS* using the MOP network protocol and *Tru64 UNIX* with the BOOTP protocol.

3.10.6.1 Updating *OpenVMS* Using the MOP Protocol

1. Download the firmware update image from the Firmware Updates Web site.
2. Copy the downloaded file to an *OpenVMS* based network server for MOP booting on the *AlphaServer* system. For details on configuring the MOP server, refer to *OpenVMS* documentation or the system's Firmware Release Notes document.
3. To ensure that the downloaded file is in a proper VMS fixed record format, enter the following command before using the file for MOP booting:

```
$ set file/attr=(rfm:fix,lrl:512,mrs:512,rat:none) "fwupdate.sys"
```

Note

Replace "fwupdate.sys" with the name of the firmware image you downloaded.

4. Boot the update file. For example:

```
P00>>> boot -file fwupdate ewa0
```

5. Enter the **update** command at the UPD> prompt.

3.10.6.2 Updating *Tru64 UNIX* Using the BOOTP Protocol

1. Download the firmware update image from the Firmware Updates Web site.
2. Copy the downloaded file to a *Tru64 UNIX* based network server for BOOTP booting on the *AlphaServer* system. For details on configuring the BOOTP server, refer to *Tru64 UNIX* documentation or the system's Firmware Release Notes document.
3. Enter the **update** command at the UPD> prompt.

4

Remote System Management

4.1 Introduction

You can manage the system through the remote management console (RMC). The RMC is implemented through an independent microprocessor that resides on the system motherboard. The RMC also provides configuration and error log functionality.

The major topics covered in this chapter include:

- RMC Overview
- Operating Modes
- Terminal Setup
- Entering the RMC
- SRM Environment Variables for COM1
- RMC Command-Line Interface
- Resetting the RMC to Factory Defaults
- Troubleshooting Tips

4.2 RMC Overview

The remote management console provides a mechanism for monitoring the system (voltages, temperatures, and fans) and manipulating it on a low level (reset, power on/off, halt).

The RMC performs monitoring and control functions to ensure the successful operation of the system.

- Monitors thermal sensors on the CPUs, the PCI backplane, and the power supplies
- Monitors voltages, power supplies, and fans
- Handles hot swap of power supplies and fans
- Controls the operator control panel (OCP) display and writes status messages on the display
- Detects alert conditions such as excessive temperature, fan failure, and power supply failure. On detection, RMC displays messages on the OCP, pages an operator, and sends an interrupt to SRM, which then passes the interrupt to the operating system or an application.
- Shuts down the system if any fatal conditions exist. For example:
  - ➢ The temperature reaches the failure limit.

    - The main fan (Fan 6) and the redundant fan (Fan 5) fail.
- Retrieves and passes information about a system shutdown to SRM at the next power-up. SRM displays a message regarding the last shutdown.
- Provides a command-line interface (CLI) for the user to control the system. From the CLI you can power the system on and off, halt or reset the system, and monitor the system environment.
- Passes error log information to shared RAM so that this information can be accessed by the system.

The RMC logic is implemented using an 8-bit microprocessor, PIC17C44, as the primary control device. The firmware code resides on the microprocessor and in flash memory. If the RMC firmware should ever become corrupted or obsolete, you can update it manually using a Loadable Firmware Update Utility. See Section 3.10.3 for details. The microprocessor can also communicate with the system power control logic to turn on or turn off power to the rest of the system.

The RMC is powered by an auxiliary 5V power supply. You can gain access to the RMC as long as AC power is available to the system (through the wall outlet). Thus, if the system fails, you can still access the RMC and gather information about the failure.

Configuration, Error Log, and Asset Information

The RMC provides additional functionality to read and write configuration and error log information to FRU error log devices. These operations are carried out via shared RAM (also called dual-port RAM or DPR).

At power-on, the RMC reads the EEPROMs in the system and dumps the contents into the DPR. These EEPROMs contain configuration information, asset inventory and revision information, and error logs. During power-up the SROM sends status and error information for each CPU to the DPR. The system also writes error log information to the DPR when an error occurs. Service providers can access the contents of the DPR to diagnose system problems.

4.3 Operating Modes

The RMC can be configured to manage different data flow paths defined by the **com1_mode** environment variable. In through mode (the default), all data and control signals flow from the system COM1 port through the RMC to the active external port. You can also set bypass modes so that the signals partially or completely bypass the RMC. The **com1_mode** environment variable can be set from either SRM or the RMC (see Section 4.7.1).

4.3.1 Through Mode

Through mode is the default operating mode. The RMC routes every character of data between the internal system COM1 port and the active external port, either the local COM1 serial port (MMJ) or the 9-pin modem port. If a modem is connected, the data goes to the modem. The RMC filters the data for a specific escape sequence. If it detects the escape sequence, it enters the RMC CLI.

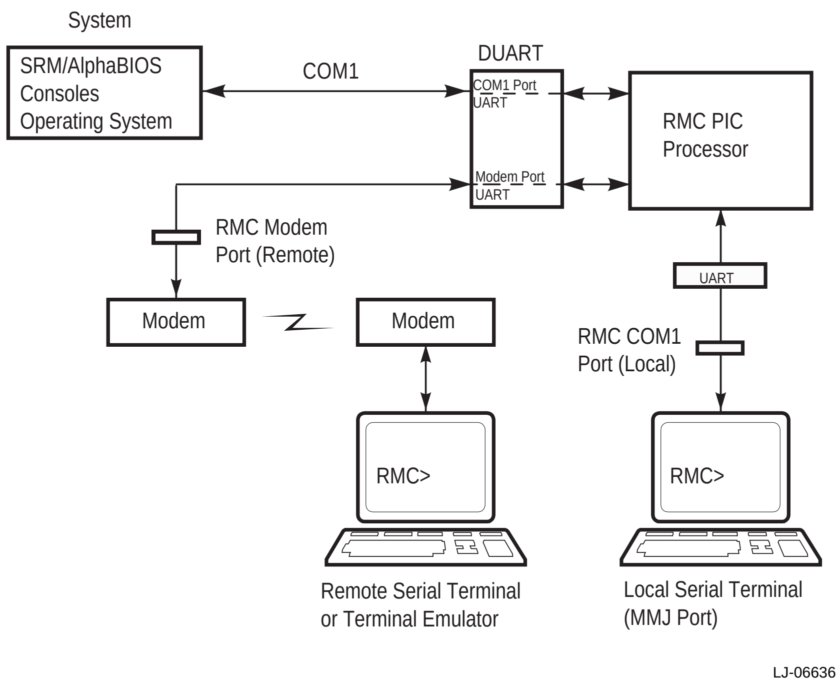
Figure 4-1 illustrates the data flow in through mode. The internal system COM1 port is connected to one port of the DUART chip, and the other port is connected to a 9-pin external modem port, providing full modem controls. The DUART is controlled by the RMC microprocessor, which moves characters between the two UART ports. The local MMJ port is always connected to the internal UART of the microprocessor. The escape sequence signals the RMC to enter the CLI. Data issued from the CLI is transmitted between the RMC microprocessor and the active port that enters the RMC.

Note

The internal system COM1 port should not be confused with the external COM1 serial port on the back of the system. The internal COM1 port is used by the system software to send data either to the COM1 port on the system or to the RMC modem port, if a modem is connected.

You can set a local mode in which only the local channel can communicate with the system COM1 port. In local mode the modem is prevented from sending characters to the system COM1 port, but you can still enter the RMC from the modem.

Figure 4-1 Data Flow in Through Mode



4.3.2 Bypass Modes

For modem connection, you can set the operating mode so that the data and control signals partially or completely bypass the RMC. The bypass modes are snoop, soft bypass, and firm bypass.

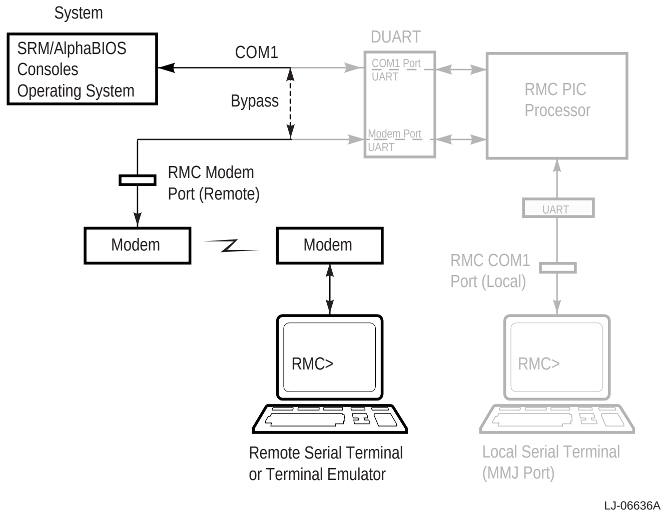
Figure 4-2 shows the data flow in the bypass modes. Note that the internal system COM1 port is connected directly to the modem port.

Note

You can connect a serial terminal to the modem port in any of the bypass modes.

The local terminal is still connected to the RMC and can still enter the RMC to switch the COM1 mode if necessary.

Figure 4-2 Data Flow in Bypass Mode



Snoop Mode

In snoop mode data partially bypasses the RMC. The data and control signals are routed directly between the system COM1 port and the external modem port, but the RMC taps into the data lines and listens passively for the RMC escape sequence. If it detects the escape sequence, it enters the RMC CLI.

The escape sequence is also passed to the system on the bypassed data lines. If you decide to change the default escape sequence, be sure to choose a unique sequence so that the system software does not interpret characters intended for the RMC.

In snoop mode the RMC is responsible for configuring the modem for dial-in as well as dial-out alerts and for monitoring the modem connectivity. Because data passes directly between the two UART ports, snoop mode is useful when you want to monitor the system but also ensure optimum COM1 performance.

Soft Bypass Mode

In soft bypass mode all data and control signals are routed directly between the system COM1 port and the external modem port, and the RMC does not listen to the traffic on the COM1 data lines. The RMC is responsible for configuring the modem and monitoring the modem connectivity. If the RMC detects loss of carrier or the system loses power, it switches automatically into snoop mode. If you have set up the dial-out alert feature, the RMC pages the operator if an alert is detected and the modem line is not in use.

Soft bypass mode is useful if management applications need the COM1 channel to perform a binary download, because it ensures that RMC does not accidentally interpret some binary data as the escape sequence. After downloading binary files, you can set the **com1_mode** environment variable from the SRM console to switch back to snoop mode or other modes for accessing the RMC, or you can hang up the current modem session and reconnect it.

Firm Bypass Mode

In firm bypass mode all data and control signals are routed directly between the system COM1 port and the external modem port. The RMC does not configure or monitor the modem. Firm bypass mode is useful if you want the system, not the RMC, to fully control the modem port and you want to disable RMC remote management features such as remote dial-in and dial-out alert. You can switch to other modes by resetting the **com1_mode** environment variable from the SRM console, but you must then set up the RMC again from the local terminal.

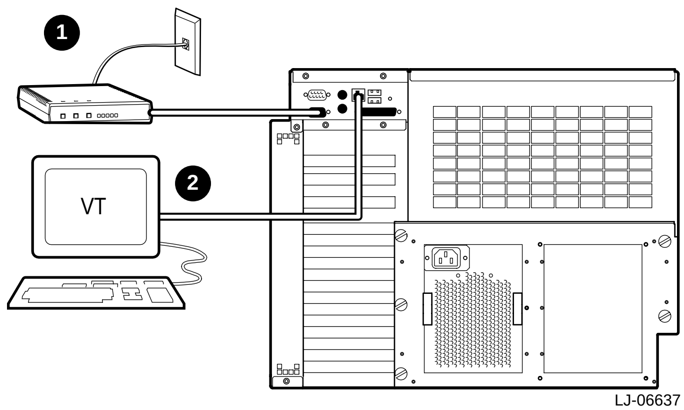
4.4 Terminal Setup

The RMC can be accessed through a modem hookup or from the serial terminal connected to the system.

The RMC connections are shown in Figure 4-3.

- For remote monitoring, connect a modem to the remote console modem port ❶.
- For local monitoring connect a serial terminal to the local COM1 serial port/terminal port (MMJ) ❷.

Figure 4-3 Setup for RMC



4.5 Entering the RMC

An escape sequence is used to invoke the remote management console. The remote management console can be accessed through a modem, from a local serial console terminal, from a local VGA monitor, or from the system. The "system" includes the operating system, the SRM console, or an application.

- You can enter the RMC from the local terminal regardless of the current operating mode
- You can enter the RMC from the modem if the RMC is in through mode, snoop mode, or local mode. In snoop mode, the escape sequence is passed to the system and displayed.

Note

Only one RMC session can be active at a time.

4.5.1 Entering RMC from a Local Serial Console Terminal

Invoke the RMC from a serial terminal by typing the following default escape sequence:

```
P00>>> ^[^[ rmc
```

This sequence is equivalent to typing Ctrl/left bracket, Ctrl/left bracket, rmc. On some keyboards, the Esc key functions like the Ctrl/left bracket combination.

To exit, enter the **quit** command. This action returns you to whatever you were doing before you entered RMC. In the following example, the **quit** command returns the user to the system COM1 port.

```
RMC> quit
Returning to COM port
```

4.5.2 Entering RMC from a Local VGA Monitor

To enter RMC from a local VGA monitor, the **console** environment variable must be set to **graphics**. Invoke the SRM console and enter the **rmc** command.

```
P00>>> rmc
You are about to connect to the Remote Management Console.  Use the RMC
reset command or press the front panel reset button to disconnect and
to reload the SRM console.
Do you really want to continue? [y/(n)] y
Please enter the escape sequence to connect to the Remote Management
Console.
```

After entering the escape sequence, the system enters the CLI and the RMC> prompt is displayed.

To exit RMC mode, use the **reset** command, which reinitializes RMC and SRM firmware.

```
RMC> reset

Returning to COM port
```

4.6 SRM Environment Variables for COM1

Several SRM environment variables allow you to set up the COM1 serial port (MMJ) for use with the RMC.

You may need to set the following environment variables from the SRM console, depending on how you decide to set up the RMC.

com1_baud	Sets the baud rate of the COM1 serial port and the modem port. The default is 9600.
com1_flow	Specifies the flow control on the serial port. The default is software.
com1_mode	Specifies the COM1 data flow paths so that data either flows through the RMC or bypasses it. This environment variable can be set from either the SRM or the RMC.
com1_modem	Specifies to the operating system whether or not a modem is present.

See the Compaq *AlphaServer ES40 User Interface Guide* for information on setting SRM environment variables.

4.7 RMC Command-Line Interface

The remote management console supports setup commands and commands for managing the system.

The RMC commands are listed below.

clear {alert, port}
deposit
disable {alert, remote}
dump
enable {alert, remote}
env
halt {in, out}
hangup
help or ?
power {on, off}
quit
reset
send alert
set {alert, com1_mode, dial, escape, init, logout, password, user}
status

Note

The **deposit** and **dump** commands are reserved for service providers.

For an RMC commands reference, see the Compaq *AlphaServer ES40 User Interface Guide*.

Command Conventions

Observe the following conventions for entering RMC commands:

- Enter enough characters to distinguish the command.

Note

The **reset** and **quit** commands are exceptions. You must enter the entire string for these commands to work.

- For commands consisting of two words, enter the entire first word and at least one letter of the second word. For example, you can enter disable a for disable alert.
- For commands that have parameters, you are prompted for the parameter.
- Use the Backspace key to erase input.
- If you enter a nonexistent command or a command that does not follow conventions, the following message is displayed:

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

- If you enter a string that exceeds 14 characters, the following message is displayed:

```
*** ERROR - overflow ***
```

4.7.1 Defining the COM1 Data Flow

Use the **set com1_mode** command from SRM or RMC to define the COM1 data flow paths.

You can set **com1_mode** to one of the following values:

through	All data passes through RMC and is filtered for the escape sequence. This is the default.
snoop	Data partially bypasses RMC, but RMC taps into the data lines and listens passively for the escape sequence.
soft_bypass	Data bypasses RMC, but RMC switches automatically into snoop mode if loss of carrier occurs.
firm_bypass	Data bypasses RMC. RMC remote management features are disabled.
local	Changes the focus of the COM1 traffic to the local MMJ port if RMC is currently in one of the bypass modes or is in through mode with an active remote session.

Example

```
RMC> set com1_mode
Com1_mode (THROUGH, SNOOP, SOFT_BYPASS, FIRM_BYPASS, LOCAL): local
```

For more details, see the Compaq *AlphaServer ES40 User Interface Guide*.

4.7.2 Displaying the System Status

The RMC **status** command displays the current RMC settings. Table 4-1 explains the status fields.

```
RMC>status
   PLATFORM STATUS
On-Chip Firmware Revision: V1.0
Flash Firmware Revision: V1.2
Server Power: ON
System Halt: Deasserted
RMC Power Control: ON
Escape Sequence: ^[^[RMC
Remote Access: Enabled
Modem Password: set
Alert Enable: Disabled
Alert Pending: YES
Init String: AT&F0E0V0X0S0=2
Dial String: ATXDT9, 15085553333
Alert String: ,,,,,,5085553332#;
Com1_mode: THROUGH
Last Alert:
Logout Timer: 20 minutes
User String:
```

Table 4-1 Status Command Fields

Field	Meaning
On-Chip Firmware Revision:	Revision of RMC firmware on the microcontroller.
Flash Firmware Revision:	Revision of RMC firmware in flash ROM.
Server Power:	On =System is on. Off = System is off..
System Halt:	Asserted = System has been halted. Deasserted = Halt has been released.
RMC Power Control:	On= System has powered on from RMC. Off = System has powered off from RMC..
Escape Sequence:	Current escape sequence for access to RMC console.
Remote Access:	Enable = Modem for remote access is enabled. Disable = Modem for remote access is disabled.
RMC Password:	Set = Password set for modem access. Not set = No password set for modem access.
Alert Enable:	Enabled = Dial-out enabled for sending alerts. Disabled = Dial-out disabled for sending alerts..
Alert Pending:	Yes = Alert has been triggered. No = No alert has been triggered..
Init String:	Initialization string that was set for modem.
Dial String:	Pager string to be dialed when an alert occurs.
Alert String:	Identifies the system that triggered the alert to the paging service. Usually the phone number of the monitored system.
Com1_mode:	Identifies the current COM1 mode..
Last Alert:	Type of alert (for example, fan problem).
Logout Timer:	The amount of time before the RMC terminates an inactive modem connection. The default is 20 minutes.
User String:	Notes supplied by user.

4.7.3 Displaying the System Environment

Use the RMC **env** command to get a snapshot of the system environment, including the state of CPUs, fans, and power supplies.

```
RMC>env

System Hardware Monitor
Temperature (warnings at 45.0°C, power-off at 50.0°C)
    CPU0: 26.0°C    CPU1: 26.0°C    CPU2: 27.0°C    CPU3: 26.0°C        ❶
    Zone0: 29.0°C    Zone1: 30.0°C    Zone2: 31.0°C                     ❷
Fans RPM
    Fan1: 2295    Fan2: 2295    Fan3: 2205                               ❸
    Fan4: 2235    Fan5: OFF     Fan6: 2518
Power Supplies(OK, FAIL, OFF, '----' means not present)                  ❹
    PS0 : OK       PS1 : OK       PS2 : ----
    CPU0: OK       CPU1: OK      CPU2: OK      CPU3: OK
CPU CORE voltage
    CPU0: +2.192V   CPU1: +2.192V   CPU2: +2.192V   CPU3: +2.192V        ❺
CPU IO voltage
    CPU0: +1.488V   CPU1: +1.488V   CPU2: +1.488V   CPU3: +1.488V
Bulk voltage
    +3.3V Bulk: +3.328V   +5V Bulk: +5.076V   +12V Bulk: +12.096V        ❻
         Vterm: +1.824V      Cterm: +2.000V   -12V Bulk: -12.480V
```

❶ CPU temperature. In this example four CPUs are present.

❷ Temperature of PCI backplane area: Zone 1 includes PCI Slots 1–3, Zone 2 includes PCI Slots 7–10, and Zone 3 includes PCI Slots 4–6.

❸ Fan RPM. With the exception of Fan 5, all fans operate as long as the system is powered on. Fan 5 is OFF unless Fan 6 fails.

❹ The normal power supply mode is either OK (system is powered on) or OFF (system is powered off or the power supply cord is not plugged in). FAIL indicates a problem with a supply.

❺ CPU CORE voltage and CPU I/O voltage. In a healthy system, the readings for all CPUs should be the same.

❻ Bulk power supply voltage.

4.7.4 Power On and Off, Reset, and Halt

The RMC power {on, off}, halt {in, out}, and reset commands perform the same functions as the buttons on the operator control panel.

Power On and Power Off

The RMC **power on** command powers the system on, and the **power off** command powers the system off. The Power button on the OCP, however, has precedence.

- If the system has been powered off with the Power button, the RMC cannot power the system on. If you enter the power on command, the message “Power button is OFF” is displayed, indicating that the command will have no effect.
- If the system has been powered on with the Power button, and the **power off** command is used to turn the system off, you can toggle the Power button to power the system back on.

When you issue the **power on** command, the terminal exits RMC and reconnects to the server’s COM1 port.

```
RMC> power on
Returning to COM port
RMC> power off
```

Halt In and Halt Out

The **halt in** command halts the system. The **halt out** command releases the halt. When you issue either the **halt in** or **halt out** command, the terminal exits RMC and reconnects to the server’s COM1 port.

```
RMC> halt in
Returning to COM port
RMC> halt out
Returning to COM port
```

The **halt out** command cannot release the halt if the Halt button is latched in. If you enter the **halt out** command, the message “Halt button is IN” is displayed, indicating that the command will have no effect. Toggling the Power button on the operator control panel overrides the **halt in** condition.

Reset

The RMC **reset** command restarts the system. The terminal exits RMC and reconnects to the server’s COM1 port.

```
RMC> reset
Returning to COM port
```

4.7.5 Configuring Remote Dial-In

Before you can dial in through the RMC modem port or enable the system to call out in response to system alerts, you must configure RMC for remote dial-in.

Connect your modem to the 9-pin modem port and turn it on. Enter the RMC from either the local serial terminal or the local VGA monitor to set up the parameters.

Example 4-1 Dial-In Configuration

```
RMC> set password                    ❶
RMC Password: ****
Verification: ****
RMC> set init                        ❷
Init String: AT&F0E0V0X0S0=2
RMC> enable remote                   ❸
RMC> status                          ❹
.
.
Remote Access: Enabled
.
.
.
```

Note

The following modems require the initialization strings shown here. For other modems, see your modem documentation.

Modem	Initialization String
Motorola 3400 Lifestyle 28.8	AT&F0E0V0X0S0=2
AT &T Dataport 14.4/FAX	AT&F0E0V0X0S0=2
Hayes Smartmodem Optima 288 V-34/V.FC + FAX	AT&FE0V0X0S0=2

❶ Sets the password that is prompted for at the beginning of a modem session. The string cannot exceed 14 characters and is not case sensitive. For security, the password is not echoed on the screen. When prompted for verification, type the password again.

❷ Sets the initialization string. The string is limited to 31 characters and can be modified depending on the type of modem used. Because the modem commands disallow mixed cases, the RMC automatically converts all alphabetic characters entered in the init string to uppercase.

The RMC automatically configures the modem's flow control according to the setting of the SRM **com1_flow** environment variable. The RMC also enables the modem carrier detect feature to monitor the modem connectivity.

❸ Enables remote access to the RMC modem port by configuring the modem with the setting stored in the initialization string.

❹ Verifies the settings. Check that the Remote Access field is set to Enabled.

Dialing In

The following example shows the screen output when a modem connection is established.

```
ATDT915085553333

RINGING

RINGING

CONNECT 9600/ARQ/V32/LAPM

RMC Password: *********

Welcome to RMC V1.2
P00>>> ^[^[rmc
RMC>
```

At the RMC> prompt, enter commands to monitor and control the remote system.

When you have finished a modem session, enter the **hangup** command to cleanly terminate the session and disconnect from the server.

4.7.6 Configuring Dial-Out Alert

When the system is not being monitored through the modem, the RMC dial-out alert feature can be used to remain informed of system status. With dial-out alert is enabled, and the RMC detects alarm conditions within the managed system, it can call a preset pager number.

You must configure remote dial-in for the dial-out feature to be enabled. See Section 4.7.5.

To set up the dial-out alert feature, enter the RMC from the local serial terminal or local VGA monitor.

Example 4-2 Dial-Out Alert Configuration

```
RMC> set dial                                ❶
Dial String: ATXDT9,15085553333
RMC> set alert                               ❷
Alert String: ,,,,,,5085553332#;
RMC> enable alert                            ❸
RMC> clear alert                             ❹
RMC> send alert                              ❺
Alert detected!
RMC> clear alert                             ❻
RMC> status                                  ❼
.
.
Alert Enable: Enabled
.
.
```

A typical alert situation might be as follows:

- The RMC detects an alarm condition, such as over temperature warning.
- The RMC dials your pager and sends a message identifying the system.
- You dial the system from a remote serial terminal.
- You enter the RMC, check system status with the **env** command, and, if the situation requires, power down the managed system.
- When the problem is resolved, you power up and reboot the system.

The elements of the dial string and alert string are shown in Table 4-2. Paging services vary, so you need to become familiar with the options provided by the paging service you will be using. The RMC supports only numeric messages.

❶ Sets the string to be used by the RMC to dial out when an alert condition occurs. The dial string must include the appropriate modem commands to dial the number.

❷ Sets the alert string, typically the phone number of the modem connected to the remote system. The alert string is appended after the dial string, and the combined string is sent to the modem when an alert condition is detected.

❸ Enables the RMC to page a remote system operator.

❹ Clears any alert that may be pending. This ensures that the send alert command will generate an alert condition.

❺ Forces an alert condition. This command is used to test the setup of the dial-out alert function. It should be issued from the local serial terminal or local VGA monitor. As long as no one connects to the modem and there is no alert pending, the alert will be sent to the pager immediately. If the pager does not receive the alert, re-check your setup.

❻ Clears the current alert so that the RMC can capture a new alert. The last alert is stored until a new event overwrites it. The Alert Pending field of the status command becomes NO after the alert is cleared.

❼ Verifies the settings. Check that the Alert Enable field is set to Enabled.

Note

If you do not want dial-out paging enabled at this time, enter the **disable alert** command after you have tested the dial-out alert function. Alerts continue to be logged, but no paging occurs.

Table 4-2 Elements of the Dial and Alert Strings

Dial String	Description
	The dial string is case sensitive. The RMC automatically converts all alphabetic characters to uppercase.
ATXDT	AT = Attention
	X = Forces the modem to dial “blindly” (not seek the dial tone). Enter this character if the dial-out line modifies its dial tone when used for services such as voice mail.
	D = Dial
	T = Tone (for touch-tone)
	, = Pause for 2 seconds.
9,	The number for an outside line (in this example, 9). Enter the number for an outside line if your system requires it.
15085553333	Phone number of the paging service.
Alert String	**Description**
,,,,,,	Pause for 12 seconds for paging service to answer. Each comma (,) provides a 2-second delay.
5085553332#	A call-back number for the paging service. The alert string must be terminated by the pound (#) character.
;	A semicolon (;) must be used to terminate the alert string.

4.7.7 Resetting the Escape Sequence

The RMC **set escape** command sets a new escape sequence. The new escape sequence can be any character string, not to exceed 14 characters. A typical sequence consists of two or more control characters. It is recommended that control characters be used in preference to ASCII characters. Use the **status** command to verify the new escape sequence before exiting the RMC.

The following example consists of two instances of the Esc key and the letters "FUN." The "F" is not displayed when you set the sequence because it is preceded by the escape character. Enter the **status** command to see the new escape sequence.

```
RMC> set escape
Escape Sequence: un
RMC> status
.
.
.
Escape Sequence: ^[^[FUN
```

Caution

Be sure to record the new escape sequence. Restoring the default sequence requires moving a jumper on the system motherboard.

4.8 Resetting the RMC to Factory Defaults

If the RMC escape sequence is set to something other than the default, and you have forgotten the sequence, RMC must be reset to its factory settings to restore the default escape sequence.

WARNING

The jumpers should only be set by service personnel who have the appropriate technical training and experience. Installers should understand the hazards of working within the system and take measures to minimize danger to themselves or other personnel.

Perform the following procedure to restore the default settings:

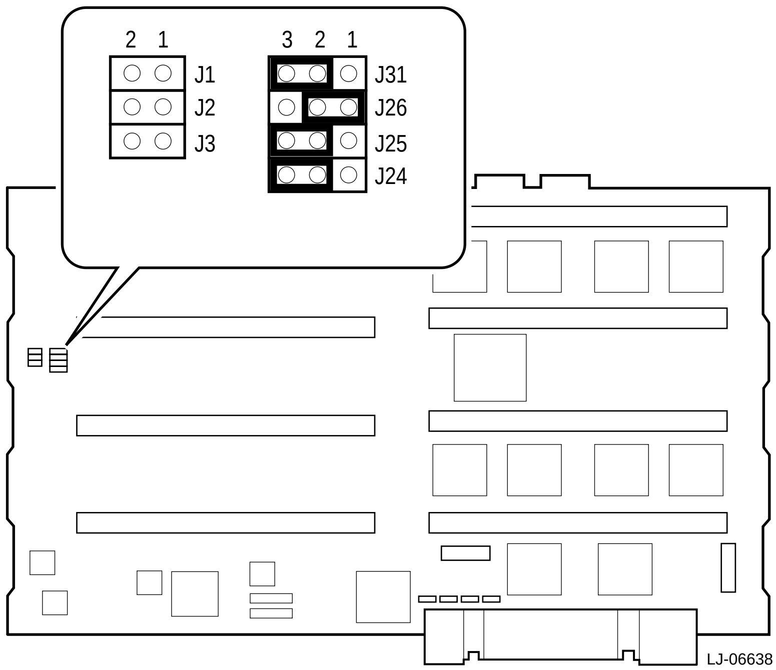
1. Perform a normal shutdown of the operating system and place the ON/Off button on the OCP to the Off position.
2. Disconnect the power cord(s) from the power supplies. Wait until the +5V Aux LEDs on the power supplies go off before proceeding.
3. Extend the chassis for service and remove the top cover as described in Chapter 5.
4. Remove the CPU module in the CPU1 slot as described in Chapter 5.
5. On the system motherboard, install jumper J25 over pins 1 and 2 (see Figure 4-4).
6. Reconnect the power cord to one of the power supplies, and then wait until the control panel displays the message "System is down."
7. Disconnect the power cord from the power supply. Wait until the +5V Aux LED on the power supply go off before proceeding.

8. Install jumper J25 over pins 2 and 3 to enable RMC mode.
9. Reinstall the CPU module in the CPU1 slot and the top cover.
10. Reconnect the power cord(s) to the power supplies.

Note

After the RMC has been reset to defaults, perform the setup procedures to enable remote dial-in and call-out alerts. See Section 4.7.5.

Figure 4-4 RMC Jumpers (Default Positions)



4.9 RMC Troubleshooting Tips

Table 4-3 lists some possible causes and suggested solutions for symptoms that might occur.

Table 4-3 RMC Troubleshooting Tips

Symptom	Possible Cause	Suggested Solution
You cannot enter the RMC from the modem.	The RMC may be in soft bypass or firm bypass mode.	Issue the show com1_mode command from SRM and change the setting if necessary. If in soft bypass mode, you can disconnect the modem session and reconnect it.
The local terminal cannot communicate with the RMC correctly.	System and terminal baud rates do not match.	Set the baud rates for the system and terminal to be the same. For first-time setup, suspect the console terminal, since the RMC and system default baud is 9600.
RMC will not answer when the modem is called.	Modem cables may be incorrectly installed.	Check modem phone lines and connections.
	RMC remote access is disabled or the modem was power cycled since last being initialized.	From the local serial terminal or VGA monitor, enter the **set password** and **set init** commands, and then enter the **enable remote** command..
	The modem is not configured correctly.	Modify the modem initialization string according to your modem documentation.
	On power-up, RMC defers initializing the modem for 30 seconds to allow the modem to complete its internal diagnostics and initializations.	Wait 30 seconds after powering up the system and RMC before attempting to dial in.
After the system is powered up, the COM1 port seems to hang or you seem to be unable to execute RMC commands.	There is a normal delay while the RMC completes the system power-on sequence.	Wait about 40 seconds.
New escape sequence is forgotten.		RMC console must be reset to factory defaults.
During a remote connection, the user sees a “+++” string on the screen.	The modem is confirming whether the modem has really lost carrier.	This is normal behavior.
The message “unknown command” is displayed when the user enters a carriage return by itself.	The terminal or terminal emulator is including a line feed character with the carriage return.	Change the terminal or terminal emulator setting so that “new line” is not selected.

5
Removal and Replacement

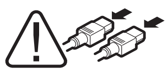
5.1 Introduction

This chapter contains the procedures for removing and replacing the following major components of the *AlphaServer* ES40sv rackmount system:

- Extending the Chassis for Service
- Front Panel
- Top Cover
- PCI Access Cover
- Removable Media Cage
- Hard Disk Drive Carrier
- Hard Disk Drive
- Receiver Frame
- Floppy Drive
- Speaker
- Power Supply
- OCP Assembly
- CPU Modules
- Memory Motherboards
- Memory DIMMs
- CPU Fan Assembly
- Front or Rear EMI/RFI Honeycomb Filter
- PCI Card-Cage Fan Assembly
- PCI Option Modules
- I/O Port Module
- PCI Backplane
- System Motherboard

WARNING

Before servicing the *AlphaServer* ES40sv rackmount system chassis or performing any of the following removal and replacement procedures, ensure that the On/Off button on the OCP is in the OFF position and disconnect the power cord(s) from the power supply ac input receptacle(s) on the rear of the chassis.



WARNING: To prevent injury, unplug the power cord from each power supply before removing and replacing components.

WARNING

Only a qualified service person should remove and replace components in the *AlphaServer* ES40sv rackmount system. A qualified service person should have the technical training and experience necessary to be aware of the hazards to which they are exposed in performing a task and the measures that should be taken to minimize the danger to themselves or other persons.

Tools Required

The following tools are required for servicing the *AlphaServer* ES40sv rackmount system chassis:

- Phillips-head and flat-blade screwdriver
- Adjustable wrench

5.2 Extending the Chassis for Service

WARNING

Before extending the chassis for service, ensure that the cabinet is stable and that all provided stabilizing features have been activated. The stabilizing features for the rack or cabinet are configuration dependent.

Perform the following procedure to extend *AlphaServer* ES40sv rackmount system chassis for service:

1. Extend the stabilizing legs at the front of the cabinet (if stabilizing legs are provided).
2. Loosen the six captive screws ❶ that secure the front panel to the front rails (see Figure 5-1).

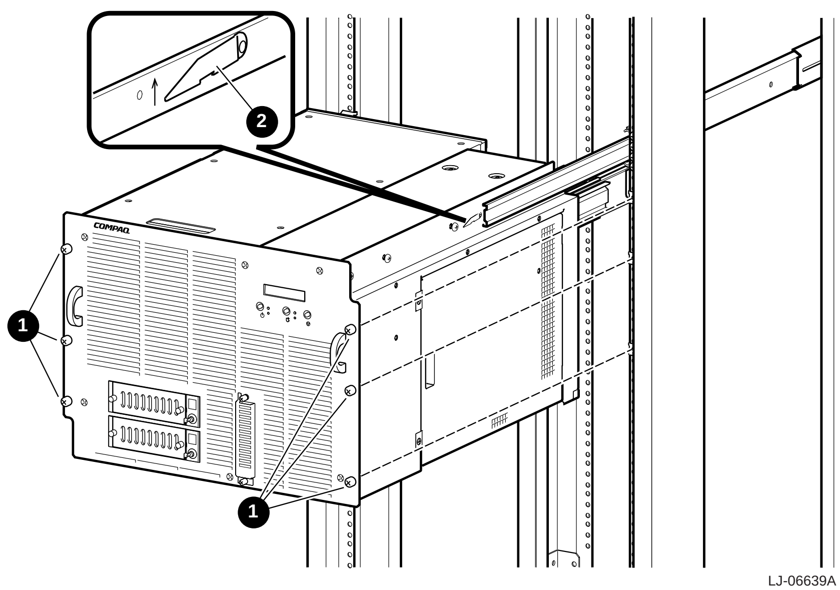
Caution

Check and ensure that all cables are free to follow the chassis before extending the chassis.

3. Carefully pull the chassis forward until the slides lock in the extended position.

To secure the chassis in the cabinet, press up on the left and right slide locks ❷, and reverse steps 1 through 3 of the extension procedure.

Figure 5-1 Extending the Chassis for Service



5.3 Front Panel

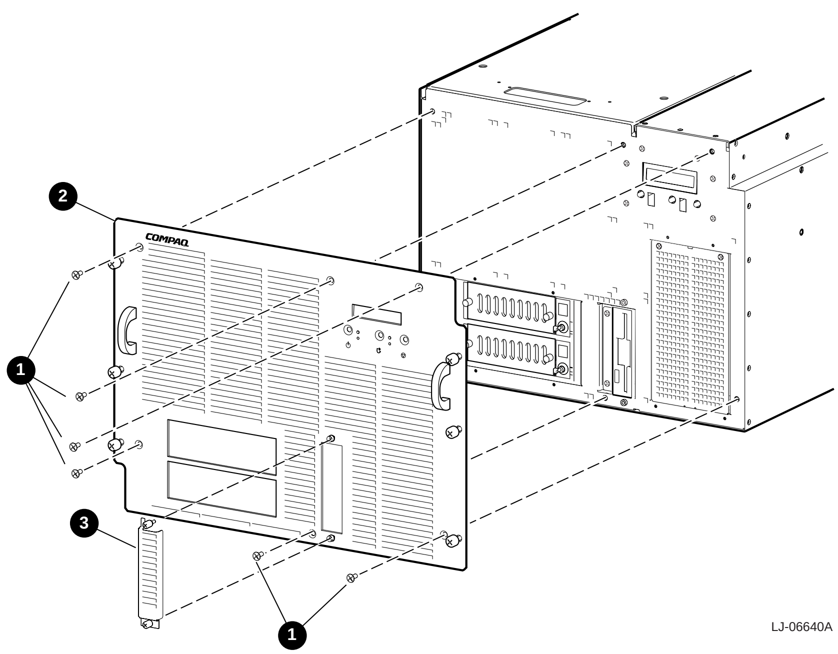
Perform the following procedure to remove the front panel from the *AlphaServer* ES40sv rackmount system chassis:

1. Extend the chassis for service (see Section 5.2).
2. Remove the six 10-32 flat-head screws ❶ that secure the front panel ❷ to the front of the chassis (see Figure 5-2).
3. Remove the two screws that secure the floppy drive cover plate ❸ and remove the cover plate.
4. Pull the front panel away from the chassis.

To replace the front panel, refer to Figure 5-2 and proceed as follows:

1. Place the front panel on the front of the chassis.
2. Secure the front panel to the front of the chassis with six 10-32 flat-head screws.
3. Secure the floppy drive cover plate with the two screws.

Figure 5-2 Removing and Replacing the Front Panel



5.4 Top Cover

Caution

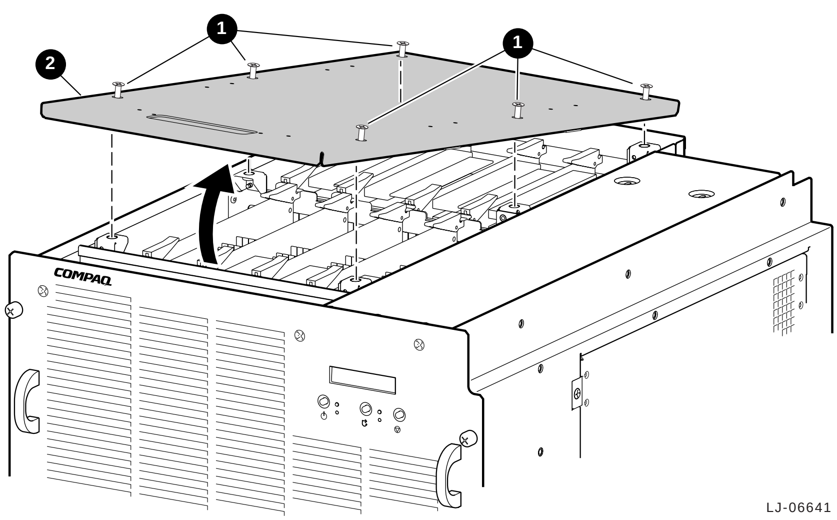
Use care when removing and replacing the cover to prevent damage to the RFI gaskets that are located around the cover and opening.

Perform the following procedure to remove the top cover:

1. Extend the chassis for service (see Section 5.2).
2. Loosen the six captive screws ❶ on the top of the top cover ❷ (see Figure 5-3).
3. Grasp the recessed hand hold along the front edge of the cover and lift the top cover off of the chassis.

To replace the top cover, refer to Figure 5-3 and reverse steps 1 through 3 of the removal procedure.

Figure 5-3 Removing and Replacing the Top Cover



5.5 PCI Access Cover

Caution

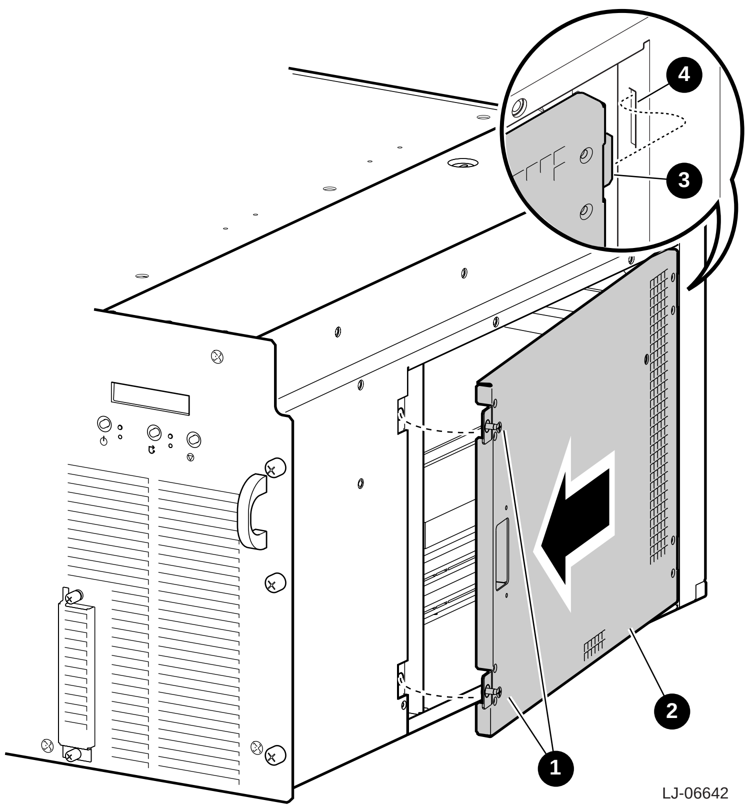
Use care when removing and replacing the cover to prevent damage to the RFI gaskets that are located around the four sides of the cover.

Perform the following procedure to remove the PCI access cover:

1. Extend the chassis for service (see Section 5.2).
2. Loosen the two captive screws ❶ along the front edge of the PCI access cover ❷ (see Figure 5-4).
3. Swing the front of the PCI access cover out from the chassis (see Figure 5-4).
4. Pull the PCI access cover forward until the tabs ❸ along the rear edge come out of the slots ❹ in the chassis, and then pull the PCI access cover out and away from the chassis.

To replace the PCI access cover, refer to Figure 5-4 and reverse steps 1 through 4 of the removal procedure.

Figure 5-4 Removing and Replacing the PCI Access Cover



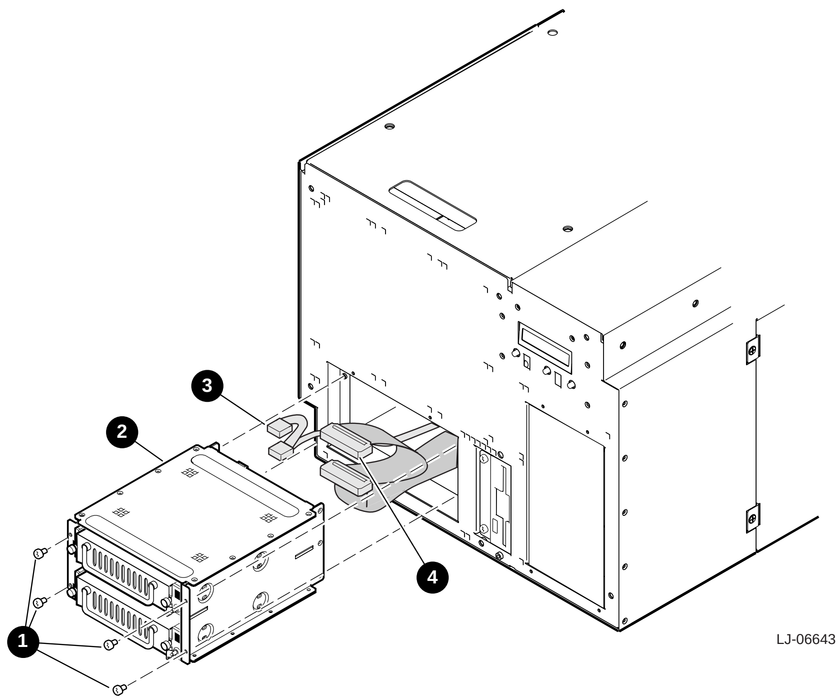
5.6 Removable Media Cage

Perform the following procedure to remove the removable media cage:

1. Extend the chassis for service (see Section 5.2).
2. Remove the front panel (see Section 5.3).
3. Remove the four screws ❶ that secure the removable media cage ❷ to the front of the chassis (see Figure 5-5).
4. Slide the removable media cage out of the chassis until the device power and data cables can be accessed.
5. Disconnect the power ❸ and data ❹ cables from the receiver frames.

To replace the removable media cage, refer to Figure 5-5 and reverse steps 1 through 5 of the removal procedure.

Figure 5-5 Removing and Replacing the Removable Media Cage



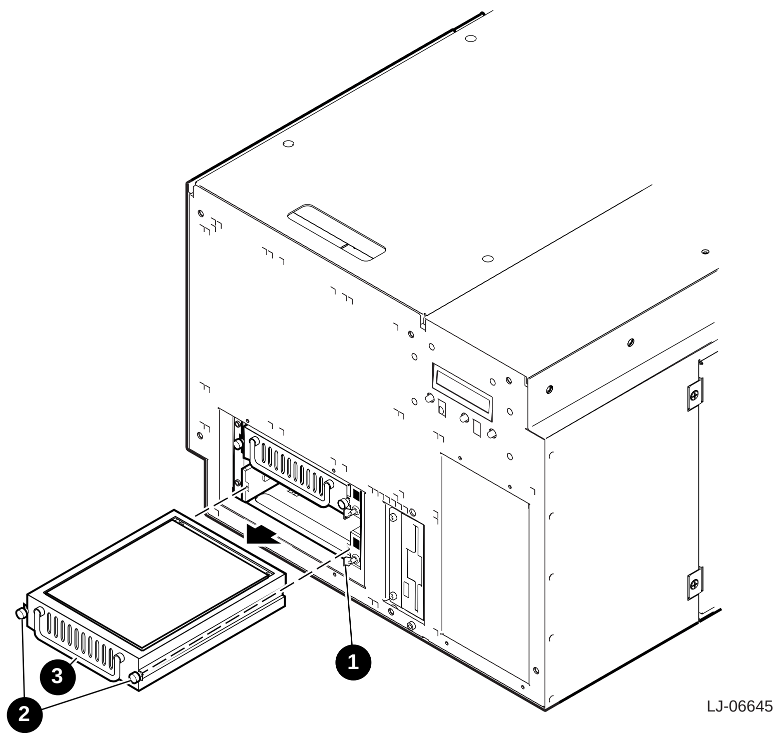
5.7 Hard Disk Drive Carrier

Perform the following procedure to remove the hard disk drive carrier:

1. Turn the key ❶ clockwise to unlock and power down the hard disk drive carrier that is being removed. **Wait until the unit display number on the front of the receiver frame stops flashing.**
2. Loosen the two captive screws ❷ that secure the hard disk drive carrier to the receiver frame (see Figure 5-6).
3. Grasp the hard disk drive carrier handle ❸ and pull the carrier out of the receiver frame (see Figure 5-6).

To replace the hard disk drive carrier, refer to Figure 5-6and reverse steps 1 through 3 of the removal procedure.

Figure 5-6 Removing and Replacing the Hard Disk Drive Carrier



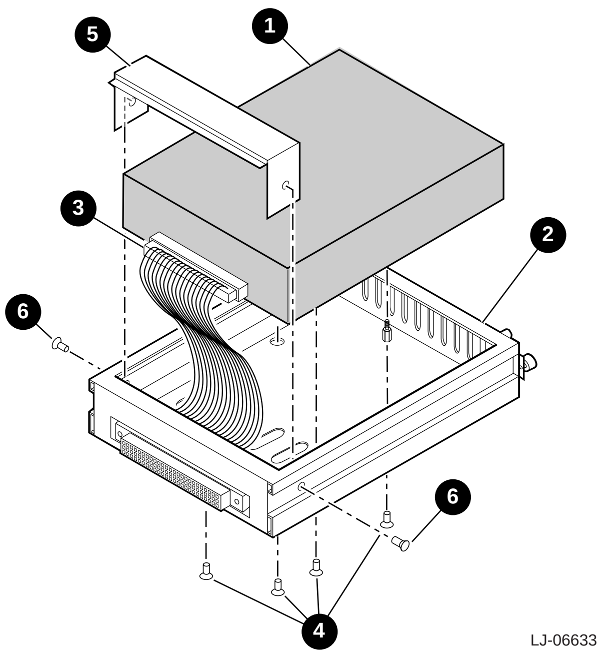
5.8 Hard Disk Drive

Perform the following procedure to remove a hard disk drive from a hard disk drive carrier:

1. Remove the hard disk drive carrier (see Section 5.7).
2. Remove the two screws ❻ that secure the cable cover ❺ to the carrier (see Figure 5-7).
3. Remove the four screws ❹ that secure the hard disk drive ❶ to the bottom of the carrier ❷ (see Figure 5-7).
4. Disconnect the 80-pin ribbon cable ❸ from the rear of the hard disk drive and lift the drive out of the carrier (see Figure 5-7).

To replace the hard disk drive in the carrier, refer to Figure 5-7 and reverse steps 1 through 4 of the removal procedure.

Figure 5-7 Removing and Replacing a Hard Disk Drive in a Carrier



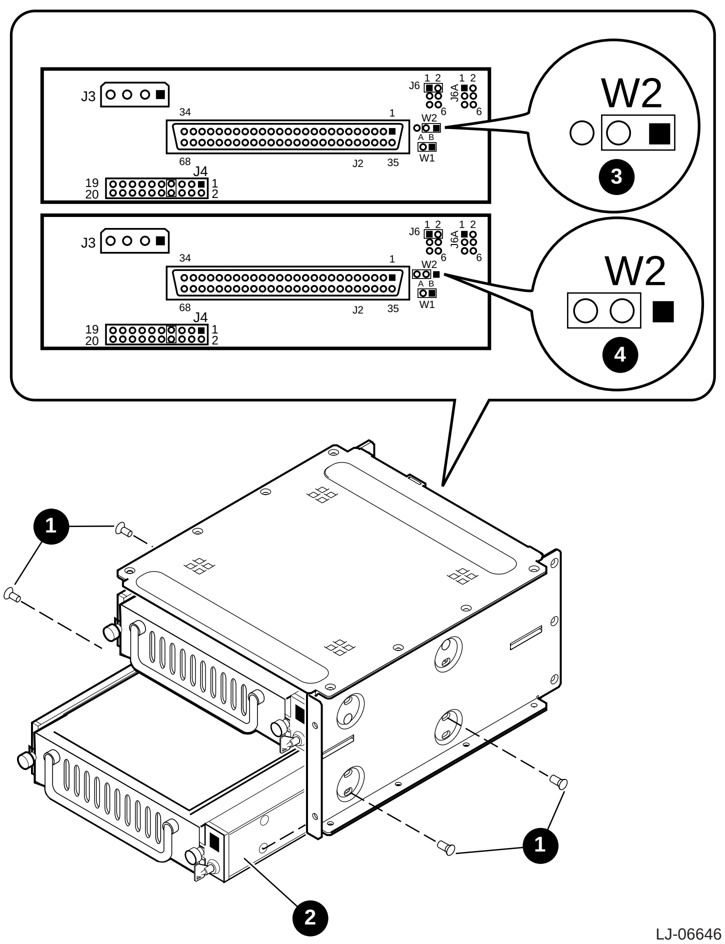
5.9 Receiver Frame

Perform the following procedure to remove the receiver frame:

1. Extend the chassis for service (see Section 5.2).
2. Remove the front panel (see Section 5.3).
3. Remove the hard disk drive carrier (see Section 5.7).
4. Remove the removable media cage (see Section 5.6).
5. Disconnect the data and power cables from the rear of the receiver frame.
6. Remove the four screws ❶ that secure the receiver frame ❷ to the removable media cage (see Figure 5-8).
7. Slide the receiver frame out of the opening in the front of the removable media cage.

To replace the receiver frame, refer to Figure 5-8 and reverse steps 1 through 7 of the removal procedure. Ensure that the new receiver frame ID jumper and termination jumper are set to match the ID jumper and termination jumper on the receiver frame being replaced. Figure 5-8 shows the on-board termination jumper in the enabled ❸ position and the disabled ❹ position.

Figure 5-8 Removing and Replacing the Receiver Frame



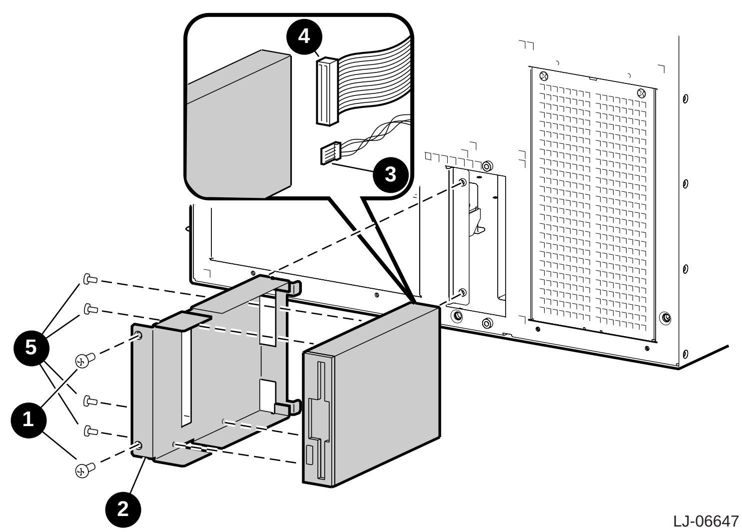
5.10 Floppy Drive

Perform the following procedure to remove the floppy drive:

1. Extend the chassis for service (see Section 5.2).
2. Remove the front panel (see Section 5.3).
3. Remove the two screws ❶ that secure the floppy drive bracket ❷ to the front of the chassis (see Figure 5-9).
4. Slide the floppy drive bracket out of the opening in the front of the chassis until the power and data cables can be accessed.
5. Disconnect the power ❸ and data ❹ cables from the floppy drive (see Figure 5-9).
6. Remove the four screws ❺ that secure the floppy drive to the bracket (see Figure 5-9).

To replace the floppy drive, refer to Figure 5-9 and reverse steps 1 through 6 of the removal procedure.

Figure 5-9 Removing and Replacing the Floppy Drive



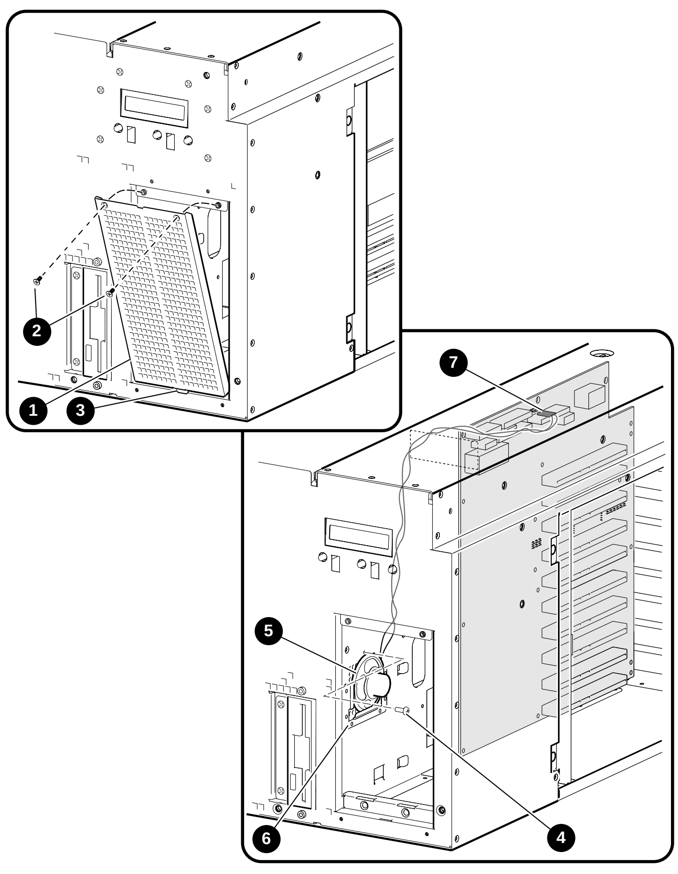
5.11 Speaker

Perform the following procedure to remove the speaker:

1. Extend the chassis for service (see Section 5.2).
2. Remove the front panel (see Section 5.3).
3. Remove the PCI access cover (see Section 5.5).
4. Remove the hard disk cover ❶ by removing the two screws ❷ that secure the top of the hard disk cover and prying the top of the cover out. Lift the hard disk cover until the metal tab ❸ on the bottom edge comes out of the slot and remove the cover from the chassis.
5. Remove the screw ❹ above the speaker ❺ that secures the speaker in the bracket ❻ (see Figure 5-10).
6. Disconnect the speaker cable ❼ from connector J12 on the PCI backplane.
7. While guiding the speaker cable through the opening in the front wall of the PCI card-cage area, slide the speaker up and out of the bracket and out of the chassis through the hard disk cover opening.

To replace the speaker, refer to Figure 5-10 and reverse steps 1 through 7 of the removal procedure.

Figure 5-10 Removing and Replacing the Speaker



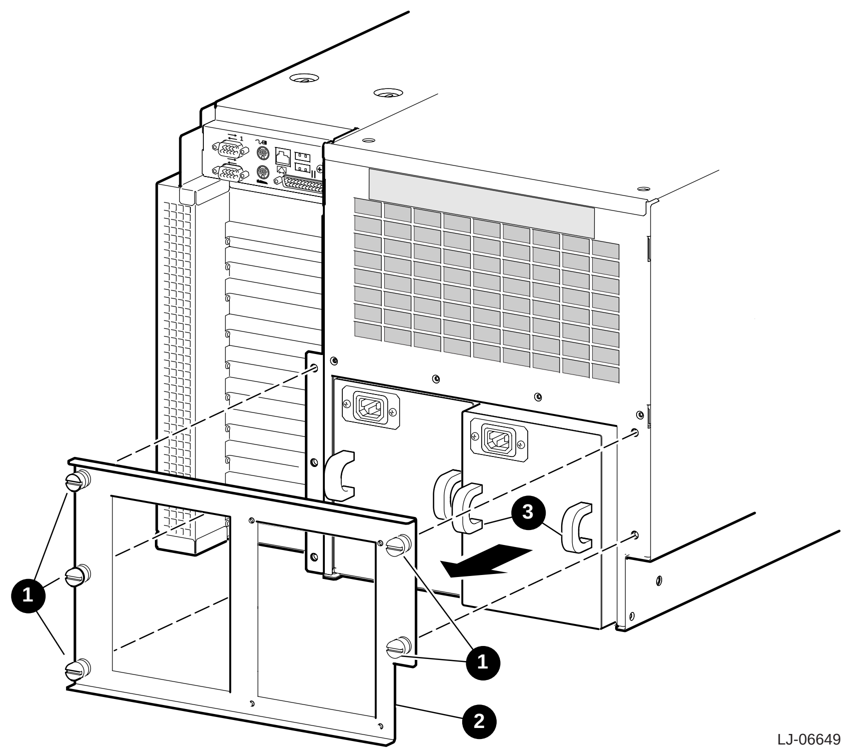
5.12 Power Supply

If there is only one power supply in the system, the system must be powered down before performing the removal and replacement procedures. If the system contains two power supplies, one power supply can be removed and replaced without powering down the system. Perform the following procedure to remove the power supply:

1. At the rear of the chassis, loosen the five captive screws ❶ that secure the power supply bracket ❷ to the chassis and remove the bracket (see Figure 5-11).
2. Grasp the two handles ❸ on the power supply and pull the power supply out of the chassis.

To replace the power supply, refer to Figure 5-11 and reverse steps 1 and 2 of the removal procedure.

Figure 5-11 Removing and Replacing the Power Supply



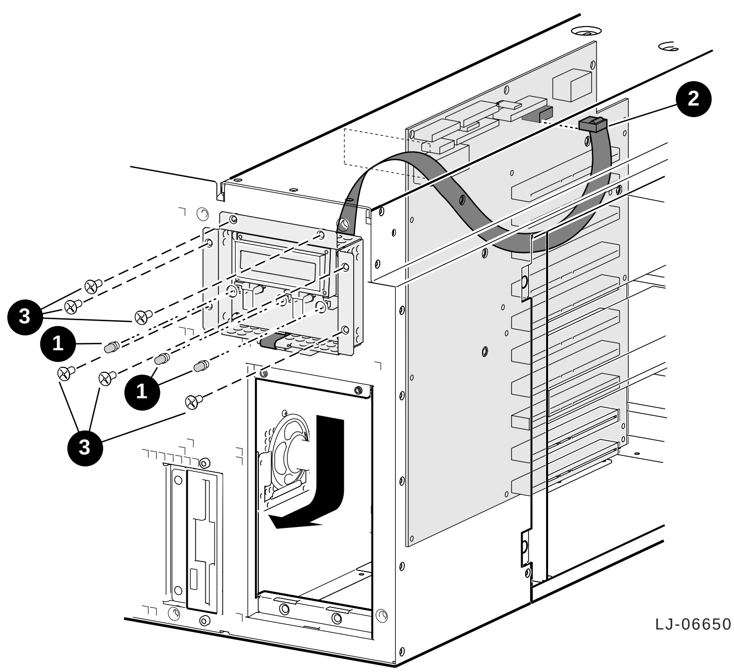
5.13 OCP Assembly

Perform the following procedure to remove the OCP assembly:

1. Extend the chassis for service (see Section 5.2).
2. Remove the front panel (see Section 5.3).
3. Remove the PCI access cover (see Section 5.5).
4. Remove the hard disk cover by removing the two screws that secure the top of the hard disk cover and prying the top of the cover out. Lift the hard disk cover until the metal tab on the bottom edge comes out of the slot and remove the cover from the chassis.
5. Remove the three push buttons ❶ from the OCP by pulling them out.
6. Disconnect the OCP assembly ribbon cable ❷ from connector J17 on the PCI backplane (see Figure 5-12).
7. While reaching through the hard disk cover opening and holding the OCP assembly, remove the six screws ❸ that secure the OCP assembly to the front of the chassis (see Figure 5-12).
8. While guiding the OCP assembly ribbon cable through the opening in the front wall of the PCI card-cage area, slide the OCP assembly to the right and then down and out of the hard disk cover opening in the front of the chassis.

To replace the OCP assembly, refer to Figure 5-12and reverse steps 1 through 8 of the removal procedure.

Figure 5-12 Removing and Replacing the OCP Assembly



5.14 CPU Modules

Perform the following procedure to remove a CPU module:

1. Extend the chassis for service (see Section 5.2).
2. Remove the top cover (see Section 5.4).
3. Put on an antistatic wriststrap.

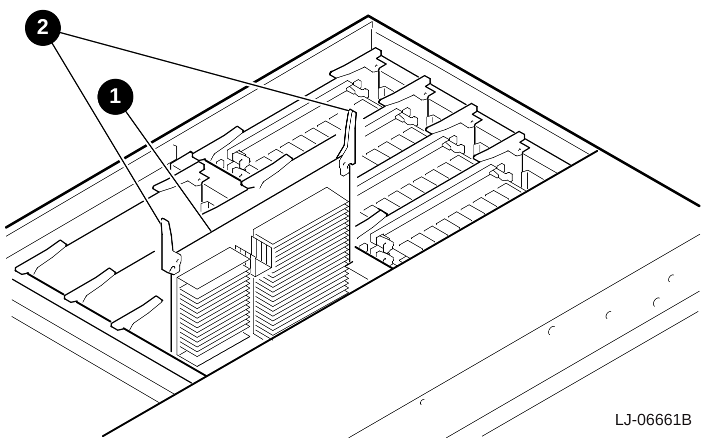
Caution

An antistatic wriststrap *must* be worn when handling any module to prevent damage to the module.

4. Pull up on the module levers ❷ located on each end of the CPU module ❶ being removed (see Figure 5-13).
5. Pull the CPU module straight up out of its slot (see Figure 5-13).

To replace the CPU module, refer to Figure 5-13 and reverse steps 1 through 5 of the removal procedure.

Figure 5-13 Removing and Replacing CPU Modules



5.15 Memory Motherboard (MMB)

Perform the following procedure to remove a memory motherboard:

1. Extend the chassis for service (see Section 5.2).
2. Remove the top cover (see Section 5.4).
3. Put on an antistatic wriststrap.

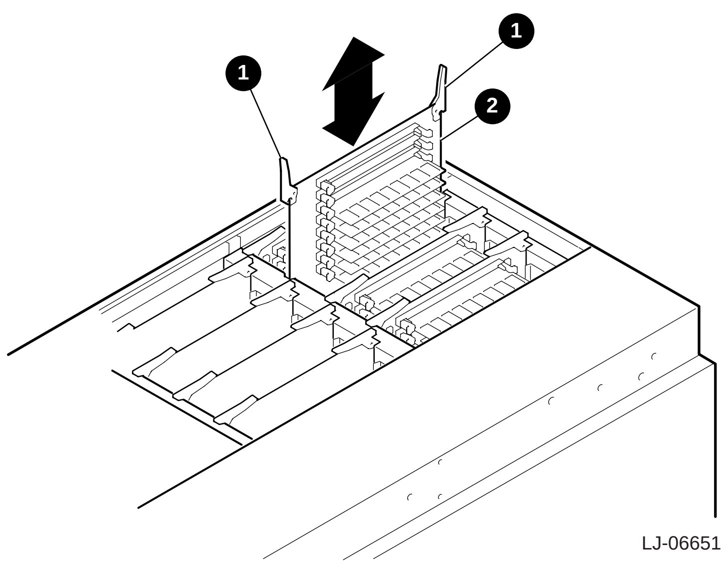
Caution

An antistatic wriststrap *must* be worn when handling any module to prevent damage to the module.

4. Pull up on the module levers ❶ located on each end of the MMB ❷ being removed (see Figure 5-14).
5. Pull the MMB straight up out of its slot (see Figure 5-14).

To replace the memory motherboard, refer to Figure 5-14 and reverse steps 1 through 5 of the removal procedure. Refer to Section 5.16 to remove the memory DIMMs from the MMB being replaced and install them on the new MMB.

Figure 5-14 Removing and Replacing a Memory Motherboard (MMB)



5.16 Memory DIMMs

Perform the following procedure to remove memory DIMMs:

1. Extend the chassis for service (see Section 5.2).
2. Remove the top cover (see Section 5.4).
3. Remove the memory motherboard (see Section 5.15).
4. Put on an antistatic wriststrap.

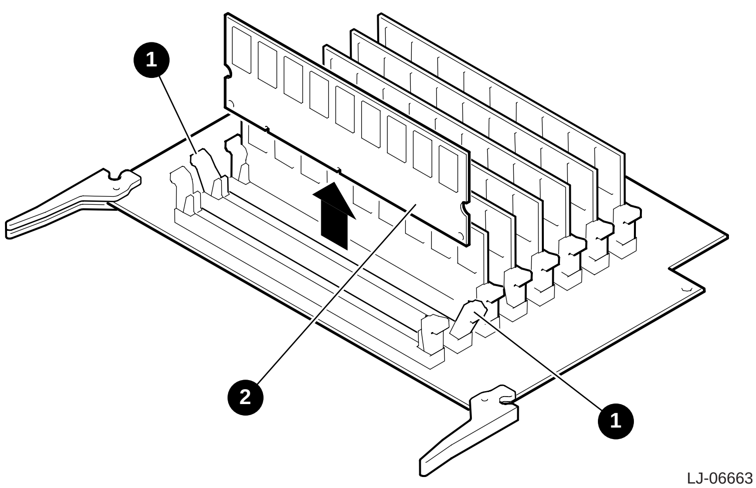
Caution

An antistatic wriststrap *must* be worn when handling any module to prevent damage to the module.

5. Push out on the locking levers ❶ located on each end of the memory DIMM ❷ being removed (see Figure 5-15).
6. Pull the memory DIMM ❷ straight up out of its slot (see Figure 5-15).

To replace the memory DIMM, refer to Figure 5-15 and align the memory DIMM with the appropriate slot and push firmly down until the DIMM is properly seated and the locking levers have closed. After replacing the memory DIMMs, reverse steps 1 through 3 of the removal procedure.

Figure 5-15 Removing and Replacing Memory DIMMs



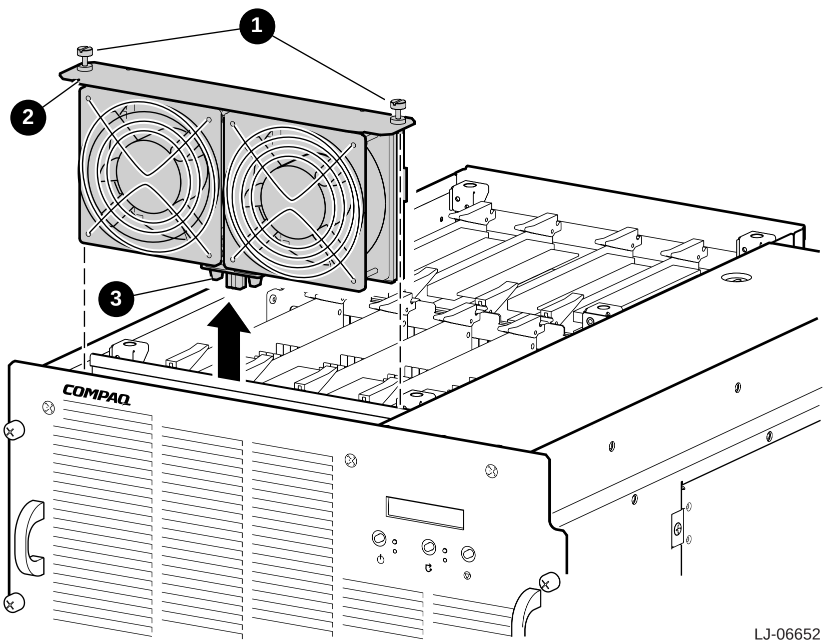
5.17 CPU Fan Assembly

Perform the following procedure to remove the CPU fan assembly:

1. Extend the chassis for service (see Section 5.2).
2. Remove the top cover (see Section 5.4).
3. Loosen the two captive screws ❶ along the top of the fan assembly ❷ (see Figure 5-16).
4. Lift the fan assembly straight up to disconnect the power connector ❸ on the bottom of the assembly and lift the assembly out of the chassis (see Figure 5-16).

To replace the CPU fan assembly, refer to Figure 5-16 and reverse steps 1 through 4 of the removal procedure.

Figure 5-16 Removing and Replacing the CPU Fan Assembly

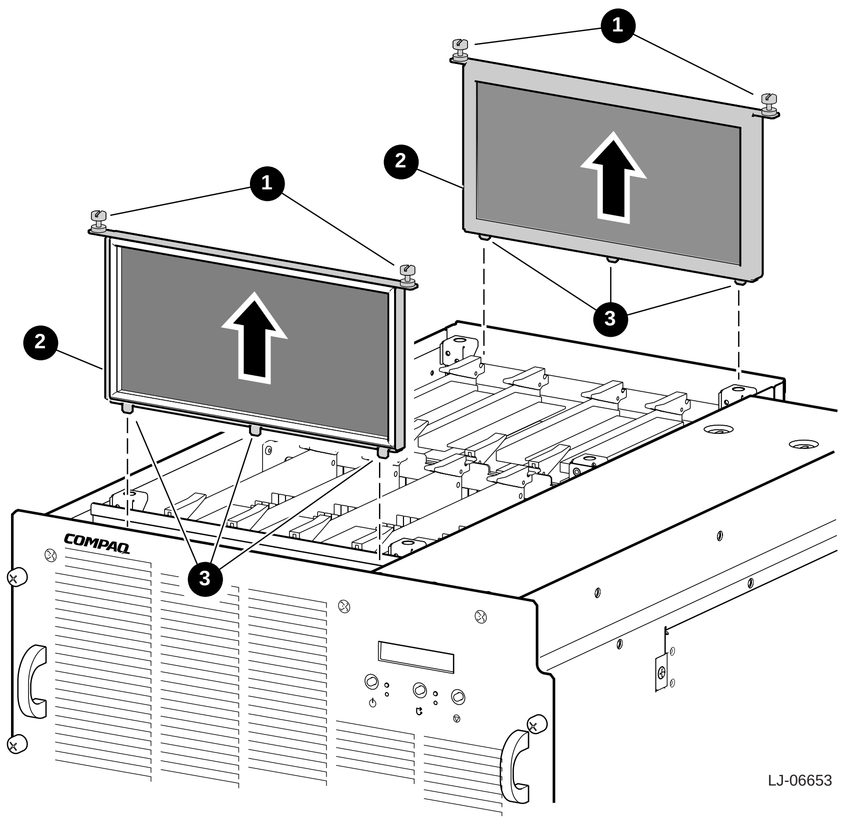


5.18 Front or Rear EMI/RFI Honeycomb Filter

Perform the following procedure to remove the front or rear EMI/RFI honeycomb filter:

1. Extend the chassis for service (see Section 5.2).
2. Remove the top cover (see Section 5.4).
3. Loosen the two captive screws ❶ along the top of the EMI/RFI honeycomb filter assembly ❷ (see Figure 5-17).
4. Lift the EMI/RFI honeycomb filter assembly until the three tabs ❸ on the bottom of the EMI/RFI honeycomb filter assembly come out of the three slots in the chassis (see Figure 5-17).

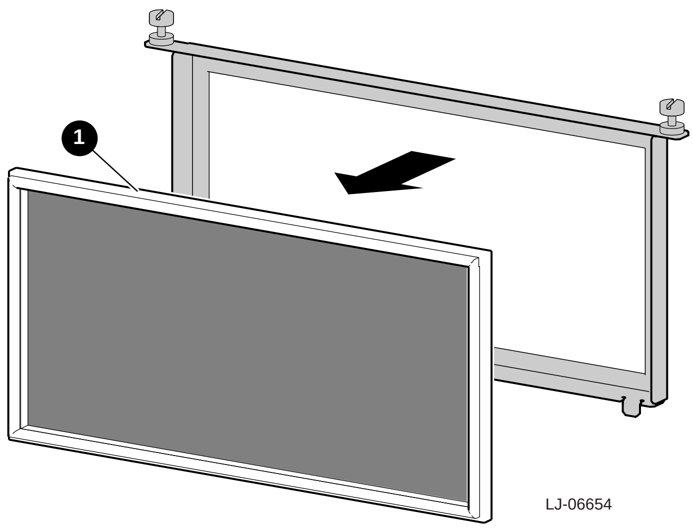
Figure 5-17 Removing and Replacing the Front or Rear EMI/RFI Honeycomb Filter Assembly



5. Remove the EMI/RFI honeycomb filter element ❶ from the EMI/RFI honeycomb filter assembly (see Figure 5-18).

To replace a front or rear EMI/RFI honeycomb filter, refer to Figure 5-17 and Figure 5-18 and reverse steps 1 through 5 of the removal procedure. Ensure that the three tabs on the bottom of the assembly go into the three slots in the chassis and that the EMI/RFI gasket on the replacement EMI/RFI honeycomb filter element is toward the outside of the chassis.

Figure 5-18 Removing the Front or Rear EMI/RFI Honeycomb Filter



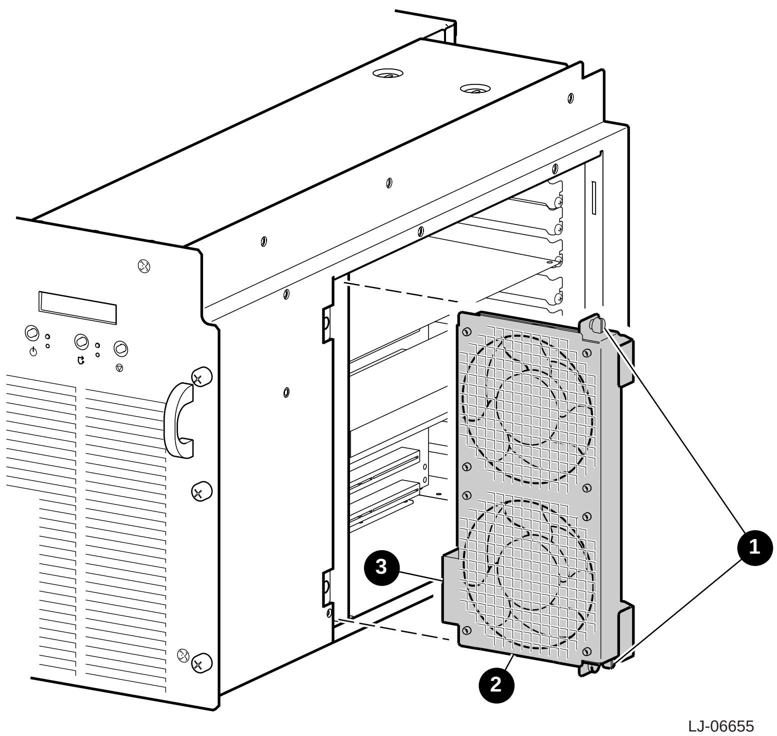
5.19 PCI Card-Cage Fan Assembly

Perform the following procedure to remove the PCI card-cage fan assembly:

1. Extend the chassis for service (see Section 5.2).
2. Remove the PCI access cover (see Section 5.5).
3. Loosen the two captive screws ❶ that secure the top and bottom of the fan assembly ❷ to the chassis (see Figure 5-19).
4. Pull the fan assembly straight out to disconnect the power connector ❸ on the rear of the assembly (see Figure 5-19).

To replace the PCI card-cage fan assembly, refer to Figure 5-19 and reverse steps 1 through 4 of the removal procedure.

Figure 5-19 Removing and Replacing the PCI Card-Cage Fan Assembly



5.20 PCI Option Modules

WARNING

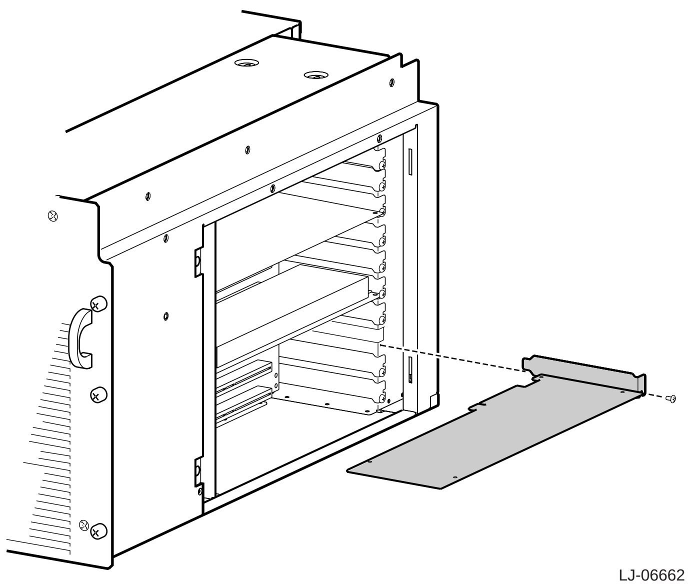
For protection against fire, only modules with current limited output should be used.

Perform the following procedure to remove PCI option modules:

1. Extend the chassis for service (see Section 5.2).
2. Remove the PCI access cover (see Section 5.5).
3. Disconnect any internal or external cables that are connected to the PCI option module.
4. Remove the screw that secures the PCI option module in its slot (see Figure 5-20).
5. Slide the PCI option module out of the slot (see Figure 5-20).

To replace the PCI option module, refer to Figure 5-20 and reverse steps 1 through 5 of the removal procedure. Ensure that the jumpers on the new module are set to match the jumpers on the module being replaced.

Figure 5-20 Removing and Replacing a PCI Option Module



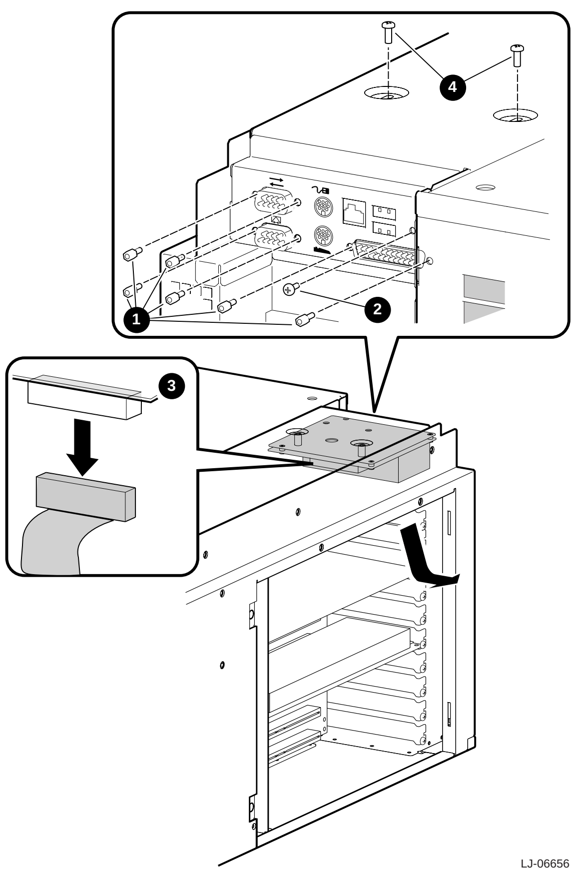
5.21 I/O Port Module

Perform the following procedure to remove the I/O port module:

1. Extend the chassis for service (see Section 5.2).
2. Remove the PCI access cover (see Section 5.5).
3. Disconnect any cables that are connected to the I/O port module from the rear of the chassis.
4. Remove the six standoffs ❶ with a 5 mm nutdriver and the one screw ❷ that secure the I/O port module to the rear of the chassis (see Figure 5-21).
5. If necessary, remove enough PCI option modules to allow access to the I/O port module.
6. Disconnect the ribbon cable ❸ from the I/O port module (see Figure 5-21).
7. Remove the two screws ❹ from the top of the chassis that secure the I/O port module (see Figure 5-21).
8. Pull the I/O port module down and out of the chassis.

To replace the I/O port module, refer to Figure 5-21 and reverse steps 1 through 8 of the removal procedure.

Figure 5-21 Removing and Replacing the I/O Port Module



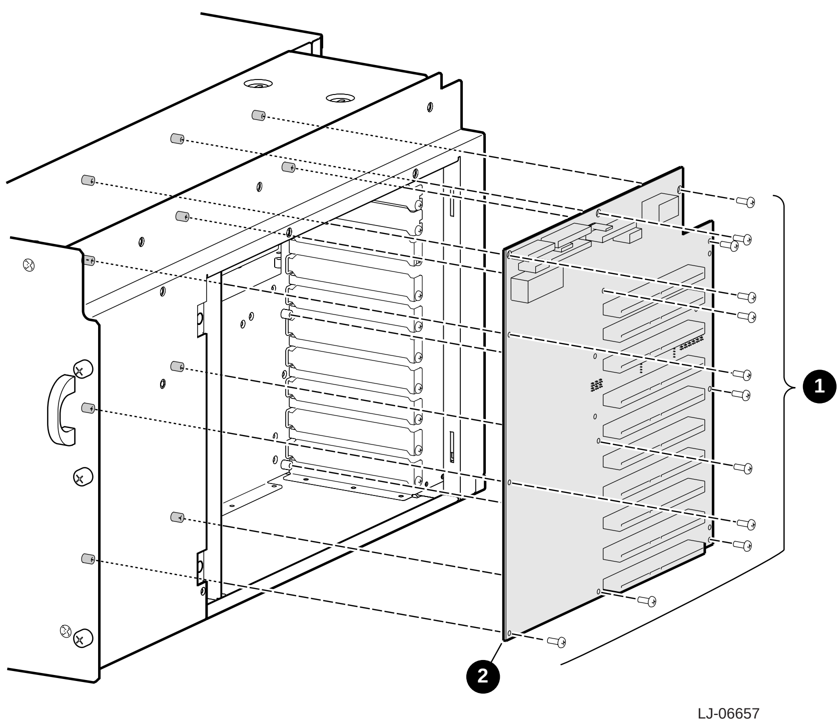
5.22 PCI Backplane

Perform the following procedure to remove the PCI backplane:

1. Extend the chassis for service (see Section 5.2).
2. Remove the PCI access cover (see Section 5.5).
3. Remove all of the PCI option modules (see Section 5.20).
4. Remove the I/O port module (see Section 5.21).
5. Disconnect the six cables from the PCI backplane.
6. Remove the 12 screws ❶ that secure the PCI backplane ❷ to the card-cage wall (see Figure 5-22).
7. Pull straight out on the PCI backplane to disconnect it from the system motherboard.
8. Remove the PCI backplane from the PCI card-cage.

To replace the PCI backplane, refer to Figure 5-22 and reverse steps 1 through 8 of the removal procedure. Ensure that the jumpers on the new PCI backplane are set to match the jumpers on the backplane being replaced.

Figure 5-22 Removing and Replacing the PCI Backplane



5.23 System Motherboard

Perform the following procedure to remove the system motherboard:

1. Extend the chassis for service (see Section 5.2).
2. Remove the top cover (see Section 5.4).
3. Remove all of the memory motherboards (see Section 5.15).
4. Remove all of the CPU modules (see Section 5.14).
5. Remove the front panel (see Section 5.3).
6. Remove the removable media cage (see Section 5.6).
7. While reaching through the removable media cage opening, disconnect the two power connectors ❶ and the two signal connectors ❷ from the front edge of the system motherboard.
8. Remove the three screws ❸ that secure the two center module support brackets ❺ to the system motherboard (see Figure 5-23).
9. Remove the six screws ❹ that secure the two center module support brackets to the left side of the chassis (see Figure 5-23).
10. Lift the two center module support brackets until the right ends of the two center module support brackets can be removed from the brackets on the right card-cage wall (see Figure 5-23).
11. Lift the two center module support brackets from the chassis.
12. Remove the four screws ❻ that secure the system motherboard to the bottom of the chassis (see Figure 5-24).
13. Slide the system motherboard to the left until it disconnects ❼ from the PCI backplane (see Figure 5-24).
14. Lift the system motherboard up and out of the chassis.

To replace the system motherboard, refer to Figure 5-23 and Figure 5-24 and reverse steps 1 through 14 of the removal procedure. Ensure that the jumpers on the new system motherboard are set to match the jumpers on the system motherboard being replaced.

Note

After replacing a system motherboard, the firmware must be updated. Refer to Section 3.10 to update the firmware on the new system motherboard.

Figure 5-23 Removing and Replacing the Two Center Module Support Brackets

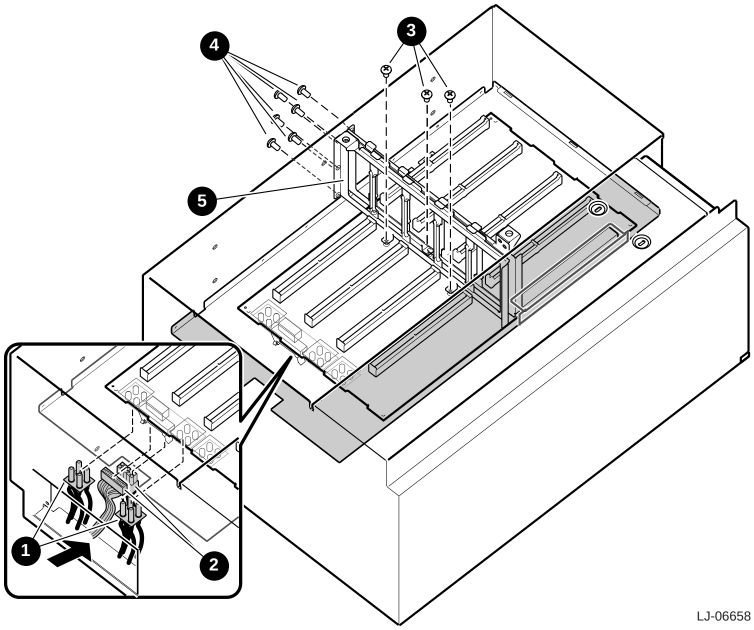
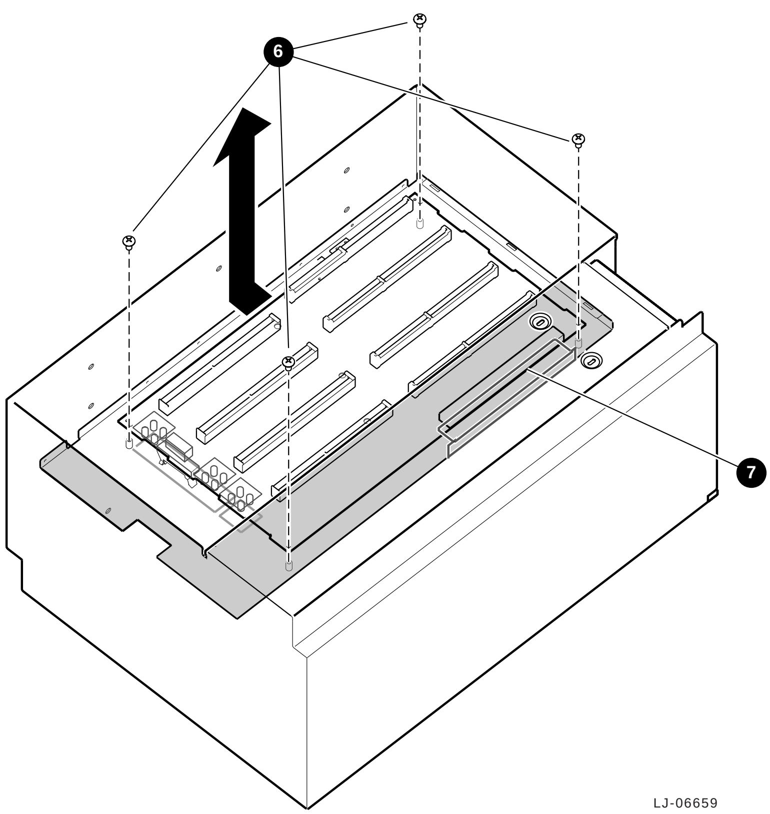


Figure 5-24 Removing and Replacing the System Motherboard



6
Troubleshooting and Diagnostics

6.1 Introduction

This chapter describes the troubleshooting strategy for *Compaq AlphaServer* ES40sv rackmount systems along with the power-up and SRM console diagnostics and event log analysis.

6.2 Questions to Consider

Before troubleshooting any system problem, first check the site maintenance log for the system's service history. Be sure to ask the system manager the following questions:

- Has the system been used and did it work correctly?
- Have changes to hardware or updates to firmware or software been made to the system recently? If so, are the revision numbers compatible for the system? (Refer to the hardware and operating system release notes).
- What is the state of the system?

  If the operating system is down, but you are able to access the SRM console, use the console environment diagnostic tools, including the power-up display, SRM diagnostics, and RMC diagnostics.

  If the operating system has crashed and rebooted, use the CANASTA-NG (Crash Analysis Troubleshooting Assistant New Generation) with the Compaq Analyze utility (to translate and interpret error logs), the SRM **crash** command, and operating system exercisers.

  If you are unable to access the SRM console, log into the RMC and issue an **env** command and a **status** command to determine if a hardware failure has occurred.

6.3 Categories of Problems

System problems can be classified into the following five categories. Using these categories, you can quickly determine a starting point for diagnosis and eliminate the unlikely sources of the problem.

1. Power problems Table 6-1
2. No access to console mode Table 6-2
3. Console-reported failures Table 6-3
4. Boot problems Table 6-4
5. Errors reported by the operating system Table 6-5

Table 6-1 Power Problems

Symptom	Action
System does not power on.	• Check that AC power is plugged in. • Check the Power setting on the control panel. Toggle the Power button to off, then back on to clear a remote power disable. • Check error messages on the OCP. • Check that the ambient room temperature is within environmental specifications (10–40°C, 50–104°F). • Check that internal power supply cables are plugged in at the system motherboard.
Power supply shuts down after a few seconds	The system may be powered off by one of the following: —RMC **power off** command —System software —Fan failure —Over-temperature condition —Power supply failure —Faulty CPU Invoke RMC and use the **status** command and the **env** command for an indication of fan failure or over-temperature condition. See Chapter 4. Check jumper J26. You can disable this jumper to override an over-temperature condition. See Appendix C

Table 6-2 Problems Getting to Console Mode

Symptom	Action
Power-up screen is not displayed.	Interpret the error beep codes and observe the OCP display for a failure detected during self-tests. See Section 6.5 for power-up error messages. Check keyboard and monitor connections. If the power-up screen is not displayed, yet the system enters console mode when you press the Return key, check that the **console** environment variable is set correctly. If you are using a VGA monitor as the console terminal, the **console** variable should be set to **graphics**. If you are using a serial console terminal, the console environment variable should be set to **serial**. If **console** is set to **serial**, the power-up screen is routed to the COM1 serial communication port or MMJ port and cannot be viewed from the VGA monitor. Try connecting a console terminal to the COM1 serial communication port. When using the COM1 port, you must set the **console** environment variable to **serial**. If the system has a customized NVRAM file, press the Halt button and then power up or reset the system. This will bypass the NVRAM script. Use the RMC **dump** command to probe the shared RAM.

Table 6-3 Problems Reported by the Console

Symptom	Action
No SRM messages are displayed after the system jumps to console.	Console firmware is corrupted. Load new firmware with fail-safe loader.
The system attempts to boot from the floppy drive after a checksum error is reported.	The system automatically reverts to the fail-safe loader to load new SRM firmware. If the fail-safe load does not work, replace the system motherboard.
Console program reports error:	
• Error beep codes report an error at power-up.	Use the error beep codes and OCP messages to determine the error. See Section 6.5.
• Power-up screen includes error messages.	Examine the console event log (with **cat el** or **more el** command).
• Power-up screen or console event log indicates problems with mass storage devices.	Refer to Section 6.5.5 to determine the problem.
• Storage devices are missing from the **show config** display.	Refer to Section 6.5.5 to determine the problem.
• PCI devices are missing from the **show config** display.	Refer to Section 6.5.5 to determine the problem.

Table 6-4 Boot Problems

Symptom	Action
System cannot find boot device.	Check the system configuration for the correct device parameters (node ID, device name, and so on). • For *Tru64 UNIX* and *OpenVMS*, use the **show config** and **show device** commands. Check the system configuration for the correct environment variable settings. • For *Tru64 UNIX* and *OpenVMS*, examine the **auto_action**, **bootdef_dev**, **boot_osflags**, and **os_type** environment variables. • For network boots, ensure that **ei*0_protocols** or **ew*0_protocols** is set to **bootp** for *Tru64 UNIX* or **mop** for *OpenVMS*.
Device does not boot.	For problems booting over a network, ensure that **ei*0_protocols** or **ew*0_protocols** is set to **bootp** for *Tru64 UNIX* or **mop** for *OpenVMS*. Run the device tests to check that the boot device is operating.

Table 6-5 Errors Reported by the Operating System

Symptom	Action
System has crashed, but SRM console is operating	Press the Halt button and enter the **crash** command to provide a crash dump file for analysis. Refer to *OpenVMS Alpha System Dump Analyzer Utility Manual* for information on how to interpret *OpenVMS* crash dump files. Refer to the *Guide to Kernel Debugging* for information on using the *Tru64 UNIX* Krash Utility. Use the SRM **info** command to get information on double error halts. If the problem is intermittent, run the SRM **test** and **sys_exer** commands.
System is hung and SRM console is not operating.	Invoke the RMC and enter the **dump** command to access shared RAM locations.
Operating system has crashed and rebooted.	Press the Halt button and enter the **crash** command to provide a crash dump file for analysis. Examine the operating system error log files to isolate the problem. If the problem is intermittent, run Compaq Analyze to determine the defective FRU.

6.4 Service Tools and Utilities

The following sections describe some of the tools and utilities available for acceptance testing and diagnosis and gives recommendations for their use.

6.4.1 Error Handling/Logging Tools (Compaq Analyze)

Tru64 UNIX and *OpenVMS* operating systems provide recovery from errors, fault handling, and event logging.

The primary tool for error handling is Compaq Analyze, a fault analysis utility designed to analyze both single error/fault events and multiple events. Compaq Analyze uses error/fault data sources in addition to the traditional binary error log. See Section 6.7.2 for information on Compaq Analyze.

6.4.2 ROM-Based Diagnostics (RBDs)

Many ROM-based diagnostics and exercisers execute automatically at power-up. RBDs can be invoked in console mode using SRM console commands. The remote management console (RMC) also provides diagnostic commands.

ROM-based diagnostics are the primary means of testing the console environment and diagnosing the CPU, memory, Ethernet, I/O buses, and SCSI subsystems. Use ROM-based diagnostics in the acceptance test procedures when you install a system or replace the following components: CPU module, memory motherboard, system motherboard, I/O bus device, or storage device. Refer to Section 6.6 for information on running ROM-based diagnostics.

6.4.3 Loopback Tests

Internal and external loopback tests are used to test the components on the I/O port module and to test Ethernet cards. The loopback tests are a subset of the ROM-based diagnostics.

Use loopback tests to isolate problems with the COM2 serial port, the parallel port, and Ethernet controllers. Refer to Section 6.6 for instructions on performing loopback tests.

6.4.4 SRM Console Commands

SRM console commands are used on systems running *Tru64 UNIX* or *OpenVMS* to set and examine environment variables and device parameters, as well as to invoke ROM-based diagnostics and exercisers. For example, the **show configuration** and **show device** commands are used to examine the configuration; the **set** command is used to set environment variables; and the **test** and **sys_exer** commands are used to test the system.

Use SRM console commands to set and examine environment variables and device parameters and to run RBDs. Refer to Section 3.4 for information on configuration-related firmware commands and Section 6.6 for information on running RBDs using SRM commands.

6.4.5 Remote Management Console (RMC)

The remote management console (RMC) is used for managing the server either locally or remotely. RMC also plays a key role in error analysis by passing error log information to shared RAM so that this information can be accessed by the system. RMC controls the control panel display and writes status messages on the control panel display.

RMC can be used to manage the server and gather error information or system state information (temperature and fan state, for example). RMC can be accessed as long as the line cord is plugged into the AC wall outlet. This feature ensures that you can gather information when the operating system is down and the SRM console is not accessible. See Chapter 4.

6.4.6 Operating System Exercisers (DEC VET)

The DIGITAL Verifier and Exerciser Tool (DEC VET) is supported by the *Tru64 UNIX* and *OpenVMS* operating systems. DEC VET is an on-line diagnostic tool used to ensure the proper installation and operation of hardware and base operating system software. Use DEC VET as part of acceptance testing to ensure that the CPU, memory, disk, tape, file system, and network are interacting properly.

6.4.7 Crash Dumps

For fatal errors, the *Tru64 UNIX* and *OpenVMS* operating systems save the contents of memory to a crash dump file. This file can be used to determine why the system crashed.

CANASTA-NG, the Crash Analysis Troubleshooting Assistant New Generation, is the primary crash dump analysis tool used by field engineers for analyzing crash dumps on Alpha systems. CANASTA-NG compares the results of a crash dump with a set of rules. If the results match one or more rules, CANASTA-NG notifies the system user of the cause of the crash and provides information to avoid similar crashes in the future.

6.4.8 Revision and Configuration Management Tool (RCM)

RCM is a tool to assist with revision and configuration management for hardware, firmware, operating system, and software products. It collects configuration and revision data from a system and stores it. A report generator produces configuration, change, and comparison reports. This is useful in finding revision incompatibilities. RCM also helps you verify service actions. For example, if a new board was supposed to be installed, you can use RCM to verify that the installation was done.

RCM is accessible using a Web browser:

http://smsat-www.ilo.dec.com/products/rcm/service/index.htm

6.4.9 *StorageWorks* Command Console (SWCC)

The *StorageWorks* Command Console (SWCC) is a storage management software tool that allows you to configure and monitor storage graphically from a single management console. It also has distributed capabilities that let you view multiple servers at the same time in a Microsoft Explorer-like navigation pane.

The *StorageWorks* Command Console's client is a graphical user interface (GUI) that can configure and monitor *StorageWorks* RAID Array solutions. This client runs on Windows NT (Intel only) or Windows 95. The Command Console agent runs on the host system and communicates with the client over a TCP/IP network connection, a SCSI connection, or a serial connection.

You can download the Command Console from the Internet at:

http://www.storage.digital.com/homepage/support/swcc/

6.5 Power-Up Diagnostics

The following sections explain how to interpret the power-up display on the console screen and the operator control panel. In addition, descriptions of the power-up sequence and firmware power-up diagnostics are provided to aid in troubleshooting. The following topics are covered:

- System startup diagnostics
- System power-on process
- RMC diagnostics
- SROM power-up display
- SRM console power-up display
- Fail-safe loader
- Identifying a bad RMC

6.5.1 System Startup Diagnostics

The power-up process begins with the power on of the power supplies. After the AC and DC power-up sequences are completed, the remote management console (RMC) reads EEROM information and passes it to SROM. The SROM minimally tests the CPUs, initializes and tests backup cache, and minimally tests memory. Finally, the SROM loads the SRM console program from flash ROM into memory and jumps to the first instruction in the console program.

There are three distinct sets of power-up diagnostics:

1. System power controller and remote management console (RMC) diagnostics—These diagnostics check the power regulators, temperature, and fans. Failures are reported in the dual-port RAM (DPR) on the control panel. Certain failures may prevent the system from powering on.
2. Serial ROM (SROM) diagnostics—SROM tests check the basic functionality of the system and load the console code from the FEPROM on the system motherboard into system memory. Failures during SROM tests are indicated by error beep codes and messages on the serial console terminal and the control panel.
3. Console firmware diagnostics—These tests are executed by the SRM console code. They test the core system, including all boot path devices. Failures during these tests are reported to the console terminal through the power-up screen or console event log.

6.5.2 System Power-On Process

The power-on sequence is as follows:

1. When the power cord is plugged into the wall outlet, 5V auxiliary AC voltage is enabled. The LEDs on the power supplies are lit, and the system power controller and RMC are initialized. The power supplies are tested. If all power supplies are bad, power-up stops.
2. Pressing the Power button on the control panel or issuing the **power-on** command from the RMC turns on power to the power supplies, CPU converters, and VTERM regulators. The converters and regulators are tested. If any converter or regulator is bad, power-up stops.
3. CPU_DCOK and SYS_DC_OK are set to “true,” which means DC power on the CPUs and system DC power are okay. All CPUs load the initial Y divisor (clock multiplier). The OCP power LED is lit.
4. SYS_RESET is set to “false.” This setting releases the system motherboard logic and PCI backplane logic from the Reset state.
5. The primary CPU is selected and CPU_(P)_RESET is set to “false.” This allows the primary CPU to attempt to load Flash SROM code.
6. If the primary CPU is good, it loads Flash SROM. If the primary CPU bad, the system tries the next available CPU and if that CPU is good, it becomes the primary. The remaining CPUs load Flash SROM. The SROM power-up then continues, as described in Section 6.5.4.

6.5.3 RMC Diagnostics

During power-up, RMC messages are sent to the operator control panel.

Table 6-6 lists the informational messages and error messages that the RMC might display on the control panel. Fatal error messages from the RMC take precedence over SROM messages and SRM messages. A fatal RMC message usually indicates a condition that causes the system to shut down.

RMC messages that indicate failed components preempt SROM and SRM error messages, even if the system is able to continue powering up. For example, if the RMC finds a bad CPU, it sends a message to the OCP. If a good CPU is found and the system successfully powers up to the console, the message indicating the failed CPU is the last message displayed on the OCP.

Table 6-6 RMC Messages

Message	Meaning
Informational Power-Up Messages	
Good bulks	Power supplies are good.
System DC is OK	DC power is on.
Good CPU	
System is up	
System is down	
Start RMC flash	
System Fatal Messages	
TIG error	Code essential to system operation is not running.
AC loss	No AC power to the system.
VTERM failed	No VTERM voltage to CPUs.
CTERM failed	No CTERM voltage to CPUs.
CPU*n* failed	A CPU failed. *n* is the CPU number: 0, 1, 2, or 3.
OverTemp failure	System temperature has passed the high threshold.
Fan 6 failed	Main CPU fan failed.
No CPU in slot 0	Configuration requires that a CPU be installed in slot 0.
Mixed CPU types	Different types of CPU are installed. Configuration requires that all CPUs be the same type.
Bad CPU ROM data	Invalid data in EEPROM on the CPU.
System Warning Messages	
PS*n* failed	A power supply failed. *n* is the power supply number: 0, 1, or 2
Fan*n* failed	A fan failed. *n* is the fan number, 1–6.
OverTemp Warning	System temperature is near the high threshold.
3.3V bulk warn	Power supply voltage over or under threshold.
5V bulk warn	Power supply voltage over or under threshold.
12V bulk warn	Power supply voltage over or under threshold.
-12V bulk warn	Power supply voltage over or under threshold.
VTERM warn	Voltage regulator over or under threshold.
CTERM warn	Voltage regulator over or under threshold.
CPU*n* VCORE warn	CPU core voltage over or under threshold. "*n*" is 0, 1, 2, or 3.
CPU*n* VIO warn	I/O voltage on CPU over or under threshold. "*n*" is 0, 1, 2, or 3.
*** ERROR - unknown command ***	You entered a command that is not in the RMC command set.
*** ERROR - illegal password ***	You entered an incorrect password.
*** ERROR - enable failed ***	
*** ERROR - Invalid character received ***	
*** ERROR - overflow ***	
*** ERROR - invalid data ***	

6.5.4 SROM Power-Up Display

Power-up information is displayed on the system's console terminal and the control panel. If the SRM **console** environment variable is set to **serial**, the entire power-up display, consisting of the SROM and SRM power-up messages, is printed on the VT terminal screen. If **console** is set to **graphics**, no SROM messages are displayed, and the SRM messages are delayed until VGA initialization has been completed. This process can take a little while

Example 6-1 shows the SROM power-up messages and corresponding operator control panel messages. See Table 6-7 for the full list of SROM and OCP start-up messages.

Example 6-1 Sample SROM Power-Up Display

SROM Power-Up Display		OCP Message
	①	PCI Test
SROM V1.00 CPU #00 @ 0500 MHz		
SROM program starting	❶	Power on
Reloading SROM		
SROM V1.00-F CPU # 00 @ 0500 MHz		
SROM program starting		
Starting secondary on CPU #1	❷	
Starting secondary on CPU #2		RelCPU
Starting secondary on CPU #3		
Bcache data tests in progress	❸	
Bcache address test in progress		BC Data
CPU parity and ECC detection in progress		
Bcache ECC data tests in progress		
Bcache TAG lines tests in progress		
Memory sizing in progress	❹	
Memory configuration in progress		Size Mem
Memory data test in progress		
Memory address test in progress		
Memory pattern test in progress		
Memory thrashing test in progress		
Memory initialization		
Loading console		
Code execution complete (transfer control)	❺	Load ROM
		Jump to Console

① When the system is powered up, the serial SROM code is loaded into the I-cache on the first available CPU, which becomes the primary CPU. The order of precedence is CPU0, CPU1, and so on. The primary CPU attempts to access the PCI bus. If it cannot, either a hang or a failure occurs and this is the only message displayed.

❶ The primary CPU interrogates the I^2C EEROM on the system motherboard and CPU modules through the shared dual-port RAM (DPR). Using the acquired data, the primary CPU determines the CPU and system configuration to jump to.

The primary CPU next checks the SROM checksum to determine the validity of the Flash SROM sectors.

If Flash SROM is invalid, the primary CPU reports the error and continues the execution of the serial SROM code. The Flash SROM must be reprogrammed in this case.

If Flash SROM is good, the primary CPU programs the appropriate registers with the values from the Flash data and selects itself as a target CPU to be loaded.

❷ The CPU loaded into Flash ROM (usually CPU0) initializes and tests the B-cache and memory, then loads the Flash SROM code to the next CPU. The loaded CPU initializes the EV6 (21264 chip) and marks itself as the secondary CPU. Once the primary CPU sees the secondary, it loads the Flash SROM code to the next CPU until all CPUs are loaded.

❸ The Flash SROM performs various B-cache tests. For example, the ECC data test verifies the detection logic for single- and double-bit errors.

❹ The primary CPU initiates all memory tests. The memory is tested for address and data errors for the first 32 MB of memory. It also initializes all of the “sized” memory in the system.

If a memory failure occurs, an error is reported. An untested memory array is assigned to address 0 and the failed memory array is de-assigned. The memory tests are re-run on the first 32 MB of memory. If all memory fails, the “No Memory Available” message is reported and the system halts.

❺ If all memory passes, the primary CPU loads the console and transfers control to it.

Table 6-7 SROM Execution Status Codes

Status Code	Full SROM Message	Corresponding OCP Message
DF	SROM program starting	Power on
DE	Init and test PCI bus	PCI Test
DD	Bcache data tests in progress	BC Data
DC	Bcache address test in progress	BC Addr
DB	CPU parity and ECC detection in progress	Par/ECC
DA	Bcache ECC data tests in progress	BC ECC
D8	Bcache TAG lines tests in progress	BC Tag
D7	Reloading SROM	Reload
D5	Memory sizing in progress	Size Mem
D4	Memory configuration in progress	Cfg Mem
D3	Memory data test in progress	Mem Data
D2	Memory address test in progress	Mem Addr
D1	Memory pattern test in progress	Mem Patt
D0	Memory thrashing test in progress	Mem thra
CF	Memory initialization	Mem Init
CD	Loading console	Load ROM
CB	Code execution complete (transfer control)	Jump to Console.
C5	Loading program from floppy	Load Flp
93	Starting secondary on CPU #3	RelCPU 3
92	Starting secondary on CPU #2	RelCPU 2
91	Starting secondary on CPU #1	RelCPU 1
90	Starting secondary on CPU #0	RelCPU 0

6.5.4.1 SROM Power-Up Error Messages

The SROM power-up identifies errors that may or may not prevent the system from coming up to the console. It is possible that these errors may prevent the system from successfully booting the operating system. Errors encountered during SROM power-up are displayed on the OCP. Some errors are also displayed on the console terminal if the console output is set to **serial**. Table 6-8 lists the SROM error messages.

Table 6-8 SROM Error Messages

Code	SROM Message	OCP Message
FD	PCI data path error	PCI Err
FA	No usable memory detected	No Mem
EF	Bcache data lines test error	BC Error
EE	Bcache data march test error	BC Error
ED	Bcache address test error	BC Error
EC	CPU Parity detection error	CPU Err
EB	CPU ECC detection error	CPU Err
EA	Bcache ECC data lines test error	BC Error
E9	Bcache ECC data march test error	BC Error
E8	Bcache TAG lines test error	BC Error
E7	Bcache TAG march test error	BC Error
E6	Console ROM checksum error	ROM Err
E5	Floppy driver error	Flpy Err
E4	No real-time clock (TOY)	TOY Err
E3	Memory data path error	Mem Err
E2	Memory address line error	Mem Err
E1	Memory pattern error	Mem Err
E0	Memory pattern ECC error	Mem Err
7F	Configuration error on CPU #3	CfgERR 3
7E	Configuration error on CPU #2	CfgERR 2
7D	Configuration error on CPU #1	CfgERR 1
7C	Configuration error on CPU #0	CfgERR 0
7B	Bcache failed on CPU #3 error	BC Bad 3
7A	Bcache failed on CPU #2 error	BC Bad 2
79	Bcache failed on CPU #1 error	BC Bad 1
78	Bcache failed on CPU #0 error	BC Bad 0
77	Memory thrash error on CPU #3	MtrERR 3
76	Memory thrash error on CPU #2	MtrERR 2
75	Memory thrash error on CPU #1	MtrERR 1
74	Memory thrash error on CPU #0	MtrERR 0
73	Starting secondary on CPU #3 error	RCPU 3 E
72	Starting secondary on CPU #2 error	RCPU 2 E
71	Starting secondary on CPU #1 error	RCPU 1 E
70	Starting secondary on CPU #0 error	RCPU 0 E
6F	Configuration error with system	CfgERR S

6.5.4.2 SROM Error Beep Codes

Error beep codes identify errors or states that are "beeped" to the user during the SROM power-up. Some are fatal and some are warnings. Most of the beeps are directly related to the SROM error codes.

Messages corresponding to the error indicated by the beep code are displayed on the console terminal if the console device is connected to the serial line and the SRM **console** environment variable is set to **serial**. Associated error messages are also displayed on the control panel.

Note

A single beep is emitted when the SROM code is successfully completed. The console firmware then continues with its power-up tests.

Table 6-9 lists the SROM error beep codes.

Table 6-9 SROM Error Beep Codes

Beep Code	Associated Messages	Meaning
1	Jump to Console	SROM code has completed execution. System jumps to console.
1-3		VGA monitor not plugged in.
1-1-4	ROM err	The SROM code is unable to load the console code; a flash ROM header area or checksum error has been detected.
1-1-7		Floppy load error, bad/wrong floppy disk.
1-2-4	BC error CPU error BC bad	Backup cache (B-cache) error. Indicates that a CPU is bad.
1-3-3	No mem	No usable memory detected. Some memory DIMMs may not be properly seated or some DIMM sets may be faulty.
2-1-2	Cfg ERR *n* Cfg ERR s	Configuration error on CPU n or a system configuration error (s). The system will still power up.

Beep Code 1-3-3: No Usable Memory Detected

If the SROM code could not find any available memory, you would hear a 1-3-3 beep code (one beep, a pause, a burst of three beeps, a pause, and another burst of three beeps). You would see a display similar to the following on the console terminal.

```
Failed  M:1 D:2        ❶
Failed  M:1 D:1
Failed  M:0 D:2
Failed  M:0 D:1
Incmpat M:3 D:6        ❷
Incmpat M:3 D:5
Incmpat M:2 D:6
Incmpat M:2 D:5
Missing M:3 D:2        ❸
Incmpat M:3 D:1
Illegal M:2 D:2        ❹
Incmpat M:2 D:1
No usable memory detected
```

❶ Indicates failed DIMMs. M identifies the MMB; D identifies the DIMM. In this line, DIMM 2 on MMB1 failed.

❷ Indicates that some DIMMs in this array are not the same. All DIMMs in the affected array are marked as incompatible (incmpat).

❸ Indicates that a DIMM in this array is missing. All missing DIMMs in the affected array are marked as missing.

❹ Indicates that the DIMM data for this array is unreadable. All unreadable DIMMs in the affected array are marked as illegal.

Finally, you would see the above messages and a final message such as “SROM 0 No Mem” on the control panel display. In such a case, the system would not come up to the console program. You would need to replace the bad DIMM sets.

6.5.5 SRM Console Power-Up Display

At the completion of SROM power-up, control is transferred to the SRM console program. The console program continues the system initialization process. Failures are reported to the console terminal through the power-up screen or a console event log.

The console program initializes the I/O system, which includes the PCI bus and connected devices such as SCSI disk controllers, the display device, the floppy drive, and the keyboard.

The display and keyboard devices are initialized first, and then a user interface (UI) is displayed. As initialization continues, startup messages are displayed within the UI.

Next, a memory test verifies that the memory is fully functioning. After memory is tested, SRM detects and initializes SCSI disk controllers connected to the computer. As each controller is detected, SRM probes its SCSI bus and reports each of the SCSI devices that it detects. After SCSI initialization is complete, *Tru64 UNIX* or *OpenVMS* systems display the SRM console prompt, P*nn*>>>. The number indicates the primary processor. In a multiprocessor system, the prompt could be P00>>>, P01>>>, P02>>>, or P03>>>.

Example 6-2 shows the messages that are displayed once the SROM has transferred control to the SRM console.

Example 6-2 SRM Power-Up Display

```
OpenVMS PALcode V1.44-1, UNIX PALcode V1.41-1

starting console on CPU 0                               ❶
initialized idle PCB
initializing semaphores
initializing heap
initial heap 200c0
memory low limit = 13e000
heap = 200c0, 17fc0
initializing driver structures
initializing idle process PID
initializing file system
initializing hardware
initializing timer data structures
lowering IPL
CPU 0 speed is 2.22 ns (500MHz)
create dead_eater
create poll
create timer
create powerup
access NVRAM
```

```
Memory size 640 MB
testing memory                                              ❷
.
.
.
probe I/O subsystem                                         ❸
probing hose 1, PCI
bus 0, slot 2, function 0 -- pka -- NCR 53C896
bus 0, slot 2, function 1 -- pkb -- NCR 53C896
bus 0, slot 4 -- ewa -- DE500-AA Network Controller
probing hose 0, PCI
probing PCI-to-ISA bridge, bus 1
bus 0, slot 2 -- vga -- DEC PowerStorm
bus 0, slot 15 -- dqa -- Acer Labs M1543C IDE
bus 0, slot 15 -- dqb -- Acer Labs M1543C IDE
starting drivers                                             ❹
entering idle loop
initializing keyboard

*** Error - CPU 1 failed powerup diagnostics ***             ①
  Secondary start error
EV6 BIST         = 1
STR status       = 1
CSC status       = 1
PChip0 status    = 1
PChip1 status    = 1
DIMx status      = 0
TIG Bus status   = 1
DPR status       = 0
CPU speed status = 0
CPU speed        = 0
Powerup time     = 00-00-00 00:00:00
CPU SROM sync    = 0
*** Error - Fan 1 failed ***
*** Error - Fan 2 failed ***
starting console on CPU 2                                    ❺
initialized idle PCB
initializing idle process PID
lowering IPL
CPU 2 speed is 2.22 ns (500MHz)
create powerup
entering idle loop
starting console on CPU 3
initialized idle PCB
initializing idle process PID
lowering IPL
CPU 3 speed is 2.22 ns (500MHz)
create powerup
initializing GCT/FRU at offset 17a000
AlphaServer ES40 Console X5.4-236, built on Mar  5 1999 at 01:42:13❻
P00>>>
```

❶ The primary CPU prints a message indicating that it is running the console. Starting with this message, the power-up display is printed to any console terminal, regardless of the state of the **console** environment variable.

If console is set to **graphics**, the display from this point to the end is saved in a memory buffer and printed to the VGA monitor after the PCI buses are sized and the VGA device is initialized.

❷ The memory size is determined and memory is tested.

❸ The I/O subsystem is probed and I/O devices are reported. I/O adapters are configured.

❹ Device drivers are started.

① The console saw that CPU 1 did not power up and fans 1 and 2 failed.

❺ The console is started on the secondary CPUs.

❻ Systems running *Tru64 UNIX* or *OpenVMS* display the SRM console banner and the prompt, P*nn*>>>. The number *n* indicates the primary processor. In a multiprocessor system, the prompt could be P00>>>, P01>>>, P02>>> and so on. From the SRM prompt, you can boot the *Tru64 UNIX* or *OpenVMS* operating system.

6.5.5.1 Console Event Log

If problems occur during SRM console power-up, standard error messages indicated by asterisks (***) may be embedded in a console event log. To display the event log, use the **more el** or **cat el** command.

Note

The ***cat el*** command displays the console event log in a scrolling display. To stop the screen display from scrolling, press Ctrl/S. To resume scrolling, press Ctrl/Q.

The ***more el*** command displays the console event log one screen at a time.

Example 6-3 Console Event Log

```
>>> more el

*** Error - CPU 1 failed powerup diagnostics ***
  Secondary start error
EV6 BIST         = 1
STR status       = 1
CSC status       = 1
PChip0 status    = 1
PChip1 status    = 1
DIMx status      = 0
TIG Bus status   = 1
DPR status       = 0
CPU speed status = 0
CPU speed        = 0
Powerup time     = 00-00-00 00:00:00
CPU SROM sync    = 0

*** Error - Fan 1 failed ***

*** Error - Fan 2 failed ***
```

6.5.6 Fail-Safe Loader

The fail-safe loader (FSL) is a software program that loads the SRM console firmware images from a floppy. From the FSL, you can update or load new SRM console firmware and FSL console firmware.

6.5.6.1 Checksum Error

A checksum error detected when the console is loading at power-up (error beep 1-1-4) indicates that the console firmware is corrupted. When the system detects the checksum error, the fail-safe loader is automatically activated so that you can load new console firmware images.

When a checksum error is detected, the sequence shown in Example 6-4 occurs.

Example 6-4 Checksum Error and Fail-Safe Load

```
Loading console
Console ROM checksum error                                          ❶
Expect: 00000000.000000FE
Actual: 00000000.000000FF
XORval: 00000000.00000001
Loading program from floppy................                         ❷
Code execution complete (transfer control)

OpenVMS PALcode V1.3-3, Digital UNIX PALcode V1.4-2

starting console on CPU 0                                           ❸
.
.
.
starting drivers
entering idle loop
P00>>>Boot (Update CD)                                              ①

OpenVMS PALcode V1.3-3, Digital UNIX PALcode V1.4-2

starting console on CPU 0                                           ②
.
.
.
starting drivers
entering idle loop

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

-------------------------------------------------------------

 Function     Description

-------------------------------------------------------------

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

UPD> update                                                         ❺
```

❶ The system detects the checksum error and writes a message to the console screen.

❷ The FSL program is loaded automatically from the floppy drive.

❸ As the FSL program is initialized, messages similar to the console power-up messages are displayed. This example shows the beginning and ending messages.

① At the console prompt, boot the LFU utility off the CD.

② As the LFU program is initialized, messages similar to the console power-up messages are displayed. This example shows the beginning and ending messages.

❹ After the "entering idle loop" message, the banner for the Loadable Firmware Update Utility is displayed.

❺ At the UPD> prompt, enter the **update** command to load the new console firmware images.

The FSL program can be downloaded from the following Internet location to create your own diskette.

http://ftp.digital.com/pub/Digital/Alpha/firmware/readme.html

6.5.6.2 Forcing a Fail-Safe Floppy Load

Under some circumstances, you may need to force the activation of the FSL. For example, if you install a system motherboard that has an older version of the firmware than your system requires, you may not be able to bring up the SRM console. In that case you need to force a floppy load so that you can update the SRM firmware.

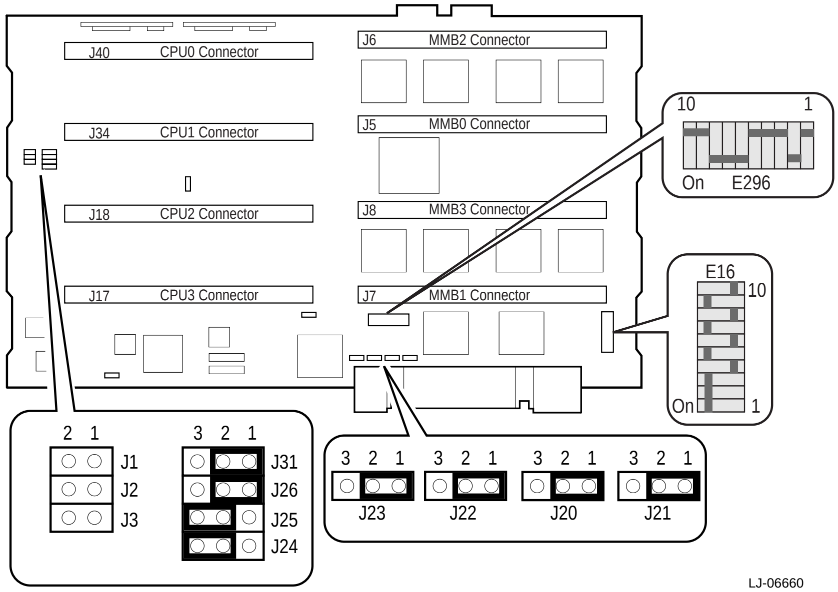
6. Perform a normal system power down.
7. Extend the chassis for service.
8. Remove the top cover, to gain access to the system motherboard.
9. Remove MMB 1 (closest to the PCI motherboard) so that you can access the function jumpers. See Figure 6-1.
10. Insert the Firmware Update Utility diskette and perform the update.
11. Locate the J22 function jumper on the system motherboard. See Figure 6-1.
12. Enable the fail-safe loader by moving the J22 jumper from pins 1 and 2 to pins 2 and 3.

Note

The J20 and J23 function jumpers must be in their default positions on pins 1 and 2.

13. Replace MMB 1 and the top cover.
14. Turn on the system and check the control panel display for progress messages. Enter "yes" at the UPD> prompt.
15. After the update utility has finished, perform a normal system power down.
16. Remove the top cover and MMB 1.
17. Move the J22 jumper back to pins 1 and 2.
18. Replace MMB 1 and the top cover.
19. Turn the system back on. The system should power up to the P00>>> console prompt.

Figure 6-1 System Motherboard Jumpers



6.5.7 Identifying a Bad RMC

Under certain circumstances, the RMC will not function. If the problem is caused by a corrupted shared RAM or corrupted RMC flash ROM, you need to update RMC firmware.

The RMC will not function if:

- No AC power is flowing from the wall outlet.
- Shared RAM does not pass its self-test (shared RAM is corrupted).
- RMC flash ROM is corrupted.

If the RMC is not working, the control panel displays the following message:

```
Bad RMC flash
```

The SRM console also sends the following message to the terminal screen:

```
*** Error - RMC detected power up error - RMC Flash corrupted ***
```

6.5.7.1 Updating the RMC

The remote management console firmware can be updated from flash ROM using the LFU.

1. Load the update medium.
2. At the UPD prompt, exit from the update utility, and answer **y** to the prompt. Enter **update RMC** to update the firmware.

```
UPD> exit

Do you want to do a manual update [y/(n)] y

***** 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.
Readme      Lists important release information.
Update      Replaces current firmware with loadable data
            image.
Verify      Compares loadable and hardware images.
? or Help   Scrolls this function table.
-----------------------------------------------------------

UPD> update RMC
.
.
.
```

6.6 SRM Console Diagnostics

The following sections describe troubleshooting with the SRM console.

The SRM console firmware contains ROM-based diagnostics that allow you to run system-specific or device-specific exercisers. The exercisers run concurrently so as to provide maximum bus interaction between the console drivers and the target devices.

Diagnostics are executed by using commands from the SRM console. Diagnostics can be run in the background by using the background operator “<” at the end of the command. Errors are reported to the console terminal, the console event log, or both.

6.6.1 Diagnostic Command Summary

Table 6-10 gives a summary of the SRM diagnostic commands and related commands.

Table 6-10 Summary of Diagnostic and Related Commands

Command	Function
buildfru	Initializes I^2Cbus EEPROM data structures for the named FRU.
cat el	Displays the console event log.
clear error <fruname>	Clears all SDD/TDD errors logged in the EEPROMs for the specified FRU. If no *fruname* is specified, it clears all SDD/TDD errors.
crash	Forces a crash dump at the operating system level.
deposit	Writes data to the specified address.
examine	Displays the contents of a memory location, register, or device.
exer	Exercises one or more devices by performing specified read, write, and compare operations.
info	Causes the SRM to read specified registers to help diagnose error halts and machine checks.
kill	Terminates a specified process.
kill_diags	Terminates all executing diagnostics.
more el	Displays the console event log, one screen at a time.
memexer	Runs a requested number of memory tests in the background..
memtest	Tests a specified section of memory.
net -ic	Initializes the MOP counters for the specified Ethernet port.
net -s	Displays the MOP counters for the specified Ethernet port.
nettest	Runs loopback tests for PCI-based Ethernet ports. Also used to test a port on a “live” network.
show error <fruname>	Reports errors logged in the EEPROMs for a specified FRU. If no *fruname* is specified, it reports all SDD/TDD errors.
show device	Lists the controllers and bootable devices in the system.
show fru	Displays information about field replaceable units (FRUs), including CPUs, memory DIMMs, and PCI cards.
show power	Determines whether a system failure was related to a fan, temperature, or power supply problem.
show_status	Displays the progress of diagnostic tests. Reports one line of information for each executing diagnostic.
sys_exer	Verifies all the devices in the system concurrently until you stop the tests with the init command.
sys_exer -lb	Conducts loopback tests for COM2 and the parallel port in addition to core system tests.
test	Verifies all the devices in the system sequentially. The status of each subsystem test is displayed to the console terminal as the test progresses.
test -lb	Conducts loopback tests for COM2 and the parallel port in addition to quick core system tests.

6.6.2 Diagnostic Commands Reference

The following sections give detailed information and examples of the diagnostic commands and related commands.

6.6.2.1 buildfru

The **buildfru** command initializes I^2Cbus EEPROM data structures for the named FRU and initializes SDD and TDD error logs. This command uses data supplied by the user to build the FRU descriptor. If the environment variable **sys_serial_num** is valid, it also initializes that field as well as the SMM value for the system.

Use the **buildfru** command if you replace a module for which the FRU information is wrong or missing. Once you replace the module, use **buildfru** to build the FRU descriptor. Use the **init** command to rebuild the FRU descriptor table, and use **show fru** to display the results.

Three areas of the EEPROM can be initialized: the FRU generic data, the FRU specific data, and the system specific data. Each area has its own checksum, which is recalculated any time that segment of the EEPROM is written.

When the **buildfru** command is executed, the FRU EEPROM is first flooded with zeroes, and then the generic data, the system specific data, and EEPROM format version information are written and checksums are updated. For certain FRUs, such as CPU modules, additional FRU "specific" data can be entered using the -S option. This data is written to the appropriate region, and its corresponding checksum is updated. Although this operation is typically performed in manufacturing, if it is performed in the field, the information supplied on the bar code label for a specific module should be used. Ensure that the data entered matches the FRU.

AlphaServer systems can be decomposed into a collection of FRUs. Some FRUs carry other FRUs. For instance, a system motherboard is a FRU, but it carries a number of sub-FRUs. A sub-FRU, such as a memory carrier module, may carry a number of its own sub-FRUs, DIMMs. The naming convention for FRUs represents the assembly hierarchy.

The following is the general form of a FRU name:

<frun>[.<frun>[.<frun>[.<frun>[<frun>]]]]

Where "fru" is a placeholder for the appropriate FRU type at that level and "n" is the number of that FRU instance on that branch of the system hierarchy.

The FRU assembly hierarchy for the *AlphaServer* ES40sv rackmount system has three levels. The FRU types from the top to the bottom of the hierarchy are as follows:

Level	FRU Type	Meaning
First Level	SMB0	System motherboard
	CPB0	PCI backplane
	SBM0	SCSI backplane
	PWR(0–2)	Power supplies
	JIO	I/O port module
Second Level	CPU(0–3)	CPUs
	MMB(0–3)	Memory motherboards
	PCI(0–9)	PCI slots
Third Level	DIM(1–8)	Memory DIMMs (1 through 8)

To build a FRU descriptor for a lower level FRU, you must point back to the higher level FRUs with which it is associated. For example, to build a descriptor for a DIMM, you point back to the MMB on which it resides and then to the system motherboard, which is where the MMB resides. Similarly, to build a descriptor for a CPU, you point back to the system motherboard. See the example at the end of this section.

Syntax

buildfru (<fru_name> <part_num> <serial_num> [<model> [<alias>]]
or
-s <fru_name> <offset> <byte> [<byte>...])

Arguments

<fru_name> Console name for this FRU. This name reflects the position of the FRU in the assembly hierarchy and has the form: <frun>[.<frun>[.<frun>[.<frun>[<frun>]]]], where “fru” is a placeholder for the FRU type at that level and “n” is the instance of that FRU type at that level.

<part_num> The FRU's Digital 2-5-2.4 part number. This ASCII string should be 16 characters (extra characters are truncated). This field should not contain any embedded spaces. If a space must be inserted, enclose the entire argument string must in double quotes. This field contains the FRU revision, and in some cases an embedded space is allowed between the part number and the revision.

<serial_num> The FRU's Digital serial number. This ASCII string must be 10 characters (extra characters are truncated). The manufacturing location and date are extracted from this field.

<model> The FRU's model name or number or a.k.a. name. This ASCII string may be up to 10 characters (extra characters are truncated). This field is optional, unless <alias> is specified.

<alias> The FRU's Compaq alias number, if one exists. This ASCII string may be up to 16 characters (extras are truncated). This field is optional.

<offset> The beginning byte offset (hex) within this FRU's EEPROM, where the following supplied data bytes are to be written.

<byte>... The data bytes to be written (up to 16). At least one data byte must be supplied after the offset.

Options

-s Writes raw data to the EEPROM. This option is typically used to apply any FRU specific data.

Example

```
P00>>>show fru | grep CPU                                                 ❶
SMB0.CPU0       0 54-12345-01          AY80112345
SMB0.CPU1       0 54-12345-01          AY80112345
SMB0.CPU2       0 54-12345-01          AY80112345
SMB0.CPU3       0 54-12345-01          AY80112345
P00>>>
P00>>>

P00>>>buildfru smb0.cpu0 30-12345-09.b01 ni90212345 digital digital❷

P00>>>gct                  <== Need to rebuild in memory FRU tree    ❸

P00>>> init                                                               ❹

...

P00>>>show fru | grep CPU                                                 ❺
SMB0.CPU0       0 30-12345-09.B01      NI90212345  DIGITAL   DIGITAL
SMB0.CPU1       0 54-12345-01          AY80112345
SMB0.CPU2       0 54-12345-01          AY80112345
SMB0.CPU3       0 54-12345-01          AY80112345
P00>>>
```

❶ The **show fru** command is issued (with **grep**) to find lines in the command output containing “CPU.”

❷ The **buildfru** command is issued to build a FRU descriptor for CPU0.

❸ The gct command is used to rebuild in memory the FRU tree.

❹ The system is initialized to rebuild the FRU descriptor table.

❺ The **show fru** command is reissued with **grep** to find the lines containing “CPU.” The new CPU0 descriptor is shown in the top line.

6.6.2.2 cat el and more el

The **cat el** and **more el** commands display the contents of the console event log. Status and error messages are logged to the console event log at power-up, during normal system operation, and while running system tests. Standard error messages are indicated by asterisks (***).

When **cat el** is used, the contents of the console event log scroll by. Use the Ctrl/S key combination to stop the screen from scrolling, and use Ctrl/Q to resume scrolling.

The **more el** command allows you to view the console event log, one screen at a time.

Syntax

cat el

or

more el

Example

In this example, the console reports that CPU 1 did not power up and fans 1 and 2 failed.

```
>>> more el

*** Error - CPU 1 failed powerup diagnostics ***               ❶
  Secondary start error
EV6 BIST         = 1
STR status       = 1
CSC status       = 1
PChip0 status    = 1
PChip1 status    = 1
DIMx status      = 0
TIG Bus status   = 1
DPR status       = 0
CPU speed status = 0
CPU speed        = 0
Powerup time     = 00-00-00 00:00:00
CPU SROM sync    = 0

*** Error - Fan 1 failed ***                                    ❷

*** Error - Fan 2 failed ***
```

❶ CPU1 failed.

❷ Fan 1 and Fan 2 failed.

6.6.2.3 clear error

The **clear error** command clears errors logged into the EEPROMs as reported by the **show error** command.

Syntax

clear error <fruname>

Fruname is the name of the specified FRU. If you do not specify a *fruname*, all SDD/TDD errors are cleared from the EEPROMs.

6.6.2.4 crash

The **crash** command forces a crash dump to the selected device for *Tru64 UNIX* and *OpenVMS* systems. Use the **crash** command when the system has hung and you are able to halt it with the Halt button or the RMC **halt in** command. The **crash** command restarts the operating system and forces a crash dump to the selected device.

- Refer to *OpenVMS Alpha System Dump Analyzer Utility Manual* for information on how to interpret *OpenVMS* crash dump files.
- Refer to the *Guide to Kernel Debugging* for information on using the *Tru64 UNIX* Krash Utility.

Syntax

crash [device]

Argument:

[device]	The device name of the device to which the crash dump is written.

Example

```
P00>>> crash dka100
CPU 0 restarting
DUMP: 19837638 blocks available for dumping.
DUMP: 118178 wanted for a partial compressed dump.
DUMP: Allowing 2060017 of the 2064113 available on 0x800001
device string for dump = SCSI 1 1 0 0 0 0 0.
DUMP.prom: dev SCSI 1 1 0 0 0 0 0, block 2178787
DUMP: Header to 0x800001 at 2064113 (0x1f7ef1)
device string for dump = SCSI 1 1 0 0 0 0 0.
DUMP.prom: dev SCSI 1 1 0 0 0 0 0, block 2178787
DUMP: Dump to 0x800001: .......: End 0x800001
device string for dump = SCSI 1 1 0 0 0 0 0.
DUMP.prom: dev SCSI 1 1 0 0 0 0 0, block 2178787
DUMP: Header to 0x800001 at 2064113 (0x1f7ef1)
succeeded

halted CPU 0

halt code = 5
HALT instruction executed
PC = fffffc0000568704
P00>>>
```

6.6.2.5 deposit and examine

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

deposit

The **deposit** command stores data in the location specified. If no options are given, the system uses the options from the preceding **deposit** command. If the specified value is too large to fit in the data size listed, the console ignores the command and issues an error. If the data is smaller than the data size, the higher order bits are filled with zeros.

examine

The **examine** command displays the contents of a memory location, a register, or a device. If no options are given, the system uses the options from the preceding **examine** command. If conflicting address space or data sizes are specified, the console ignores the command and issues an error. For data lengths longer than a longword, each longword of data should be separated by a space.

The syntax for the **deposit** and **examine** commands is:

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

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

-b	Defines data size as byte.
-w	Defines data size as word.
-l (default)	Defines data size as longword.
-q	Defines data size as quadword.
-o	Defines data size as octaword.
-h	Defines data size as hexword.
-d	Instruction decode (examine command only)
-n *value*	The number of consecutive locations to modify.
-s *value*	The address increment size. The default is the data size.
space:	Device name (or address space) of the device to access.
address	Offset within a device to which data is deposited. Can be:

dev_name	A device name.
dpr	The dual-port RAM.
econfig	
fpr- *name*	The floating-point register set; name is F0 to F31.
gpr- *name*	The general register set; name is R0 to R31.
ipr- *name*	The internal processor registers.
pconfig	The PCI configuration space.
pt- *name*	The PALtemp register set; name is PT0 to PT23.
pmem	Physical memory (default).
vmem	Virtual memory.

data	Data to be deposited.

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

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

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

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

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

Examples

deposit

```
P00>>> dep -b -n 1ff pmem:0 0                    ❶
P00>>> d -l -n 3 vmem:1234 5                     ❷
P00>>> d -n 8 r0 ffffffff                        ❸
P00>>> d -l -n 10 -s 200 pmem:0 8                ❹
P00>>> d -l pmem:0 0                             ❺
P00>>> d + ff                                    ❻
P00>>> d scbb 820000                             ❼
```

❶ Clear first 512 bytes of physical memory.

❷ Deposit 5 into four longwords starting at virtual memory address 1234.

❸ Load GPRs R0 through R8with -1.

❹ Deposit 8 in the first longword of the first 17pages in physical memory.

❺ Deposit 0 to physical memory address 0.

❻ Deposit FF to physical memory address 4.

❼ Deposit 820000 to SCBB.

examine

```
P00>>> examine pc                                ❶
PC psr: 0 (    PC)  0000000000001170
P00>>> examine sp                                ❷
gpr:   F0 (   R30)  0000000000072A60
P00>>> e -n 6 r4                                 ❸
grp: 20 (    R4) 0000000000005000
grp: 28 (    R5) 000000000FFFE000
grp: 30 (    R6) 00000003F8000C00
grp: 38 (    R7) 0000000053F761AE
grp: 40 (    R8) 0000010000000000
grp: 48 (    R9) 00000003F7800100
grp: 50 (   R10) 00000000000C7FFC
P00>>> examine pmem:400EC                        ❹
pmem:  400EC A49D0078A47D0070
```

❶ Examine the program counter.

❷ Examine the stack pointer.

❸ Examine register R4 and the next six registers.

❹ Examine physical memory.

6.6.2.6 exer

The **exer** command exercises one or more devices by performing specified read, write, and compare operations.

- A read operation reads from a device that you specify into a buffer.
- A write operation writes from a buffer to a device that you specify.
- A compare operation compares the contents of the two buffers.

The **exer** command uses two buffers to carry out the operations, buffer1 and buffer2. A read or write operation can be performed using either buffer; a compare operation uses both buffers.

You can use options to specify the following:

- An address range to test within the test device(s).
- The packet size (or I/O size) which is the number of bytes read or written in one I/O operation.
- The number of passes to run.
- How many seconds to run for a sequence of individual operations performed on the test devices. The qualifier used to specify this is called the action string qualifier.

Syntax

Exer ([-sb <start_block>] [-eb <end_block>] [-p <pass_count>]
[-l <blocks>] [-bs <block_size>] [-bc <blocks_per_io>]
[-d1 <buf1_string>] [-d2 <buf2_string>] [-a <action_string>]
[-sec <seconds>] [-m] [-v] [-delay <milliseconds>] <device_name>…)

Arguments:

<device_name>…	Specifies the names of the devices or filestreams to be exercised.

Options

-sb <start_block>	Specifies the starting block number (hex) within filestream. The default is 0.
-eb <end_block>	Specifies the ending block number (hex) within filestream. The default is 0
-p <pass_count>	Specifies the number of passes to run the exerciser. If 0, then run continuously or until Ctrl/C is pressed. The default is 1.
-l <blocks>	Specifies the number of blocks (hex) to exercise. l has precedence over eb. If only reading, then specifying neither l nor eb defaults to read until eof. If writing, and neither l nor eb is specified, then exer writes for the size of the device. The default is 1.
-bs <block_size>	Specifies the block size (hex) in bytes. The default is 200 (hex).
-bc <block_per_io>	Specifies the number of blocks (hex) per I/O. On devices without length (tape), use the specified packet size or default to 2048. The maximum block size allowed with variable length block reads is 2048. The default is 1.
-d1 <buf1_string>	String arg for eval to gen buffer1 data pattern from buffer1 is initialized only once and that is before any I/O occurs. Default = all bytes set to hex 5A's.
-d2 <buf2_string>	String arg for eval to gen buffer2 data pattern from buffer2 is initialized only once and that is before any I/O occurs. Default = all bytes set to hex 5A's.

-a <action_string>	Specifies an exerciser "action string," which determines the sequence of reads, writes, and compares to various buffers. The default action string is ?r. The action string characters are:

- r – Read into buffer1.
- w – Write from buffer1.
- R – Read into buffer2.
- W – Write from buffer2.
- n – Write without lock from buffer1.
- N – Write without lock from buffer2.
- c – Compare buffer1 with buffer2.
- **-** – Seek to file offset prior to last read or write.
- ? – Seek to a random block offset within the specified range of blocks. **Exer** calls the program, random, to "deal" each of a set of numbers once. **Exer** chooses a set that is a power of two and is greater than or equal to the block range. Each call to random results in a number that is then mapped to the set of numbers that are in the block range and **exer** seeks to that location in the filestream. Since **exer** starts with the same random number seed, the set of random numbers generated is always over the same set of block range numbers.
- s – Sleep for a number of milliseconds specified by the delay qualifier. If no delay is present, sleep for 1 millisecond.

  NOTE: *Times as reported in verbose mode are not necessarily accurate when this action character is used.*

- z – Zero buffer 1
- Z – Zero buffer 2
- b – Add constant to buffer 1
- B – Add constant to buffer 2

-sec <seconds>	Specifies to terminate the exercise after the number of seconds have elapsed. By default the exerciser continues until the specified number of blocks or passcount are processed.
-m	Specifies metrics mode. At the end of the exerciser a total throughput line is displayed.
-v	Specifies verbose mode. Data read is also written to stdout. This is not applicable on writes or compares. The default is verbose mode off.
-delay <millisecs>	Specifies the number of milliseconds to delay when "s" appears as a character in the action string.

Example Sequence

```
>>>exer du*.* dk*.* -p 0 –secs 36000
```

Read all DSSI and SCSI type disks for the entire length of each disk. Repeat this until 36000 seconds, 10 hours, have elapsed. All disks will be read concurrently. Each block read will occur at a random block number on each disk.

```
>>>exer –l 2 duc0
```

Read block numbers 0 and 1 from device duc0.

```
>>>exer –sb 1 –eb 3 –bc 4 –a ’w’ –d1 ’0x5a’ duc0
```

Write hex 5a’s to every byte of blocks 1, 2, and 3. The packet size is bc * bs, 4 * 512, 2048 for all writes.

```
>>>ls –l du*.* dk*.*
d**.* no such file
r---    dk           0/0            0
```

```
dka0.0.0.0.0
```

```
>>>exer dk*.* -bc 10 –sec 20 -m –a ’r’
dka0.0.0.0.0 exer completed

packet                IOs           elapsed idle

size   IOs     bytes read     bytes written     /sec
```

```
bytes/sec seconds secs
```

```
8192    3325    27238400     0   166     1360288    20
```

```
19
```

```
>>>exer –eb 64 –bc 4 –a ’?w-Rc’ duc0
```

A destructive write test over block numbers 0 thru 100 on disk duc0. The packet size is 2048 bytes. The action string specifies the following sequence of operations:

1. Set the current block address to a random block number on the disk between 0 and 97. A four block packet starting at block numbers 98, 99, or 100 would access blocks beyond the end of the length to be processed so 97 is the largest possible starting block address of a packet.
2. Write a packet of hex 5a’s from buffer1 to the current block address.
3. Set the current block address to what it was just prior to the previous write operation.
4. From the current block address read a packet into buffer2.
5. Compare buffer1 with buffer2 and report any discrepancies.
6. Repeat steps 1 thru 5 until enough packets have been written to satisfy the length requirement of 101 blocks.

```
>>>exer –a ’?r-w-Rc’ duc0
```

A non-destructive write test with packet sizes of 512 bytes. The action string specifies the following sequence of operations:

1. Set the current block address to a random block number on the disk.
2. From the current block address on the disk, read a packet into buffer1.
3. Set the current block address to the device address where it was just before the previous read operation occurred.
4. Write a packet of hex 5a’s from buffer1 to the current block address.
5. Set the current block address to what it was just prior to the previous write operation.
6. From the current block address on the disk, read a packet into buffer2.
7. Compare buffer1 with buffer2 and report any discrepancies.

8. Repeat the above steps until each block on the disk has been written once and read twice.

```
>>>set myd 0
>>>exer –bs 1 –bc a –l a –a ’w’ –d1 ’myd myd ~ =’ foo
>>>clear myd
>>>hd foo –l a
00000000 ff 00 ff 00 ff 00 ff 00 ff 00 ...
```

Use an environment variable, myd, as a counter. Write 10 bytes of the pattern ff 00 ff 00 … to RAM disk file foo. A packet size of 10 bytes is used. Since the length specified is also 10 bytes, then only one write occurs. Delete the environment variable, myd. The hd, hex dump of foo shows the contents of foo after exer is run.

```
>>>set myd 0
>>>exer –bs 1 –bc a –l a –a ’w’ –d1 ’myd myd 1 + =’ foo
>>>hd foo –l a
00000000 01 02 03 04 05 06 07 08 09 0a ...
```

Write a pattern of 01 02 03 …0a to file foo.

```
>>>set myd 0
>>>exer –bs 1 –bc 4 –l a –a ’w’ –d1 ’myd myd 1 + =’ foo -m
foo exer completed

packet                IOs                elapsed idle
size         IOs    bytes read    bytes written        /sec
bytes/sec seconds sec
    4      3      0       10      3001     10001          00
>>>hd foo
00000000 01 02 03 04 01 02 03 04 01 02 ...
>>>show myd
myd              4

>>>echo ’0123456789abcdefghijklmnopqrstAB’ –n foo3
>>>exer –bs 1 –v –m foo3
b21kfmp8jatsnA1gri54B69o3qdc7eh0foo3 exer completed

packet                IOs                elapsed idle
size         IOs    bytes read    bytes written        /sec
bytes/sec seconds sec
  1    32          32      0     5333      5333          00
```

6.6.2.7 info

The **info** command causes the SRM to read specified registers to help diagnose error halts and machine checks.

Example

```
P00>>> info

0.  HWRPB MEMDSC

1.  Console PTE

2.  GCT/FRU 5

Enter selection:
```

6.6.2.8 kill and kill_diags

The **kill** and **kill_diags** commands terminate diagnostics that are currently executing.

The **kill** command terminates a specified process.

The **kill_diags** command terminates all diagnostics.

Syntax

kill_diags

kill [PID. . .]

Argument:

[PID. . .]	The process ID of the diagnostic to terminate. Use the **show_status** command to determine the process ID.

Example

```
P00>>>  memexer 4
P00>>>show_status
 ID       Program      Device       Pass  Hard/Soft Bytes Written  Bytes Read
-------- ------------ ------------ ------ --------- ------------- -----------
00000001         idle system          0    0    0             0           0
0000125e      memtest memory         12    0    0    6719275008  6719275008
00001261      memtest memory         12    0    0    6689914880  6689914880
00001268      memtest memory         11    0    0    6689914880  6689914880
0000126f     exer_kid dka0.0.0.2.1    0    0    0             0     8612352
00001270     exer_kid dka100.1.0.2    0    0    0             0     8649728
00001271     exer_kid dka200.2.0.2    0    0    0             0     8649728
00001278     exer_kid dqa0.0.0.15.    0    0    0             0     3544064
00001280     exer_kid dfa0.0.0.2.1   84    0    0             0     8619520
00001281     exer_kid dfb0.0.0.102 1066    0    0             0   109256192
0000128e     exer_kid dva0.0.0.100    0    0    0             0      980992
00001381      nettest ewa0.0.0.4.1  362    0    1       1018720     1018496
P00>>> kill_diags
```

6.6.2.9 memexer

The **memexer** command runs a specified number of memory exercisers in the background. Nothing is displayed unless an error occurs. Each exerciser tests all available memory in twice the backup cache size blocks for each pass.

Use the **kill** command to terminate an individual diagnostic or the **kill_diags** command to terminate all diagnostics. Use the **show_status** command to display the progress of diagnostic tests.

Syntax

memexer [number]

Argument:

[number] Number of memory exercisers to start. The default is 1.

The number of exercisers, as well as the length of time for testing, depends on the context of the testing. Generally, running three to five exercisers for 15 minutes to 1 hour is sufficient for troubleshooting most memory problems.

Examples

The following example shows no errors.

```
P00>>>  memexer 4
P00>>>show_status
 ID       Program      Device       Pass  Hard/Soft Bytes Written  Bytes Read
-------- ------------ ------------ ------ --------- ------------- -----------
00000001         idle system          0    0    0             0           0
0000125e      memtest memory         12    0    0    6719275008  6719275008
00001261      memtest memory         12    0    0    6689914880  6689914880
00001268      memtest memory         11    0    0    6689914880  6689914880
0000126f     exer_kid dka0.0.0.2.1    0    0    0             0     8612352
00001270     exer_kid dka100.1.0.2    0    0    0             0     8649728
00001271     exer_kid dka200.2.0.2    0    0    0             0     8649728
00001278     exer_kid dqa0.0.0.15.    0    0    0             0     3544064
00001280     exer_kid dfa0.0.0.2.1   84    0    0             0     8619520
00001281     exer_kid dfb0.0.0.102 1066    0    0             0   109256192
0000128e     exer_kid dva0.0.0.100    0    0    0             0      980992
00001381      nettest ewa0.0.0.4.1  362    0    1       1018720     1018496
```

The following example shows a memory compare error indicating bad DIMMs. In most cases, the failing bank and DIMM position are specified in the error message.

```
P00>>> memexer 3

*** Hard Error - Error #41 - Memory compare error

Diagnostic Name      ID      Device  Pass  Test  Hard/Soft   11-FEB-1999
memtest    00000193        brd0  114    1        0   12:00:01
Expected value:    25c07
Received value     35c07
Failing addr:     a11848

*** End of Error ***

P00>>> kill_diags
P00>>>
```

6.6.2.10 memtest

The **memtest** command exercises a specified section of memory.

Memtest may be run on any specified address. If the **-z** option is not included (default), the address is verified and allocated from the firmware's memory zone. If the **-z** qualifier is included, the test is started without verification of the starting address.

When a starting address is specified, the memory is allocated beginning at the starting address **-** 32 bytes for the length specified. The extra 32 bytes that are allocated are reserved for the allocation header information. Therefore, if a starting address of 0xa00000 and a length of 0x100000 is requested, the area from 0x9fffe0 through 0xb00000 is reserved. This may be confusing if you try to begin two **memtest** processes simultaneously with one beginning at 0xa00000 for a length of 0x100000 and the other at 0xb00000 for a length of 0x100000. The second **memtest** process will send a message that it is "Unable to allocate memory of length 100000 at starting address b00000". Instead, the second process should use the starting address of 0xb00020.

Use the **kill** command to terminate an individual diagnostic or the **kill_diags** command to terminate all diagnostics. Use the **show_status** display to determine the progress of tests.

Note

If **memtest** is used to test large sections of memory, testing may take a while to be completed. If you issue a Ctrl/C or **kill PID** in the middle of testing, **memtest** may not abort right away. For speed reasons, a check for a Ctrl/C or **kill** is done outside of any test loops. If this is not satisfactory, you can run concurrent **memtest** processes in the background with shorter lengths within the target range.

Memtest Test 1

Memtest Test 1 uses a graycode algorithm to test a specified section of memory. The graycode algorithm used is: data = (x>>1)^x, where x is an incrementing value.

Three passes are made of the memory under test.

- The first pass writes alternating graycode inverse graycode to each four longwords. This causes many data bits to toggle between each 16 byte write.

  For example graycode patterns for a 32 byte block would be:

  Graycode(0) 00000000 Graycode(1) 00000001 Graycode(2) 00000003 Graycode(3) 00000002 Inverse Graycode(4) FFFFFFF9 Inverse Graycode(5) FFFFFFF8 Inverse Graycode(6) FFFFFFFA Inverse Graycode(7) FFFFFFFB

- The second pass reads each location, verifies the data, and writes the inverse of the data, one longword at a time. This causes all data bits to be written as a one and zero.

- The third pass reads and verifies each location.

You can specify the **-**f "fast" option so that the explicit data verify sections of the second and third loops is not performed. This does not catch address shorts but stresses memory with a higher throughput. The ECC/EDC logic can be used to detect failures.

Memtest Test 2

Memtest Test 2 uses a marching 1's/0's algorithm to test a specified section of memory. The same range can be tested as in the graycode, test 1. The default data pattern used by this test is 0x55555555 and its inverse 0xAAAAAAAA. The data pattern can be altered with the **-d** qualifier. The pattern entered and its compliment are then used instead.

Three passes are also made of the memory under test.

- The first pass writes the data pattern entered (or default) beginning at the starting address and marching through for the entire length specified.
- The second pass begins again at the starting address, reads the previously written data pattern and writes back its inverse, a longword at a time for the entire specified length.
- The third pass begins at the end of the testing region, again reads back the previously written inverse pattern, and writes back 0's, a longword at a time, decrementing up through memory until the starting address is reached.

Memtest Test 3

The random test, Memtest Test 3, performs writes with random data to random addresses using random data size, lengths, and alignments. The run time of the random test may be noticeably longer than that of the other tests because it requires two calls to the console firmware's random number generator every time data is written.

The random test accesses every memory location within the boundaries specified by the **-sa** and **-l** qualifiers (as long as the length is less than 8 megabytes — until a 64 bit compiler exists. With lengths greater than 8 megabytes, a modulo function is required on the seed and therefore some addresses may get repeated and some not tested at all). It first obtains an address index into the Linear Congruential Generator structure dependent on the length specified. It obtains the data index as a function of the entered random data seed and the maximum 32 bit data pattern. Using the address index and an initial address seed of 0, the random number generator is called to obtain a random address. It is then called again, using the data index and initial user entered data seed, (**-rs** qualifier), or default of 0, to get the longword of data to use in testing.

The lower bit of the random data returned is also used to determine whether to perform longword or quadword transactions. (Using the lower bit saves another call to the random function to help speed up the test). The data is then stored to the random address, and memory barrier is performed to flush it out to the B-cache and then a read is done to read it back in. A compare is then done on the data written and the data read. In the case of quadword writes and reads, the longword of random data is shifted left by 32 and or'd with the original's compliment to form the quadword.

Memtest Test 4

Memtest test 4 is a DVT test at this time. Test 4 requires that an actual memory module be present in the system. You must first set up a block of data to use in the test. The address of this block of data is read as an input to the test using the **-ba** qualifier. (default is block of data containing 4 longwords of 0xF's, then 4 longwords of 0's, then 4 longwords of 0xF's and lastly 4 longwords of 0's.)

First, the test performs a write of the block of data to the specified starting address. It then adds a B-cache offset to the starting address and performs another write of arbitrary data. This causes the original data to be 'victimized' to memory. A read is then performed of the original starting address and verifies that it is correct. The starting address is then incremented by a block and the write/write/read procedure repeated for the specified length of memory.

Syntax

memtest ([-sa <start_address>] [-ea <end_address>] [-l <length>]
[-bs <block_size>] [-i <address_inc>] [-p <pass_count>]
[-d <data_pattern>] [-rs <random_seed>] [-ba <block_address>]
[-t <test_mask>] [-se <soft_error_threshold>]
[-g <group_name>] [-rb] [-f] [-m] [-z] [-h] [-mb])

Options

-sa Start address. Default is first free space in memzone.

-ea End address. Default is start address plus length size.

-l Length of section to test in bytes, default is the zone size with the ---**rb** option and the block_size for all other tests. **-l** has precedence over **-ea**.

-bs Block (packet) size in bytes in hex, default 8192 bytes This is used only for the random block test. For all other tests the block size equals the length.

-i Specifies the address increment value in longwords. This value is used to increment the address through the memory to be tested. The default is 1 (longword). This is only implemented for the graycode test. An address increment of 2 tests every other longword. This option is useful for multiple CPUs testing the same physical memory.

-p Passcount If 0 then run forever or until Ctrl/C. Default = 1

-t Test mask. Default = run all tests in selected group.

-g Group name

-se Soft error threshold

-f Fast. If **-f** is included in the command line, the data compare is omitted. Detects only ECC/EDC errors.

-m Timer. Prints out the run time of the pass. Default = off .

-z Tests the specified memory address without allocation. Bypasses all checking, but allows testing in addresses outside of the main memory heap. Also allows unaligned input.

WARNING: This flag can trash anything and everything!!

-d Used only for march test (2). Uses this pattern as test pattern. Default = 5's

-h Allocates test memory from the firmware heap.

-rs Used only for random test (3). Uses this data as the random seed to vary random data patterns generated. Default = 0.

-rb Randomly allocates and tests all of the specified memory address range. Allocations are done of block_size.

-mb Memory barrier flag. Used only in the Alpha -f graycode test. When set an mb is done after every memory access. This guarantees serial access to memory.

-ba Used only for block test (4). Uses the data stored at this address to write to each block.

Example

```
P00>>>memtest -sa 4000000 -l 4000000 -p 0<
P00>>>show_status
 ID       Program      Device       Pass  Hard/Soft Bytes Written  Bytes Read
-------- ------------ ------------ ------ --------- ------------- -------------
00000001         idle system           0    0    0             0           0
000000f9      memtest memory           2    0    0      67108864    67108864
P00>>>
*** Hard Error - Error #43 - Memory compare error
```

```
Diagnostic Name        ID          Device  Pass  Test  Hard/Soft   1-JAN-1999
memtest          000000f9            brd0     9     1     1    0     12:00:01
Expected value:           ffffffff
Received value:                  0
Failing addr:               400000

*** End of Error ***
P00>>>
```

6.6.2.11 net

The **net** command performs maintenance operations on a specified Ethernet port. **Net -ic** initializes the MOP counters for the specified Ethernet port, and **net -s** displays the current status of the port, including the contents of the MOP counters.

Syntax

net -ic <port_name>

or

net –s <port_name>

Arguments

<port_name>	Specifies the Ethernet port on which to operate, either ei*0 or ew*0.

Example

```
P00>>> net -ic ewa0
P00>>> net -s ewa0
Status counts:
ti: 72 tps: 0 tu: 47 tjt: 0 unf: 0 ri: 70 ru: 0
rps: 0 rwt: 0 at: 0 fd: 0 lnf: 0 se: 0 tbf: 0
tto: 1 lkf: 1 ato: 1 nc: 71 oc: 0

MOP BLOCK:
 Network list size: 0

MOP COUNTERS:
Time since zeroed (Secs): 3

TX:
 Bytes: 0 Frames: 0
 Deferred: 0 One collision: 0 Multi collisions: 0
TX Failures:
 Excessive collisions: 0 Carrier check: 0 Short circuit: 0
 Open circuit: 0 Long frame: 0 Remote defer: 0
 Collision detect: 0

RX:
 Bytes: 0 Frames: 0
 Multicast bytes: 0 Multicast frames: 0
RX Failures:
 Block check: 0 Framing error: 0 Long frame: 0
 Unknown destination: 0 Data overrun: 0 No system buffer: 0
 No user buffers: 0
P00>>>
```

6.6.2.12 nettest

The **nettest** command tests the network ports using MOP loopback.

Nettest performs a network test. It can test the ei* or ew* ports in internal loopback, external loopback, or live network loopback mode.

Nettest contains the basic options to run MOP loopback tests. It is assumed that this command will be included in a script for most applications. Many environment variables can be set from the console to customize **nettest** before **nettest** is started. The environment variables, a brief description, and their default values are listed below.

Note

Each variable name is preceded by e*a0_ or e*b0_ to specify the desired port.

You can change other network driver characteristics by modifying the port mode. See the **-mode** option.

Use the **kill** command to terminate an individual diagnostic or the **kill_diags** command to terminate all diagnostics. Use the **show_status** display to determine the process ID when terminating an individual diagnostic test.

Syntax

nettest ([-f <file>] [-mode <port_mode>] [-p <pass_count>]
[-sv <mop_version>] [-to <loop_time>] [-w <wait_time>]
[<port>])

Arguments

<port>	Specifies the Ethernet port on which to run the test.
Options	
-f <file>	Specifies the file containing the list of network station addresses to loop messages to. The default file name is lp_nodes_e*a0 for port e*a0. The default file name is lp_nodes_e*b0 for port e*b0. The files by default have their own station address.
-mode <port_mode>	Specifies the mode to set the port adapter (TGEC). The default is ex (external loopback). Allowed values are: df : default, use environment variable values ex : external loopback in : internal loopback nm : normal mode nf : normal filter pr : promiscuous mc : multicast ip : internal loopback and promiscuous fc : force collisions nofc : do not force collisions nc : do not change mode
-p <pass_count>	Specifies the number of times to run the test. If 0, then run forever. The default is 1. **NOTE*:*** This is the number of passes for the diagnostic. Each pass will send the number of loop messages as set by the environment variable, ***eia*_loop_count*** or ***ewa*_loop_count***.

-sv <mop_version>	Specifies which MOP version protocol to use. If 3, then MOP V3 (DECNET Phase IV) packet format is used. If 4, then MOP V4 (DECNET Phase V IEEE 802.3) format is used.
-to <loop_time>	Specifies the time in seconds allowed for the loop messages to be returned. The default is 2 seconds.
-w <wait_time>	Specifies the time in seconds to wait between passes of the test. The default is 0 (no delay). The network device can be very CPU intensive. This option will allow other processes to run.
Environment Variables	
e*a*_loop_count	Specifies the number (hex) of loop requests to send. The default is 0x3E8 loop packets.
e*a*_loop_inc	Specifies the number (hex) of bytes the message size is increased on successive messages. The default is 0xA bytes.
e*a*_loop_patt	Specifies the data pattern (hex) for the loop messages. The following are legitimate values. 0 : all zeros 1 : all ones 2 : all fives 3 : all 0xAs 4 : incrementing data 5 : decrementing data ffffffff : all patterns
loop_size	Specifies the size (hex) of the loop message. The default packet size is 0x2E.

Example Sequence

```
>>nettest            - internal loopback test on port ei*0 or ew*0
>>nettest ez*        - internal loopback test on ports eia0/eib0 or ewa0/ewb0
>>nettest -mode ex      - external loopback test on port eia0 or ewa0
>>nettest -mode ex -w 10  - external loopback test on port eia0 or ewa0; wait
                            10 seconds between tests
>>nettest -f foo -mode nm -   normal mode loopback test on port eia0 or ewa0
                              using the list of nodes contained in the file foo
```

6.6.2.13 show error

The **show error** command displays the SDD/TDD errors on the specified FRU.

Use the **clear error** command to clear SDD/TDD errors.

Syntax

show error <fruname>

Fruname is the name of the specified FRU. If you do not specify a *fruname*, all errors are displayed.

6.6.2.14 show device

The **show device** command lists the controllers and bootable devices in the system.

Storage devices are identified by the following letters:

dk	SCSI disk or CD	ew	Ethernet port
dq	IDE CD-ROM	fw	FDDI device
dr	RAID set device	mk	SCSI tape
du	DSSI disk	mu	DSSI tape
dv	Diskette drive	pk	SCSI port
ei	Network card	pu	DSSI port

```
P00>>>show device
dka0.0.0.2.1          DKA0                    RZ1CB-CS  0844
dka100.1.0.2.1        DKA100                  RZ1CB-CA  LYJ0
dka200.2.0.2.1        DKA200                  RZ1CB-CA  LYJ0
dkb400.4.0.102.1      DKB400                     RRD46  1337
dqa0.0.0.15.0         DQA0     TOSHIBA CD-ROM XM-5702B  3476
dva0.0.0.1000.0       DVA0
ewa0.0.0.4.1          EWA0           00-00-F8-1F-29-97
pka0.7.0.2.1          PKA0               SCSI Bus ID 7
pkb0.7.0.102.1        PKB0               SCSI Bus ID 7
```

6.6.2.15 show fru

The **show fru** command displays a table of the Field Replaceable Units (FRUs) in the system. Typing **show fru** displays all the FRU information for the system.

Syntax

show fru ([-e])

Options

-e Specifies that only FRUs that have errors logged are displayed.

Example

```
P00>>> show fru

    ❶           ❷    ❸                   ❹          ❺          ❻
FRU Name        E  Part#              Serial#    Model#/MFG Alias#/YY-WW

SMB0            0 54-25385-01.C02       NI90360843
SMB0.CPU0       0 54-12345-01           AY80112345
SMB0.CPU1       0 54-12345-01           AY80112345
SMB0.CPU2       0 54-12345-01           AY80112345
SMB0.CPU3       0 54-12345-01           AY80112345
SMB0.MMB0       0 54-25582-01.B02       NI84960460
SMB0.MMB0.DIM1  0 54-12345-01CPQ        AY90112345
SMB0.MMB0.DIM2  0 54-12345-01CPQ        AY90112345
SMB0.MMB0.DIM3  0 20-12345-01.A001CPQ   AY90112345  DIGITAL   DIGITAL
SMB0.MMB0.DIM4  0 20-12345-01.A001CPQ   AY90112345  DIGITAL   DIGITAL
SMB0.MMB0.DIM5  0 20-12345-01.A001CPQ   AY90112345  DIGITAL   DIGITAL
SMB0.MMB0.DIM6  0 20-12345-01.A001CPQ   AY90112345  DIGITAL   DIGITAL
SMB0.MMB1       0 54-25582-01.B02       NI84960352
SMB0.MMB1.DIM1  0 54-12345-01CPQ        AY90112345
SMB0.MMB1.DIM2  0 20-12345-01.A001CPQ   AY90112345  DIGITAL   DIGITAL
SMB0.MMB1.DIM3  0 20-12345-01.A001CPQ   AY90112345  DIGITAL   DIGITAL
SMB0.MMB1.DIM4  0 20-12345-01.A001CPQ   AY90112345  DIGITAL   DIGITAL
SMB0.MMB1.DIM5  0 20-12345-01.A001CPQ   AY90112345  DIGITAL   DIGITAL
SMB0.MMB1.DIM6  0 20-12345-01.A001CPQ   AY90112345  DIGITAL   DIGITAL
SMB0.MMB2       0 54-25582-01.B02       NI84960315
SMB0.MMB2.DIM1  0 54-12345-01CPQ        AY90112345
SMB0.MMB2.DIM2  0 20-12345-01.A001CPQ   AY90112345  DIGITAL   DIGITAL
SMB0.MMB2.DIM3  0 20-12345-01.A001CPQ   AY90112345  DIGITAL   DIGITAL
SMB0.MMB2.DIM4  0 20-12345-01.A001CPQ   AY90112345  DIGITAL   DIGITAL
SMB0.MMB2.DIM5  0 20-12345-01.A001CPQ   AY90112345  DIGITAL   DIGITAL
SMB0.MMB2.DIM6  0 20-12345-01.A001CPQ   AY90112345  DIGITAL   DIGITAL
SMB0.MMB3       0 54-25582-01.B02       NI84960254
SMB0.MMB3.DIM1  0 54-12345-01CPQ        AY90112345
SMB0.MMB3.DIM2  0 20-12345-01.A001CPQ   AY90112345  DIGITAL   DIGITAL
SMB0.MMB3.DIM3  0 20-12345-01.A001CPQ   AY90112345  DIGITAL   DIGITAL
SMB0.MMB3.DIM4  0 20-12345-01.A001CPQ   AY90112345  DIGITAL   DIGITAL
SMB0.MMB3.DIM5  0 20-12345-01.A001CPQ   AY90112345  DIGITAL   DIGITAL
SMB0.MMB3.DIM6  0 40-98765-55.G098CPQ   AY99999999
SMB0.CPB0       0 54-30156-01.AO3       NI84760022
SMB0.CPB0.PCI1  0 DE500-BA Network Cont
SMB0.CPB0.PCI4  0 ELSA GLoria Synergy
SMB0.CPB0.PCI5  0 NCR 53C895
PWR0            0 API-7650              NI320PSC-X  2P850000  30-49448-0
PWR1            0 API-7650              NI320PSC-X  2P850000  30-49448-0
FAN1            0 930329                0           B32262-3  Digital
FAN2            0 930329                0           B32262-3  Digital
```

```
FAN3            0 930329                0           B32262-3  Digital
FAN4            0 930329                0           B32262-3  Digital
FAN5            0 930329                0           B32262-3  Digital
FAN6            0 930329                0           B32262-3  Digital
JIO0            0 54-25575-01           0           Junk I/O  Digital
```

The following example shows that CPU1 and fan 4 have errors.

```
P00>>> show fru –e
SMB0.CPU1       1  54-24773-02.A01    AY76543211 CPU Module COMPAQ9876543210
FAN4            1  B2262-33                      Fan
P00>>>
```

The column headings have the following meanings:

❶	FRU Name	The FRU name recognized by the SRM console. The name also indicates the location of that FRU in the physical hierarchy of the system. SMB = system motherboard; CPU = CPUs; MMB = memory motherboard; DIM = DIMMs; CPB = PCI backplane; PCI = PCI option; PWR = power supply; FAN = fans; JIO= I/O port module (junk I/O).
❷	E	Indicates whether the FRU has any errors logged against it. Healthy FRUs have a zero (0) in this column; FRUs with errors have a one (1).
❸	Part #	The part number of the FRU in ASCII, either a Compaq part number oe a vendor part number.
❹	Serial #	The serial number. For Compaq FRUs, the serial number has the form XXYWWNNNNN. XX = manufacturing location code YWW = year and week NNNNN = sequence number. For vendor FRUs, the four byte sequence number is displayed in hex.
❺	Model #/MFG	Option FRU information. For Compaq FRUs, a model name, number, or an "a.k.a" name. For vendor FRUs, the manufacturer's name.
❻	Alias #/YY-WW	Option data. For Compaq FRUs, the Compaq part alias number (if one exists). For vendor FRUs, the year and week number.

6.6.2.16 show power

The **show power** command is used to determine whether the failure of a system is related to a fan, temperature, or power supply problem. You can use this command if you are able to restart the system.

```
P00>>> show power

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

The system was last reset via a system software reset

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

6.6.2.17 show_status

The **show_status** command displays the progress of diagnostics. The **show_status** command reports one line of information per executing diagnostic.

Many of the diagnostics run in the background and provide information only if an error occurs.

The following command string is useful for periodically displaying diagnostic status information for diagnostics running in the background:

```
P00>>> while true;show_status;sleep n;done
```

Where *n* is the number of seconds between **show_status** displays.

Syntax

show_status

Example

```
❶          ❷          ❸           ❹     ❺        ❻             ❼
P00>>>show_status
 ID       Program      Device       Pass  Hard/Soft Bytes Written  Bytes Read
-------- ------------ ------------ ------ --------- ------------- -----------
00000001         idle system          0    0    0             0           0
0000125e      memtest memory         12    0    0    6719275008  6719275008
00001261      memtest memory         12    0    0    6689914880  6689914880
00001268      memtest memory         11    0    0    6689914880  6689914880
0000126f     exer_kid dka0.0.0.2.1    0    0    0             0     8612352
00001270     exer_kid dka100.1.0.2    0    0    0             0     8649728
00001271     exer_kid dka200.2.0.2    0    0    0             0     8649728
00001278     exer_kid dqa0.0.0.15.    0    0    0             0     3544064
00001280     exer_kid dfa0.0.0.2.1   84    0    0             0     8619520
00001281     exer_kid dfb0.0.0.102 1066    0    0             0   109256192
0000128e     exer_kid dva0.0.0.100    0    0    0             0      980992
00001381      nettest ewa0.0.0.4.1  362    0    1       1018720     1018496
P00>>>
```

❶ Process ID.

❷ The SRM diagnostic for the particular device.

❸ ID of the device under test.

❹ Number of diagnostic passes that have been completed.

❺ Error count (hard and soft): soft errors are not usually fatal; hard errors halt the system or prevent completion of the diagnostics.

❻ Bytes successfully written by the diagnostic.

❼ Bytes successfully read by the diagnostic.

6.6.2.18 sys_exer

The **sys_exer** command exercises the devices displayed with the **show config** command. Tests are run concurrently and in the background. Nothing is displayed after the initial test startup messages unless an error occurs.

Use the **show_status** command to display the progress of diagnostic tests.

The diagnostics started by the **sys_exer** command automatically reallocate memory resources, because these tests require additional resources. Use the **init** command to reconfigure memory before booting an operating system.

Because the **sys_exer** tests are run concurrently and indefinitely (until you stop them with the **init** command), they are useful in flushing out intermittent hardware problems.

When using the **sys_exer** command after shutting down an operating system, you must initialize the system to a quiescent state. Enter the following command at the SRM console:

```
P00>>> init
.
.
.
P00>>> sys_exer
```

By default, no write tests are performed on disk and tape drives. Media must be installed to test the floppy drive and tape drives. When the *-lb* argument is used, a loopback connector is required for the COM2 port (9-pin loopback connector, 12-27351-01) and parallel port (25-pin loopback connector).

Syntax

sys_exer [-lb]

Argument:

[-lb]	The loopback option runs console loopback tests for the COM2 serial port and the parallel port during the test sequence.

Example

```
P00>>>sys_exer
Default zone extended at the expense of memzone.
Use INIT before booting
Exercising the Memory
Exercising the DK* Disks(read only)
Exercising the DQ* Disks(read only)
Exercising the DF* Disks(read only)
Exercising the Floppy(read only)
Testing the VGA (Alphanumeric Mode only)
Exercising the EWA0 Network

Type "show_status" to display testing progress
Type "cat el" to redisplay recent errors
Type "init" in order to boot the operating system
P00>>>show_status
 ID       Program      Device       Pass  Hard/Soft Bytes Written  Bytes Read
-------- ------------ ------------ ------ --------- ------------- -----------
00000001         idle system          0     0    0             0           0
0000125e      memtest memory         12     0    0    6719275008  6719275008
00001261      memtest memory         12     0    0    6689914880  6689914880
00001268      memtest memory         11     0    0    6689914880  6689914880
```

```
0000126f     exer_kid dka0.0.0.2.1    0    0    0           0       8612352
00001270     exer_kid dka100.1.0.2    0    0    0           0       8649728
00001271     exer_kid dka200.2.0.2    0    0    0           0       8649728
00001278     exer_kid dqa0.0.0.15.    0    0    0           0       3544064
00001280     exer_kid dfa0.0.0.2.1   84    0    0           0       8619520
00001281     exer_kid dfb0.0.0.102 1066    0    0           0     109256192
0000128e     exer_kid dva0.0.0.100    0    0    0           0        980992
00001381      nettest ewa0.0.0.4.1  362    0    1     1018720       1018496

P00>>>init
OpenVMS PALcode V1.44-1, Tru64 UNIX PALcode V1.41-1
...
starting console on CPU 0
```

6.6.2.19 test

The **test** command verifies all the devices in the system. This command can be used on all supported operating systems: *Tru64 UNIX* and *OpenVMS*.

The **test** command also does a quick test on the system speaker. A beep is emitted as the command starts to run.

The tests are run sequentially for a minimum of 10 seconds per test. The status of each subsystem test is displayed to the console terminal as the test progresses. If a particular device is not available to test, a message is displayed. The test script does no destructive testing, that is, it does not write to disk drives.

To run a complete diagnostic test using the **test** command, the system configuration must include:

- A serial loopback connected to the COM2 port
- A parallel loopback connected to the parallel port
- A trial diskette with files installed
- A trial CD-ROM with files installed

The test script tests devices in the following order:

1. Memory tests (one pass).
2. Read-only tests: DK* disks, DR* disks, DQ* disks, DU* disks, MK* tapes, DV* floppy.
3. Console loopback tests if *-lb* argument is specified: COM2 serial port and parallel port.
4. VGA console tests. These tests are run only if the **console** environment variable is set to **serial**. The VGA console test displays rows of the word “compaq.”
5. Network internal loopback tests for EW* networks.

Note

By default, no write tests are performed on disk and tape drives. Media must be installed to test the floppy drive and tape drives.

Syntax

test [-lb]

Argument:

[-lb]	The loopback (-lb) option runs console loopback tests for the COM2 serial port and the parallel port during the test sequence.

Example

```
P00>>>test
Testing the Memory
Testing the DK* Disks(read only)
No DU* Disks available for testing
No DR* Disks available for testing
Testing the DQ* Disks(read only)
Testing the DF* Disks(read only)
No MK* Tapes available for testing
No MU* Tapes available for testing
Testing the DV* Floppy Disks(read only)
Testing the VGA (Alphanumeric Mode only)
Testing the EWA0 Network
Testing the EWB0 Network
P00>>>
```

6.7 Error Log Analysis

The following sections provide information on how to interpret error logs reported by the operating system. The following topics are covered

- RMC and Error Information
- Error Log Analysis with Compaq Analyze
- Fault detection and reporting
- Machine checks/interrupts
- Errors Captured by the SRM Console
- Error Analysis with RMC

6.7.1 RMC and Error Information

The remote management console (RMC) manages an extensive network of FRU I_2C EEPROMs. Information from these EEPROMs is stored in dual-port RAM (DPR) and can be accessed to diagnose hardware failures.

At system power-up, the RMC reads 256 bytes of data from each FRU EEPROM and stores it in the dual-port RAM, a shared RAM that facilitates interaction between the RMC and the system. The EEPROM data contains information on configuration and errors. The data is accessible through the TIG chip on the system motherboard.

As one of its functions, the TIG provides interfaces for the firmware and the operating system to communicate with the server management logic. The data accessed from DPR provides configuration information to the firmware during start-up. Remote or local applications can read the error log and configuration information. The error log information is written to the dual-port RAM by an error handling agent and then written back to the EEPROMs by the RMC. This arrangement ensures that the error log is available on a FRU after power has been lost.

The RMC provides several commands for accessing error information in the shared RAM. Compaq Analyze, described in Section 6.7.2, can also access the FRU EEPROM error logs to diagnose hardware failures.

6.7.2 Error Log Analysis with Compaq Analyze

Compaq Analyze is a fault management diagnostic tool that is used to determine the cause of hardware failures. Compaq Analyze analyzes both single error/fault events and multiple events. It uses error/fault data sources in addition to the traditional binary error log.

Compaq Analyze is installed on the customer's system with the operating system. The Compaq Analyze Director starts automatically as part of the system start-up. Compaq Analyze provides automatic background analysis, constantly viewing and reading the error log file. If an event occurs, it triggers the firing of an analysis rule. The analysis engine collects the information on the error and generates a problem report. The report can be sent to users on a notification mailing list and, if DSNlink is installed, a call can be logged with the local customer support center.

Compaq Analyze supports the *Tru64 UNIX*, and *OpenVMS* operating systems on *AlphaServer* platforms. It uses a Java-based graphical user interface.

Note

Compaq Analyze is not backward compatible with *DECevent*.

6.7.2.1 Features of Compaq Analyze

Graphical User Interface (GUI)

The user interface for Compaq Analyze is a browser-style GUI. From the interface you can:

- View event problem reports
- View translated output of event logs
- Configure tool components
- Create and populate user groups
- Connect to specific nodes

Error Log Decomposition

- Provides a formatted display of binary error log entries for *Tru64 UNIX* and *OpenVMS*, for 620/630/660/670/680 entries with subpackets.
- Provides a formatted display of binary error log entries for *Tru64 UNIX* Common Access Method (CAM) error log entries.

Machine Check Error Log Entry Types

Tru64 UNIX and *OpenVMS*:

CPU Correctable Error (630)
CPU Uncorrectable Error (670)
System Correctable Error (620)
System Uncorrectable Error (660)
System Environmental (680)

I/O Error Log Entry Types

Supports CAM SCSI Entry Type 199 (*Tru64 UNIX* only)

Notification

Compaq Analyze can send problem reports to recipients via SMTP mail.

Scavenge (*Tru64 UNIX* Only)

On *Tru64 UNIX* systems, Compaq Analyze can scavenge for events that are still pending processing (that is, the system was rebooted and new events occurred).

6.7.2.2 Installing Compaq Analyze

Kits for installing Compaq Analyze can be downloaded from the Product Service Engineering Web site:

http://pinkft.cxo.dec.com/compaq_analyze

6.7.2.3 Starting and Stopping Compaq Analyze

Tru64 UNIX

The installation process for *Tru64 UNIX* starts the Compaq Analyze Director automatically. To restart the Director, enter:

```
/usr/sbin/ca start
```

To verify that the Director is running, enter:

```
ps -ef | grep ca
```

To launch the GUI, enter:

```
ca &
```

To quit the GUI, do one of the following:

- Click the toolbar icon
- From the System menu, select **Quit GUI**
- Press Alt/Q

To stop Compaq Analyze, enter:

```
# /usr/sbin/ca stop
```

OpenVMS

To start and stop Compaq Analyze on *OpenVMS*, you need a startup command file and a shutdown command file.

Startup .COM file:

@SYS$STARTUP:CompaqAnalyze$STARTUP.COM

Shutdown .COM file:

@SYS$MANAGER:CompaqAnalyze$SHUTDOWN.COM

To start the Director and launch the GUI, enter:

```
$@SYS$STARTUP:CompaqAnalyze$STARTUP
%DCL-S-SPAWNED, process Compaq_Analyze spawned
$@sys$startup:ca_gui$startup
```

To quit the GUI, do one of the following:

- Click the toolbar icon
- From the System menu, select **Quit GUI**
- Press Alt/Q

To stop Compaq Analyze, enter

```
$@SYS$MANAGER:CompaqAnalyze$SHUTDOWN
```

6.7.2.4 Event Problem Reports

If an event occurs, it triggers the firing of an analysis rule. The analysis engine collects the information on the error and generates a problem report. The report is listed in the GUI window for the local host.

6.7.2.5 Viewing Translated Output

Events are listed as separate entries in the GUI window for the local host. To view the translated output, select the entry from the list and click Display or double-click the entry. The second column in Example 6-5 shows the translated output from an event.

Example 6-5 Translated Output

```
Event Header Common Fields [Fields Not Specific to an Operating System
(CEH) V2.0]

Event_Leader                xFFFFFFFE
Header_Length               680
Event_Length                1616
Header_Rev_Major            2
Header_Rev_Minor            0
OS_Type                     1
Hardware_Arch               4
CEH_Vendor_ID               3564
Hdwr_Sys_Type               34
Logging_CPU                 1
CPUs_In_Active_Set          2
Major_Class                 100
Minor_Class                 3
DSR_Msg_Num                 1813
CEH_Device                  5
Chip_Type                   8
CEH_Device_ID_0             x0000000B
CEH_Device_ID_1             x0000000C
CEH_Device_ID_2             x0000000D
Unique_ID_Count             14
Unique_ID_Prefix            15
Num_Strings                 15
```

6.7.3 Fault Detection and Reporting

Table 6-11 provides a summary of the fault detection and correction components of Compaq *AlphaServer* ES40 systems.

Generally, PALcode handles exceptions as follows:

1. The PALcode determines the cause of the exception.
2. If possible, it corrects the problem and passes control to the operating system for reporting before returning the system to normal operation.

   If PALcode is unable to correct the problem, it

   - Logs double error halts into the flash ROM
   - Logs uncorrectable errors to the shared RAM
   - For single halts, logs the uncorrectable logout frame into shared RAM.
3. If error/event logging is required, control is passed through the system control block (SCB) to the appropriate operating system exception handler. The operating system exception handler logs the error condition into the binary error log. If the error is not an uncorrectable CPU error or system error, Compaq Analyze should diagnose the error to the defective FRU.

Table 6-11 *Compaq AlphaServer* ES40 Fault Detection and Correction

Component	Fault Detection/Correction Capability
Alpha 21264 microprocessor	Contains error checking and correction (ECC) logic for data cycles. Check bits are associated with all data entering and exiting the microprocessor. A single-bit error on any of the four longwords being read can be corrected (per cycle). A double-bit error on any of the four longwords being read can be detected (per cycle).
Backup cache (B-cache)	ECC check bits on the data store, and parity on the tag address store and tag control store.
Memory DIMMs	ECC logic protects data by detecting and correcting data cycle errors. A single-bit error on any of the four longwords can be corrected (per cycle). A double-bit error on any of the four longwords being read can be detected (per cycle).
PCI SCSI controller adapter	SCSI data parity is generated.
ISA-to-PCI bridge chip	PCI data parity is generated.

6.7.4 Machine Checks/Interrupts

The exceptions that result from hardware system errors are called machine checks/interrupts. They occur when a system error is detected during the processing of a data request.

During the error-handling process, errors are first handled by the appropriate PALcode error routine and then by the associated operating system error handler. PALcode transfers control to the operating system through the system control block (SCB) vector. The SCB vectors pertain to *Tru64 UNIX* and *OpenVMS* systems only.

The following types of machine checks/interrupts are related to system events.

- CPU Correctable Error (630)
- CPU Uncorrectable Error (670)
- System Correctable Error (620)
- System Uncorrectable Error (660)
- System Environmental Error (680)

6.7.4.1 CPU Correctable Errors (SCB: 630)

CPU correctable errors are generic Alpha 21264 microprocessor correctable errors. The following EV6-detected conditions cause PALcode to invoke the operating system 630 error handler:

Error Code (Hex)	Machine Check Type	Error Description
86	Correctable	B-cache probe hit single-bit ECC error
9E	Correctable	D-cache tag parity error on issue
86	Correctable	I-cache tag or data parity error
86	Correctable	D-cache victim single-bit ECC error
86	Correctable	B-cache single-bit ECC fill error to I-stream or D-stream
86	Correctable	Memory single-bit ECC fill error to I-stream or D-stream

6.7.4.2 CPU Uncorrectable Error (SCB: 670)

CPU uncorrectable errors are fatal microprocessor machine check errors that result in a system crash. The error conditions are:

Error Code (Hex)	Machine Check Type	Error Description
8E	CPU uncorrect	PAL detected bugcheck error
90	CPU uncorrect	Operating system detected bugcheck error
98	CPU uncorrect	EV6 detected second D-cache store EEC error
98	CPU uncorrect	EV6 detected D-cache tag parity error in pipeline 0 or 1
98	CPU uncorrect	EV6 detected duplicate d-cache tag parity error
98	CPU uncorrect	EV6 detected double-bit ECC memory fill error
98	CPU uncorrect	EV6 detected double-bit probe hit EEC error
98	CPU uncorrect	EV6 detected B-cache tag parity error

6.7.4.3 System Correctable Error (SCB: 620)

These nonfatal errors are *Compaq AlphaServer* ES40-specific correctable errors. The following condition causes the PALcode to invoke the operating system 620 error handler:

Error Code (Hex)	Machine Check Type	Error Description
204	System correct	System detected ECC single-bit error

6.7.4.4 System Uncorrectable Error (SCB: 660)

A system uncorrectable error is a system-detected machine check that occurred as a result of an “off-chip” request to the system. The following conditions cause PALcode to build the 660/670 machine check logout frame and invoke the 660 error handler:

Error Code (Hex)	Machine Check Type	Error Description
202	System uncorr	Uncorrectable ECC error
202	System uncorr	Nonexistent memory reference
202	System uncorr	PCI system bus error (SERR)
202	System uncorr	PCI read data parity error (RDPE)
202	System uncorr	PCI address/command parity error (APE)
202	System uncorr	PCI no device select (NDS)
202	System uncorr	PCI target abort (TA)
202	System uncorr	Invalid scatter/gather page table entry (SGE) error
202	System uncorr	PCI data parity error (PERR)
		Flash ROM write error
202	System uncorr	PCI target delayed completion retry time-out (DCRTO)
202	System uncorr	PCI master retry time-out (RTO 2**24) error
		PCI-ISA software NMI error

6.7.4.5 System Environmental Error (SCB: 680)

A system environmental error is a system detected machine check caused by an overtemperature condition, fan failure, or power supply failure. The error conditions are:

Error Code (Hex)	Machine Check Type	Error Description
206	Uncorrectable	Over-temperature failure (>50°C)
206	Uncorrectable	Complete power supply failure
206	Correctable	Fan failure
206	Correctable	Power supply failure
206	Correctable	High temperature warning (>45°C and <50°C)

6.7.4.6 Error Logging and Event Log Entry Format

The *Tru64 UNIX* and *OpenVMS* error handlers generate several entry types. Error entries, except for correctable memory errors, are logged immediately. Entries can be of variable length based on the number of registers within the entry. Each entry consists of an operating system header, several device frames, and an end frame. Most entries have a PAL-generated logout frame, and may contain frames for CPU, memory, and I/O.

Figure 6-2 shows an error log event structure map for a *Tru64 UNIX* or *OpenVMS* system uncorrectable PCI target abort error.

Figure 6-2 Sample Error Log Event Structure Map

OFFSET(hex)	63 56	55 48	47 40	39 32	31 24	23 16	15 8	7 0
ech0000 ech+nnnn	NEW COMMON OS HEADER							
lfh0000 lfh+nnnn	STANDARD LOGOUT FRAME HEADER							
lfev60000 lfev6+nnnn	COMMON PAL EV6 SECTION (first 8 QWs Zeroed)							
lfctt_A0[u]	SESF<63:32> = Reserved(MBZ)		<39:32>= (MBZ)		SESF<31:16> = Reserved(MBZ)		SESF<15:0>= 0002(hex)	
lfctt_A8[u]	Cchip CPUx Device Interrupt Request Register (DIRx<61> = 1)							
lfctt_B0[u]	Cchip Miscellaneous Register (MISC)							
lfctt_B8[u]	Pchip0 Error Register (P0_PERROR<63:0> = 0)							
lfctt_C0[u]	Pchip1 Error Register (P1_PERROR<51>=0;<47:18>=PCI Addr;<17:16>=PCI Opn; <6>=1)							
lfett_C8[u] lfett_138[u]	Pchip1 Extended Tsunami/Typhoon System Packet							
eelcb_140 eelcb_190 eelcb_1E0 eelcb_230 eelcb_280 eelcb_2D0	Pchip 1 PCI Slot 4 Single Device Bus Snapshot Packet Pchip 1 PCI Slot 5 Single Device Bus Snapshot Packet Pchip 1 PCI Slot 6 Single Device Bus Snapshot Packet Pchip 1 PCI Slot 7 Single Device Bus Snapshot Packet Pchip 1 PCI Slot 8 Single Device Bus Snapshot Packet Pchip 1 PCI Slot 9 Single Device Bus Snapshot Packet							
2D8	Termination or End Packet							

6.7.5 Errors Captured by SRM Console

When the system is running PALcode, neither double error halts nor machine checks result in error log entries. These errors cause the machine to return to the SRM console, not to the operating system. The SRM console displays an error message. Use the SRM **info** command to help determine the error.

6.7.5.1 PALcode Overview

PALcode, privileged architecture library code, is used to implement a number of functions at the machine level without the use of microcode. This allows operating systems to make common calls to PALcode routines without knowing the hardware specifics of each system the operating system is running on. PALcode routines handle:

- Instructions that require complex sequencing, such as atomic operations
- Instructions that require VAX-style interlocked memory access
- Privileged instructions
- Memory management
- Context swapping
- Interrupt and exception dispatching
- Power-up initialization and booting
- Console functions
- Emulation of instructions with no hardware support

6.7.5.2 Double Error Halt While in PAL

A double error halt occurs under the following conditions:

- A machine check occurs.
- PAL completes its tasks and returns control of the system to the operating system.
- A second machine check occurs before the operating system completes its tasks.

The machine returns to the console and displays the following message:

```
halt code = 6
double error halt
PC  =  20000004
Your system has halted due to an irrecoverable error. Record the
error halt code and PC and contact your Compaq Services
representative. In addition, type INFO 5 and INFO 8 at the console
and record the results.
```

The **info 5** command causes the SRM console to read the PAL-built logout area that contains all the data used by the operating system to create the error entry.

The **info 8** command causes the SRM console to read the IOD 0 and IOD 1 registers.

6.7.5.3 Machine Check While in PAL

If a machine check occurs while the system is running PALcode, the machine returns to the SRM console, not to the operating system. The SRM console writes the following message:

```
halt code = 7
machine check while in PAL mode
PC  =  20000004
Your system has halted due to an irrecoverable error. Record the
error halt code and PC and contact your Compaq Services
representative. In addition, type INFO 3 and INFO 8 at the console
and record the results.
```

The **info 3** command causes the SRM console to read the “impure area,” which contains the state of the CPU before it entered PAL.

6.7.6 Errors Captured by RMC

The remote management console provides several commands for accessing error information in the shared RAM. These commands can be entered at a remote console and can be used if you are not able to access the SRM console because of a system crash. The commands are **dump**, **env**, and **status.**

6.7.6.1 dump

The **dump** command dumps unformatted data from the shared RAM. The data can then be analyzed by service providers. The **dump** command allows you to dump data remotely from the RMC. You can also use this command if you are not able to access the SRM console because of a system crash. For example, you can dump the data from an uncorrected machine check.

The **dump** command accepts two arguments:

addr> prompts for the starting address

count> prompts for the number of following consecutive bytes. If no count is specified, the count defaults to 0.

In the following example, the **dump** command dumps 5 bytes from address 10.

```
RMC>dump
Address: 10
Count: 5
0010:13 17 58 11 02 47
```

In the following example, the **dump** command dumps 20 bytes from address 3200.

```
RMC>dump
Address: 3200
Count: 20
3200:20 01 2F 43 01 00 88 00 01 10 00 01 20 00 01 32
3210:00 00 00 00 00 12 20 00 10 20 00 13 00 01 76 00
3220:02
```

In the following example, the **dump** command dumps 10 bytes from address 3400.

```
RMC>dump
Address: 3400
Count: 10
3400:04 08 80 00 00 00 00 00 00 00 00 00 00 00 00 00
3410:00
RMC>
```

In the following example, the **dump** command dumps FE bytes from address 10.

```
RMC>dump
addr> 10
count> fe
0010:03 31 07 28 01 09 00 00 00 00 00 00 00 00 00 00
0020:00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0030:00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0040:01 80 01 01 01 01 01 01 00 00 00 00 00 00 00 00
0050:00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0060:00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0070:00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0080:00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0090:00 00 00 00 00 00 00 00 00 00 1D 00 19 18 19 00
00A0:00 00 00 00 00 00 00 00 00 00 00 FF FF FA FA 3B
00B0:00 00 00 00 00 00 00 00 00 00 BA 00 00 00 00 00
00C0:00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00D0:00 00 00 00 00 00 00 00 00 00 22 00 00 00 00 00
00E0:00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00F0:00 00 00 00 00 00 00 00 00 10 00 00 00 0A 03 0A
RMC>
```

6.7.6.2 env

The **env** command displays the system environmental status, including power supplies, voltages, fans, and temperatures. If a fault has occurred, the reading blinks. The **env** command is fully described in Chapter 4.

6.7.6.3 error

The **error** command is similar to the SRM **show error** command. The RMC reads information in the shared RAM FRU EEROM locations to determine what errors have occurred. The following example shows an error on CPU0.

```
RMC>error
dev>cpu0
Entry  Fail Address  Bank   Source Event Type
0      0be21e00      1      1      00
RMC>
```

6.7.6.4 fru

The **fru** command is similar to the SRM **show fru** command. It is used to access information in the shared RAM FRU EEROM locations based on the logical device name you enter at the dev> prompt. The logical device names are:

Name	Meaning
cpu0, cpu1, cpu2, cpu3	CPUs
dimm {0-31}	Memory DIMMs
mmb {0, 1, 2, 3}	Memory motherboards
sys	System board
pci	PCI backplane

The following example shows information on the CPU0 module.

```
RMC>fru
dev>cpu0
Slot   Part#         Rev           Serial#       Events Logged
                                                 SDD    TDD
0      25241111      A1            KA426C0457    00     00
RMC>
```

6.8 Option Card Problems

Option card problems can include problems related to network options and PCI options.

Network Problems

Network problems can vary, depending on the type of network option card that you have installed. See the option card documentation for information on troubleshooting network problems. Make sure you have correctly set the network type for the network interface card.

PCI Parity Errors

Some PCI devices do not implement PCI parity, and some have a parity-generating scheme that may not comply with the PCI Specification. In such cases, the device functions properly as long as parity is not checked.

You can turn off parity checking so that false PCI parity errors do not result in machine check errors. When you disable PCI parity, no parity checking is implemented for any PCI device.

For *Tru64 U*NIX and *OpenVMS* systems, use the **set pci_parity off** command from the SRM console.

PCI Bus Problems

PCI bus problems at startup are usually indicated by the inability of the system to detect the PCI device. Use Table 6-12 to diagnose the likely cause of the problem.

Table 6-12 Troubleshooting PCI Bus Problems

Step	Action
1	Check the cabling and confirm that the PCI card is correctly seated.
2	Run system console PCI diagnostics for devices on the Supported Options List. (If the device is not on the list, refer to the device's documentation.) • Storage adapter—Run the **test** command to exercise the storage devices off the PCI controller option. • Ethernet adapter—Run the **test** command to exercise an Ethernet adapter.
3	Check for a bad slot by moving the suspected controller to a different slot.
4	Contact the option manufacturer.

A
Hardware Specifications

Table A-1 contains the physical, environmental, and electrical specifications for the *AlphaServer* ES40sv rackmount system.

Table A-1 *AlphaServer* ES40sv Rackmount System Specifications

Physical Characteristics	
Chassis	
Weight	61.24 kg (135 lb.)
Length	63.5 cm (25 in.)
Width	44.45 cm (17.5 in.)
Height	35.56 cm (14.00 in.)
Environmental Requirements	
Temperature range	10°C to 35°C (50°F to 95°F)
Temperature change rate	11°C/hr (20°F/hr) maximum
Wet bulb temperature (maximum)	28°C (82°F)
Dew point temperature (minimum)	2°C (36°F)
Relative humidity	10% to 95% (noncondensing)
Air intake location	Front
Air exhaust location	Rear
Heat dissipation (maximum)	3917 BTU/hr
Altitude	-60 m to 3048 m (-200 ft to 10,000 ft)
Electrical Requirements	
Nominal voltage	120 to 240 Vac
Operational voltage range	90 to 264 Vac
Frequency	50 – 60 Hz
Input current (maximum)	10.3 A at 120 Vac
	5.15 A at 240 Vac
Input power (maximum)	1250 W

B

Field Replaceable Units (FRUs)

The major field replaceable units (FRUs) and their part numbers for the *AlphaServer* ES40sv rackmount system are listed in Table B-1.

Table B-1 *AlphaServer* ES40sv Rackmount System FRUs

Part Description	Part Number
CPU Module (replacement for failed modules)	54-30158-03
Memory Motherboard	54-25582-01
DIMM pair (32 MB) (64 MB) (128 MB) (256 MB) (512 MB) (1 GB)	54-25066-BA 54-25053-BA 54-24941-EA 54-24941-FA 54-xTBDx-xx * 54-xTBDx-xx *
Floppy Drive	RX23L-AA or current model
CD-ROM Drive	RRD47-VA or current model
Power Supply	30-50703-01
OCP Assembly	70-33989-01
CPU Fan Assembly	70-33995-01
PCI Card-Cage Fan Assembly	70-33996-01
I/O Port Module	70-34016-01
PCI Backplane	54-30156-01
System Motherboard	54-25385-01
Speaker	70-31349-01
Receiver Frame	30-50963-02

* Contact Compaq for part availability.

C
Jumpers and Switches

C.1 System Motherboard Jumpers and Switches

Table C-1 lists the jumpers on the system motherboard along with the name of the jumper and a description of the settings. Figure C-1 shows the location of the system motherboard jumpers.

Table C-1 System Motherboard Jumpers

Jumper	Name	Description
J1	Debug	Used for debug. When installed, bypasses power-up checks of processors by system power controller.
J2	Reserved	Reserved (not installed).
J3	Reserved	Reserved (not installed).
J12/13	Flash Write	1–2: Normal operation 2–3: Disable Flash write
J21	Load TIG	1–2: Load TIG from flash RAM (default) 2–3: Load TIG from serial ROM. This setting allows you to load the TIG if the flash RAM is corrupted.
J20	Fail-safe Loader	Must be in default positions over pins 1 and 2 to enable FSL. FIR_FUNC2 1–2 = 0, 2–3 = 1
J22	Fail-safe Loader	Jumper for enabling fail-safe loader (FSL) FIR_FUNC1 1–2= 0, 2–3= 1
J23	Fail-safe Loader	Must be in default positions over pins 1 and 2 to enable FSL. FIR_FUNC0 1–2= 0, 2–3 = 1
J24	RMC Flash Update	1-2: Disables RMC flash update 2-3: Enables RMC flash update (default)
J25	Setting RMC to Defaults	1-2: Sets RMC back to defaults 2-3: Normal RMC operating mode (default)
J26	Over-Temperature Shutdown	1-2: Causes system to shut down if over-temperature limit is reached (default) 2-3: Permits system to continue running at over-temperature
J31	COM1 Bypass	1-2: Allows RMC to control COM1 bypass (default) 2-3: Disables COM1 bypass No jumper installed: Forces COM1 bypass

The following are the settings for the FIR_FUNC jumpers (J20, J22, and J23).

Setting	Meaning
000	Normal
001	Prevent flash loads. Load from SROM.
010	Load from floppy
111	Lock console. This setting prevents the writing of flash from CPUs.

Table C-2 lists the switches on the system motherboard switchpack E296 along with the switch name and the default position. Figure C-1 shows the location of switchpack E296.

Table C-2 System Motherboard Switchpack E296 Settings

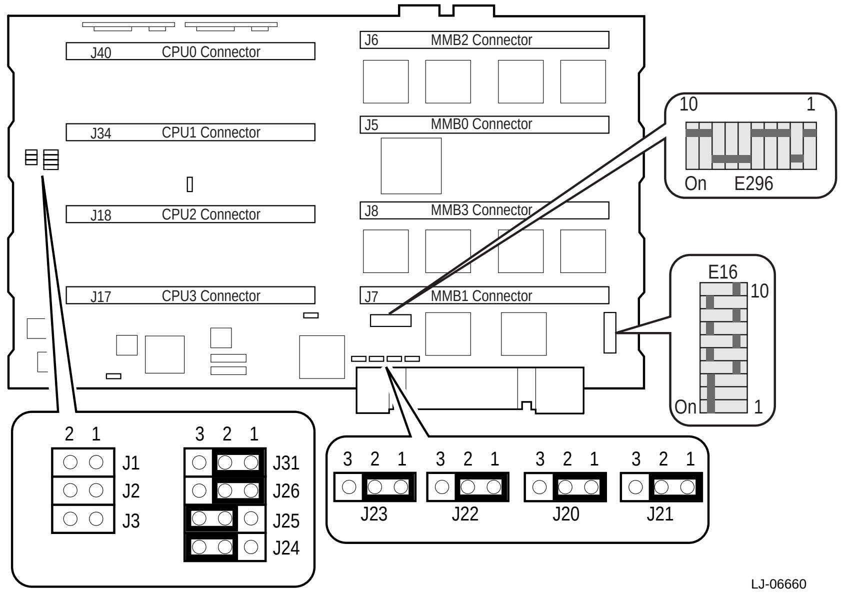
Switch	Name	Default Position
SW1	SYS_EXT_DELAY1	Off
SW2	SYS_EXT_DELAY0	On
SW3	SYS_FILL_DELAY	Off
SW4	CPU_CFWD_PSET	Off
SW5	PCI_CLK_DIV_IN1	Off
SW6	PCI_CLK_DIV_IN0	On
SW7	Y_DIV3	On
SW8	Y_DIV2	On
SW9	Y_DIV1	Off
SW10	Y_DIV0	Off

Table C-3 lists the switches on the system motherboard switchpack E16 along with the switch name and the default position. This switchpack is used to set the system motherboard MAIN CLOCK frequency. Figure C-1 shows the location of switchpack E16.

Table C-3 System Motherboard Switchpack E16 Settings

Switch	Name	Default Position
SW1	M0	On
SW2	M1	On
SW3	M2	On
SW4	M3	Off
SW5	M4	On
SW6	M5	Off
SW7	M6	On
SW8	N0	Off
SW9	N1	On
SW10	XTAL_SEL	Off

Figure C-1 System Motherboard Jumper and Switch Locations



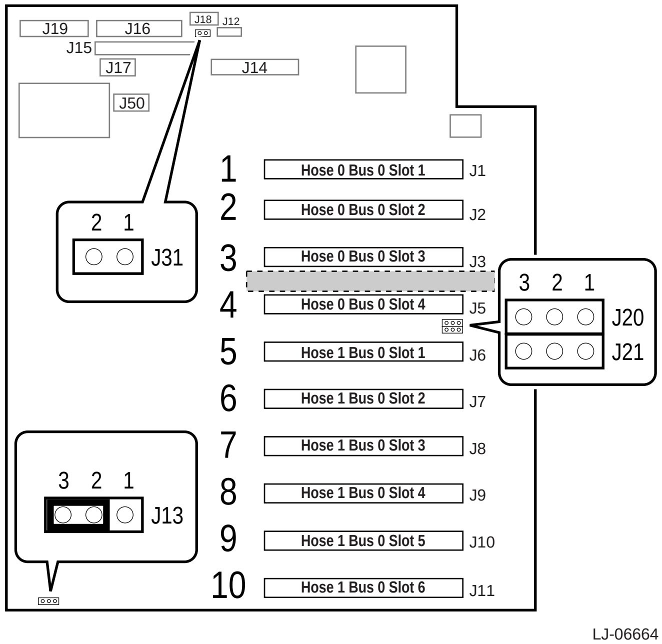
C.2 PCI Backplane Jumpers

Table C-4 lists the jumpers on the PCI backplane along with the name of the jumper and a description of the settings. Figure C-2 shows the location of the PCI backplane jumpers.

Table C-4 PCI Backplane Jumpers

Jumper	Name	Description
J13	VBAT to RTC	1–2: Enable VBAT to real-time clock (RTC) chip 2–3: Disable VBAT to RTC chip. The setting on this jumper should be checked if you lose time between power cycles or if the operating system comes up with a very inaccurate time.
J20	PCI 0 PME	1–2: Enable PCI 0 power management events (PME). 2–3: Disable PCI 0 PME (default)
J21	PCI 1 PME	1–2: Enable PCI 1 PME 2–3: Disable PCI 1 PME
J31	COM1 DTR	1–2: Do not force COM1 DTR 2–3: Force COM1 DTR (default) This jumper is provided so that you can force DTR. The default position prevents disconnection of the modem on a power cycle.

Figure C-2 PCI Backplane Jumper Locations

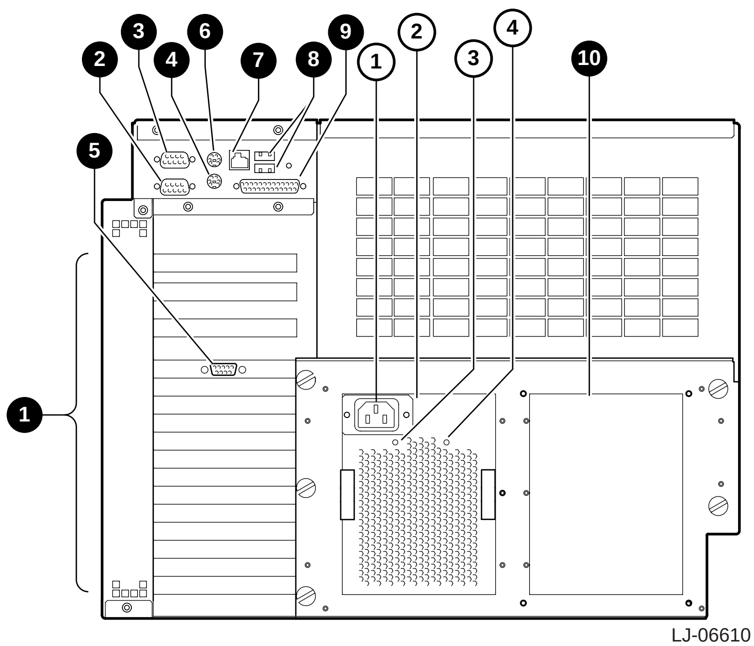


D
Connectors

D.1 Rear Chassis Connectors and Components

Figure D-1 shows the rear connectors and components on the *AlphaServer* ES40sv rackmount system chassis.

Figure D-1 Rear Chassis Connectors and Components

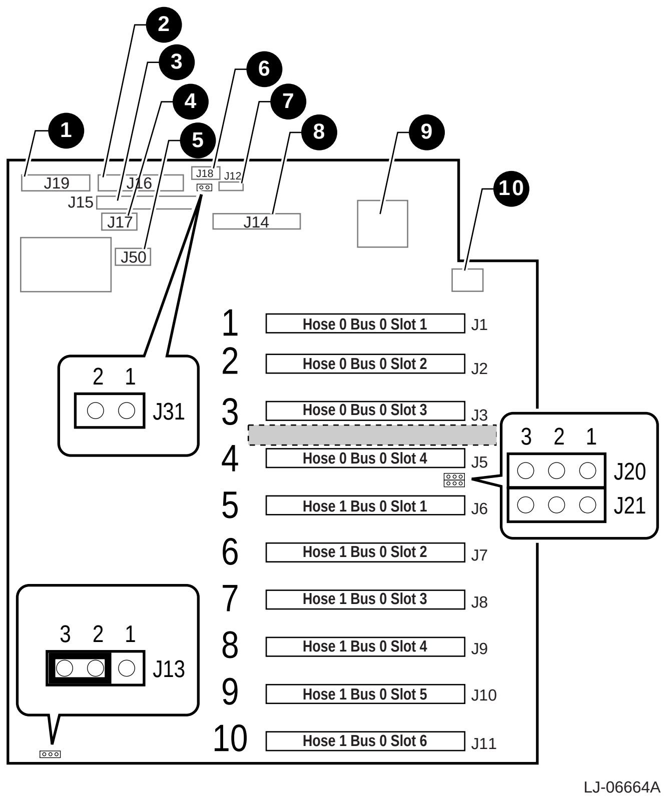


❶ PCI board slots
❷ Modem port
❸ Serial port (COM2)
❹ Keyboard port
❺ PCI slot for VGA controller
❻ Mouse port
❼ Serial port (COM1)
❽ USB ports
❾ Parallel port
❿ Optional power supply cover
① AC input receptacle
② Power supply
③ Power OK LED
④ +5V Aux LED

D.2 PCI Backplane Connectors

Figure D-2 shows the PCI backplane connectors.

Figure D-2 PCI Backplane Connectors



❶ Fan connector J19

❷ Floppy drive connector J16

❸ IDE connector J15

❹ OCP connector J17

❺ Safety interlock connector J50 (not used)

❻ Door connector J18 (bypassed)

❼ Speaker connector J12

❽ I/O port module connector J14

❾ 1543C PCI-ISA and Super I/O Chip

❿ RTC Chip