COMPAQ

# AlphaServer ES45

## Service Guide

Order Number: EK-ES450-SV. A01

This manual is for service providers and self-maintenance customers responsible for ES45 systems.

Compaq Computer Corporation

**First Printing, July 2001**

Compaq shall not be liable for technical or editorial errors or omissions contained herein. The information in this document is provided "as is" without warranty of any kind and is subject to change without notice. The warranties for Compaq products are set forth in the express limited warranty statements accompanying such products. Nothing herein should be construed as constituting an additional warranty.

### FCC Notice

This equipment generates, uses, and may emit radio frequency energy. The equipment has been type tested and found to comply with the limits for a Class A digital device pursuant to Part 15 of FCC rules, which are designed to provide reasonable protection against such radio frequency interference.

Operation of this equipment in a residential area may cause interference in which case the user at his own expense will be required to take whatever measures may be required to correct the interference.

Any modifications to this device—unless expressly approved by the manufacturer—can void the user's authority to operate this equipment under part 15 of the FCC rules.

### Modifications

The FCC requires the user to be notified that any changes or modifications made to this device that are not expressly approved by Compaq Computer Corporation may void the user's authority to operate the equipment.

### Cables

Connections to this device must be made with shielded cables with metallic RFI/EMI connector hoods in order to maintain compliance with FCC Rules and Regulations.

### Taiwanese Notice

警告使用者:

這是甲類的資訊產品，在居住的環境中使用時，可能會造成射頻干擾，在這種情況下，使用者會被要求採取某些適當的對策。

### Japanese Notice

に基づくクラスＡ情報技術装置です。この装置を家庭環境で使用すると電波妨害を引き起こすことがあります。この場合には使用者が適切な対策を講ずるよう要求されることがあります。

### Canadian Notice

This Class A digital apparatus meets all requirements of the Canadian Interference-Causing Equipment Regulations.

### Avis Canadien

Cet appareil numérique de la classe A respecte toutes les exigences du Règlement sur le matériel brouilleur du Canada.

### European Union Notice

Products with the CE Marking comply with both the EMC Directive (89/336/EEC) and the Low Voltage Directive (73/23/EEC) issued by the Commission of the European Community.

Compliance with these directives implies conformity to the following European Norms (in brackets are the equivalent international standards):

EN55022 (CISPR 22) - Electromagnetic Interference

EN50082-1 (IEC801-2, IEC801-3, IEC801-4) - Electromagnetic Immunity

EN60950 (IEC950) - Product Safety

### Warning!

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

### Achtung!

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

### Attention!

Ceci est un produit de Classe A.  Dans un environnement domestique, ce produit risque de créer des interférences radioélectriques, il appartiendra alors à l'utilisateur de prendre les mesures spécifiques appropriées.

# Contents

## Chapter 3 Power-Up Diagnostics and Display

## Chapter 4 SRM Console Diagnostics

## Chapter 5 Error Logs

## Chapter 6 System Configuration and Setup

## Chapter 7 Using the Remote Management Console

## Chapter 8 FRU Removal and Replacement

## Appendix A SRM Console Commands

## Appendix B Jumpers and Switches

## Appendix C DPR Address Layout

## Appendix D Registers

## Appendix E Isolating Failing DIMMs

## Examples

## Figures

## Tables

# Preface

## Intended Audience

This manual is for service providers and self-maintenance customers who are responsible for servicing ES45 systems.

## Document Structure

This manual uses a structured documentation design. Topics are organized into small sections, usually consisting of two facing pages. Most topics begin with an abstract that provides an overview of the section, followed by an illustration or example. The facing page contains descriptions, procedures, and syntax definitions.

This manual has eight chapters and five appendixes.

- Chapter 1, System Overview, provides system overview information.
- Chapter 2, Troubleshooting, describes the starting points for diagnosing problems.
- Chapter 3, Power-up Diagnostics and Display, describes the power-up process and RMC, SROM, and SRM power-up diagnostics.
- Chapter 4, SRM Console Diagnostics, provides troubleshooting information with the SRM console.
- Chapter 5, Error Logs, provides information for interpreting error logs.
- Chapter 6, System Configuration and Setup, describes how to configure and set up systems.
- Chapter 7, Using the Remote Management Console, provides information for managing the system through remote management console.
- Chapter 8, FRU Removal and Replacement, describes the procedures for removing and replacing FRUs.

- Appendix A, SRM Console Commands, lists the SRM console commands most frequently used with the ES4x family of systems.
- Appendix B, Jumpers and Switches, lists and describes the configuration jumpers and switches on the system motherboard and PCI board.
- Appendix C, DPR Address Layout, shows the address layout of the dual-port RAM.
- Appendix D, Registers, describes 21264 EV68 internal processor registers.
- Appendix E, Isolating Failing DIMMs, explains how to manually isolate a failing DIMM from the failing address and failing data bits.

## Documentation Titles

### Table 1 Compaq AlphaServer ES45 Documentation

| Title | Order Number |
|---|---|
| **User Documentation Kit** | QA–6NUAB–G8 |
| Owner’s Guide | EK–ES450–UG |
| Documentation CD (6 languages) | AG–RPJ8A–TS |
| **Maintenance Kit** | QA–6NUAA–G8 |
| Service Guide | EK–ES450–SV |
| Service Guide HTML CD (IPBs included) | AG–RPJ5A–TS |
| **Loose Piece Items** | |
| Basic Installation Card | EK–ES450–PD |
| Rackmount Installation Guide | EK–ES450–RG |
| Rackmount Installation Template | ES–ES450–TP |

## Information on the Internet

Visit the Compaq Web site at www.compaq.com for service tools and more information about the *AlphaServer* ES45 system.

# Chapter 1
# System Overview

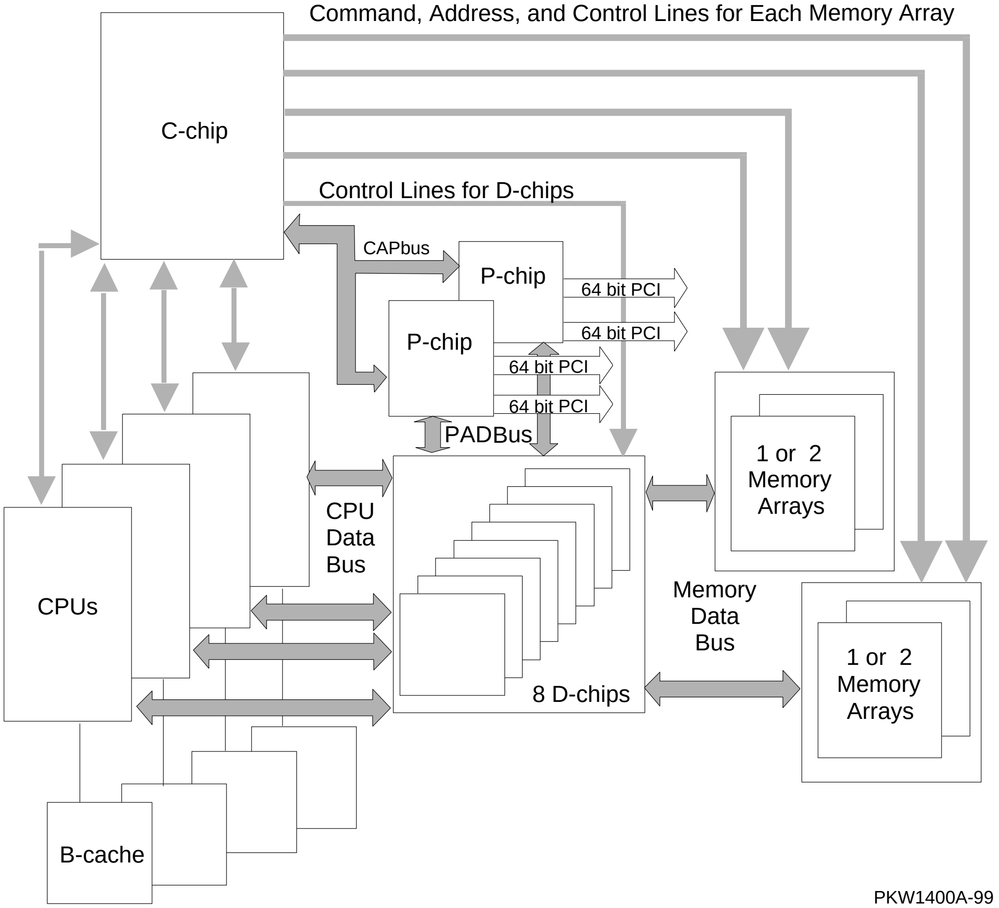
This chapter provides an overview of the system in these sections:

- System Architecture
- System Enclosures
- System Chassis—Front View/Top View
- System Chassis—Rear View
- Hot Swap Module
- I/O Ports and Slots
- Control Panel
- System Motherboard
- CPU Card
- Memory Architecture and Options
- PCI Backplane
- Remote System Management Logic
- Power Supplies
- Fans
- Removable Media Storage
- Hard Disk Drive Storage
- System Access
- Console Terminal

## 1.1 System Architecture

**The system uses a switch-based interconnect system that maintains constant performance even as the number of transactions multiplies.**

### Figure 1–1 System Block Diagram



This system is designed to fully exploit the potential of the Alpha EV68 CB chip by using a switch-based (or point-to-point) interconnect system. With a traditional bus design, the processors, memory, and I/O modules share the bus. As the number of bus users increases, the transactions interfere with one another, increasing latency and decreasing aggregate bandwidth. With a switch-based system, speed is maintained and little degradation in performance occurs as the number of CPUs, memory, and I/O users increases.

The switched system interconnect uses a set of complex microprocessor support chips that route the traffic over multiple paths. This chipset consists of one C-chip, two P-chips, and eight D-chips.

- C-chip. Provides the command interface from the CPUs and main memory. The C-chip allows each CPU to do transactions simultaneously.
- D-chips. Provide the data path for the CPUs, main memory, and I/O.
- P-chips. Provide the interface to the I/O with two buses per chip.

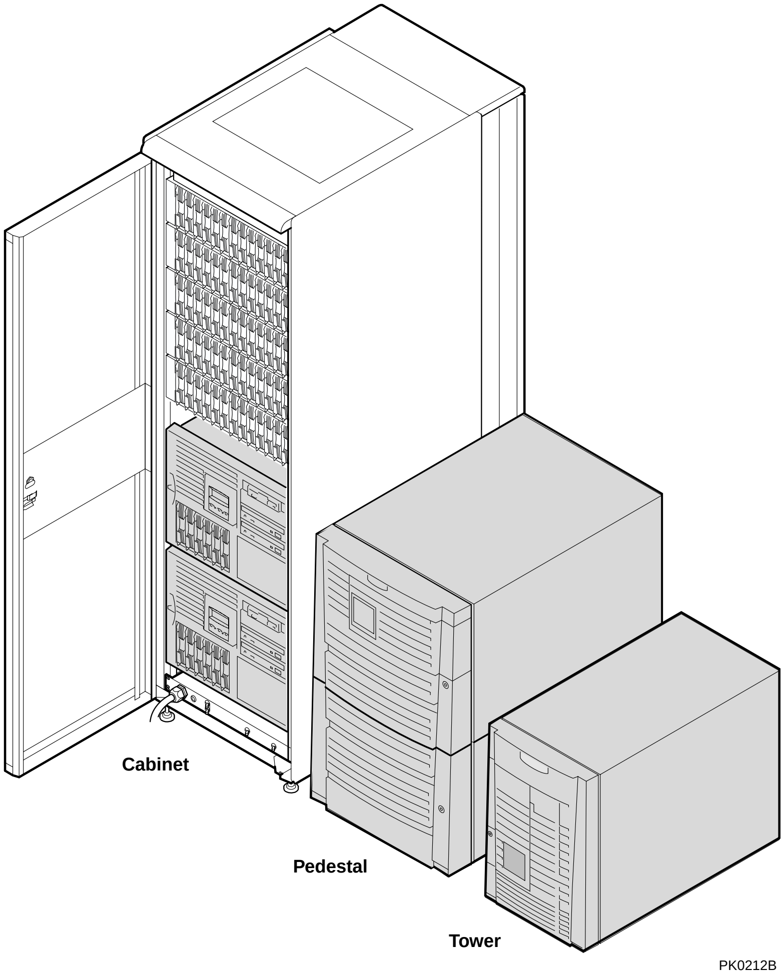
The chipset supports up to four CPUs and up to 32 Gbytes of memory. Interleaving occurs when at least two sibling or nonsibling memory arrays are used.

Two 256-bit memory buses support four memory arrays, yielding a maximum 8 Gbytes/sec system bandwidth. Transactions are ECC protected. Upon the receipt of data, the receiver checks for data integrity and corrects any errors.

## 1.2 System Enclosures

**The ES45 family consists of a standalone tower, a pedestal with expanded storage capacity, and a cabinet.**

Figure 1–2 ES45 Systems



The ES45 system provides connectors for eight DIMMs on each of the memory motherboards (MMBs) and connectors for ten PCI options on the PCI backplane. The system comes with the following:

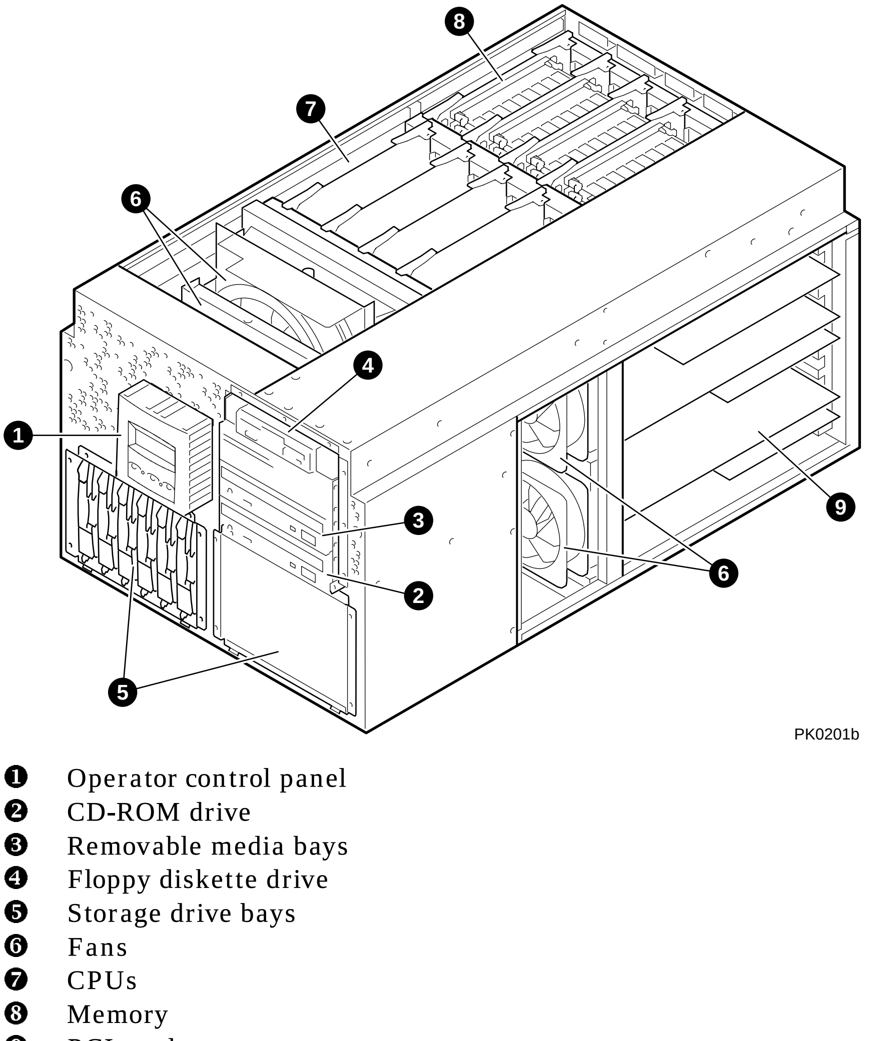
- 1–4 CPUs
- Up to 32 DIMMs (8 DIMMs on each MMB)
- 10 PCI slots (4 – 33 MHz and 6 – 66 MHz)

The following components are common to all ES45 systems:

- Up to four CPUs, based on the EV68 Alpha chip
- Memory DIMMs (200-pin)
- Floppy diskette drive (3.5-inch, high density)
- CD-ROM drive
- Two half-height or one full-height removable media bays
- Up to two storage drive cages that house up to six 1-inch drives per cage
- Up to three power supplies, offering N+1 power
- A 25-pin parallel port, two 9-pin serial ports, mouse and keyboard ports, and one MMJ connector for a local console terminal
- An operator control panel with a 16-character back-lit display and a Power button, Halt button, and Reset button

## 1.3 System Chassis—Front View/Top View

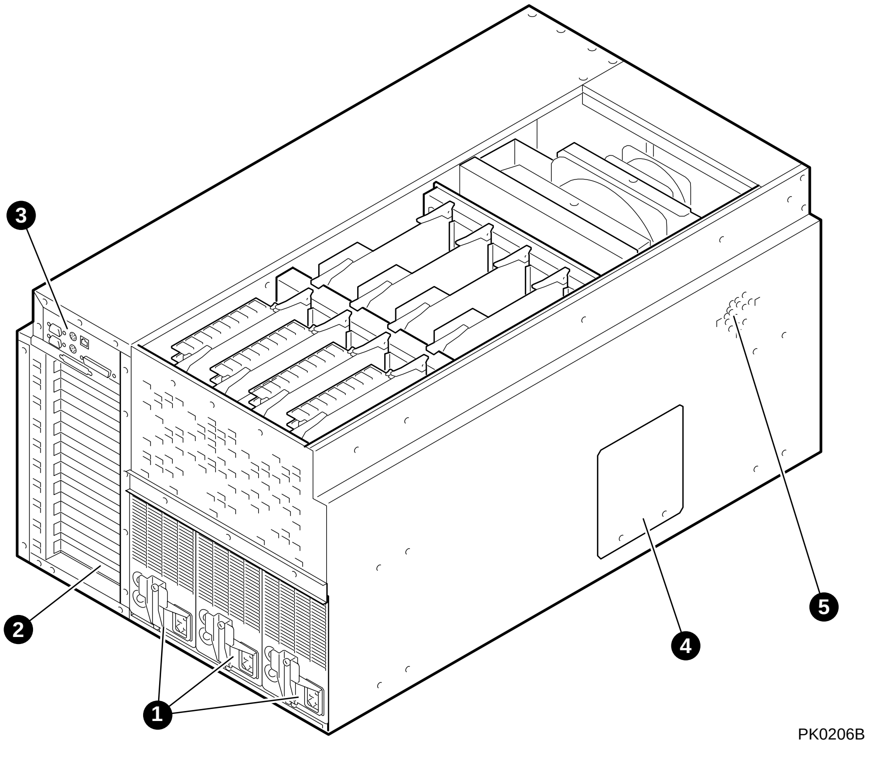
### Figure 1–3 Components Top/Front View (Pedestal/Rackmount Orientation)



❶ Operator control panel
❷ CD-ROM drive
❸ Removable media bays
❹ Floppy diskette drive
❺ Storage drive bays
❻ Fans
❼ CPUs
❽ Memory
❾ PCI cards

## 1.4 System Chassis—Rear View

### Figure 1–4 Rear Components (Pedestal/Rackmount Orientation)



❶ Power supplies

❷ PCI bulkhead

❸ I/O ports

❹ Power harness access cover

❺ Speaker

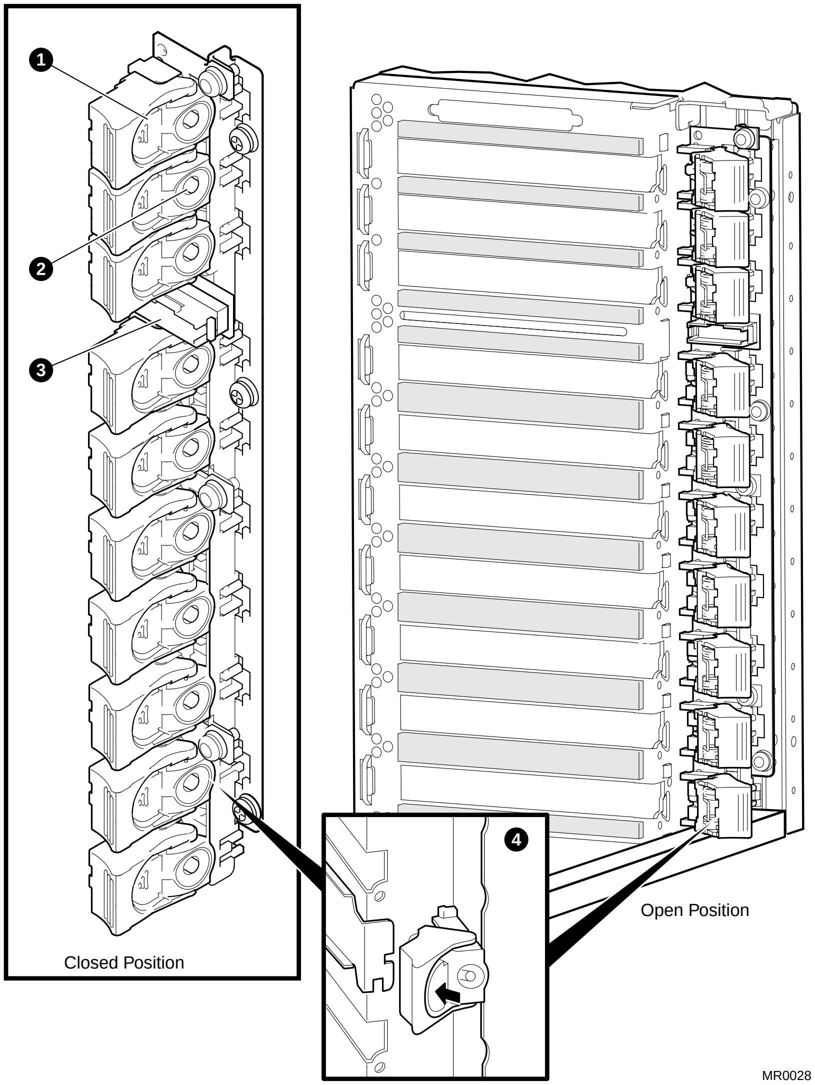
## 1.5 Hot Swap Module

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

**WARNING: To prevent injury, unplug the power cord from each power supply before installing components.**

**CAUTION:** *Hot swap is not currently supported by the operating systems. Do not press switches ❶ and ❷ on the hot swap module while the system is powered. Pressing the switches may result in loss of data.*

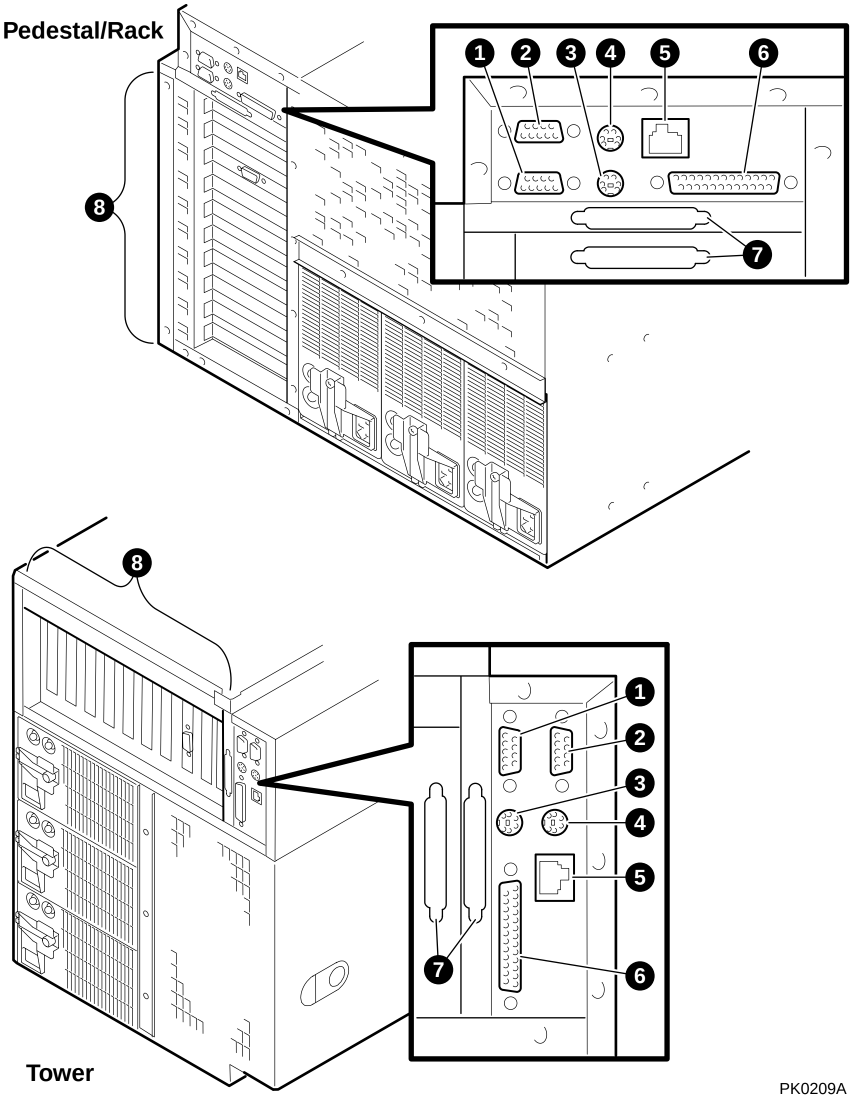
## Figure 1–5 Hot Swap Module



❶ Module release button
❷ Momentary hot swap power switch (Not supported)
❸ Communication connector
❹ Module release button connection

## 1.6 I/O Ports and Slots

### Figure 1–6 Rear Connectors



### Rear Panel Connections

❶ Modem port—Dedicated 9-pin port for connection by modem to remote management console.

❷ COM2 serial port—Extra port to modem or any serial device.

❸ Keyboard port—To PS/2-compatible keyboard.

❹ Mouse port—To PS/2-compatible mouse.

❺ COM1 MMJ-type serial port/terminal port —For connecting a console terminal.

❻ Parallel port—To parallel device such as a printer.

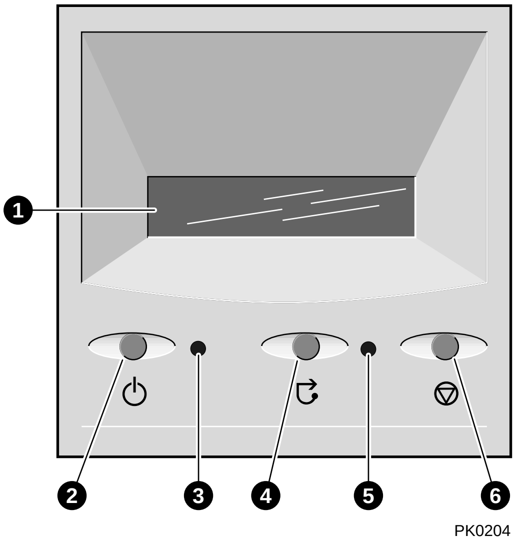
❼ SCSI breakouts.

❽ PCI slots—For option cards for high-performance network, video, or disk controllers.

## 1.7 Control Panel

**The control panel provides system controls and status indicators. The controls are the Power, Halt, and Reset buttons. A 16-character back-lit alphanumeric display indicates system state. The panel has two LEDs: a green Power OK indicator and an amber Halt indicator.**

### Figure 1–7 Control Panel



❶ Control panel display. A one-line, 16-character alphanumeric display that indicates system status during power-up and testing.

❷ Power button. Powers the system on and 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 **env** command from the remote management console (RMC) command line. The RMC is powered separately from the rest of the system and can operate as long as one power supply is plugged in. (See Chapter 7.)

❸ Power LED (green). Lights when the power button is depressed and system power passes initial checks.

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 you press the Halt button.

❻ Halt button. Halts the system.

- If the operating system is running, pressing the Halt button halts the operating system and returns to the SRM console.
- If the Halt button is latched when the system is reset or powered up, the system halts in the SRM console. Systems that are configured to autoboot cannot boot until the Halt button is unlatched.

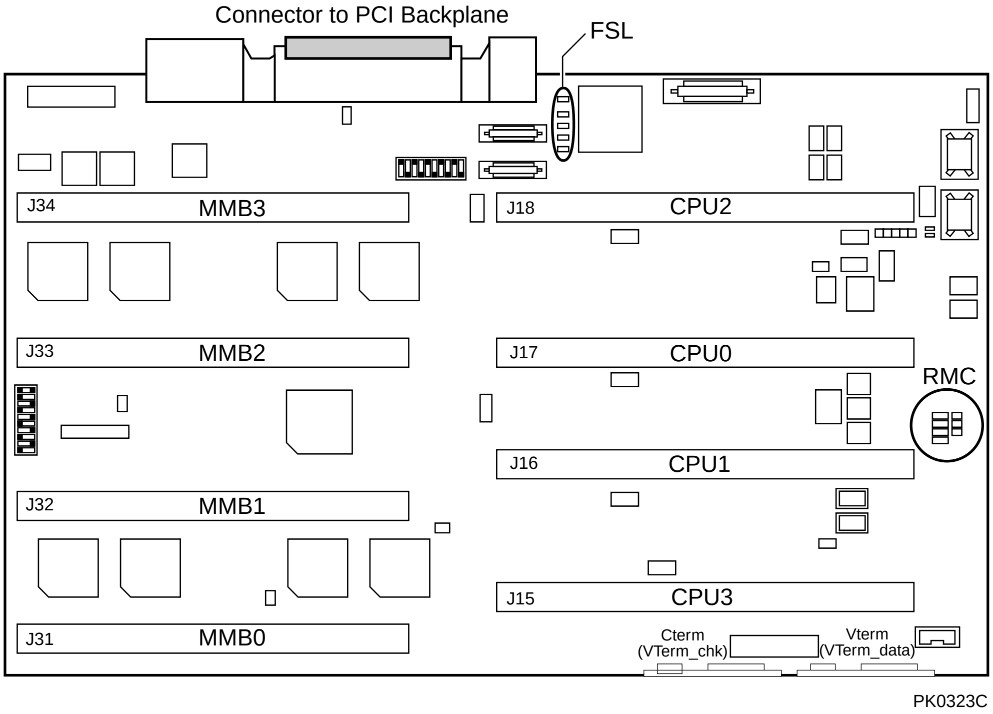
Commands issued from the remote management console (RMC) can be used to reset, halt, and power the system on or off.

| RMC Command | Function |
|---|---|
| **Power {off, on}** | Equivalent to pressing the Power button on the system. If the Power button is in the Off position, the RMC **power on** command has no effect. |
| **Halt {in, out}** | Equivalent to pressing the Halt button on the control panel to cause a halt (**halt in**) or releasing it from the latched position to deassert the halt (**halt out**). |
| **Reset** | Equivalent to pressing the Reset button on the control panel. |

## 1.8 System Motherboard

**The system motherboard is located on the floor of the system card cage. It has slots for the CPUs and memory motherboards (MMBs) and has the PCI backplane interconnect.**

### Figure 1–8 Component and Connector Locations



The system motherboard has the majority of the logic for the system, including:

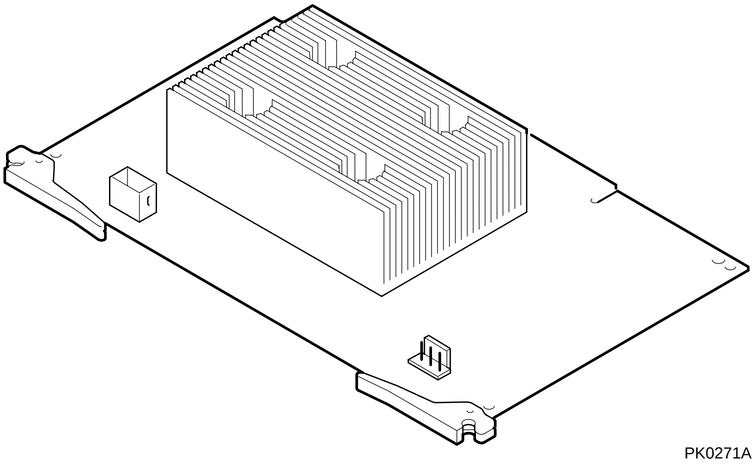
- CPU connectors
- MMB connectors
- Connector to PCI backplane
- RMC jumpers
- Fail-safe loader (FSL) jumpers
- Vterm and Cterm regulators

Figure 1–8 shows the location of components and connectors on the system motherboard.

## 1.9 CPU Card

**An ES45 can have up to four CPU cards. The CPU card has an 8-Mbyte second-level cache and a DC-to-DC converter that provides the required voltage to the Alpha chip. Power-up diagnostics are stored in a flash SROM on the card.**

Figure 1–9 CPU Card



The EV68 CB microprocessor is a superscalar CPU with out-of-order execution and speculative execution to maximize speed and performance. It contains four integer execution units and dedicated execution units for floating-point add, multiply, and divide. It has an instruction cache and a data cache on the chip. Each cache is a 64 KB, two-way, set associative, virtually addressed cache that has 64-byte blocks. The data cache is a physically tagged, write-back cache.

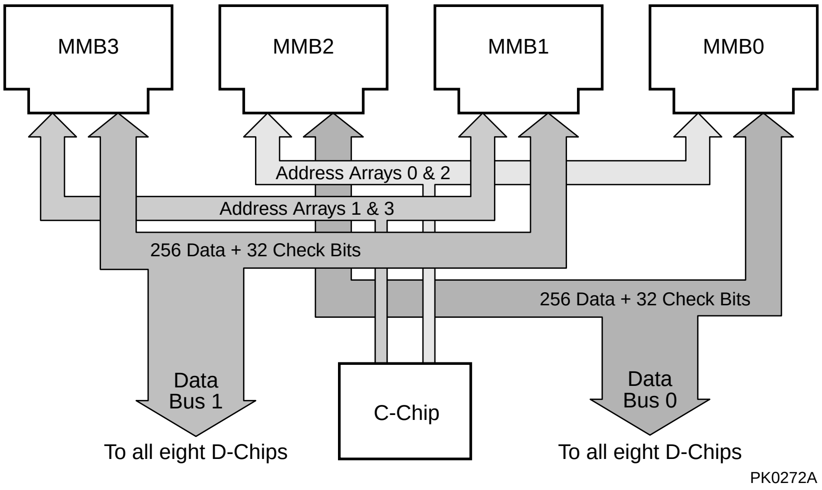
Each CPU card has an 8 MB secondary B-cache (backup cache) consisting of dual data rate (DDR) static RAMs (SRAMs) that provide low latency and high bandwidth. Each CPU card also has a DC–to–DC converter that provides the required voltage to the Alpha chip.

See Chapter 6 for CPU configuration.

## 1.10 Memory Architecture and Options

**The system has two 256-bit wide memory data buses, which can move large amounts of data simultaneously.**

### Figure 1–10 Memory Architecture



### Memory Architecture

Memory throughput in this system is maximized by the following features:

- Two independent, wide memory data buses
- Very low memory latency (120 ns) and high bandwidth with 125 MHz clock
- ECC memory

Each data bus is 256 bits wide (32 bytes). The memory bus speed is 125 MHz. This yields 4 GB/sec bandwidth per bus (32 x 125 MHz = 4 GB/sec). The maximum bandwidth is 8 GB/sec.

The switch interconnect design takes full advantage of the capabilities of the two wide data buses. The 256 data bits are distributed equally over two memory motherboards (MMBs). Simultaneously, in a read operation, 128 bits come from one MMB and the other 128 bits come from another MMB, to make one 256-bit read. Another 256-bit read operation can occur at the same time on the other independent data bus.

In addition, two address buses per MMB (one for each array) allow overlapping/ pipelined accesses to maximize use of each data bus. When all arrays are identical (same size), the memory is interleaved; that is, sequential blocks of memory are distributed across all four arrays.

### Memory Options

Each memory option consists of a set of four 125 MHz, 200-pin JEDEC-standard DIMMs with PECL clocks. The DIMMs are synchronous DRAMs. Memory options are available in the following sizes:

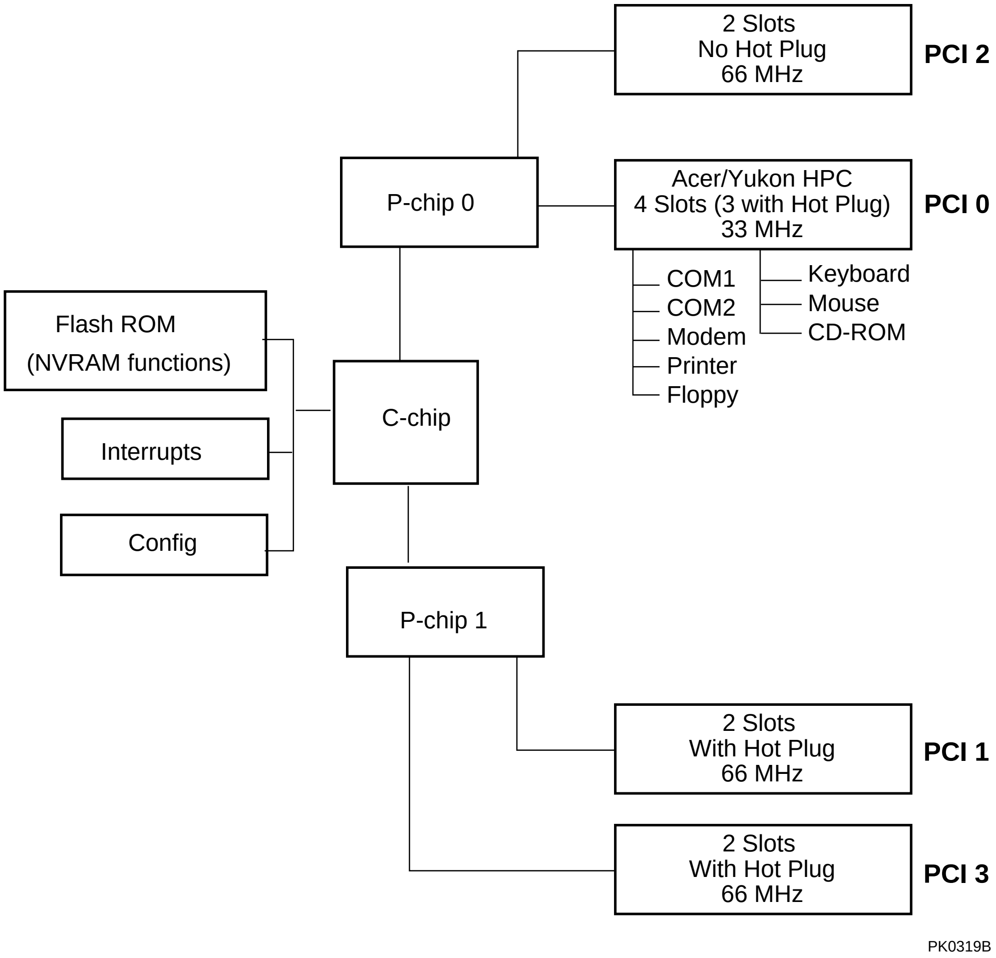
- 512 Mbytes (128 MB DIMMs)
- 1 Gbyte (256 MB DIMMs)
- 2 Gbytes (512 MB DIMMs)
- 4 Gbytes (1 GB DIMMs)

Memory options are installed into memory motherboards (MMBs) located on the system motherboard (see Figure 1–8). There are four MMBs. The MMBs have either four or eight slots for installing DIMMs. See Chapter 6 for memory configuration.

## 1.11 PCI Backplane

**The PCI backplane has four independent 64-bit, PCI buses, one at 33 MHz and three at 66 MHz. The PCI buses support 3.3 volt and 5 volt options.**

### Figure 1–11 I/O Control Logic



PCI modules are either designed specifically for 5.0 or 3.3 volt slots, or are universal in design and can plug into either 3.3 or 5.0 volt slots.

**CAUTION:** *PCI modules designed specifically for 5.0 volts or 3.3 volts are keyed differently. Check the keying before you install the PCI module and do not force it in. Plugging a module into a wrong slot can damage it.*

### PCI Bus Implementation

- Is fully compliant with the PCI Version 2.2 Specification.
- Operates at delivering a peak bandwidth of 1.8 GB/sec; over 533 Mbytes/sec for each 66 MHz bus and over 266 Mbytes/sec for the 33 MHz bus.
- Has ten option slots (seven of which are hot swap).
- Supports three address spaces: PCI I/O, PCI memory, and PCI configuration space
- Supports byte/word, tri-byte, quadword, and longword operations
- Exists in noncached address space only

### I/O Implementation

The system has 10 I/O slots, with six slots at 66 MHz and four slots at 33 MHz. The Acer Labs 1543C chip provides the bridge from PCI 0 to ISA. The C-chip controls accesses to memory on behalf of both P-chips.

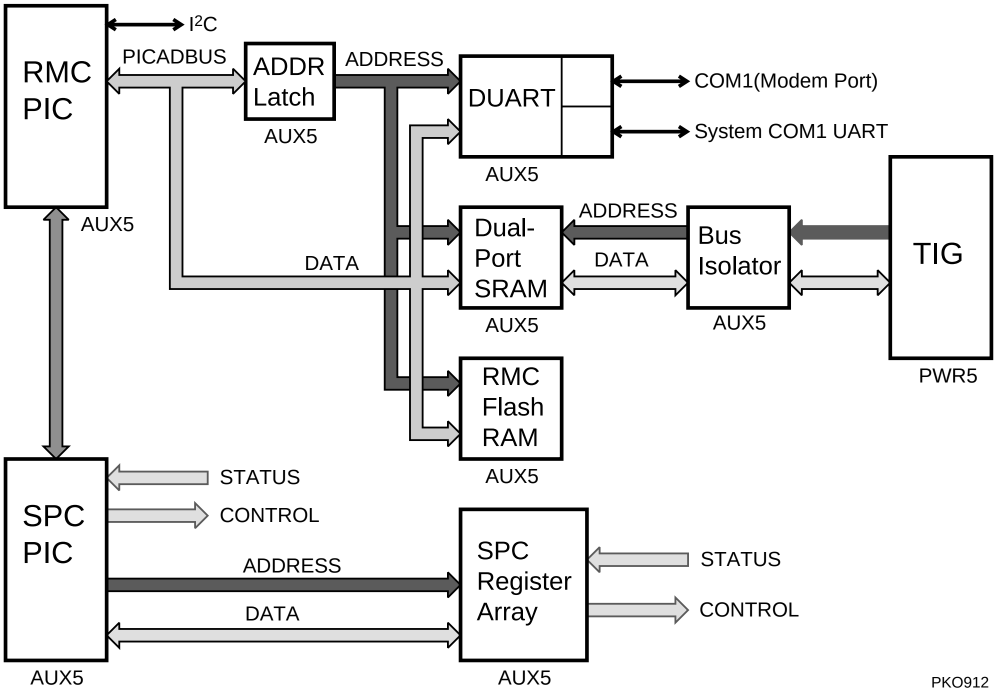
### I/O Ports

The I/O ports are shown in Section 1.6.

## 1.12 Remote System Management Logic

**The remote system management logic consists of two major elements: the system power controller (SPC), used to monitor and control system power supplies, regulators, and cooling apparatus; and the remote management console (RMC), which facilitates remote interrogation and control of the system. The components used within the remote system management logic are powered by the AUX_5V supply, which is always present whenever AC input power is available to the system.**

### Figure 1–12 Remote System Management Logic Diagram



### Dual-Port RAM (DPR)

The ES45 system features a dual-port RAM—RAM that is shared between the RMC and the system motherboard logic—to ease communication between the system and the RMC.  This book refers to the dual-port RAM as the DPR.

The RMC reads 256 bytes of data from each FRU EEPROM at power-up and stores it in the DPR. This data contains configuration and possibly error log information. The data is accessible via the TIG chip to the firmware for configuration information during start-up. Remote or local applications can read the error log  and configuration information. The error log information is written to the DPR by Compaq Analyze (see Chapter 5) and then written back to the EEPROMs by the RMC. This ensures that the error log is available on a FRU after power has been lost.

- Section 1.12.1 describes the SPC logic.
- Section 1.12.2 describes the RMC logic.

## 1.12.1 System Power Controller (SPC)

**The system power controller (SPC) is responsible for sequencing the turn-on/turn-off of all power supplies and regulators, monitoring all system power supplies and regulators, generating hardware resets to all logic elements, and generating power system status signals for use by other functional units within the system. Additionally, it is responsible for emergency shutdown if the internal system temperature exceeds permissible limits.**

An 8-bit CMOS microprocessor (PIC 17C44) with associated programming controls the functions of the SPC. The PIC processor receives inputs from:

- Operator control panel (power-on, reset)
- Power supplies and DC/DC regulators (Power-OK)
- Thermal sensors (temperature failure)
- TIG chip (command bus from the firmware)
- Remote management console logic (remote power up/down, reset)

It provides outputs to:

- Power supplies and DC/DC regulators (power supply enables)
- Processors (DC_OK, reset)
- TIG bus chip (handshake)
- Remote management console (power status)

## 1.12.2 Remote Management Console (RMC)

**The remote management console (RMC) provides a mechanism for remotely monitoring a system and manipulating it on a very low level. It also provides access to the repository for all error information in the system. This provides the operator, either remotely or locally, with the ability to monitor the system (voltages, temperature, fans, error status) and manipulate it (reset, power on/off, halt) without any interaction on the part of the operating system.**

The RMC can also detect alert conditions such as overtemperature, fan failure, and power supply failure and automatically dial a user-defined pager phone number or another computer system to make the remote operator aware of the alert condition.

The RMC logic is implemented using an 8-bit microprocessor (PIC 17C44) as the primary control device. Support devices include:

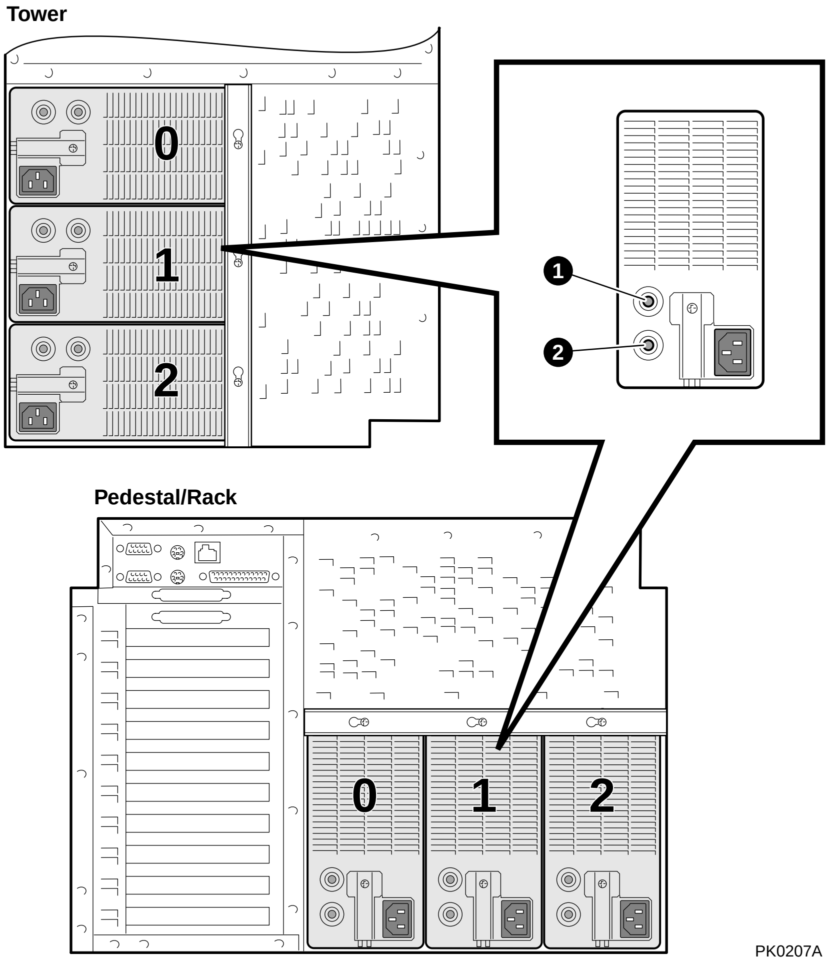
- Flash RAM (for code storage)
- Address latch
- Dual universal asynchronous receiver/transmitter (DUART)
- 8-bit $I^2C$ port expanders
- $I^2C$ temperature sensors
- $I^2C$ nonvolatile memories (NVRAM)
- Programmable array logic (PAL)
- Dual-port RAM (DPR)
- RS232 drivers and receivers

Chapter 7 describes the operation and use of the RMC.

## 1.13 Power Supplies

**The power supplies provide power to components in the system box. The number of power supplies required depends on the system configuration.**

### Figure 1–13 Power Supplies



Two to three power supplies provide power to components in the system box. The system supports redundant power configurations to ensure continued system operation if a power supply fails. See Chapter 6 for power supply configurations.

The power supplies select line voltage automatically (100V to 240V and 50 Hz or 60 Hz).

### Power Supply LEDs

Each power supply has two green LEDs that indicate the state of power to the system.

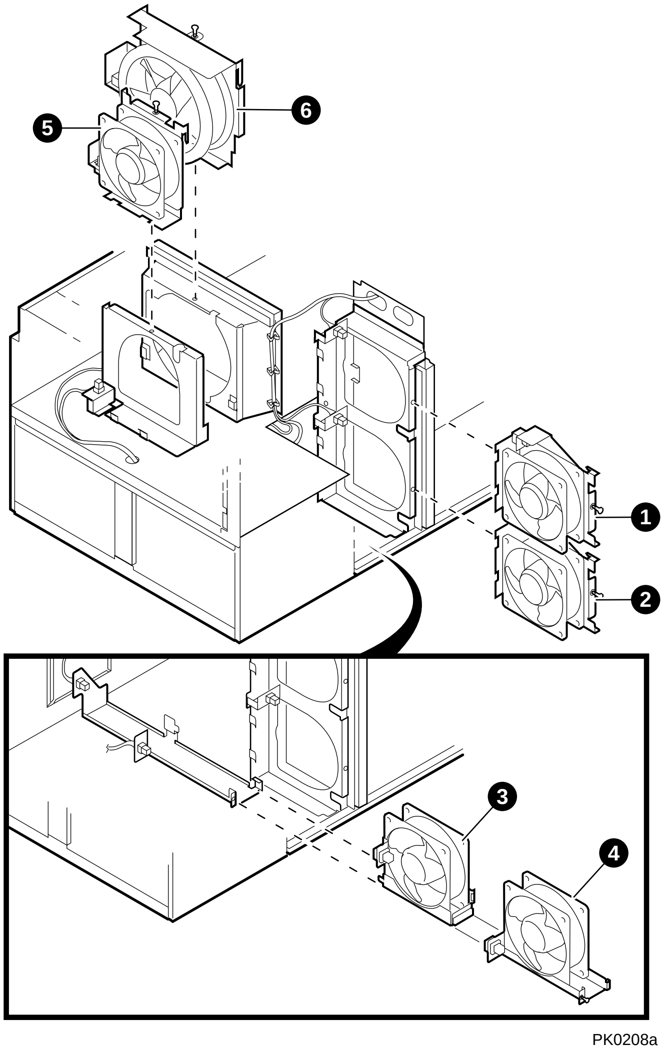
❶ POK (Power OK) Indicates that the power supply is providing power. The POK LED is on when the system is running. When the system power is on and a POK LED is off, that supply is not contributing to powering the system.

❷ +5 V Auxiliary Indicates that AC power is flowing from the wall outlet. As long as the power supply cord is plugged into the wall outlet, the +5V Aux LED is always on, even when the system power is off.

## 1.14 Fans

**The system has six hot-plug fans that provide front-to-back airflow.**

### Figure 1–14 System Fans



The system fans are shown in Figure 1–14 and described in Table 1–1.

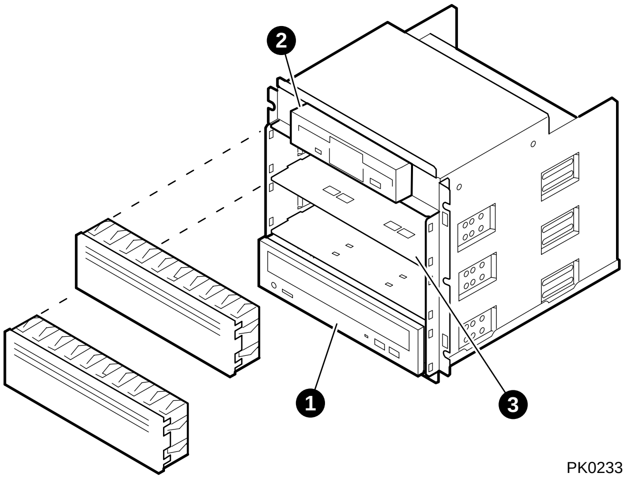
### Table 1–1 Fan Descriptions

| Fan Number | Area Cooled | Fan Failure Scenario |
|---|---|---|
| ❶,❷<br>4.5-in. | PCI card cage<br>Removable media<br>Right drive cage | Both fans are powered at all times. If one fan fails, all other system fans run at maximum speed to provide adequate cooling. You can replace either fan while the system is running. |
| ❸,❹<br>4.5-in. | Power supplies<br>Left drive cage | Both fans are powered at all times. If one fan fails, all other system fans run at maximum speed to provide adequate cooling. You can replace either fan while the system is running. |
| ❺<br>4.5-in.<br>redundant | CPU and memory card cage | Not powered except during a fan failure. |
| ❻<br>6.75-in.<br>main fan | CPU and memory card cage | Fan is powered at all times. If it fails, all other system fans run at maximum speed to provide adequate cooling. You can replace the fan while the system is running. |

## 1.15 Removable Media Storage

**The system box houses a CD-ROM drive ➊ and a high-density 3.5-inch floppy diskette drive ➋ and supports two additional 5.25-inch half-height drives or one additional full-height drive. The 5.25-inch half height area has a divider ➌ that can be removed to mount one full-height 5.25-inch device.**

Figure 1–15 Removable Media Drive Area

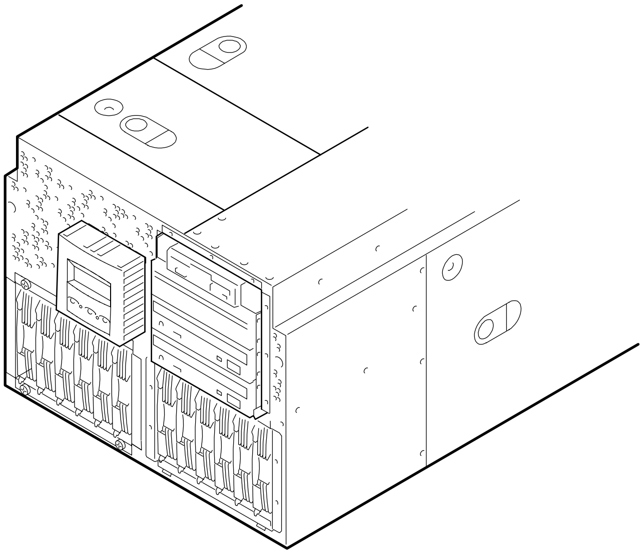


## 1.16 Hard Disk Drive Storage

**The system chassis can house up to two storage disk cages. The storage subsystem supports “hot pluggable" universal hard disk drives that can be replaced while the storage backplane is powered and operating.**

You can install six 1-inch universal hard drives in each storage disk cage. See Chapter 8 for information on replacing hard disk drives.

### Figure 1–16 Hard Disk Storage Cage with Drives (Pedestal/Rack)

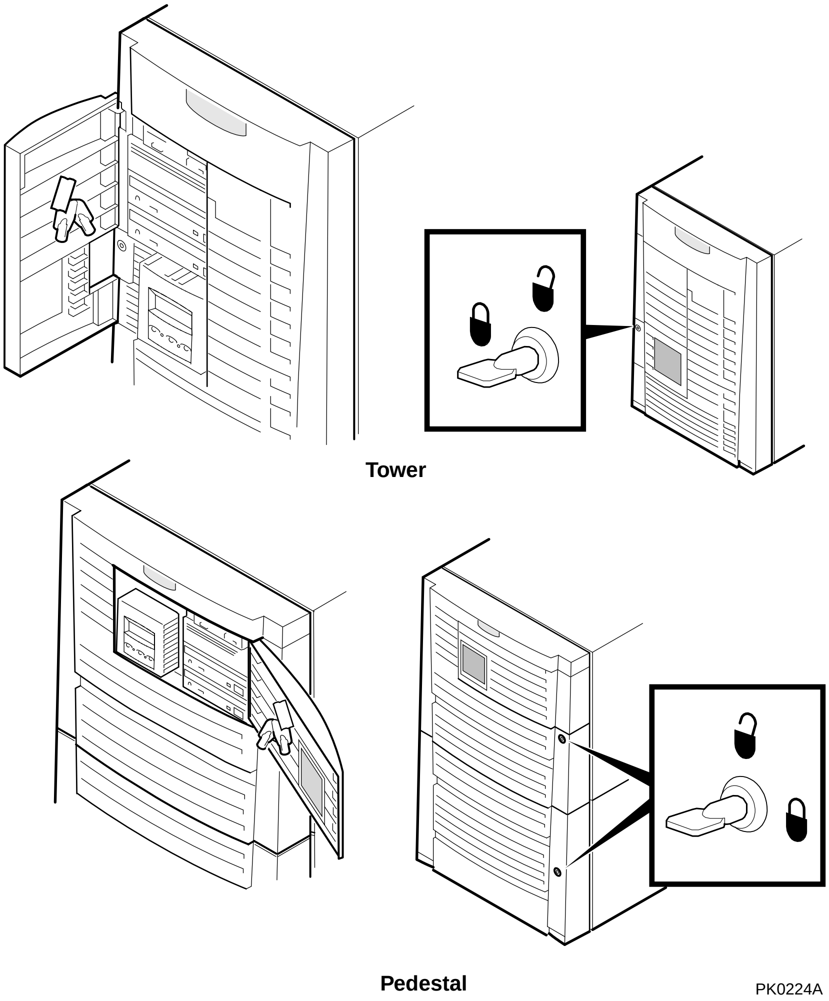


MR0046

## 1.17 System Access

**At the time of delivery, the system keys are taped inside the small front door that provides access to the operator control panel and removable media devices.**

### Figure 1–17 System Lock and Key



Both the tower and pedestal systems have a small front door through which the control panel and removable media devices are accessible. At the time of delivery, the system keys are taped inside this door.

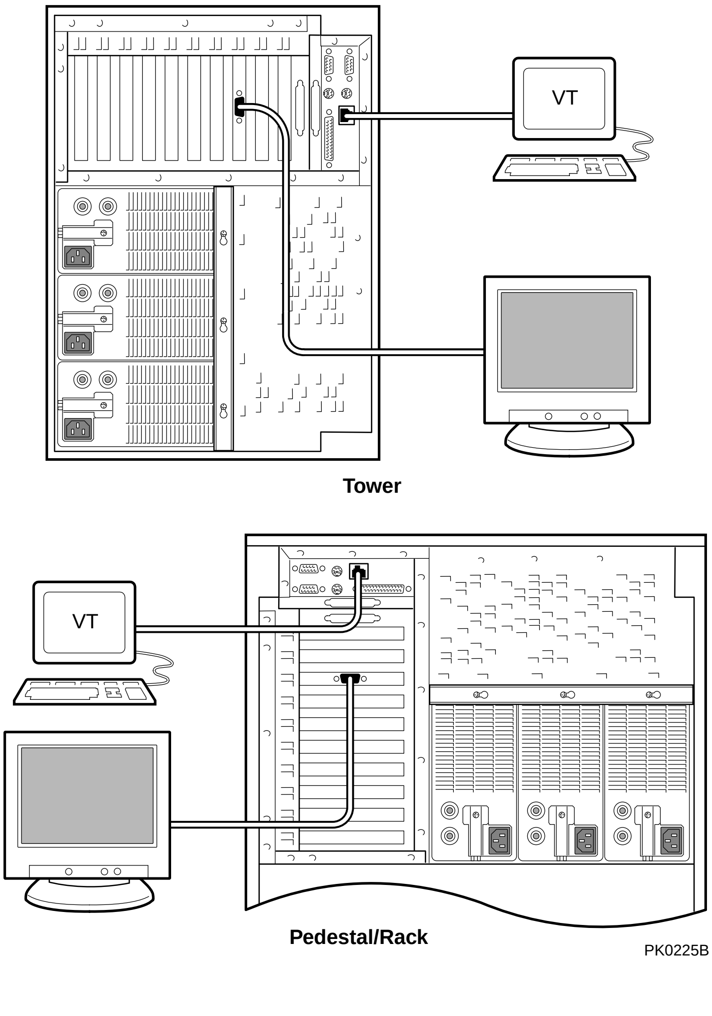
The tower front door has a lock that lets you secure access to the universal disk drives and to the rest of the system.

The pedestal has two front doors, both of which can be locked. The upper door secures the disk drives and access to the rest of the system, and the lower door secures the disk expanded storage cages.

## 1.18 Console Terminal

**The console terminal can be a serial (character cell) terminal connected to the COM1 or COM2 port or a VGA monitor connected to a VGA adapter. A VGA monitor requires a keyboard and mouse.**

### Figure 1–18 Console Terminal Connections (Local)



# Chapter 2
# Troubleshooting

This chapter describes the starting points for diagnosing problems on ES45 systems. The chapter also provides information resources.

- Questions to Consider
- Diagnostic Tables
- Service Tools and Utilities
- Q-Vet Installation Verification
- Information Resources

## 2.1 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 system release notes.)
- What is the current 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 OCP display, power-up display, and SRM commands.
    - ➢ If you are unable to access the SRM console, enter the RMC CLI and issue commands to determine the hardware status. See Chapter 7.
    - ➢ If the operating system has crashed and rebooted, the CCAT (Compaq Crash Analysis Tool), the Compaq Analyze service tools (to interpret error logs), the SRM **crash** command, and operating system exercisers can be used to diagnose system problems.

## 2.2 Diagnostic Tables

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

**WARNING: To prevent injury, access is limited to persons who have appropriate technical training and experience. Such persons are expected to understand the hazards of working within this equipment and take measures to minimize danger to themselves or others. These measures include:**
**1. Remove any jewelry that may conduct electricity.**
**2. If accessing the system card cage, power down the system and wait 2 minutes to allow components to cool.**
**3. Wear an anti-static wrist strap when handling internal components.**

### Table 2–1 Power Problems

| Symptom | Action | Reference |
|---|---|---|
| System does not power on. | • Check error messages on the OCP.<br>• Check that AC power is plugged in.<br>• Check that the ambient room temperature is within environmental specifications (10–35° C, 50–95° F).<br>• Check the Power setting on the control panel. Toggle the Power button to off, then back on to clear a remote power disable.<br>• 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:<br>Loss of AC power<br>RMC **power off** command<br>System software<br>Multiple fan failure<br>Overtemperature condition<br>Power supply failure (If N+1 config. multiple power supply failure)<br>Faulty CPU (CPU DC/DC converter failure) | |
| | If AC power is present, use the RMC **env** command to check environmental status. | Chapter 7 |
| | Check jumper J5. If the system must be kept running, this jumper can be positioned to override an overtemperature condition. | Appendix B |

### Table 2–2 Problems Getting to Console Mode

| Symptom | Action | Reference |
|---|---|---|
| Power-up screen is not displayed at system console. | Note any error beep codes and observe the OCP display for a failure detected during self-tests. | Chapter 3 |
| | Check keyboard and monitor connections. | Chapter 1 |
| | Press the Return key. If the system enters console mode, check that the **console** environment variable is set correctly. | |
| | If the console terminal is a VGA monitor, the **console** variable should be set to **graphics**. If it is a serial terminal, the console environment variable should be set to **serial**. | Chapter 6 |
| | 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 set the **console** environment variable to **serial**. | Chapter 6 |
| | Use RMC commands to determine status. | Chapter 7 |

**Table 2–3 Problems Reported by the Console**

| Symptom | Action | Reference |
|---|---|---|
| No SRM messages are displayed after the "jump to console" message. | Console firmware is corrupted. Load new firmware with fail-safe loader. | Chapter 3 |
| 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 firmware. If the fail-safe load does not work, replace the system motherboard. | Chapter 3 and Chapter 8 |
| 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. | Chapter 3 |
| • Power-up screen includes error messages. | Examine the console event log (**more el** command). | Chapter 4 |
| • Power-up screen or console event log indicates problems with mass storage devices. | Check cables and seating of drives. Check power to an external storage box. | |
| • Storage devices are missing from the **show config** display. | Check cables and seating of drives. Check power to an external storage box. | |
| • PCI devices are missing from the **show config** display. | Checking seating of modules. | |

Table 2–4 Boot Problems

| Symptom | Action | Reference |
|---|---|---|
| System cannot find boot device. | Use the **show config** and **show device** commands to check the system configuration for the correct device parameters (node ID, device name, and so on). | Chapter 6 |
| | Examine the **auto_action**, **bootdef_dev**, **boot_osflags**, and **os_type** environment variables. | |
| | For network boots, make sure **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, make sure **ei*0_protocols** or **ew*0_protocols** is set to **bootp** for *Tru64 UNIX* or **mop** for *OpenVMS*. | Chapter 6 |
| | Run the **test** command to see if the boot device is operating. | Chapter 4 |

### Table 2–5 Errors Reported by the Operating System

| Symptom | Action | Reference |
|---|---|---|
| System is hung, but SRM console is operating | Press the Halt button and enter the **crash** command to provide a crash dump file for analysis.<br>• Refer to *OpenVMS Alpha System Dump Analyzer Utility Manual* for information on how to interpret OpenVMS crash dump files.<br>• Refer to the *Guide to Kernel Debugging* for information on using the UNIX Krash Utility.<br>Use the SRM **info** command to display registers and data structures.<br>If the problem is intermittent, run the SRM **test** and **sys_exer** commands. | Chapter 4 |
| System is hung and SRM console is not operating. | Invoke the RMC CLI and enter the **dump** command to access DPR locations. | Chapter 7 |
| Operating system has crashed and rebooted. | Examine the operating system error log files to isolate the problem. | |
| | If the problem is intermittent, ensure that Compaq Analyze has been installed and is running in background mode (GUI does not have to be running) to determine the defective FRU. | Chapter 5 |

## 2.3 Service Tools and Utilities

**This section lists some of the tools and utilities available for acceptance testing and diagnosis and gives recommendations for their use.**

### 2.3.1 Error Handling/Logging Tools (Compaq Analyze)

The operating systems provide fault management error detection, handling, notification, and logging.

The primary tool for error handling is Compaq Analyze, a fault analysis utility designed to analyze both single and multiple error/fault events. Compaq Analyze uses error/fault data sources other than the traditional binary error log. See Chapter 5.

### 2.3.2 Loopback Tests

Internal and external loopback tests are used to test the components on the I/O connector assembly ("junk I/O") and to test Ethernet cards. The loopback tests are a subset of the SRM diagnostics.

Use loopback tests to isolate problems with the COM2 serial port, the parallel port, and Ethernet controllers. See the **test** command in Chapter 4 for instructions on performing loopback tests.

### 2.3.3 SRM Console Commands

SRM console commands are used to set and examine environment variables and device parameters. For example, the **show configuration** and **show device** commands are used to examine the configuration, and the **set *envar* and show *envar*** commands are used to set and view environment variables.

SRM commands are also used to invoke ROM-based diagnostics and to run native exercisers. For example, the **test** and **sys_exer** commands are used to test the system.

See Chapter 6 for information on configuration-related console commands and environment variables. See Chapter 4 for information on running console exercisers. See Appendix A for a list of console commands used most often on ES45 systems.

### 2.3.4 Remote Management Console (RMC)

The remote management console (RMC) is used for managing the server either locally or remotely. It also plays a key role in error analysis by passing error log information to the dual-port RAM (DPR), which is shared between the RMC and the system motherboard logic, so that this information can be accessed by the system. RMC also controls the control panel display. RMC has a command-line interface from which you can enter a few diagnostic commands.

RMC can be accessed as long as the power cord for a working supply is plugged into the AC wall outlet and a console terminal is attached to the system. This feature ensures that you can gather information when the operating system is down and the SRM console is not accessible. See Chapter 7.

### 2.3.5 Crash Dumps

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

CCAT, the Compaq Crash Analysis Tool, is the primary crash dump analysis tool for analyzing crash dumps on Alpha systems. CCAT compares the results of a crash dump with a set of rules. If the results match one or more rules, CCAT notifies the system user of the cause of the crash and provides information to avoid similar crashes in the future.

### 2.3.6 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 that are 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 from the following Web site:

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

## 2.4 Q-Vet Installation Verification

**CAUTION: Customers are not authorized to access, download, or use Q-Vet. Q-Vet is for use by Compaq engineers to verify the system installation. Misuse of Q-Vet may result in loss of customer data.**

Q-Vet is the Qualification Verifier Exerciser Tool that is used by Compaq engineers to exercise systems under development. Compaq recommends running the latest Q-Vet released version to verify that hardware is installed correctly and is operational. Q-Vet does not verify specific operating system or layered product configurations.

The latest Q-Vet release, information, Release Notes, and documentation are located at http://chump2.mro.cpqcorp.net/qvet/.

If the system has been partitioned, Q-Vet must be installed and run separately on each partition to verify the complete system. Compaq recommends that Compaq Analyze be installed on the operating system prior to running Q-Vet.

**CAUTION**:

*Do **not** install the Digital System Verification Software (DECVET) on the system; use Q-Vet instead.*

*Non-IVP Q-Vet scripts verify disk operation for some drives with "write enabled" techniques. These are intended for Engineering and Manufacturing Test. Run ONLY IVP scripts on systems that contain customer data or any other items that must not be written over. See the Q-Vet Disk Testing Policy Notice on the Q-Vet Web site for details. All Q-Vet IVP scripts use Read Only and/or File I/O to test hard drives. Floppy and tape drives are always write tested and should have scratch media installed.*

*Q-Vet **must** be de-installed upon completion of system verification.*

### Swap or Pagefile Space

The system must have adequate swap space (on *Tru64 UNIX*) or pagefile space (on *OpenVMS*) for proper Q-Vet operation. You can set this up either before or after Q-Vet installation.

During initialization, Q-Vet will display a message indicating the minimum amount of swap/pagefile needed, if it determines that the system does not have enough. You can then reconfigure the system.

If you wish to address the swap/pagefile size before running Q-Vet, see the Swap/Pagefile Estimates on the Q-Vet Web site.

### 2.4.1 Installing Q-Vet

**The procedures for installation of Q-Vet differ between operating systems. You must install Q-Vet on each partition in the system.**

Install and run Q-Vet from the **SYSTEM** account on VMS and the **root** account on UNIX. Remember to install Q-Vet in each partition.

#### Tru64 UNIX

1. Make sure that there are no old Q-Vet or DECVET kits on the system by using the following command:
   ```
   setld -i  |  grep  VET
   ```

   Note the names of any listed kits, such as OTKBASExxx etc., and remove the kits using **qvet_uninstall** if possible. Otherwise use the command
   ```
   setld -d    kit1_name  kit2_name   kit3_name
   ```
2. Copy the kit tar file (*QVET_Vxxx.tar*) to your system.
3. Be sure that there is no directory named output. If so move to another directory or remove the output directory.
   ```
   rm -r output
   ```
4. Untar the kit with the command
   ```
   tar xvf   QVET_Vxxx.tar
   ```
   *Note: The case of the file name may be different depending upon how it was stored on the system. Also, you may need to enclose the file name in quotation marks if a semi-colon is used.*
5. Install the kit with the command
   ```
   setld -l  output
   ```
6. During the install, if you intend to use the GUI you must select the optional GUI subset (QVETXOSFxxx).
7. The Q-Vet installation will size your system for devices and memory. It also runs qvet_tune. You should answer '`y`' to the questions that are asked about setting parameters. If you do not, you may have trouble running Q-Vet. After the installation completes, you should delete the output directory with `rm -r output.` You can also delete the kit tar file.
8. You ***must*** reboot the system before starting Q-Vet.
9. On reboot you can start Q-Vet GUI via `vet&` or you can run non GUI (command line) via `vet –nw.`

### OpenVMS

1. Delete any *QVETAXPxxx.A* or *QVETAXPxxx.EXE* file from the current directory.
2. Copy the self-extracting kit image file (*QVETAXPxxx.EXE*) to the current directory.
3. It is highly recommended, but not required, that you purge the system disk before installing Q-Vet. This will free up space that may be needed for pagefile expansion during the AUTOGEN phase.
   **$purge sys$sysdevice:[*...]*.***
4. Extract the kit saveset with the command **`$run QVETAXPxxx.EXE`** and verify that the kit saveset was extracted by checking for the "Successful decompression" message.
5. Use **`@sys$update:vmsinstal`** for the Q-Vet installation. The installation will size your system for devices and memory. You should choose all the default answers during the Q-Vet installation. This will verify the Q-Vet installation, tune the system, and reboot. During the install, if you ***do not*** intend to use the GUI, you can answer **`no`** to the question "Do you want to install Q-Vet with the DECwindows Motif interface?"
6. After the installation completes you should delete the *QVETAXP0xx.A* file and the *QVETAXPxxx.EXE* file.
7. On reboot you can start Q-Vet GUI via **$vet** or the command interface via **$vet/int=char.**

### 2.4.2 Running Q-Vet

**You must run Q-Vet on each partition in the system to verify the complete system.**

Compaq recommends that you review the Special Notices and the Testing Notes section of the Release Notes located at http://chump2.mro.cpqcorp.net/qvet/ before running Q-Vet.

Follow the instructions listed for your operating system to run Q-Vet in each partition.

#### Tru64 UNIX

Graphical Interface

1. From the Main Menu, select **IVP, Load Script** and select **Long IVP** (the IVP tests will then load into the Q-Vet process window).
2. Click the **Start All** button to begin IVP testing.

Command-Line Interface

```
> vet -nw
Q-Vet_setup> execute .Ivp.scp
Q-Vet_setup> start
```

Note that there is a "**.**" in front of the script name, and that commands are case sensitive.

### OpenVMS

Graphical Interface

1. From the Main Menu, select **IVP, Load Script** and select **Long IVP** (the IVP tests will then load into the Q-Vet process window).
2. Click the **Start All** button to begin IVP testing.

Command-Line Interface

```
$ vet /int=char
Q-Vet_setup> execute ivp.vms
Q-Vet_setup> start
```

Note that commands are case sensitive.

**NOTE:** *A short IVP script is provided for a simple verification of device setup. It is selectable from the GUI IVP menu, and the script is called* ***.Ivp_short.scp*** *(**ivp_short.vms**). This script will run for 15 minutes and then terminate with a Summary log. The short script may be run prior to the long IVP script if desired, but not in place of the long IVP script, which is the full IVP test.*

The long IVP will run until the slowest device has completed one pass (typically 2 to 12 hours). This is called a Cycle of Testing.

### 2.4.3 Reviewing Results of the Q-Vet Run

**After running Q-Vet, check the results of the run by reviewing the summary log.**

If you follow the above steps, Q-Vet will run all exercisers until the slowest device has completed one full pass. Depending on the size of the system (number of CPUs and disks), this will typically take 2 to 12 hours. Q-Vet will then terminate testing and produce a summary log. The termination message will tell you the name and location of this file.

All exerciser processes can also be manually terminated with the Suspend and Terminate buttons (**stop** and **terminate** commands).

After all exercisers report “Idle,” the summary log is produced containing Q-Vet specific results and statuses.

A. If there are no Q-Vet errors, no system event appendages, and testing ran to the specified completion time, the following message will be displayed:

```
″Q-Vet Tests Complete: Passed″
```

B. Otherwise, a message will indicate:

```
″Additional information may be available from Compaq
Analyze″
```

It is recommended that you run Compaq Analyze to review test results. The testing times (for use with Compaq Analyze) are printed to the Q-Vet run window and are available in the summary log.

## 2.4.4 De-Installing Q-Vet

**The procedures for de-installation of Q-Vet differ between operating systems. You must de-install Q-Vet from each partition in the system. Failure to do so may result in the loss of customer data at a later date if Q-Vet is misused.**

Follow the instructions listed under your operating system to de-install Q-Vet from a partition. The **qvet_uninstall** programs will remove the Q-Vet supplied tools and restore the original system tuning/configuration settings.

### Tru64 UNIX

1. **Stop, Terminate,** and **Exit** from Q-Vet testing.
2. Execute the command **`qvet_uninstall.`** This will also restore the system configuration/tuning file `sysconfigtab`.
3. Note: log files are retained in `/usr/field/tool_logs`
4. Reboot the system. You must reboot in any case, even if Q-Vet is to be reinstalled.

### OpenVMS

1. **Stop**, **Terminate,** and **Exit** from any Q-Vet testing.
2. Execute the command **@sys$manager:qvet_uninstall.** This will restore system tuning (modparams.dat) and the original UAF settings.
3. Note: log files are retained in **sys$specific:[sysmgr.tool_logs]**
4. Reboot the system. You must reboot in any case, even if Q-Vet is to be reinstalled.

## 2.5 Information Resources

**Many information resources are available, including tools that can be downloaded from the Internet, firmware updates, a supported options list, and more.**

### 2.5.1 Compaq Service Tools CD

The Compaq Service Tools CD-ROM enables field engineers to upgrade customer systems with the latest version of software when the customer does not have access to Compaq Web pages. The Web site is:

http://caspian1.zko.dec.com/service_tools/

### 2.5.2 ES45 Service HTML Help File

The information contained in this guide, including the FRU procedures and illustrations, is available in HTML Help format as part of the Maintenance Kit. It can also be accessed from the Learning Utility and ProSIC Web sites.

### 2.5.3 Alpha Systems Firmware Updates

The firmware resides in the flash ROM on the system motherboard. You can obtain the latest system firmware from CD-ROM or over the network.

#### Quarterly Update Service

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

**Alpha Firmware Internet Access**

- You can obtain Alpha firmware updates from the following Web site:

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

The README file describes the firmware directory structure and how to download and use the files.

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

http://gatekeeper.research.compaq.com/pub/Digital/Alpha/firmware/

Individual Alpha system firmware releases that occur between releases of the firmware CD are located in the interim directory:

http://gatekeeper.research.compaq.com/pub/Digital/Alpha/firmware/interim/

### 2.5.4 Fail-Safe Loader

The fail-safe loader (FSL) allows you to boot the firmware update utility in an attempt to repair corrupted console files that reside within the flash ROMs on the system motherboard. You can download the fail-safe loader from the Internet (using the firmware update URL) to create your own fail-safe loader diskettes. See Chapter 3 for information on forcing a fail-safe floppy load.

### 2.5.5 Software Patches

Software patches for the supported operating systems are available from:

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

### 2.5.6 Learning Utility

The Learning Utility provides information about various technical topics.

http://learning1.americas.cpqcorp.net/mcsl-html/home.asp

### 2.5.7 Late-Breaking Technical Information

You can download up-to-date files and late-breaking technical information from the Internet.

The information includes firmware updates, the latest configuration utilities, software patches, lists of supported options, and more.

http://www.compaq.com/alphaserver/es40/es40.html

### 2.5.8 Supported Options

A list of options supported on the system is available on the Internet:

http://www.compaq.com/alphaserver/es40/

# Chapter 3
# Power-Up Diagnostics and Display

This chapter describes the power-up process and RMC, SROM, and SRM power-up diagnostics. The following topics are covered:

- Overview of Power-Up Diagnostics
- System Power-Up Sequence
- Power-Up Displays
- Power-Up Error Messages
- Forcing a Fail-Safe Floppy Load
- Updating the RMC

## 3.1 Overview of Power-Up 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 deposits it into the DPR. The SROM minimally tests the CPUs, initializes and tests backup cache, and minimally tests memory. Finally, the SROM loads the SRM console program into memory and jumps to the first instruction in the console program.**

There are three distinct sets of power-up diagnostics:

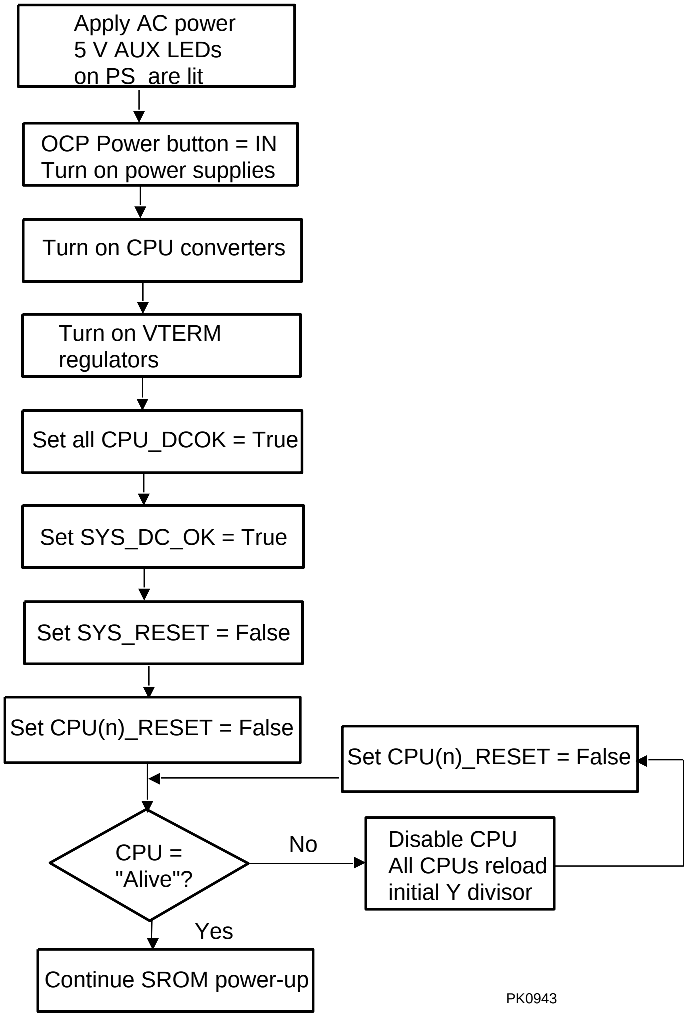
1. System power controller and remote management console diagnostics—These diagnostics check the power regulators, temperature, and fans. Failures are reported in the dual-port RAM (DPR) and on the OCP display. Certain failures may prevent the system from powering on.
2. Serial ROM (SROM) diagnostics—SROM 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 OCP.
3. Console firmware diagnostics—These tests are executed by the SRM console code. They test the core system, including boot path devices. Failures during these tests are reported to the console terminal through the power-up screen or console event log.

## 3.2 System Power-Up Sequence

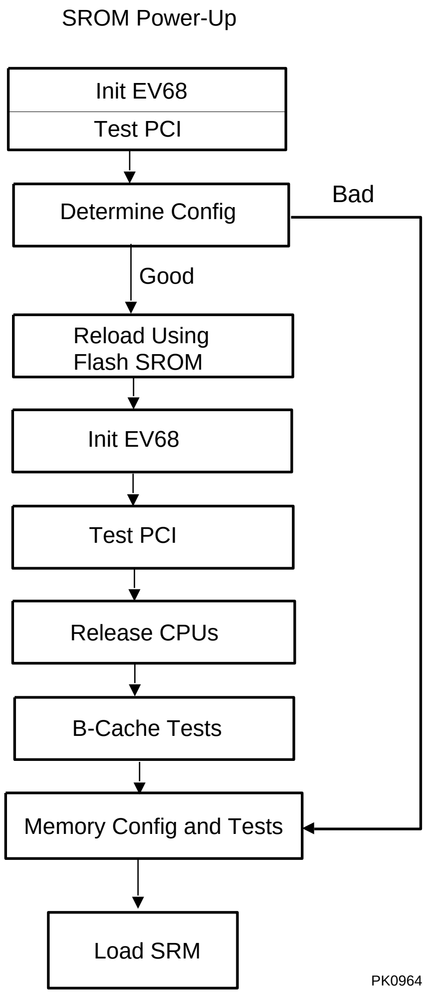
**The power-up sequence is described below and illustrated in Figure 3–1.**

1. When the power cord is plugged into the wall outlet, 5V auxiliary AC voltage is enabled. The 5 V AUX LEDs on the power supplies are lit, and the system power controller and RMC are initialized.
2. Pressing the Power button on the control panel or subsequently issuing the **power-on** command from the RMC turns on power to the power supplies, CPU converters, and VTERM regulators. The POK LEDs on the power supplies are lit and the power supplies are tested. If all power supplies are bad, power-up stops. All DC/DC converters and regulators are then tested. If any converter or regulator is bad, power-up stops.
3. CPU_DCOK and SYS_DC_OK are set to “true,” which means that DC power on the CPUs and system is 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 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 3.3.

## Figure 3–1 Power-Up Sequence



### Figure 3–1 Power-Up Sequence (Continued)



## 3.3 Power-Up Displays

**Power-up information is displayed on the operator control panel and on the console terminal startup screen. Messages sent from the RMC and SROM programs are displayed first, followed by messages from the SRM console.**

### 3.3.1 SROM Power-Up Display

The following example describes the SROM power-up sequence and shows the SROM power-up messages and corresponding OCP messages.

#### Example 3–1 Sample SROM Power-Up Display

```
SROM V2.15 CPU # 00 @  1000  MHz                PCI Test     ❶
SROM program starting                           Power on     ❷
Reloading SROM
SROM V2.15 CPU # 00 @  1000  MHz
System Bus Speed @ 0125  MHz
SROM program starting                           RelCPU       ❸
PCI66 bus speed check
Reloading SROM
SROM V2.15 CPU # 00 @  1000  MHz
System Bus Speed @ 0125  MHz
SROM program starting
PCI66 bus speed check
Starting secondary on CPU #1
Starting secondary on CPU #2                    BC Data
Starting secondary on CPU #3
Bcache data tests in progress
Bcache address test in progress
CPU parity and ECC detection in progress
```

### SROM Power-Up Sequence

❶ When the system powers up, the SROM code is loaded into the I-cache (instruction 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 board 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 SROM code. Invalid flash SROM must be reprogrammed.

If flash SROM is good, the primary CPU programs appropriate registers with the values from the flash data and selects itself as the 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 EV68 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 B-cache tests. For example, the ECC data test verifies the detection logic for single- and double-bit errors.

### Example 3–1 Sample SROM Power-Up Display (Continued)

```
Bcache ECC data tests in progress               Size Mem    ❺
Bcache TAG lines tests in progress
Memory sizing in progress
Memory configuration in progress
Testing AAR3
Memory data test in progress
Memory address test in progress
Memory pattern test in progress
Testing AAR2
Memory data test in progress
Memory address test in progress
Memory pattern test in progress
Testing AAR1
Memory data test in progress
Memory address test in progress
Memory pattern test in progress
Testing AAR0
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    ❻
Code execution complete (transfer control)      Jump to
                                                Console
```

**NOTE:** *The power-up text that is displayed on the screen depends on what kind of terminal is connected as the console terminal: VT or VGA.*

*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 displayed 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.*

### SROM Power-Up Sequence

❺ The primary CPU initiates all memory tests. The memory is tested for address and data errors for the first 32 MB of memory in each array. It also initializes all 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 rerun on the first 32 MB of memory in the remaining arrays. 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 console and transfers control to it.

### 3.3.2 SRM Console Power-Up Display

**When SROM power-up is complete, 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.**

The following section shows the messages that are displayed once the SROM has transferred control to the SRM console.

#### Example 3–2 SRM Power-Up Display

```
OpenVMS PALcode V1.88-28, Tru64 UNIX PALcode V1.83-24              ❶
starting console on CPU 0
initialized idle PCB
initializing semaphores
initializing heap
initial heap 240c0
memory low limit = 1e6000
heap = 240c0, 17fc0
initializing driver structures
initializing idle process PID
initializing file system
initializing timer data structures
lowering IPL
CPU 0 speed is 1000 MHz
create dead_eater
create poll
create timer
create powerup
access NVRAM
4096 MB of System Memory                                           ❷
Testing Memory
...
probe I/O subsystem                                                ❸
```

## Example 3–2 SRM Power-Up Display (Continued)

```
Hose 0 - PCI bus running at 33Mhz
entering idle loop
probing hose 0, PCI
probing PCI-to-ISA bridge, bus 1
probing PCI-to-PCI bridge, bus 2
bus 0, slot 8 -- pka -- NCR 53C895
bus 0, slot 9 -- eia -- DE600-AA
bus 2, slot 0 -- pkb -- NCR 53C875
bus 2, slot 1 -- pkc -- NCR 53C875
bus 2, slot 2 -- ewa -- DE500-AA Network Controller
bus 0, slot 16 -- dqa -- Acer Labs M1543C IDE
bus 0, slot 16 -- dqb -- Acer Labs M1543C IDE
Hose 1 - PCI bus running at 66Mhz
probing hose 1, PCI
bus 0, slot 2 -- vga -- 3Dlabs OXYGEN VX1
Hose 2 - AGP bus
probing hose 2, PCI
Hose 3 - PCI bus running at 33Mhz
probing hose 3, PCI
probing PCI-to-PCI bridge, bus 2
bus 2, slot 4 -- eib -- DE602-AA
bus 2, slot 5 -- eic -- DE602-AA
bus 2, slot 6 -- eid -- DE602-FA
bus 0, slot 2 -- fwa -- DEFPA
starting drivers
                                                                 ❹
```

### SRM Power-Up Sequence

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

If console is set to **graphics**, the display from this point on is saved in a memory buffer and displayed on 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 started.

### Example 3–2 SRM Power-Up Display (Continued)

```
initializing keyboard                                              ❺
starting console on CPU 1
initialized idle PCB
initializing idle process PID
lowering IPL
CPU 1 speed is 1000 MHz
create powerup
entering idle loop
starting console on CPU 2
initialized idle PCB
initializing idle process PID
lowering IPL
CPU 2 speed is 1000 MHz
create powerup
starting console on CPU 3
initialized idle PCB
initializing idle process PID
lowering IPL
CPU 3 speed is 1000 MHz
create powerup
initializing GCT/FRU at 220000
initializing pka pkb pkc ewa fwa dqa dqb eia eia0: link up :
Negotiated  100Basx
eib eic eid
Memory Testing and Configuration Status                            ❻
  Array       Size       Base Address    Intlv Mode
---------  ----------  ----------------  ----------
    0        4096Mb     0000000000000000    2-Way
    1        1024Mb     0000000200000000    2-Way
    2        4096Mb     0000000100000000    2-Way
    3        1024Mb     0000000240000000    2-Way

       10240 MB of System Memory
AlphaServer ES45 Console V5.9-9, built on June 2001 at 17:09:49    ❼
```

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

❻ Various diagnostics are performed.

❼ The console terminal displays 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 operating system.

---

**NOTE:** *If the console requires the heap to be expanded, it restarts.*

---

### 3.3.3 SRM Console Event Log

**The SRM console event log helps you troubleshoot problems that do not prevent the system from coming up to the SRM console. The console event log consists of status messages received during power-up self-tests.**

#### Example 3–3 Sample 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 ***
```

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

Example 3–3 shows a console event log that shows errors. The console reported that CPU 1 did not power up and fans 1 and 2 failed.

## 3.4 Power-Up Error Messages

**Error messages at power-up may be displayed by the RMC, SROM, and SRM. A few SROM messages are announced by beep codes.**

### 3.4.1 SROM Messages with Beep Codes

Table 3–1 Error Beep Codes

| Beep Code | Associated Messages | Meaning |
|---|---|---|
| 1 | Jump to Console | SROM code has completed execution. System jumps to SRM console. SRM messages should start to be displayed. If no SRM messages are displayed, it may indicate corrupted firmware. See Section 3.4.2. |
| 1-1-4 | ROM err | The ROM err message is displayed briefly, then a single beep is emitted, and Jump to Console is displayed. The SROM code is unable to load the console code; a flash ROM header area or checksum error has been detected. See Section 3.4.2. |
| 2-1-2 | Cfg ERR *n*<br>Cfg ERR s | Configuration error on CPU *n* (*n* is 0, 1, 2, or 3) or a system configuration error. The system will still power up. |

### Table 3–1 Error Beep Codes (Continued)

| Beep Code | Associated Messages | Meaning |
|---|---|---|
| 1-2-4 | BC error<br>CPU error<br>BC bad | Backup cache (B-cache) error. Indicates a bad CPU. |
| 1-3-3 | No mem | No usable memory detected. Some memory DIMMs may not be properly seated or some DIMM sets may be faulty. See Section 3.4.3. |

A few SROM error messages that appear on the operator control panel are announced by audible error beep codes, an indicated in Table 3–1. For example, a 1-1-4 beep code consists of one beep, a pause (indicated by the hyphen), one beep, a pause, and a burst of four beeps. This beep code is accompanied by the message “ROM err.”

Related messages are also 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**.

### 3.4.2 Checksum Error

**If no messages are displayed on the operator control panel after the Jump to Console message, the console firmware is corrupted. When the system detects the error, it attempts to load a utility called the fail-safe loader (FSL) so that you can load new console firmware images. A sequence similar to the one in Example 3–4 occurs.**

#### Example 3–4 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 V1.91-33, Tru64 UNIX PALcode V1.87-27

starting console on CPU 0                                              ❸
.
.
.
starting drivers
entering idle loop
P00>>> boot update_cd                                                  ❹
```

The sequence shown in Example 3–4 is as follows:

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

❷ The system attempts to automatically load the FSL program 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.

❹ At the P00>>> console prompt, boot the Loadable Firmware Update Utility (LFU) from the Alpha Systems Firmware CD (shown in the example as the variable *update_cd*).

---

**NOTE:** *For more information on the LFU, see the Firmware Updates Web site: http://ftp.digital.com/pub/digital/Alpha/firmware/*

---

### 3.4.3 No MEM Error

**If the SROM code cannot find any usable memory, a 1-3-3 beep code is issued (one beep, a pause, a burst of three beeps, a pause, and another burst of three beeps), and the message "No MEM" is displayed on the OCP. The system does not come up to the console program. This error indicates missing or bad DIMMs.**

The OCP and console terminal display text similar to the following:

```
Failed  MMB-2 J9  ❶
Failed  MMB-2 J5
Failed  MMB-0 J9
Failed  MMB-0 J5
Incmpat MMB-3 J9  ❷
Incmpat MMB-3 J5
Incmpat MMB-1 J9
Incmpat MMB-1 J5
Missing MMB-2 J6  ❸
Incmpat MMB-2 J2
Illegal MMB-0 J6  ❹
Incmpat MMB-0 J2
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 mismatched. 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.

See Chapter 6 for memory configuration rules.

### 3.4.4 RMC Error Messages

**Table 3–2 lists the fatal error messages that could potentially be displayed on the OCP by the remote management console during power-up. Most fatal error messages prevent the system from completing power-up. The warning messages listed in Table 3–3 require prompt attention but might not prevent the system from completing power-up or booting the operating system.**

**NOTE:** *The VTERM and CTERM regulators referenced in Table 3–2 are located on the system motherboard.*

*The "* CPUn *failed" message does not necessarily prevent the completion of power-up. If the system finds a good CPU, it continues the power-up process.*

Table 3–2 RMC Fatal Error Messages

| Message | Meaning |
|---|---|
| AC loss | No AC power to the system. |
| CPU*n* failed | CPU failed. "n" is 0, 1, 2, or 3. |
| VTERM failed | No VTERM voltage to CPUs. |
| CTERM failed | No CTERM voltage to CPUs. |
| Fan5, 6 failed | Main fan (6) and redundant fan (5) failed. |
| OverTemp failure | System temperature has passed the high threshold. |
| No CPU in slot 0 | Configuration requires that a CPU be installed in slot 0. |
| CPU door opened | System card cage cover off. Reinstall cover. |
| TIG error | Code essential to system operation is not loaded and/or running or TIG flash is corrupt. |
| Mixed CPU types | Different types of CPU are installed. Configuration requires that all CPUs be the same type. |

### Table 3–2 RMC Fatal Error Messages (Continued)

| Message | Meaning |
|---|---|
| Bad CPU ROM data | Invalid data in EEROM on the CPU. |
| 2.5V bulk failed | 2.5V regulator failed |
| AGP config error | Power consumption requirement for AGP failed |

### Table 3–3 RMC Warning Messages

| Message | Meaning |
|---|---|
| PS*n* failed | Power supply failed. “n” is 0, 1, or 2. |
| OverTemp Warning | System temperature is near the high threshold. |
| Fan*n* failed | Fan failed. “n” is 0 through 6. |
| PCI door opened | Cover to PCI card cage is off. Reinstall cover. |
| Fan door opened | Cover to main fan area (fans 5 and 6) is off. Reinstall cover. |
| 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. |

*Continued on next page*

### Table 3–3 RMC Warning Messages (Continued)

| Message | Meaning |
|---|---|
| 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. |
| CPU*n* VCACHE warn | Cache voltage on CPU over or under threshold. “n” is 0, 1, 2, or 3. |
| 1.5V bulk warn | Power supply voltage over or under threshold. For AGP backplane only. |
| 2.5V bulk warn | Power supply voltage over or under threshold. |

### 3.4.5 SROM 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 screen if the console output is set to serial.**

Table 3–4 lists the SROM error messages. The code numbers shown in the Code column are displayed in place of OCP or SROM messages if the SROM flash is invalid.

Table 3–4 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 |

*Continued on next page*

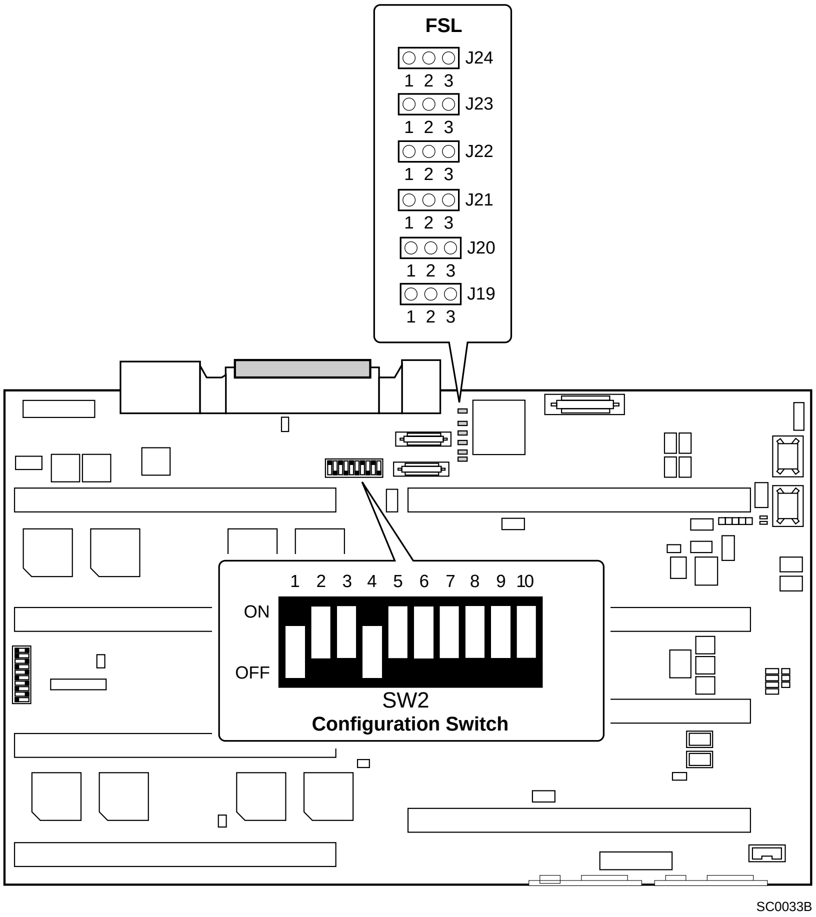
Table 3–4 SROM Error Messages (Continued)

| Code | SROM Message | OCP Message |
|---|---|---|
| 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 |

## 3.5 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.**

Figure 3–2 Function Jumpers (FSL)



1. Turn off the system. Unplug the power cord from each power supply and wait for the 5V AUX indicators to extinguish.
2. Remove enclosure covers (tower and pedestal) or the front bezel (rackmount) to access the system chassis. See Chapter 8 for illustrations.
3. Remove the fan cover and the system card cage cover to gain access to the system motherboard. See Chapter 8 for illustrations.
4. Remove CPU 2 (closest to the PCI backplane) so that you can access the function jumpers.
5. Locate the J22 function jumper on the system motherboard. See Figure 3–2.
6. Enable the fail-safe loader by moving the J22 jumper from pins 1 and 2 to pins 2 and 3.

**NOTE:** *The J21 and J23 function jumpers must be in their default positions over pins 1 and 2.*

7. Replace the chassis covers and enclosure covers. Plug in the power supplies.
8. Insert the LFU diskette into the floppy drive, and insert the update CD into the CD-ROM drive.
9. Power up the system and check the control panel display for progress messages.
10. At the P00>>> prompt, boot the update CD. Enter **update** at the UPD> prompt and press Return. Enter **yes** at the “Confirm update” prompt.
11. After the update is complete, turn off the system and unplug the power supplies.
12. Place J22 over pins 1 and 2.
13. Replace CPU 2.
14. Replace the chassis covers and enclosure covers, plug in the power supplies, and power up the system.

**NOTE:** *For more information on the LFU, see the Firmware Updates Web site: http://ftp.digital.com/pub/digital/Alpha/firmware/*

## 3.6 Updating the RMC

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

The RMC will not function if:

- No AC power is provided to any of the power supplies.
- DPR does not pass its self-test (DPR 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 a message to the terminal screen:

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

**NOTE:** *If the RMC is corrupt, you may not get an output from Com 1 (MMJ). If this occurs, remove J4 and move the serial line to the modem port.*

You can update the remote management console firmware 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 manual update 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
.
.
.
```

**NOTE:** *For more information on the LFU, see the Firmware Updates Web site: http://ftp.digital.com/pub/digital/Alpha/firmware/*

# Chapter 4
# SRM Console Diagnostics

This chapter describes 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 to provide maximum bus interaction between the console drivers and the target devices.

Run the diagnostics by using commands from the SRM console. To run the diagnostics in the background, use the background operator “&” at the end of the command. Errors are reported to the console terminal, the console event log, or both.

If you are not familiar with the SRM console, see the *ES45 Owner’s Guide*.

## 4.1 Diagnostic Command Summary

**Diagnostic commands are used to test the system and help diagnose failures. Table 4–1 gives a summary of the SRM diagnostic commands and related commands. See Chapter 6 for a list of SRM environment variables, and see Appendix A for a list of SRM commands most commonly used for the ES45 system.**

Table 4–1 Summary of Diagnostic and Related Commands

| Command | Function |
|---|---|
| **buildfru** | Initializes I$^2$Cbus EEPROM data structures for the named FRU. |
| **cat el** | Displays the console event log. Same as **more el**, but scrolls rapidly. The most recent errors are at the end of the event log and are visible on the terminal screen. |
| **clear_error** | Clear errors logged in the FRU EEPROMs as reported by the **show error** command. |
| **crash** | Forces a crash dump at the operating system level. |
| **deposit** | Writes data to the specified address of a memory location, register, or device. |
| **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. |
| **floppy_write** | Runs a write test on the floppy drive to determine whether you can write on the diskette. |

*Continued on next page*

### Table 4–1 Summary of Diagnostic and Related Commands (Continued)

| Command | Function |
|---|---|
| **grep** | Searches for “regular expressions”—specific strings of characters—and prints any lines containing occurrences of the strings. |
| **hd** | Dumps the contents of a file (byte stream) in hexadecimal and ASCII. |
| **hose_x_default_ speed** | Controls the default PCI bus speed for the specified hose when no PCI devices are present. If PCI devices are present on the specified hose, this EV is ignored and the bus speed is negotiated based on whether each device is 66 MHz capable or not. |
| **info** | Displays registers and data structures. |
| **kill** | Terminates a specified process. |
| **kill_diags** | Terminates all executing diagnostics. |
| **more el** | Same as **cat el**, but 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. |
| **php_led_test** | Tests the LEDs of all hot-plug slots on a specified I/O hose. Each LED is tested with four available patterns (blink A, blink B, On and Off for 5 seconds in each pattern). |

*Continued on next page*

### Table 4–1 Summary of Diagnostic and Related Commands (Continued)

| Command | Function |
|---|---|
| **php_button_test** | Tests the attention switch of each hot-plug slot on a specified I/O hose. The user is prompted to press the attention switch for each slot that has a blinking green LED. The user has 10 seconds to press the switch before the test declares a failure and moves onto the next slot. |
| **scsi_poll** | Controls whether or not a particular SCSI device driver polls for devices on the bus when the driver is started. This device is supported by some, but not all, console SCSI device drivers. |
| **scsi_reset** | Controls whether or not a particular SCSI device driver resets the SCSI bus when the driver is started. This EV is supported by some, but not all, console SCSI device drivers. |
| **set sys_serial_ num** | Sets the system serial number, which is then propagated to all FRUs that have EEPROMs. |
| **sys_com1_rmc** | Enables/disables internal COM1 access to the RMC. |
| **show error** | Reports errors logged in the FRU EEPROMs. |
| **show fru** | Displays information about field replaceable units (FRUs), including CPUs, memory DIMMs, and PCI cards. |
| **show_status** | Displays the progress of diagnostic tests. Reports one line of information for each executing diagnostic. |
| **sys_exer** | Exercises the devices displayed with the **show config** command. |
| **sys_exer -lb** | Runs console loopback tests for the COM2 serial port and the parallel port during the **sys_exer** test sequence. |

*Continued on next page*

### Table 4–1 Summary of Diagnostic and Related Commands (Continued)

| Command | Function |
|---|---|
| **test** | Verifies the configuration of the devices in the system. |
| **test -lb** | Runs loopback tests for the COM2 serial port and the parallel port in addition to verifying the configuration of devices. |

## 4.2 buildfru

**The buildfru command initializes I$^2$C bus EEPROM descriptive data structures for the named FRU and initializes its SDD and TDD error logs. This command uses data supplied on the command line to build the FRU descriptor. Buildfru is used by Manufacturing, FRU repair operations, or Field Service.**

### Example 4–1 buildfru

```
P00>>> buildfru smb0.mmb0.J3 54-24941-EA NI90200100         ❶
P00>>> buildfru smb0.cpu0 30-30158-05.AX05 NI94060554 Compaq❷
P00>>> buildfru -s smb0.mmb0.J3 80 45                       ❸
P00>>> buildfru -s smb0.mmb0.J3 80 47 46 45 44 43 42 41     ❹
```

❶ Building of the FRU descriptor on a DIMM, passing a part number and a serial number

❷ Building of the FRU descriptor on a CPU, passing a part number, serial number, and miscellaneous string

❸ Building of the FRU descriptor on a DIMM with the **-s** qualifier, pass offset 80, and value of 45

❹ Building of the FRU descriptor on a DIMM with the **-s** qualifier, pass offset 80, and many sequential data bytes

The **buildfru** command is used for several purposes:

- By Manufacturing to build a FRU table containing a description of each FRU in the system
- By FRU repair operations for initializing good stocking spares
- By Field Service to make any FRU descriptor adjustments required by the customer.

The information supplied on the **buildfru** command line includes the console name for the FRU, part number, serial number, model number, and optional information. The **buildfru** command facilitates writing the FRU information to the EEPROM on the device.

Use the **show fru** command to display the FRU table created with **buildfru**. Use the **show error** command to display FRUs that have errors logged to them.

Typically, you only need to use **buildfru** in Field Service if you replace a device for which the information displayed with the **show fru** command is inaccurate or missing. After replacing the device, use **buildfru** to build the new FRU descriptor.

---

**NOTE:** *Be sure to enter the FRU information carefully. If you enter incorrect information, the callout used by Compaq Analyze will not be accurate.*

---

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 zeros 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.

### FRU Assembly Hierarchy

Alpha-based systems can be decomposed into a collection of FRUs. Some FRUs carry various levels of nested FRUs. For instance, the system motherboard is a FRU that carries a number of "child" FRUs. A child, such as a memory motherboard (MMB), may carry a number of its own children, DIMMs. The naming convention for FRUs represents the assembly hierarchy.

The following is the general form of a FRU name:

**<fru*n*>[.<fru*n*>[.<fru*n*>]]**

The *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 ES45 FRU assembly hierarchy has three levels. The FRU types from the top to the bottom of the hierarchy are as follows:

| Level | FRU Type | Meaning |
|---|---|---|
| First Level | SMB | System motherboard |
| | JIO | I/O connector module (junk I/O) |
| | OCP | Operator control panel |
| | PWR (0–2) | Power supplies |
| | FAN | Fans |
| Second Level | CPU (0–3) | CPUs |
| | MMB (0–3) | Memory motherboards |
| | CPB | PCI backplane |
| Third Level | J (2–9) | Memory DIMMs |
| | PCI (0–9) | PCI slots |
| | SBM (0–1) | SCSI backplane |

To build a FRU descriptor for a lower level FRU, point back to the higher level FRUs to which it is associated. For example, to build a descriptor for a DIMM, point back to the MMB on which it resides and then to the system motherboard. All fields are automatically set to uppercase before writing to EEPROM. See Example 4–1.

If you enter the **buildfru** data correctly for a device that has an EEPROM to program, nothing is displayed after you enter the command. If you enter incorrect data or the device does not have an EEPROM to program, an error message similar to the following is displayed:

```
P00>>>
P00>>> buildf fan4 54-12345-01.a001 ay84412345
Device FAN4 does not support setting FRU values
P00>>>
```

**Syntax**

**buildfru ( <fru_name> <part_num> <serial_num> [<misc> [<other>]]**
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.

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

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

**<other>** 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 (0–255 hex) within this FRU's EEPROM, where the following supplied data bytes are to be written.

**<byte>...** The data bytes to be written. 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.

## 4.3 cat el and more el

**The cat el and more el commands display the contents of the console event log.**

In Example 4–2, the console reports that CPU 1 did not power up and fans 1 and 2 failed.

### Example 4–2 more el

```
>>> 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 CPU 1 failed.

❷ Fan 1 and Fan 2 failed.

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**

## 4.4 clear_error

**The clear_error command clears errors logged in the FRU EEPROMs as reported by the show error command.**

### Example 4–3 clear_error

```
P00>>> clear_error smb0          ❶
P00>>>

P00>>> clear_error all           ❷
P00>>>
```

❶ Clears all errors logged in the FRU EEPROM on the system motherboard (SMB0).

❷ Clears all errors logged to all FRU EEPROMs in the system.

The **clear_error** command clears TDD, SDD, and checksum errors. Hardware failures and unreadable EEPROM errors are not cleared. See Table 4–2.

**Syntax**

**clear_error <*fruname*>** Clears all errors logged to a specific FRU. *Fruname* is the name of the specified FRU. If you do not specify a FRU, you must use **clear_error all** to clear errors.

**clear_error all** Clears all errors logged to all system FRUs.

See the **show error** command for information on the types of errors that might be logged to the FRU EEPROMs.

## 4.5 crash

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

```
P00>>> crash

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>>>
```

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.

- See the *OpenVMS Alpha System Dump Analyzer Utility Manual* for information on how to interpret OpenVMS crash dump files.

- See the *Guide to Kernel Debugging* for information on using the Tru64 UNIX Krash Utility.

## 4.6 deposit and examine

**The deposit command writes data to the specified address of a memory location, register, or device. The examine command displays the contents of a memory location, register, or a device.**

### Example 4–4 deposit and examine

#### 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
```

#### examine

```
P00>>> e dpr:34f0 -l -n 5         ❶
dpr:             34F0 00000000
dpr:             34F4 00000000
dpr:             34F8 00000000
dpr:             34FC 00000000
dpr:             3500 204D5253
dpr:             3504 352E3558
P00>>>
```

### 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.

In Example 4–4:

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 R8 with -1.

❹ Deposit 8 in the first longword of the first 17 pages 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

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.

In Example 4–4:

❶ Examine the DPR starting at location 34f0 and continuing through the next 5 locations, and display the data size in longwords.

#### Syntax

**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.

*dev_name* Device name (address space) of the device to access. Device names are:

- **dpr** Dual-port RAM. See Appendix C for the DPR address layout.
- **eerom** Nonvolatile ROM used for EV storage.
- **fpr** Floating-point register set; name is F0 to F31. Alternatively, can be referenced by name.
- **gpr** General register set; name is R0 to R31. Alternatively, can be referenced by name.
- **ipr** Internal processor registers. Alternatively, some IPRs can be referenced by name.
- **pcicfg** PCI configuration space.
- **pciio** PCI I/O space.
- **pcimem** PCI memory space
- **pt** The PALtemp register set; name is PT0 to PT23.
- **pmem** Physical memory (default).
- **vmem** Virtual memory.

*offset* Offset within a device to which data is deposited.

*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.

## 4.7 exer

**The exer command exercises one or more devices by performing specified read, write, and compare operations. Typically exer is run from the built-in console script. Advanced users may want to use the specific options described here. Note that running exer on disks can be destructive.**

Optionally, **exer** reports performance statistics:

- 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, buffer1 and buffer2, to carry out the operations. A read or write operation can be performed using either buffer. A compare operation uses both buffers.

### Example 4–5 exer

```
P00>>> exer dk*.* -p 0 -secs 36000
```

Read SCSI 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.

```
P00>>> exer -l 2 dka0
```

Read block numbers 0 and 1 from device dka0.

```
P00>>> exer -sb 1 -eb 3 -bc 4 -a 'w' -d1 '0x5a' dka0
```

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.

```
P00>>> ls -l dk*.*
r---   dk                  0/0              0   dka0.0.0.0.0
P00>>> exer dk*.* -bc 10 -sec 20 -m -a 'r'
dka0.0.0.0.0 exer completed
packet                                    IOs            elapsed
idle
8192  3325    27238400              0    166   1360288     20    19

P00>>> exer -eb 64 -bc 4 -a '?w-Rc' dka0
```

A destructive write test over block numbers 0 through 100 on disk dka0. 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 through 5 until enough packets have been written to satisfy the length requirement of 101 blocks.

```
P00>>> exer -a '?r-w-Rc' dka0
```

A nondestructive write test with packet sizes of 512 bytes. Use this test only if the customer has a current backup of any disks being tested. 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 the contents of buffer1 back 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.

You can tailor the behavior of **exer** by using options to specify the following:

- An address range to test within the test device(s)
- The packet size, also known as the 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
- A sequence of individual operations performed on the test devices. The qualifier is called the action string qualifier.

### Syntax

**exer ( [-sb start_block>] [-eb end_block>] [-p pass_count>]**
**[-l blocks>] [-bs block_size>] [-bc block_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 forever or until Ctrl/C. 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 till eof. If writing, and neither **-l** nor **-eb** are specified then exer will write for the size of 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 bytes. The default is 1.

**-d1 <buf1_string>** String argument for eval to generate buffer1 data pattern from. Buffer1 is initialized only once before any I/O occurs. Default = all bytes set to hex 5A's.

**-d2 <buf2_string>** String argument for eval to generate buffer2 data pattern from. Buffer2 is initialized only once 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.

**-a <action_string>**
(continued)

- ? 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 will always be over the same set of block range numbers.
- s Sleep for a number of milliseconds specified by the delay qualifier. If no delay qualifier is present, sleep for 1 millisecond. Times as reported in verbose mode will not necessarily be 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.

## 4.8 floppy_write

**The floppy_write script runs a write test on the floppy drive to determine whether or not you can write on the diskette. Use this script if a customer is unable to write data to the floppy. This is a destructive test, so use a blank floppy.**

### Example 4–6 floppy_write

```
P00>>> floppy_write
Destructive Test of the Floppy started
P00>>> show_status
ID        Program     Device        Pass  Hard/Soft Bytes Written  Bytes Read
-------- ------------ ------------ ------ --------- ------------- -----------
00000001  idle system               0     0    0     0             0
00000c37  exer_kid    dva0.0.0.100  0     0    0     6656          6656
```

The **floppy_write** script uses **exer** to run a write test on the floppy. The test runs in the background. Use the **show_status** command to display the progress of the test. Use the **kill** or **kill_diags** command to terminate the test.

## 4.9 grep

**The grep command is very similar to the UNIX grep command. It allows you to search for “regular expressions”—specific strings of characters—and prints any lines containing occurrences of the strings. Using grep is similar to using wildcards.**

### Example 4–7 grep

```
P00>>> show fru | grep PCI
SMB0.CPB0.PCI1  0 DE500-BA Network Cont
SMB0.CPB0.PCI4  0 DEC PowerStorm
SMB0.CPB0.PCI5  0 NCR 53C895
P00>>>
```

In Example 4–7 the output of the **show fru** command is piped into **grep** (the vertical bar is the piping symbol), which filters out only lines with “PCI.”

**Grep** supports the following metacharacters:

**^** Matches beginning of line

**$** Matches end of line

**.** Matches any single character

**[ ]** Set of characters; [ABC] matches either 'A' or 'B' or 'C'; a dash (other than first or last of the set) denotes a range of characters: [A-Z] matches any uppercase letter; if the first character of the set is '^' then the sense of match is reversed: [^0-9] matches any non-digit; several characters need to be quoted with backslash (\) if they occur in a set: '\', ']', '-', and '^'

***** Repeated matching; when placed after a pattern, indicates that the pattern should match any number of times. For example, '[a-z][0-9]*' matches a lowercase letter followed by zero or more digits.

**+** Repeated matching; when placed after a pattern, indicates that the pattern should match one or more times '[0-9]+' matches any non-empty sequence of digits.

**?** Optional matching; indicates that the pattern can match zero or one times. '[a-z][0-9]?' matches lowercase letter alone or followed by a single digit.

**\** Quote character; prevent the character that follows from having special meaning.

### Syntax

**grep ( [-{c|i|n|v}] [-f <file>] [<expression>] [<file>...] )**

### Arguments

**<expression>** Specifies the target regular expression. If any regular expression metacharacters are present, the expression should be enclosed with quotes to avoid interpretation by the shell.

**<file>...** Specifies the files to be searched. If none are present, then standard input is searched.

### Options

**-c** Print only the number of lines matched.

**-i** Ignore case. By default **grep** is case sensitive.

**-n** Print the line numbers of the matching lines.

**-v** Print all lines that do not contain the expression.

**-f <file>** Take regular expressions from a file, instead of command.

## 4.10 hd

**The hd command dumps the contents of a file (byte stream) in hexadecimal and ASCII.**

### Example 4–8 hd

```
P00>>> hd -eb 0 dpr:2b00                                                       ❶
block 0
00000000  48 45 4C 4C 4F FF FF FF FF FF FF FF FF FF FF FF   HELLO...........
00000010  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
00000020  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
00000030  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF 3A   ...............:
00000040  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
00000050  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
00000060  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
00000070  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
00000080  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
00000090  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
000000a0  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
000000b0  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
000000c0  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
000000d0  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
000000e0  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
000000f0  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
00000100  48 45 4C 4C 4F FF FF FF FF FF FF FF FF FF FF FF   HELLO...........
00000110  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
00000120  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
00000130  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF 3A   ...............:
00000140  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
00000150  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
00000160  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
00000170  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
00000180  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
00000190  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
000001a0  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
000001b0  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
000001c0  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
000001d0  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
000001e0  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
000001f0  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
P00>>>
```

❶ Example 4–8 shows a hex dump to DPR location 2b00, ending at block 0.

### Syntax

**hd [-{byte|word|long|quad}] [-{sb|eb} <n>] <file>[:<offset>].**

### Arguments

| | |
|---|---|
| **<file>**[:**<offset>**] | Specifies the file (byte stream) to be displayed. |

### Options

| | |
|---|---|
| **-byte** | Print out data in byte sizes |
| **-word** | Print out data by word |
| **-long** | Print out data by longword |
| **-quad** | Print out data by quadword |
| **-sb <n>** | Start block |
| **-eb <n>** | End block |

## 4.11 info

**The info command displays registers and data structures. You can enter the command by itself or followed by a number (0 – 8). If you do not specify a number, a list of selections is displayed and you are prompted to enter a selection.**

### Example 4–9 info 0

```
P00>>> info
         0. HWRPB MEMDSC
         1. Console PTE
         2. GCT/FRU 5
         3. Dump System CSRs
         4. IMPURE area (abbreviated)
         5. IMPURE area (full)
         6. LOGOUT area
         7. Dump Error Log
         8. Clear Error Log
Enter selection: 0

HWRPB: 2000    MEMDSC:2d40   Cluster count: 3

Cluster: 0, Usage: Console
START_PFN: 00000000  PFN_COUNT: 0000016e PFN_TESTED: 00000000
     366 pages from 0000000000000000 to 00000000002dbfff

Cluster: 1, Usage: System
START_PFN: 0000016e  PFN_COUNT: 0013fe75 PFN_TESTED: 00003e92
BITMAP_VA: 0000000000000000 BITMAP_PA: 000000027ffd8000
  1310327 good pages from 00000000002dc000 to 00000027fffc5fff

Cluster: 2, Usage: Console
START_PFN: 0011ffe3  PFN_COUNT: 0000001d PFN_TESTED: 00000000
      29 pages from 000000027ffc6000 to 000000027fffffff

P00>>>
```

For information about the data displayed by the **info** commands, see the following documents:

- For **info 0**, **info 1**, and **info 4**, see the *Alpha System Reference Manual, Third Edition* (EY-W938E-DP), available from Digital Press, an imprint of Butterworth-Heinemann.
- For **info 2,** see the Galaxy Console and Alpha Systems V5.0 FRU Configuration Tree Specification.
- For **info 3,** see the Titan 21274 Chipset Functional Specification.

**`info 0`** Displays the SRM memory descriptors as described in the *Alpha System Reference Manual*.

**`info 1`** Displays the page table entries (PTE) used by the console and operating system to map virtual to physical memory. Valid data is displayed only after a boot operation.

**`info 2`** Dumps the Galaxy Configuration Tree (GCT) FRU table. Galaxy is a software architecture that allows multiple instances of *OpenVMS* to execute cooperatively on a single computer.

**`info 3`** Dumps the contents of the system control status registers (CSRs) for the C-chip, D-chip, and P-chips.

**`info 4`** Displays the per CPU impure area in abbreviated form. The console uses this scratch area to save processor context.

**`info 5`** Displays the per CPU impure area in full form.

**`info 6`** Displays the per CPU machine check logout area.

**`info 7`** Displays the contents of the Console Data Log.

**`info 8`** Clears all event frames in the Console Data Log.

For more information on:

- info 0, 1, 4, and 5 see the *Alpha System Reference Manual*
- info 2 see the Galaxy console and Alpha Systems V5.0 FRU configuration Tree Specification.
- info 3 see the Titan Chipset Engineering Specification.
- info 6 and 7 see the AlphaServer ES45 Platform Fault Management Specification.

Example 4–10 shows an **info 1** display.

### Example 4–10 info 1

```
P00>>> info 1
pte 000000003FFA8000 0000000100001101  va 0000000010000000  pa 0000000000002000
pte 000000003FFA8008 0000000200001101  va 0000000010002000  pa 0000000000004000
pte 000000003FFA8010 0000000300001101  va 0000000010004000  pa 0000000000006000
pte 000000003FFA8018 0000000400001101  va 0000000010006000  pa 0000000000008000
pte 000000003FFA8020 0000000500001101  va 0000000010008000  pa 000000000000A000
pte 000000003FFA8028 0000000600001101  va 000000001000A000  pa 000000000000C000
pte 000000003FFA8030 0000000700001101  va 000000001000C000  pa 000000000000E000
pte 000000003FFA8038 0000000800001101  va 000000001000E000  pa 0000000000010000
pte 000000003FFA8040 0000000900001101  va 0000000010010000  pa 0000000000012000
pte 000000003FFA8048 0000000A00001101  va 0000000010012000  pa 0000000000014000
pte 000000003FFA8050 0000000B00001101  va 0000000010014000  pa 0000000000016000
pte 000000003FFA8058 0000000C00001101  va 0000000010016000  pa 0000000000018000
pte 000000003FFA8060 0000000D00001101  va 0000000010018000  pa 000000000001A000
pte 000000003FFA8068 0000000E00001101  va 000000001001A000  pa 000000000001C000
pte 000000003FFA8070 0000000F00001101  va 000000001001C000  pa 000000000001E000
pte 000000003FFA8078 0000001000001101  va 000000001001E000  pa 0000000000020000
pte 000000003FFA8080 0000001100001101  va 0000000010020000  pa 0000000000022000
pte 000000003FFA8088 0000001200001101  va 0000000010022000  pa 0000000000024000
pte 000000003FFA8090 0000001300001101  va 0000000010024000  pa 0000000000026000
pte 000000003FFA8098 0000001400001101  va 0000000010026000  pa 0000000000028000
pte 000000003FFA80A0 0000001500001101  va 0000000010028000  pa 000000000002A000
pte 000000003FFA80A8 0000001600001101  va 000000001002A000  pa 000000000002C000
pte 000000003FFA80B0 0000001700001101  va 000000001002C000  pa 000000000002E000
pte 000000003FFA80B8 0000001800001101  va 000000001002E000  pa 0000000000030000
pte 000000003FFA80C0 0000001900001101  va 0000000010030000  pa 0000000000032000
pte 000000003FFA80C8 0000001A00001101  va 0000000010032000  pa 0000000000034000
pte 000000003FFA80D0 0000001B00001101  va 0000000010034000  pa 0000000000036000
pte 000000003FFA80D8 0000001C00001101  va 0000000010036000  pa 0000000000038000
pte 000000003FFA80E0 0000001D00001101  va 0000000010038000  pa 000000000003A000
pte 000000003FFA80E8 0000001E00001101  va 000000001003A000  pa 000000000003C000
        .
        .
        .
```

Example 4–11 shows an **info 2** display.

## Example 4–11 info 2

```
P00>>> info 2
GCT_ROOT_NODE

GCT_NODE                  21e000
Type                      1
Subtype                   0
Hd_extension              0
Size                      c000
Rev_major                 6
Rev_minor                 0
Id                        0000000000000000
node_flags                0
saved_owner               0
affinity                  0
parent                    0
child                     2c0
fw_usage                  0
Root->lock                ffffffff
Root->transient_level     1f
Root->current_level       1f
Root->console_req         200000
Root->min_alloc           100000
Root->min_align           100000
Root->base_alloc          2000000
Root->base_align          2000000
Root->max_phys_addr       7fffffffff
Root->mem_size            27fffffff
Root->platform_type       40500000022
Root->platform_name       0000000000000280
Root->primary_instance    0
Root->first_free          69c0
Root->high_limit          bcc0
Root->lookaside           0
Root->available           2d00
Root->max_partition       1
Root->partitions          0000000000000180
Root->communities         00000000000001c0
Root->bindings            0000000000000200
Root->max_plat_partition  1
Root->max_desc            1
Root->galaxy_id           21e128
Root->root_flags          1

dump depth view ? (Y/<N> N

dump each node ? (Y/<N>)  N

dump binary ? (Y/<N>) N

show flags ? (Y/<N>) N

Dump a Node – Enter Handle (hex) ?

P00>>>
P00>>>
```

Example 4–12 shows an **info 3** display.

## Example 4–12 info 3

```
P00>>> info

         0. HWRPB MEMDSC
         1. Console PTE
         2. GCT/FRU 5
         3. Dump System CSRs
         4. IMPURE area (abbreviated)
         5. IMPURE area (full)
         6. LOGOUT area
         7. Dump Error Log
         8. Clear Error Log
Enter selection: 3

CCHIP   CSRs:             801a0000000
CSC                  705B80000919792F : 0000
MTR                  00002F641E001225 : 0040
MISC                 0000001100000000 : 0080
AAR0                 0000000000009305 : 0100
AAR1                 0000000200007105 : 0140
AAR2                 0000000100009305 : 0180
AAR3                 0000000240007105 : 01c0
DIM0                 F884003010011000 : 0200
DIM1                 0000000000000000 : 0240
DIM2                 0000000000000000 : 0600
DIM3                 0000000000000000 : 0640
DIR0                 0000000000000000 : 0280
DIR1                 0000000000000000 : 02c0
DIR2                 0000000000000000 : 0680
DIR3                 0000000000000000 : 06c0
DRIR                 0300000000000000 : 0300
TTR                  000000000000077F : 0580
TDR                  F7FFF7FFF7FFF7FF : 05c0


DCHIP   CSRs:             801b0000000
DSC                  7F7F7F7F7F7F7F7F : 0800
DSC2                 7F7F7F7F7F7F7F7F : 08c0
STR                  3939393939393939 : 0840
DREV                 1111111111111111 : 0880

PCHIP 0 CSRs:             80180000000
GWSBA0               0000000000800000 : 0000
GWSBA1               0000000080000001 : 0040
GWSBA2               0000000000000000 : 0080
GWSBA3               0000000000000002 : 00c0
GWSM0                0000000000700000 : 0100
GWSM1                000000003FF00000 : 0140
GWSM2                0000000000000000 : 0180
GWSM3                0000000000000000 : 01c0
GTBA0                0000000000000000 : 0200
GTBA1                0000000000000000 : 0240
GTBA2                0000000000000000 : 0280
GTBA3                0000000000000000 : 02c0
GPCTL                00000004C38000C0 : 0300
GPLAT                000000000000FF00 : 0340
SERROR               0000000000000000 : 0400
```

```
SERREN               000000000000000E   0440
GPERROR              0000000000400000   0500
GPERREN              00000000000007F6   0540
SCTL                 0000000002831611   0700
AWSBA0               0000000000800000 : 0000
AWSBA1               0000000080000001 : 0040
AWSBA2               0000000000000000 : 0080
AWSBA3               0000000000000002 : 10c0
AWSM0                0000000000700000 : 1100
AWSM1                000000003FF00000 : 1140
AWSM2                0000000000000000 : 1180
AWSM3                0000000000000000 : 11c0
ATBA0                0000000000000000 : 1200
ATBA1                0000000000000000 : 1240
ATBA2                0000000000000000 : 1280
ATBA3                0000000000000000 : 12c0
APCTL                48000004C2C200C0 : 1300
APLAT                000000000000FF00 : 1340
AGPERROR             0000054112908000 : 1400
AGPERREN             0000000000000010   1440
APERROR              0000000000000000   1500
APERREN              00000000000007F6   1540

PCHIP 1 CSRs:             80380000000
GWSBA0               0000000000800000   0000
GWSBA1               0000000080000001   0040
GWSBA2               0000000000000000   0080
GWSBA3               0000000000000002   00c0
GWSM0                0000000000700000   0100
GWSM1                000000003FF00000   0140
GWSM2                0000000000000000   0180
GWSM3                0000000000000000   01c0
GTBA0                0000000000000000   0200
GTBA1                0000000000000000   0240
GTBA2                0000000000000000   0280
GTBA3                0000000000000000   02c0
GPCTL                00000004C34000C0   0300
GPLAT                000000000000FF00   0340
SERROR               0000000000000000   0400
SERREN               000000000000000E   0440
GPERROR              0000080000004000   0500
GPERREN              00000000000007F6   0540
SCTL                 0000000002831711   0700
AWSBA0               0000000000800000 : 1000
AWSBA1               0000000080000001 : 1040
AWSBA2               0000000000000000 : 1080
AWSBA3               0000000000000002 : 10c0
AWSM0                0000000000700000 : 1100
AWSM1                000000003FF00000 : 1140
AWSM2                0000000000000000 : 1180
AWSM3                0000000000000000 : 12c0
ATBA0                0000000000000000 : 1200
ATBA1                0000000000000000 : 1240
ATBA2                0000000000000000 : 1280
ATBA3                0000000000000000 : 12c0
APCTL                00000004C1C200C0 : 1300
APLAT                000000000000FF00 : 1340
AGPERROR             0008000000000000 : 1400
AGPERREN             0000000000000010   1440
APERROR              0000000000010000   1500
APERREN              00000000000007F6   1540
```

Example 4–13 shows an **info 4** display.

## Example 4–13 info 4

```
P00>>> info 4
                      cpu00    cpu01    cpu02    cpu03
per_cpu impure area  00004200 00004800 00004e00 00005400
cns$flag             00000001 00000001 00000001 00000001 : 0000
cns$flag+4           00000000 00000000 00000000 00000000 : 0004
cns$hlt              00000000 00000000 00000000 00000000 : 0008
cns$hlt+4            00000000 00000000 00000000 00000000 : 000c
cns$mchkflag         00000210 00000210 00000210 00000210 : 0210
cns$mchkflag+4       00000000 00000000 00000000 00000000 : 0214
cns$fpcr             00000000 00000000 00000000 00000000 : 0318
cns$fpcr+4           8ff00000 8ff00000 8ff00000 8ff00000 : 031c
cns$va               ffffffec fe00385f fe00385f fe00385f : 0320
cns$va+4             ffffffff 00000801 00000801 00000801 : 0324
cns$va_ctl           00000000 00000000 00000000 00000000 : 0328
cns$va_ctl+4         00000000 00000000 00000000 00000000 : 032c
cns$exc_addr         00600930 00000000 00000000 00000000 : 0330
cns$exc_addr+4       00000000 00000000 00000000 00000000 : 0334
cns$ier_cm           00000000 00000000 00000000 00000000 : 0338
cns$ier_cm+4         00000020 00000020 00000020 00000020 : 033c
cns$sirr             00000000 00000000 00000000 00000000 : 0340
cns$sirr+4           00000000 00000000 00000000 00000000 : 0344
cns$isum             00000000 00000000 00000000 00000000 : 0348
cns$isum+4           00000020 00000020 00000020 00000020 : 034c
cns$exc_sum          00001fc0 000010c0 000010c0 000010c0 : 0350
cns$exc_sum+4        00000000 00000000 00000000 00000000 : 0354
cns$pal_base         00008000 00008000 00008000 00008000 : 0358
cns$pal_base+4       00000000 00000000 00000000 00000000 : 035c
cns$i_ctl            16300386 16300386 16300386 16300386 : 0360
cns$i_ctl+4          00000000 00000000 00000000 00000000 : 0364
cns$pctr_ctl         00000000 00000000 00000000 00000000 : 0368
cns$pctr_ctl+4       00000000 00000000 00000000 00000000 : 036c
cns$process_context  00000004 00000004 00000004 00000004 : 0370
cns$process_context+ 00000000 00000000 00000000 00000000 : 0374
cns$i_stat           c0000000 80000000 00000000 80000000 : 0378
cns$i_stat+4         0000013d 00000142 00000174 0000017F : 037c
cns$dtb_alt_mode     00000000 00000000 00000000 00000000 : 0380
cns$dtb_alt_mode+4   00000000 00000000 00000000 00000000 : 0384
cns$mm_stat          00000290 000000e1 000000e1 000000e1 : 0388
cns$mm_stat+4        00000000 00000000 00000000 00000000 : 038c
cns$m_ctl            00000020 00000020 00000020 00000020 : 0390
cns$m_ctl+4          00000000 00000000 00000000 00000000 : 0394
cns$dc_ctl           000000c3 000000c3 000000c3 000000c3 : 0398
cns$dc_ctl+4         00000000 00000000 00000000 00000000 : 039c
cns$dc_stat          00000000 00000000 00000000 00000000 : 03a0
cns$dc_stat+4        00000000 00000000 00000000 00000000 : 03a4
cns$write_many       00000000 00000000 00000000 00000000 : 03a8
cns$write_many+4     00000000 00000000 00000000 00000000 : 03ac
cns$virbnd           00000000 00000000 00000000 00000000 : 03b0
cns$virbnd+4         00000000 00000000 00000000 00000000 : 03b4
cns$sysptbr          00000000 00000000 00000000 00000000 : 03b8
cns$sysptbr+4        00000000 00000000 00000000 00000000 : 03bc
cns$report_1am       00000000 00000000 00000000 00000000 : 03c0
cns$report_1am+4     00000000 00000000 00000000 00000000 : 03c4
P00>>>
```

Example 4–14 shows an **info 5** display.

### Example 4–14 info 5

```
P00>>> info 5
                                cpu00   cpu01    cpu02    cpu03
per_cpu impure area           00004200 00004800 00004e00 00005400
cns$flag                      00000001 00000001 00000001 00000001 : 0000
cns$flag+4                    00000000 00000000 00000000 00000000 : 0004
cns$hlt                       00000000 00000000 00000000 00000000 : 0008
cns$hlt+4                     00000000 00000000 00000000 00000000 : 000c
cns$gpr[0]                    00018000 00018000 00018000 00018000 : 0010
cns$gpr[0]+4                  00000000 00000000 00000000 00000000 : 0014
cns$gpr[1]                    0000001f 0000001f 0000001f 0000001f : 0018
cns$gpr[1]+4                  00000000 00000000 00000000 00000000 : 001c
cns$gpr[2]                    00004180 00004180 00004180 00004180 : 0020
cns$gpr[2]+4                  00000000 00000000 00000000 00000000 : 0024
cns$gpr[3]                    00001101 00001101 00001101 00001101 : 0028
cns$gpr[3]+4                  00000000 00000000 00000000 00000000 : 002c
cns$gpr[4]                    00000000 00000000 00000000 00000000 : 0030
cns$gpr[4]+4                  00000000 00000000 00000000 00000000 : 0034
cns$gpr[5]                    00000000 00000000 00000000 00000000 : 0038
cns$gpr[5]+4                  00000000 00000000 00000000 00000000 : 003c
cns$gpr[6]                    00000000 00000000 00000000 00000000 : 0040
cns$gpr[6]+4                  00000000 00000000 00000000 00000000 : 0044
cns$gpr[7]                    00000000 00000000 00000000 00000000 : 0048
cns$gpr[7]+4                  00000000 00000000 00000000 00000000 : 004c
cns$gpr[8]                    00000000 00000000 00000000 00000000 : 0050
cns$gpr[8]+4                  00000000 00000000 00000000 00000000 : 0054
cns$gpr[9]                    00000000 00008000 00008000 00008000 : 0058
cns$gpr[9]+4                  00000000 00000000 00000000 00000000 : 005c
cns$gpr[10]                   00008000 00008000 00008000 00008000 : 0060
cns$gpr[10]+4                 00000000 00000000 00000000 00000000 : 0064
cns$gpr[11]                   00000008 00000008 00000008 00000008 : 0068
cns$gpr[11]+4                 00000000 00000000 00000000 00000000 : 006c
cns$gpr[12]                   00004200 00004800 00004e00 00005400 : 0070
cns$gpr[12]+4                 00000000 00000000 00000000 00000000 : 0074
cns$gpr[13]                   00000000 ffffffff fffffffe fffffffd : 0078
cns$gpr[13]+4                 00000000 ffffffff ffffffff ffffffff : 007c
cns$gpr[14]                   00000000 00000000 00000000 00000000 : 0080
cns$gpr[14]+4                 00000000 00000000 00000000 00000000 : 0084
cns$gpr[15]                   00000001 000048d8 000048d8 000048d8 : 0088
cns$gpr[15]+4                 00000000 00000000 00000000 00000000 : 008c
cns$gpr[16]                   00000000 00000000 00000000 00000000 : 0090
cns$gpr[16]+4                 00000000 00000000 00000000 00000000 : 0094
cns$gpr[17]                   00000000 00000000 00000000 00000000 : 0098
cns$gpr[17]+4                 00000000 2e313300 2e313300 2e313300 : 009c
cns$gpr[18]                   000000b2 00000000 00000000 00000000 : 00a0
cns$gpr[18]+4                 00000000 00000000 00000000 00000000 : 00a4
cns$gpr[19]                   00000091 00800000 00800000 00800000 : 00a8
cns$gpr[19]+4                 00000000 00000008 00000008 00000008 : 00ac
cns$gpr[20]                   00000005 00000000 00000000 00000000 : 00b0
cns$gpr[20]+4                 00000000 00000000 00000000 00000000 : 00b4
cns$gpr[21]                   00000000 00000000 00000000 00000000 : 00b8
cns$gpr[21]+4                 00000000 00000000 00000000 00000000 : 00bc

        .
        .
        .
```

```
cns$shadow23+4                00000000 00000000 00000000 00000000 : 0314
cns$fpcr                      00000000 00000000 00000000 00000000 : 0318
cns$fpcr+4                    8ff00000 8ff00000 8ff00000 8ff00000 : 031c
cns$va                        ffffffec fe00385f fe00385f fe00385f : 0320
cns$va+4                      ffffffff 00000801 00000801 00000801 : 0324
cns$va_ctl                    00000000 00000000 00000000 00000000 : 0328
cns$va_ctl+4                  00000000 00000000 00000000 00000000 : 032c
cns$exc_addr                  00600930 00000000 00000000 00000000 : 0330
cns$exc_addr+4                00000000 00000000 00000000 00000000 : 0334
cns$ier_cm                    00000000 00000000 00000000 00000000 : 0338
cns$ier_cm+4                  00000020 00000020 00000020 00000020 : 033c
cns$sirr                      00000000 00000000 00000000 00000000 : 0340
cns$sirr+4                    00000000 00000000 00000000 00000000 : 0344
cns$isum                      00000000 00000000 00000000 00000000 : 0348
cns$isum+4                    00000020 00000020 00000020 00000020 : 034c
cns$exc_sum                   00001fc0 000010c0 000010c0 000010c0 : 0350
cns$exc_sum+4                 00000000 00000000 00000000 00000000 : 0354
cns$pal_base                  00008000 00008000 00008000 00008000 : 0358
cns$pal_base+4                00000000 00000000 00000000 00000000 : 035c
cns$i_ctl                     16300386 16300386 16300386 16300386 : 0360
cns$i_ctl+4                   00000000 00000000 00000000 00000000 : 0364
cns$pctr_ctl                  00000000 00000000 00000000 00000000 : 0368
cns$pctr_ctl+4                00000000 00000000 00000000 00000000 : 036c
cns$process_context           00000004 00000004 00000004 00000004 : 0370
cns$process_context+          00000000 00000000 00000000 00000000 : 0374
cns$i_stat                    c0000000 80000000 00000000 80000000 : 0378
cns$i_stat+4                  0000013d 00000142 00000174 0000017f : 037c
cns$dtb_alt_mode              00000000 00000000 00000000 00000000 : 0380
cns$dtb_alt_mode+4            00000000 00000000 00000000 00000000 : 0384
cns$mm_stat                   00000290 000000e1 000000e1 000000e1 : 0388
cns$mm_stat+4                 00000000 00000000 00000000 00000000 : 038c
cns$m_ctl                     00000020 00000020 00000020 00000020 : 0390
cns$m_ctl+4                   00000000 00000000 00000000 00000000 : 0394
cns$dc_ctl                    000000c3 000000c3 000000c3 000000c3 : 0398
cns$dc_ctl+4                  00000000 00000000 00000000 00000000 : 039c
cns$dc_stat                   00000000 00000000 00000000 00000000 : 03a0
cns$dc_stat+4                 00000000 00000000 00000000 00000000 : 03a4
cns$write_many                00000000 00000000 00000000 00000000 : 03a8
cns$write_many+4              00000000 00000000 00000000 00000000 : 03ac
cns$virbnd                    00000000 00000000 00000000 00000000 : 03b0
cns$virbnd+4                  00000000 00000000 00000000 00000000 : 03b4
cns$sysptbr                   00000000 00000000 00000000 00000000 : 03b8
cns$sysptbr+4                 00000000 00000000 00000000 00000000 : 03bc
cns$report_lam                00000000 00000000 00000000 00000000 : 03c0
cns$report_lam+4              00000000 00000000 00000000 00000000 : 03c4
```

Example 4–15 show an **info 6** display.

### Example 4–15 info 6

```
P00>>> info 6
                                       cpu00
per_cpu logout area                   00006000
mchk_crd__flag_frame                  00000000 : 0000
mchk_crd__flag_frame+4                00000000 : 0004
mchk_crd__offsets                     00000000 : 0008
mchk_crd__offsets+4                   00000000 : 000c
mchk_crd__mchk_code                   00000000 : 0010
mchk_crd__mchk_code+4                 00000000 : 0014
mchk_crd__i_stat                      00000000 : 0018
mchk_crd__i_stat+4                    00000000 : 001c
mchk_crd__dc_stat                     00000000 : 0020
mchk_crd__dc_stat+4                   00000000 : 0024
mchk_crd__c_addr                      00000000 : 0028
mchk_crd__c_addr+4                    00000000 : 002c
mchk_crd__dc1_syndrome                00000000 : 0030
mchk_crd__dc1_syndrome+4              00000000 : 0034
mchk_crd__dc0_syndrome                00000000 : 0038
mchk_crd__dc0_syndrome+4              00000000 : 003c
mchk_crd__c_stat                      00000000 : 0040
mchk_crd__c_stat+4                    00000000 : 0044
mchk_crd__c_sts                       00000000 : 0048
mchk_crd__c_sts+4                     00000000 : 004c
mchk_crd__mm_stat                     00000000 : 0050
mchk_crd__mm_stat+4                   00000000 : 0054
mchk_crd__os_flags                    00000000 : 0058
mchk_crd__os_flags+4                  00000000 : 005c
mchk_crd__cchip_dirx                  00000000 : 0060
mchk_crd__cchip_dirx+4                00000000 : 0064
mchk_crd__cchip_misc                  00000000 : 0068
mchk_crd__cchip_misc+4                00000000 : 006c
mchk_crd__pachip0_serror              00000000 : 0070
mchk_crd__pachip0_serror+             00000000 : 0074
mchk_crd__pachip0_aperror             00000000 : 0080
mchk_crd__pachip0_aperror             00000000 : 0084
mchk_crd__pachip0_gperror             00000000 : 0078
mchk_crd__pachip0_gperror             00000000 : 007c
mchk_crd__pachip0_agperro             00000000 : 0088
mchk_crd__pachip0_agperro             00000000 : 008c
mchk_crd__pachip1_serror              00000000 : 0090
mchk_crd__pachip1_serror+             00000000 : 0094
mchk_crd__pachip1_aperror             00000000 : 00a0
mchk_crd__pachip1_aperror             00000000 : 00a4
mchk_crd__pachip1_gperror             00000000 : 0098
mchk_crd__pachip1_gperror             00000000 : 009c
mchk_crd__pachip1_agperro             00000000 : 00a8
mchk_crd__pachip1_agperro             00000000 : 00ac
mchk__flag_frame                      000000f8 : 00b0
mchk__flag_frame+4                    00000000 : 00b4
mchk__offsets                         00000018 : 00b8
mchk__offsets+4                       000000a0 : 00bc
mchk__mchk_code                       00000202 : 00c0
mchk__mchk_code+4                     00000001 : 00c4
mchk__i_stat                          00000000 : 00c8
mchk__i_stat+4                        00000000 : 00cc
mchk__dc_stat                         00000000 : 00d0
```

```
mchk__dc_stat+4                       00000000 : 00d4
mchk__c_addr                          00000000 : 00d8
mchk__c_addr+4                        00000000 : 00dc
mchk__dc1_syndrome                    00000000 : 00e0
mchk__dc1_syndrome+4                  00000000 : 00e4
mchk__dc0_syndrome                    00000000 : 00e8
mchk__dc0_syndrome+4                  00000000 : 00ec
mchk__c_stat                          00000000 : 00f0
mchk__c_stat+4                        00000000 : 00f4
mchk__c_sts                           00000000 : 00f8
mchk__c_sts+4                         00000000 : 00fc
mchk__mm_stat                         00000000 : 0100
mchk__mm_stat+4                       00000000 : 0104
mchk__exc_addr                        0009c250 : 0108
mchk__exc_addr+4                      00000000 : 010c
mchk__ier_cm                          80000000 : 0110
mchk__ier_cm+4                        00000022 : 0114
mchk__isum                            00000000 : 0118
mchk__isum+4                          00000002 : 011c
mchk__reserved_0                      00000000 : 0120
mchk__reserved_0+4                    00000000 : 0124
mchk__pal_base                        00008000 : 0128
mchk__pal_base+4                      00000000 : 012c
mchk__i_ctl                           16304386 : 0130
mchk__i_ctl+4                         00000000 : 0134
mchk__process_context                 00000004 : 0138
mchk__process_context+4               00000000 : 013c
mchk__reserved_1                      00000000 : 0140
mchk__reserved_1+4                    00000000 : 0144
mchk__reserved_2                      00000000 : 0148
mchk__reserved_2+4                    00000000 : 014c
mchk__os_flags                        00000001 : 0150
mchk__os_flags+4                      00000000 : 0154
mchk__cchip_dirx                      00000000 : 0158
mchk__cchip_dirx+4                    40000000 : 015c
mchk__cchip_misc                      00000000 : 0160
mchk__cchip_misc+4                    00000011 : 0164
mchk__pachip0_serror                  00000000 : 0168
mchk__pachip0_serror+4                00000000 : 016c
mchk__pachip0_aperror                 00000000 : 0178
mchk__pachip0_aperror+4               00000000 : 017c
mchk__pachip0_gperror                 00400002 : 0170
mchk__pachip0_gperror+4               00000000 : 0174
mchk__pachip0_agperror                00000000 : 0180
mchk__pachip0_agperror+4              00000000 : 0184
mchk__pachip1_serror                  00000000 : 0188
mchk__pachip1_serror+4                00000000 : 018c
mchk__pachip1_aperror                 00000000 : 0198
mchk__pachip1_aperror+4               00000000 : 019c
mchk__pachip1_gperror                 00000000 : 0190
mchk__pachip1_gperror+4               00000000 : 0194
mchk__pachip1_agperror                00000000 : 01a0
mchk__pachip1_agperror+4              00000000 : 01a4
```

Example 4–16 shows as **info 7** display.

### Example 4–16 info 7

```
P00>>> info 7
Number of Errors Saved = 3
Error 1
 0000 : 0001000400050018       Console Uncorrectable Error Frame Header
 0008 : 0000300a190f1324        OCT 25 15:19:36
 0010 : 0000000300000170

 0000 : 00010001000c0108    Processor Machine Check Frame
 0008 : 0000000000000000        CPU ID
 0010 : 00000000000000f8        Frame Flag/Size
 0018 : 000000a000000018        Frame Offsets
 0020 : 0000000100000098        Frame Revision/Code
 0028 : 0000000020000000        I_STAT
 0030 : 0000000000000000        DC_STAT
 0038 : 0000000000040000        C_ADDR
 0040 : 0000000000000000        DC1_SYNDROME
 0048 : 0000000000000000        DC0_SYNDROME
 0050 : 0000000000000000        C_STAT
 0058 : 000000000000000d        C_STS
 0060 : 00000000000002d1        MM_STAT
 0068 : 00000000001caf00        EXC_ADDR
 0070 : 0000002280000000        IER_CM
 0078 : 0000000200000000        ISUM
 0080 : 0000000000000000        RESERVED
 0088 : 0000000000008000        PAL_BASE
 0090 : 0000000016304386        I_CTL
 0098 : 0000000000000004        PROCESS_CONTEXT
 00a0 : 0000000000000000        Reserved
 00a8 : 0000000000000000        Reserved
 00b0 : 0000000000000004        OS Flags
 00b8 : 0000000000000000        Cchip DIRx
 00c0 : 0000000000000000        Cchip MISC
 00c8 : 0000000000000000        PChip 0 SERROR
 00d0 : 0000000000000000        PChip 0 GPERROR
 00d8 : 0000000000000000        PChip 0 APERROR
 00e0 : 0000000000000000        PChip 0 AGPERROR
 00e8 : 0000000000000000        PChip 1 SERROR
 00f0 : 0000000000000000        PChip 1 GPERROR
 00f8 : 0000000000000000        PChip 1 APERROR
 0100 : 0000000000000000        PChip 1 AGPERROR
 0000 : 00010004000c0010    Clipper DPR Extended Memory Frame
 0008 :                      f0  DPR AAR0 Config
 0009 :                      40  DPR AAR0 Size
 000a :                      d2  DPR AAR1 Config
 000b :                      10  DPR AAR1 Size
 000c :                      f1  DPR AAR2 Config
 000d :                      40  DPR AAR2 Size
 000e :                     d3  DPR AAR3 Config
 000f :                      10  DPR AAR3 Size

 0000 : 0001000a000c0058    Titan Extended Memory Frame
 0008 : 0000000000009305        AAR0
 0010 : 0000000200007105        AAR1
 0018 : 0000000100009305        AAR2
 0020 : 0000000240007105        AAR3
 0028 : 0000000002831611        SCTL
 0030 : 00000004c38000c0        GPCTL
 0038 : 48000004c2c200c0        APCTL
 0040 : 0000000002831711        SCTL
 0048 : 00000004c34000c0        GPCTL
 0050 : 00000004c1c200c0        APCTL

Error 2

 0000 : 0001000400050018    Console Uncorrectable Error Frame Header
 0008 : 0000300a190f1426        OCT 25 15:20:38
 0010 : 0000000200000218

 0000 : 00010002000c0108    System Machine Check Frame
 0008 : 0000000000000000        CPU ID
 0010 : 00000000000000f8        Frame Flag/Size
        .
        .
        .


0108 : 0000000000000000        GTBA3

Error 3

 0000 : 0001000200050018    System Event Frame Header
 0008 : 0000300a190f1523        OCT 25 15:21:35
 0010 : 0000000100000080

 0000 : 00010003000c0080    System Event Frame
 0008 : 0000000000000000        CPU ID
 0010 : 0000000000000070        Frame Flag/Size
 0018 : 0000001800000018        Frame Offsets
 0020 : 0000000100000206        Frame Revision/Code
 0028 : 0000000000000000        OS Flags
 0030 : 0000000000000680        Cchip DIRx
 0038 : 0000000000000060        TIG SMIR
 0040 : 000000000000000f        TIG CPUIR
 0048 : 0000000000000007        TIG PSIR
 0050 : 0000000000000000        LM78 ISR
 0058 : 0000000000000000        Door Open
 0060 : 0000000000000000        Temperature Warning
 0068 : 0000000000000000        Fan Fault
 0070 : 0000000000000000        Power Down Code
 0078 : 0000000000000000        Reserved
```

Example 4–17 shows an **info 8**.

### Example 4–17 info 8

```
P00>>> info 8
         0. HWRPB MEMDSC
         1. Console PTE
         2. GCT/FRU 5
         3. Dump System CSRs
         4. IMPURE area (abbreviated)
         5. IMPURE area (full)
         6. LOGOUT area
         7. Dump Error Log
         8. Clear Error Log
Enter selection:
```

## 4.12 kill and kill_diags

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

### Example 4–18 kill and kill_diags

```
P00>>> memexer 3
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

dva0.0.0.1000.0 exer completed

packet       IOs            elapsed  idle
size         IOs     bytes read  bytes written    /sec bytes/sec seconds  secs
512          112          28672          28672      5      2748      21     16
```

The **kill** command terminates a specified process. The **kill_diags** command terminates all diagnostics.

**Syntax**

**kill_diags**

**kill [PID. . . ]**

**Arguments**

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

## 4.13 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.**

The following example shows no errors.

### Example 4–19 memexer

```
P00>>> memexer 3
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

*** ERROR - DIMM J2 on MMB 1 Failed ***

P00>>> kill_diags
P00>>>
```

If the memory configuration is very large, the console might not test all of the memory. The upper limit is 1 GB.

Use the **show_status** command to display the progress of the tests. Use the **kill** or **kill_diags** command to terminate the test.

### Syntax

**memexer [number]**

### Arguments

**[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.

## 4.14 memtest

**The memtest command exercises a specified section of memory. Typically memtest is run from the built-in console script. Advanced users may want to use the specific options described here.**

### Example 4–20 memtest

```
P00>>> sh mem                         ❶
 Array      Size       Base Address    Intlv Mode
---------  ----------  ----------------  ----------
    0        256Mb     0000000060000000    2-Way
    1        512Mb     0000000040000000    2-Way
    2        256Mb     0000000070000000    2-Way
    3       1024Mb     0000000000000000    2-Way

     2048 MB of System Memory
                      ❷        ❸        ❹
P00>>> memtest -sa 400000 -l 2000000 -p 10&
*** Hard Error - Error #43 - Memory compare error

Diagnostic Name  ID             Device  Pass  Test  Hard/Soft   2-JAN-2000
memtest          00000118       brd0     1     1     1    0     12:00:01
Expected value:                ffffffffe
Received value:                ffffffff
Failing addr:                    400004
*** Error - DIMM 3 on MMB 2 Failed ***❺
```

❶ Use the **show memory** command or an **info 0** command to see where memory is located.

❷ Starting address

❸ Length of the section to test in bytes

❹ Passcount. In this example, the test will run for 10 passes.

❺ The test detected a failure on DIMM 3, which is located on MMB 2.

Use the **show_status** command to display the progress of the test. Use the **kill** or **kill_diags** command to terminate the test.

**Memtest** provides a graycode memory test. The test writes to memory and then reads the previously written value for comparison. The section of memory that is tested has its data destroyed. The **-z** option allows testing outside of the main memory pool. Use caution because this option can overwrite the console.

**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.

**NOTE:** *If **memtest** is used to test large sections of memory, testing may take a while to complete. 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 — Graycode Test

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 are 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.

### 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 indefinitely or until Ctrl/C is issued. 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.

**Options**

**-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.

**CAUTION:** *This flag can overwrite the console. If the system hangs, press the Reset button.*

**-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 **-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.

## 4.15 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.**

### Example 4–21 net -ic and net -s

```
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>>>
```

**Syntax**

**net [-ic]**

**net [-s]**

**Arguments**

| | |
|---|---|
| **<port_name>** | Specifies the Ethernet port on which to operate, either ei*0 or ew*0. |

## 4.16 nettest

**The nettest command tests the network ports using MOP loopback. Typically nettest is run from the built-in console script. Advanced users may want to use the specific options and environment variables described here.**

### Example 4–22 nettest

```
P00>>> nettest ei*                ❶
P00>>> nettest -mode in    ew*    ❷
P00>>> nettest -mode ex -w 10  e* ❸
```

❶ Internal loopback test on port ei*0

❷ Internal loopback test on ports ewa0/ewb0

❸ External loopback test on port eia0 or ewa0; wait 10 seconds between tests

**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. 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 in the syntax table in this section. 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 **show_status** display to determine the process ID when terminating an individual diagnostic test. Use the **kill** or **kill_diags** command to terminate tests.

### 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 until terminated by a **kill** or **kill_diags** command 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.

## 4.17 set sys_serial_num

**The set sys_serial_num command sets the system serial number. This command is used by Manufacturing for establishing the system serial number, which is then propagated to all FRU devices that have EEPROMs. The sys_serial_num environment variable can be read by the operating system.**

**IMPORTANT:** *The system serial number must be set correctly. Compaq Analyze will not work with an incorrect serial number.*

### Example 4–23 set sys_serial_num

```
P00>>> set sys_serial_num NI900100022
```

When the system motherboard (SMB) is replaced, you must use the **set sys_serial_num** command to restore the master setting.

**Syntax**

**set sys_serial_num *value***

*Value* is the system serial number, which is on a sticker on the back of the system chassis.

## 4.18 show error

**The show error command reports errors logged to the FRU EEPROMs.**

### Example 4–24 show error

```
P00>>> show error
 ❶
SMB0           TDD - Type: 15 Test: 15  SubTest: 15  Error: 15              ❷
001f8408  0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F   ................
SMB0           SDD - Type: 14 LastLog: 0   Overwrite: 0                     ❸
001f8408  0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F   ................
001f8418  0F 0F 0F 0F 0F 0F 0F 0F 0F 00 00 00 00 00 00 00   ................
001f8428  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
001f8438  00 00 00 00 00 00 00 00 00 00 FF 00 00 00 00 00   ................
001f8448  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
001f8458  00 00 00 00 00 00 00 00                           ........
SMB0           Bad checksum   0 to 64  EXP:dc  RCV:dd                       ❹
001f8408  80 08 00 01 53 00 01 00 00 00 00 00 00 00 00 00   ....S...........
001f8418  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
001f8428  FF 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
001f8438  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 DD   ...............Y
SMB0           Bad checksum  64 to 126  EXP:e1  RCV:0f
001f8408  4A FF FF FF FF FF FF FF 02 35 34 2D 31 32 33 34   J........54-1234
001f8418  35 2D 30 31 2E 41 30 30 31 20 20 00 00 09 44 91   5-01.A001  ...D.
001f8428  34 51 15 41 41 41 41 41 41 41 41 41 41 41 41 41   4Q.AAAAAAAAAAAAA
001f8438  0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F   ................
SMB0           Bad checksum 128 to 254  EXP:0c  RCV:0d
001f8408  0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F   ................
001f8418  0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F   ................
001f8428  0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F   ................
001f8438  0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 00 00   ................
001f8408  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
001f8418  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
001f8428  FF 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
001f8438  00 00 00 00 00 00 00 00 00 00 00 00 00 4A 21 0D   .............J!.
SMB0           SYS_SERIAL_NUM Mismatch                                      ❺
P00>>>
```

The output of the **show error** command is based on information logged to the serial control bus EEPROMs on the system FRUs. Both the operating system and the ROM-based diagnostics log errors to the EEPROMs. This functionality allows you to generate an error log from the console environment. No errors are displayed for fans or the OCP because these components do not have an EEPROM.

### Syntax

**show error**

All FRUs with errors are displayed. If no errors are logged, nothing is displayed and you are returned to the SRM console prompt.

Example 4–24 shows TDD, SDD, checksum, and sys_serial_num mismatch errors logged to the EEPROM on the system motherboard (SMB0). Table 4–2 shows a reference to these errors. The bit masks correspond to the bit masks that would be displayed in the E field of the **show fru** command.

❶ FRU to which errors are logged; in this example the system motherboard, SMB0.

❷ A TDD error has been logged. TDDs (test-directed diagnostics) test specific functions sequentially. Typically, nothing else is running during the test. TDDs are performed in SROM or XSROM or early in the console power-up flow.

❸ An SDD error has been logged. SDDs (symptom-directed diagnostics) are generic diagnostic exercisers that try to cause random behavior and look for failures or “symptoms.” All SDDs are logged by Compaq Analyze.

❹ Three checksum errors have been logged.

❺ There was a mismatch between the serial number on the system motherboard and the system serial number. This could occur if a motherboard from a system with a different serial number was swapped into this system.

### Table 4–2 Show Error Message Translation

| Bit Mask<br>(E Field) | Text Message | Meaning and Action |
|---|---|---|
| **01** | <*fruname*> Hardware Failure | Module failure. FRUs that are known to be connected but are unreadable are considered hardware failures. An example is power supplies. |
| **02** | <*fruname*> TDD - Type:0 Test: 0 SubTest: Error: 0 | Serious error. Run the Compaq Analyze GUI, if necessary, to determine what action to take. If you cannot run Compaq Analyze, replace the module. |
| **04** | <*fruname*> SDD - Type:0 LastLog: 0 Overwrite: 0 | Serious error. Compaq Analyze (CA) has written a FRU callout into the SDD area and DPR global area. Follow the instructions given by Compaq Analyze. |
| **08** | <*fruname*> EEPROM Unreadable | Reserved. |
| **10** | <*fruname*> Bad checksum 0 to 64 EXP:01 RCV:02 | Informational. Use the **clear_error** command to clear the error unless TDD or SDD is also set. |
| **20** | <*fruname*> Bad checksum 64 to 126 EXP:01 RCV:02 | Informational. Use the **clear_error** command to clear the error unless TDD or SDD is also set. |
| **40** | <*fruname*> Bad checksum 128 to 254 EXP:01 RCV:02 | Informational. Use the **clear_error** command to clear the error unless TDD or SDD is also set. |
| **80** | <*fruname*> SYS_SERIAL_NUM Mismatch | Informational. Use the **clear_error** command to clear the error unless TDD or SDD is also set. |

## 4.19 show fru

**The show fru command displays the physical configuration of FRUs. Use show fru -e to display FRUs with errors.**

### Example 4–25 show fru

```
P00>>> build smb0 54-25385-01.e01 ay94412345
P00>>> show fru
❶               ❷  ❸                     ❹           ❺        ❻

FRUname         E  Part#              Serial#    Model/Other Alias/Misc
SMB0            00 54-30292-02.A01    SW03300011
SMB0.CPU0       00 54-30466-01.A01    SW03200044
SMB0.CPU1       00 54-30466-01.A01    SW03200042
SMB0.CPU2       00 54-30466-01.A01    SW03200037
SMB0.CPU3       00 54-30466-02.A01    SW04300170
SMB0.MMB0       00 54-30348-02.A01    SW02800955
SMB0.MMB0.J4    00 54-30350-DB.A02 CP SW03600466
SMB0.MMB0.J8    00 54-30350-DB.A02 CP SW03600482
SMB0.MMB0.J5    00 54-30350-DB.A02 CP SW03600472
SMB0.MMB0.J9    00 54-30350-DB.A02 CP SW03600525
SMB0.MMB0.J2    00 54-30350-DB.A02 CP SW03600468
SMB0.MMB0.J6    00 54-30350-DB.A02 CP SW03600471
SMB0.MMB0.J3    00 54-30350-DB.A02 CP SW03600473
SMB0.MMB0.J7    00 54-30350-DB.A02 CP SW03600522
SMB0.MMB1       00 54-30348-02.A01    SW02800940
SMB0.MMB1.J4    00 54-30350-AA.A01 CP SW02100172
SMB0.MMB1.J8    00 54-30350-AA.A01 CP SW02100172
SMB0.MMB1.J5    00 54-30350-AA.A01 CP NI01400121
SMB0.MMB1.J9    00 54-30350-AA.A01 CP SW01400169
SMB0.MMB1.J2    00 54-30350-AA.A01 CP SW01400214
SMB0.MMB1.J6    00 54-30350-AA.A01 CP SW01400205
SMB0.MMB1.J3    00 54-30350-AA.A01 CP SW02100172
SMB0.MMB1.J7    00 54-30350-AA.A01 CP SW02100172
SMB0.MMB2       00 54-30348-02.A01    SW02800932
SMB0.MMB2.J4    00 54-30350-DB.A02 CP SW03600523
SMB0.MMB2.J8    00 54-30350-DB.A02 CP SW03600519
SMB0.MMB2.J5    00 54-30350-DB.A02 CP SW03600527
SMB0.MMB2.J9    00 54-30350-DB.A02 CP SW03600469
SMB0.MMB2.J2    00 54-30350-DB.A02 CP SW03600465
SMB0.MMB2.J6    00 54-30350-DB.A02 CP SW03600467
SMB0.MMB2.J3    00 54-30350-DB.A02 CP SW03600479
SMB0.MMB2.J7    00 54-30350-DB.A02 CP SW03600477
SMB0.MMB3       00 54-30348-02.A01    SW02800927
SMB0.MMB3.J4    00 54-30350-AA.A01 CP SW02100172
SMB0.MMB3.J8    00 54-30350-AA.A01 CP SW02100172
SMB0.MMB3.J5    02 54-30350-AA.A01 CP NI01400118
SMB0.MMB3.J9    00 54-30350-AA.A01 CP SW01400170
SMB0.MMB3.J2    02 54-30350-AA.A01 CP NI01400122
SMB0.MMB3.J6    00 54-30350-AA.A01 CP SW01400210
SMB0.MMB3.J3    00 54-30350-AA.A01 CP SW02100172
SMB0.MMB3.J7    00 54-30350-AA.A01 CP SW02100172
SMB0.CPB0       00 54-30418-01.A01    SW03000307
```

```
JIO0            00 54-25575-01        -          Junk I/O
SMB0.CPB0.PCI1  00                               NCR 53C895
SMB0.CPB0.PCI2  00                               DECchip 21
SMB0.CPB0.PCI3  00                               DE500-AA N
SMB0.CPB0.PCI5  00                               ELSA GLori
SMB0.CPB0.PCI7  00                               DEGPA-SA
SMB0.CPB0.PCI8  00                               DEGPA-SA
SMB0.CPB0.PCI9  00                               NCR 53C895
OCP0            00 70-33894-0x        -          OCP
PWR0            00 30-49448-01. C05   30-49448-0 API-76      7f
PWR1            00 30-49448-01. C05   30-49448-0 API-76      7f
PWR2            00 30-49448-01. C05   30-49448-0 API-76      7f
FAN1            00 70-40073-01        -          Fan
FAN2            00 70-40073-01        -          Fan
FAN3            00 70-40072-01        -          Fan
FAN4            00 70-40071-01        -          Fan
FAN5            00 70-40073-02        -          Fan
FAN6            00 70-40074-01        -          Fan

P00>>>
```

❶ FRUname The FRU name recognized by the SRM console. The name also indicates the location of that FRU in the physical hierarchy.

SMB = system motherboard; CPU = CPUs; MMB = memory motherboard; DIM = DIMMs; CPB = PCI backplane; PCI = PCI option; SBM = SCSI backplane; PWR = power supply; FAN = fans; JIO= I/O connector module (junk I/O).

❷ E Error field. Indicates whether the FRU has any errors logged against it. FRUs without errors show 00 (hex). FRUs with errors have a non-zero value that represents a bit mask of possible errors. See Table 4–3.

❸ Part # The part number of the FRU in ASCII, either a Compaq part number or 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 4-byte sequence number is displayed in hex.

❺ Model/Other Optional data. For Compaq FRUs, the Compaq part alias number (if one exists). For vendor FRUs, the year and week of manufacture.

❻ Alias/Misc Miscellaneous information about the FRUs. For Compaq FRUs, a model name, number, or the common name for the entry in the Part # field. For vendor FRUs, the manufacturer's name.

Table 4–3 lists bit assignments for failures that could potentially be listed in the E (error) field of the **show fru** command. Because the E field is only two characters wide, bits are “or’ed” together if the device has multiple errors. For example, the E field for a FRU with both TDD (02) and SDD (04) errors would be 06:

010 | 100 = 110 (6)

### Table 4–3 Bit Assignments for Error Field

| Bit Mask (E Field) | Meaning |
|---|---|
| 01 | Hardware failure |
| 02 | TDD error has been logged |
| 04 | SDD error has been logged |
| 08 | Reserved |
| 10 | Checksum failure on bytes 0-62 |
| 20 | Checksum failure on bytes 64-126 |
| 40 | Checksum failure on bytes 128-254 |
| 80 | FRU’s system serial number does not match system’s |

## 4.20 show_status

**The show_status command displays the progress of diagnostics. The 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.**

### Example 4–26 show status

```
P00>>> show_status

❶           ❷           ❸           ❹     ❺         ❻              ❼
 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 The 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.

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**

## 4.21 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.**

### Example 4–27 sys_exer

```
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 V1.91-33, Tru64 UNIX PALcode V1.87-27
...
starting console on CPU 0
```

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] [-t]**

### Arguments

**[-lb]** The loopback option runs console loopback tests for the COM2 serial port and the parallel port during the test sequence.

**[-t]** Number of seconds to run. The default is run until terminated by a **kill** or **kill_diags** command.

## 4.22 test

**The test command verifies all the devices in the system. This command can be used on all supported operating systems.**

### Example 4–28 test -lb

```
P00>>> test -lb
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 Serial Port 1(external loopback)
Testing the parallel Port(external loopback)
Testing the VGA (Alphanumeric Mode only)
Testing the EW* Network
P00>>>
```

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, and the status of each subsystem test is displayed to the console terminal as the tests progress. 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.

### Syntax

**test [*argument*]**

Use the **-lb** (loopback) argument for console loopback tests.

To run a complete diagnostic test using the **test** command, the system configuration must include:

- A serial loopback connected to the COM2 port (not included)
- A parallel loopback connected to the parallel port (not included)
- 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, MK* tapes, DV* floppy.

---

**NOTE:** *You must install media to test disks, tapes, and the floppy drive. Since no write tests are performed, it is safe to test disks and tapes that contain data.*

---

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.

# Chapter 5
# Error Logs

This chapter tells how to interpret error logs reported by the operating system. The following topics are covered:

- Error Log Analysis with Compaq Analyze
- Fault Detection and Reporting
- Machine Checks/Interrupts
- Environmental Errors Captured by SRM

## 5.1 Error Log Analysis with Compaq Analyze

**Compaq Analyze (CA) is a fault management diagnostic tool that is used to determine the cause of hardware failures. Compaq Analyze performs system diagnostic processing of both single and multiple error/fault events.**

Compaq Analyze may or may not be installed on the customer's system with the operating system, depending on the release cycle. If CA is installed, the Compaq Analyze Director starts automatically as part of the system start-up. CA provides automatic background analysis. When an error event occurs, it triggers the firing of an analysis rule. The analysis engine collects and processes the information and typically generates a “problem found” report, if appropriate. The report can be automatically sent to users on a notification mailing list and, if DSNlink is installed, a call can be logged with the customer support center.

Compaq Analyze has the capability to support the *Tru64 UNIX* and *OpenVMS* operating systems on Alpha platforms.

**NOTE:** *Compaq Analyze is a successor tool to DECevent and typically does not support the same systems as DECevent.*

### UNIX Indictment

For each CPU indictment that is sent to the operating system a callout report is generated. After the bad component is replaced the following commands must be executed to bring the new components on-line for use. The following is an example of using the Indictment command.

```
#hwmgr –status comp -ngood
      STATUS        ACCESS                 INDICT
HWID: HOSTNAME      SUMMARY    STATE       STATE     LEVEL   NAME

2:     mcsse1     critical    offline   available   high   CPU0

#hwmgr –online –name CPU0
hwmgr:CPU0 is now online
```

**NOTE:** *The indicted problem state attached to the previous component is still in effect even though a new component may have been inserted. Use the command "* `hwmgr –unindict –id <hwid>` *" to clear the problem state when the component is operating properly.*

The following is an example of a CPU `unindict` command.

```
#hwmgr –unindict –id 2
hwmgr:Unindict operation was successful
```

### 5.1.1 WEB Enterprise Service (WEBES) Director

**Compaq Analyze uses the functionality contained in the WEBES Director, a process that manages all other WEBES processes and executes continuously on the machine when configured to do so. The Director manages the decomposition processing of system error events, provides required information to the analysis engine, and performs notification message routing for the system. Compaq Analyze provides the functionality for system event analysis and Bit-To-Text (BTT) translation.**

Compaq Analyze, includes common WEBES code. Subsequent releases of Compaq Analyze will continue to ship with the common WEBES code.

The Director is started when the system is booted. Normally you do not need to start the Director. If the Director has stopped running, restart it by following the instructions in the WEBES Compaq Analyze User Guide documentation.

Compaq Analyze includes a graphical user interface (WUI) that allows the user to interact with the Director. While only one Director process executes on the machine at any time, many WUI processes can run at the same time, connected to the single Director. Refer to the Compaq Analyze installation and user manuals for the respective operating system to launch the Compaq Analyze WUI. The Compaq service tools Web site available to customers is:

http://www.support.compaq.com/svctools

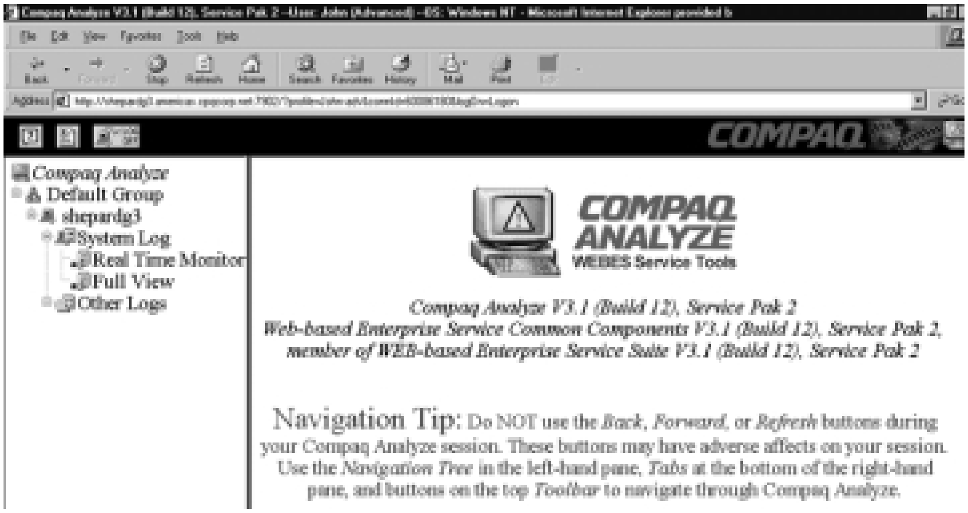
The applicable Compaq Analyze documentation includes the following:

- *Compaq Analyze User' s Guide*
- *Compaq Analyze Installation Guide for Tru64 UNIX*
- *Compaq Analyze Installation Guide for OpenVMS*
- *Compaq Analyze Releases Notes*

### 5.1.2 Using Compaq Analyze

**After you have logged on to Compaq Analyze the following screen appears. If an event has occurred, it is listed under “localhost” events. See Figure 5–1.**

Figure 5–1 Compaq Analyze Initial Screen

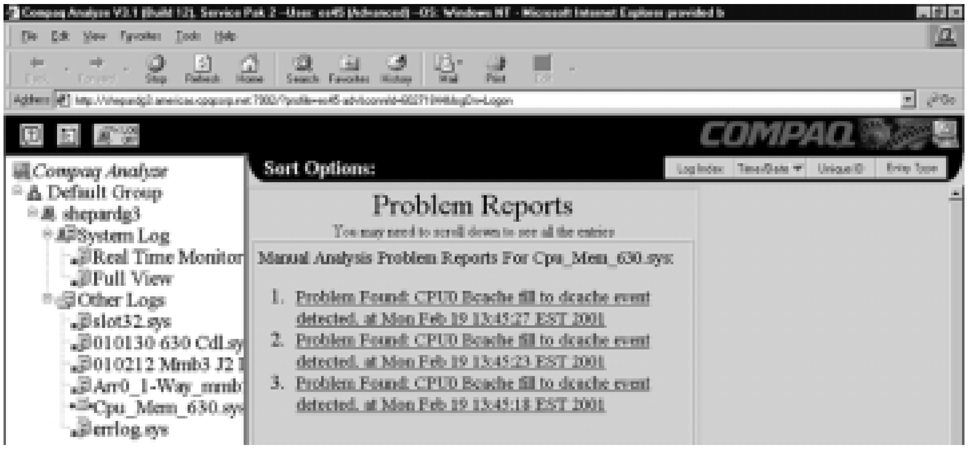


*Compaq Analyze V3.1 (Build 12), Service Pak 2*
*Web-based Enterprise Service Common Components V3.1 (Build 12), Service Pak 2, member of WEB-based Enterprise Service Suite V3.1 (Build 12), Service Pak 2*

Navigation Tip: Do NOT use the *Back*, *Forward*, or *Refresh* buttons during your Compaq Analyze session. These buttons may have adverse affects on your session. Use the *Navigation Tree* in the left-hand pane, *Tabs* at the bottom of the right-hand pane, and buttons on the top *Toolbar* to navigate through Compaq Analyze.

1. In this example, the Other Logs file is selected and the Problem Reports display in Figure 5–2 appears.

### Figure 5–2 Problem Reports Screen



Compaq Analyze V3.1 (Build 12), Service Pak 2 –User: es45 (Advanced) –OS: Windows NT - Microsoft Internet Explorer provided b
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Edit
Address
Go
COMPAQ
Compaq Analyze
Default Group
shepardg3
System Log
Real Time Monitor
Full View
Other Logs
slot32.sys
010130-630 Cdl.sy
010212 Mmb3 J2 I
Arr0_1-Way_mmb
Cpu_Mem_630.sys
errlog.sys
Sort Options:
Log Index
Time/Date
Unique ID
Entry Type

Problem Reports

You may need to scroll down to see all the entries

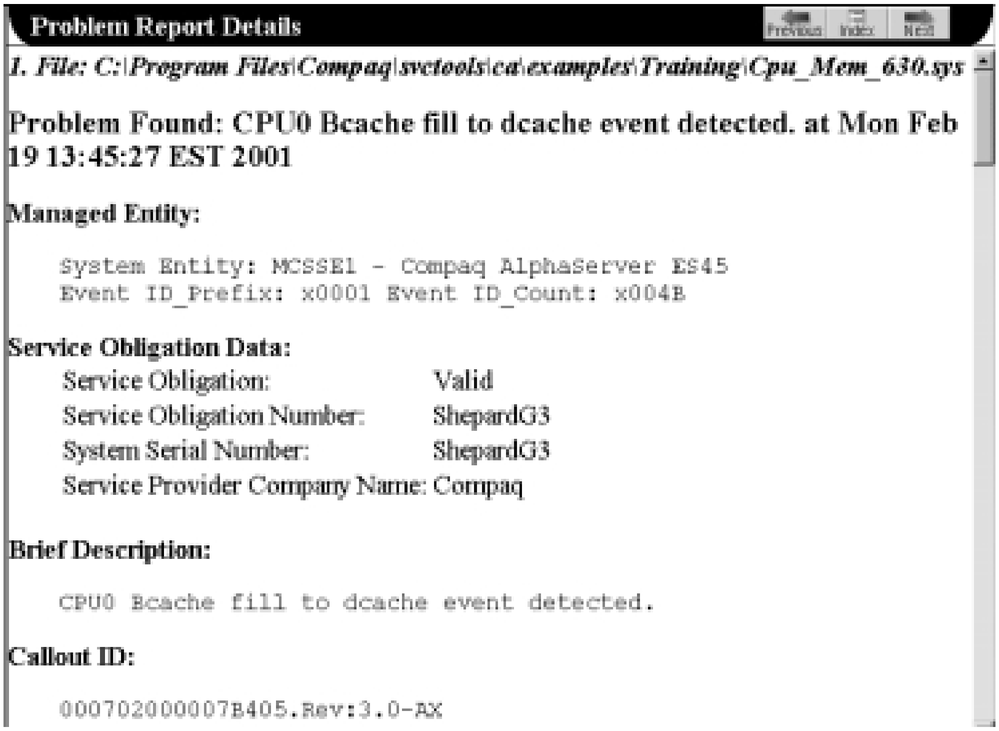
Manual Analysis Problem Reports For Cpu_Mem_630.sys:

1. Problem Found: CPU0 Bcache fill to dcache event detected, at Mon Feb 19 13:45:27 EST 2001
2. Problem Found: CPU0 Bcache fill to dcache event detected, at Mon Feb 19 13:45:23 EST 2001
3. Problem Found: CPU0 Bcache fill to dcache event detected, at Mon Feb 19 13:45:18 EST 2001

2. Cpu_Mem_630.sys is selected and the problem reports are listed. You may select any log listed in Other Logs to view a list of all problems found. You may also view each report by clicking on the underlined hot link under Problem Reports.

3. Figure 5–3 provides an example problem report.

### Figure 5–3 Compaq Analyze Problem Report Details



***1. File: C:\Program Files\Compaq\svctools\ca\examples\Training\Cpu_Mem_630.sys***

**Problem Found: CPU0 Bcache fill to dcache event detected. at Mon Feb 19 13:45:27 EST 2001**

**Managed Entity:**

```
    System Entity: MCSSE1 - Compaq AlphaServer ES45
    Event ID_Prefix: x0001 Event ID_Count: x004B
```

**Service Obligation Data:**

| | |
|---|---|
| Service Obligation: | Valid |
| Service Obligation Number: | ShepardG3 |
| System Serial Number: | ShepardG3 |
| Service Provider Company Name: | Compaq |

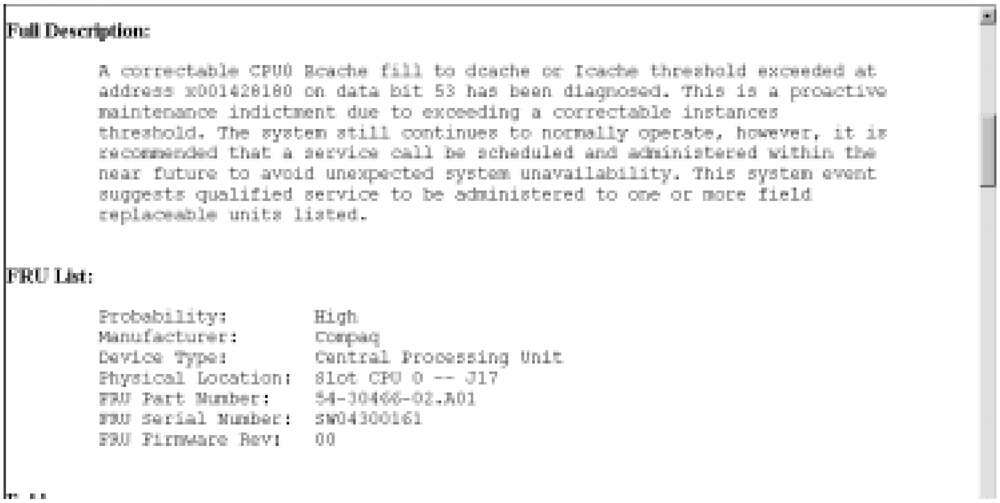
**Brief Description:**

```
    CPU0 Bcache fill to dcache event detected.
```

**Callout ID:**

```
    000702000007B405.Rev:3.0-AX
```

### Figure 5–3 Compaq Analyze Problem Report Details (Continued)



**Full Description:**

```
        A correctable CPU0 Bcache fill to dcache or Icache threshold exceeded at
        address x001428180 on data bit 53 has been diagnosed. This is a proactive
        maintenance indictment due to exceeding a correctable instances
        threshold. The system still continues to normally operate, however, it is
        recommended that a service call be scheduled and administered within the
        near future to avoid unexpected system unavailability. This system event
        suggests qualified service to be administered to one or more field
        replaceable units listed.
```

**FRU List:**

```
        Probability:        High
        Manufacturer:       Compaq
        Device Type:        Central Processing Unit
        Physical Location:  Slot CPU 0 -- J17
        FRU Part Number:    54-30466-02.A01
        FRU Serial Number:  SW04300161
        FRU Firmware Rev:   00
```

#### Managed Entity

The Managed Entity designator includes the system host name (typically a computer name for networking purposes), the type of computer system (“Compaq AlphaServer ES45”), and the error event identification. The error event identification uses new common event header Event_ID_Prefix and Event_ID_Count components. The Event_ID_Prefix refers to an OS specific identification for this event type. The Event_ID_Count indicates the number of this event and the event type.

#### Service Obligation Data

Provides Obligation number and validity, system serial number, and company name of service provider.

#### Brief Description

The Brief Description designator indicates whether the error event is related to the CPU, system (PCI, storage, and so on), or environmental subsystem.

### Callout ID

The last 12 characters of the Callout ID designator can be used to determine the revision level of the analysis rule-set that is being used.

### Full Description

The Full Description designator provides detailed error information, which can include a description of the detected fault or error condition, the specific address or data bit where this fault or error occurred, the probable FRU list, and service related information.

### FRU List

The FRU List designator lists the most probable defective FRUs.  This list indicates that one or more of these FRUs needs to be serviced.  The information typically includes the FRU probability, manufacturer, system device type, system physical location, part number, serial number, and firmware revision level (if applicable).

### 5.1.3 Bit to Test

**The following table is an example of the Common Event Header (CEH) for Cpu_Mem_630.sys. To access the CEH, select the Events tab for the problem report selected.**

#### Table 5–1 Common Event Header Example Table (CEH) V2.0

```
OS_Type                  2                  -- OpenVMS AXP
Hardware_Arch            4                  -- Alpha
CEH_Vendor_ID            3,564              -- Compaq Computer Corp
Hdwr_Sys_Type            38                 -- Titan Corelogic
Logging_CPU              0                  -- CPU Logging this Event
CPUs_In_Active_Set       2
                                            -- Correctable Processor
Entry_Type               630                Event
                                            -- Compaq AlphaServer
DSR_Msg_Num              1,972              ES45
                                            .... CPU Slots: 2 (1000
                                            Mhz)
                                            .... AGP Slots: 1
                                            .... PCI Slots: 8
                                            .... MMB Slots: 8 (DIMMs)
Chip_Type                12                 -- EV68CB - 21264C
CEH_Device               0
CEH_Device_ID_0          x0000 0000
CEH_Device_ID_1          x0000 0000
CEH_Device_ID_2          x0000 0000
Unique_ID_Count          79
Unique_ID_Prefix         1


TLV Section of CEH
TLV_DDR_String            Generic IDE/ATAPI disk
TLV_DSR_String            Compaq AlphaServer ES45
TLV_Sys_Serial_Num        prqv
TLV_Time_as_Local         Tue, 30 Jan 2001 14:20:17 -0500
TLV_OS_Version            X765-SSB
TLV_Computer_Name         MCSSE1
```

```
Logout_Frame_CPU_Section
Frame_Size               x0000 00B0
Frame_Flags              x8000 0000
CPU_Area_Offset          x0000 0018
System_Area_Offset       x0000 0058
                                               Machine Check Logout
Mchk_Error_Code          x0000 0086
                                               Frame Error Code
Value[31:0]              x86                   CPU Non-Fatal

Frame_Rev                x0000 0001
I_STAT                   x0000 0000 0000 0000  Ibox Status Register
DC_STAT                  x0000 0000 0000 0008  Dcache Status Register
                                               Dcache ECC during load
ECC_Err_Ld[3]            x1
                                               instruction

                                               Cbox Read Erred
C_ADDR                   x0000 0000 3BDC 63C0
                                               Address Register
                                               Access Reference
Error_Ref[42:6]          xEF 718F
                                               Location
Io_M[43]                 x0                    System Memory Access

C_SYNDROME_1             x0000 0000 0000 0098  High QW Data Syndrome
QW_Upper[7:0]            x98                   Data Bit 53

C_SYNDROME_0             x0000 0000 0000 0000  Low QW Data Syndrome
QW_Lower[7:0]            x0                    No Syndrome

                                               Cbox Read Status
C_STAT                   x0000 0000 0000 000C
                                               Register
                                               Single-bit Bcache ECC
Cbox_Error[4:0]          xC
                                               Fill to Icache

                         x0000 0000 0000       Cache Block Access
C_STS
                         000D                  Status Register
                                               Shared, Valid,
Cblock_Status[3:0]       xD
                                               Parity

                         x0000 0000 0000       Memory Management
MM_STAT
                         0000                  Status Register
Logout_Frame_System_Section
SW_Error_Sum_Flags     x0000 0000 0000 0004
                                               No Pchip0 PCI Error
Pchip0_PCI_Error[0]    x0
                                               Detected
                                               No Pchip1 PCI Error
Pchip1_PCI_Error[1]    x0
                                               Detected
                                               Pchip or CPU Memory
Pchip_Mem_Error[2]     x1
                                               Error Detected
                                               No PCI Hot Plug Slot
Hot_Plug_Slot[39:32]   x0
                                               Intervention

                                               Cchip Device Interrupt
Cchip_DIRx             x0000 0000 0000 0000
                                               Request Register
Cchip_MISC             x0000 0000 0000 0000    Cchip Miscellaneous
```

```
                                               Register
Nxs[31:29]             x0                      CPU 0 Source Device

P0_Serror              x0000 0000 0000 0000    No Error Detected
Bus_Source[53:52]      x0                      GPCI Bus
TransAction_Cmd[55:54] x0                      DMA Read
ECC_Syndrome[63:56]    x0                      No Data Bit Error

P0_GPerror             x0000 0000 0000 0000    No Error Detected
PCI_Cmd[55:52]         x0                      Interrupt Acknowledge

P0_APerror             x0000 0000 0000 0000    No Error Detected
PCI_Cmd[55:52]         x0                      Interrupt Acknowledge

P0_AGPerror            x0000 0000 0000 0000    No Error Detected
AGP_Lost_Err[0]        x0
AGP_Cmd[52:50]         x0                      Read

P1_Serror              x0000 0000 0000 0000    No Error Detected
Bus_Source[53:52]      x0                      GPCI Bus
TransAction_Cmd[55:54] x0                      DMA Read
ECC_Syndrome[63:56]    x0                      No ECC Error

P1_GPerror             x0000 0000 0000 0000    No Error Detected
PCI_Cmd[55:52]         x0                      Interrupt Acknowledge

P1_APerror             x0000 0000 0000 0000    No Error Detected
PCI_Cmd[55:52]         x0                      Interrupt Acknowledge

P1_AGPerror            x0000 0000 0000 0000    No Error Detected

START OF SUBPACKETS IN THIS EVENT

ES4X Dual Port RAM Subpacket, Version 1
                                     Non - Split, Set0 - 4 Dimms
DPR_0                x40             Only, configured as lowest array
                                     Non - Split, Set0 - 4 Dimms
DPR_2                x41             Only, configured as next lowest
                                     array
                                     Non - Split, Set0 - 4 Dimms
DPR_4                x42             Only, configured as next highest
                                     array
                                     Non - Split, Set0 - 4 Dimms
DPR_6                x43             Only, configured as highest
                                     array
```

```
System Memory / IO Configuration Subpacket, Version 1
AAR_0                x0000 0000 0000 6005 Memory Array 0
                                          Configuration Register
Sa0[8]               x0                   Non - Split Array
Asiz0[15:12]         x6                   512 Mb
Addr0[34:24]         x0                   Array0 Base Address [34:24]
                                          Bits

AAR_1                x0000 0000 2000 6005 Memory Array 1
                                          Configuration Register
Sa1[8]               x0                   Non - Split Array
Asiz1[15:12]         x6                   512 Mb
Addr1[34:24]         x20                  Array1 Base Address [34:24]
                                          Bits

AAR_2                x0000 0000 4000 6005 Memory Array 2
                                          Configuration Register
Sa2[8]               x0                   Non - Split Array
Asiz2[15:12]         x6                   512 Mb
Addr2[34:24]         x40                  Array2 Base Address [34:24]
                                          Bits

AAR_3                x0000 0000 6000 6005 Memory Array 3
                                          Configuration Register
Sa3[8]               x0                   Non - Split Array
Asiz3[15:12]         x6                   512 Mb
Addr3[34:24]         x60                  Array3 Base Address [34:24]
                                          Bits

P0_SCTL              x7265 5361 6870 6C41 Pchip0 System Control
                                          Register
REV[7:0]             x41
PID[8]               x0                   Pchip ID Value
RPP[9]               x0
ECCEN[10]            x1                   DMA ECC Enabled
SWARB[12:11]         x1                   GPCI/APCI (RR) > AGPX
CRQMAX[19:16]        x0
CDQMAX[23:20]        x7
PTPMAX[27:24]        x8
INUM[28]             x0                   256K Max Downstream PTP/PIO
                                          Writes to bypass PIO Read
NEWAMU[29]           x1                   GPCI Enabled to Perform PTE
                                          Fetch Xactions
PTPWAR[30]           x1                   PTP Writes Enabled During
                                          Pending Reads

P0_GPCTL             x3534 5345 2072 6576 Pchip 0 Gport Control
                                          Register
FBTB[0]              x0
THDIS[1]             x1                   TLB Anti-Thrash Disabled
CHAINDIS[2]          x1                   PIO Write Chaining Disabled
TGTLAT[4:3]          x2                   Target Latency Timer = 32
                                          PCI Clocks
Win_HOLE[5]          x1                   512K - 1Mb Win-Hole ENabled
MnStr_WIN_Enable[6]  x1                   Monster Window Enabled
ARBENA[7]            x0
```

```
PRIGRP[15:8]         x65
PPRI[16]             x0
PCISPD66[17]         x1                    GPCI Frequency = 66 MHz
CNGSTLT[21:18]       xC                    12 DMA Reads Retry w/no
                                           delayed Completion
PTPDESTEN[29:22]     x81
DPCEN[30]            x0                    Data Parity Checking
                                           Disabled
APCEN[31]            x0                    Address Parity Checking
                                           Disabled
DCR_Timer[33:32]     x1                    DCR Timer Count = 2^11
EN_Stepping[34]      x1                    Address Stepping Enabled

P0_APCTL             x3320 6C65 646F 4D20  Pchip0 Aport Control
                                           Register
FBTB[0]              x0
THDIS[1]             x0
CHAINDIS[2]          x0
TGLAT[4:3]           x0                    TGLAT = 128 PCI Clocks
HOLE_Enable[5]       x1                    Window-Hole Enabled
MWIN_Enable[6]       x0
ARBENA[7]            x0
PRIGRP[15:8]         x4D
PCISPD66[17]         x1                    APCI = 66MHz
CNGSTLT[21:18]       xB                    11 DMA Reads Retry w/no
                                           delayed Completion
PTPDESTEN[29:22]     x91
DPCEN[30]            x1
APCEN[31]            x0                    Address Parity Error
                                           Checking Disabled
DCR_Timer[33:32]     x1                    DCRT Count = 2^11
EN_Stepping[34]      x1                    PCI Config Address Stepping
                                           Enabled
AGP_Rate[53:52]      x2                    AGP Rate = 4X
AGP_SBA_Enabled[54]  x0
AGP_Enabled[55]      x0
AGP_Present[57]      x1                    AGP Bus Enabled
AGP_HP_RD[60:58]     x4                    4 Cchip HP Outstanding
                                           Reads
AGP_LP_RD[63:61]     x1                    1 Cchip LP Outstanding Read

P1_SCTL              x3A54 4556 2D51 0046  Pchip1 System Control
                                           Register
REV[7:0]             x46
PID[8]               x0                    Pchip PID = 0
RPP[9]               x0
ECCEN[10]            x0                    Pchip1 ECC Disabled
SWARB[12:11]         x0                    GPCI > APCI > AGPX
CRQMAX[19:16]        x1                    Max Cchip Requests from
                                           both Pchips
CDQMAX[23:20]        x5                    Max Dchip Data Xfrs from
                                           both Pchips
PTPMAX[27:24]        xD                    Max PTP Reqs from both
                                           Pchips
INUM[28]             x0                    256K MAX PTP/PIO Writes
                                           Enabled to bypass PIO Read
```

```
NEWAMU[29]           x1                   AMU Enabled to Perform PTE
                                          Fetch Xactions
PTPWAR[30]           x0                   PTP Writes Disabled During
                                          Pending Reads

P1_GPCTL             x7261 7473 2E2E 2E0A Pchip1 Gport Control
                                          Register
FBTB[0]              x0                   PCI Fast Back-To_Back
                                          Xactions Disabled
THDIS[1]             x1                   TLB Anti-Thrashing Enabled
CHAINDIS[2]          x0                   GPCI PIO Write Chaining
                                          Disabled
TGLAT[4:3]           x1                   Target RetryTimer = 64 PCI
                                          Clocks
WIN_Hole[5]          x0                   512Kb - 1Mb Window Hole
                                          Disabled
Mnstr_Win_Enable[6]  x0                   Monster Window Disabled
ARBENA[7]            x0                   Internal Arbitor Disabled
PRIGRP[15:8]         x2E
PCISPD66[17]         x1                   GPCI Frequency = 66 Mhz
CNGSTLT[21:18]       xB                   11 DMA Reads Retry w/No
                                          Delayed Completion Enabled
PTPDESTEN[29:22]     xB8
DPCEN[30]            x0                   Data Parity Error Detection
                                          Disabled
APCEN[31]            x0                   Address Parity Error
                                          Detection Disabled
DCRTV[33:32]         x3                   DCR Timer = 2^8 Counts
EN_Stepping[34]      x0                   Address Stepping Disabled

P1_APCTL             x7250 5B20 676E 6974 Pchip1 Aport Control
                                          Register
FBTB[0]              x0                   PCI Fast Back-To-Back
                                          Xactions Disabled
THDIS[1]             x0                   TLB Anti-Thrashing Enabled
CHAINDIS[2]          x1                   APCI PIO Write Chaining
                                          Disabled
TGLAT[4:3]           x2                   Target Latency Timer = 32
                                          PCI Clocks
Win_Hole[5]          x1                   512Kb - 1Mb Hole Enabled
Mnstr_Win_Enable[6]  x1                   Monster Window Enabled
ARBENA[7]            x0                   Arbitor Disabled
PRIGRP[15:8]         x69
PPRI[16]             x0
PCISPD66[17]         x1                   APCI Frequency = 66 Mhz
CNGSLT[21:18]        xB                   11 DMA Read Retry w/No
                                          Delayed Completion Enabled
PTPDESTEN[29:22]     x9D
DPCEN[30]            x1                   Data Parity Error Detection
                                          Enabled
APCEN[31]            x0                   Address Command Parity
                                          Error Detection Disabled
DCRTV[33:32]         x0                   DCR Timer = 2^15 Counts
EN_Stepping[34]      x0                   Address Stepping Disabled
AGP_Rate[53:52]      x1                   AGP Rate = 2X
```

```
AGP_SBA_EN[54]       x1                   SideBand Addressing Enabled
AGP_EN[55]           x0                   AGP Xactions Disabled
AGP_Present[57]      x1                   agp_present = 1
AGP_HP_RD[60:58]     x4                   4 Cchip Pending HP Reads
AGP_LP_RD[63:61]     x3                   3 Cchip Pending LP Reads
```

## 5.2 Fault Detection and Reporting

**Table 5–2 provides a summary of the fault detection and correction components of ES45 systems.**

Generally, PALcode handles exceptions/interrupts as follows:

1. The PALcode determines the cause of the exception/interrupt.
2. If possible, it corrects the problem and passes control to the operating system for error notification, reporting, and logging before returning the system to normal operation.

   If PALcode is unable to correct the problem, it

   - Logs double error halt error frames into the flash ROM
   - Logs uncorrectable error logout frames to the DPR
   - For single error halts, logs the uncorrectable logout frame into the DPR.
3. If error/event logging is required, control is passed through the OS Privileged Architecture Library (PAL) handler. The operating system error handler logs the error condition into the binary error log. Compaq Analyze should then diagnose the error to the defective FRU.

### Table 5–2 ES45 Fault Detection and Correction

| Component | Fault Detection/Correction Capability |
|---|---|
| Alpha 21264 (EV68) microprocessor | Contains error checking and correction (ECC) logic for data cycles. Check bits are associated with all data entering and exiting the microprocessor.<br>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. |

## 5.3 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 PAL handler.

Table 5–3 lists the machine checks/interrupts that are related to error events. The designations — 630, 670, 620, 660, and 680 — indicate a system control block (SCB) offset to the fatal system error handler for *Tru64 UNIX* and *OpenVMS*.

### Table 5–3 Machine Checks/Interrupts

| Error Type | Error Descriptions |
|---|---|
| CPU Correctable Error (630)<br>Generic Alpha 21264 (EV68 ) correctable errors. | B-cache probe hit single-bit ECC error<br>D-cache tag parity error on issue<br>I-cache tag or data parity error<br>D-cache victim single-bit ECC error<br>B-cache single-bit ECC fill error to I-stream or D-stream<br>Memory single-bit ECC fill error to I-stream or D-stream |
| CPU Uncorrectable Error (670)<br>Fatal microprocessor machine check errors that result in a system crash. | PAL detected bugcheck error<br>Operating system detected bugcheck error<br>EV68 detected second D-cache store EEC error<br>EV68 detected D-cache tag parity error in pipeline 0 or 1<br>EV68 detected duplicate D-cache tag parity error<br>EV68 detected double-bit ECC memory fill error<br>EV68 detected double-bit probe hit EEC error<br>EV68 detected B-cache tag parity error |

### Table 5–3 Machine Checks/Interrupts (Continued)

| Error Type | Error Descriptions |
|---|---|
| System Correctable Error (620)<br>ES45-specific correctable errors. | System detected ECC single-bit error |
| System Uncorrectable Error (660)<br>A system-detected machine check that occurred as a result of an “off-chip” request to the system. | Uncorrectable ECC error<br>Nonexistent memory reference<br>PCI system bus error (SERR)<br>PCI read data parity error (RDPE)<br>PCI address/command parity error (APE)<br>PCI no device select (NDS)<br>PCI target abort (TA)<br>Invalid scatter/gather page table entry (SGE) error<br>PCI data parity error (PERR)<br>Flash ROM write error<br>PCI target delayed completion retry time-out (DCRTO)<br>PCI master retry time-out (RTO 2**24) error<br>PCI-ISA software NMI error |
| System Environmental Error (680)<br>System-detected machine check caused by an overtemperature condition, fan failure, or power supply failure. | Overtemperature failure (>50° C) (see Note)<br>Uncorrectable Fan 5 failure<br>Complete power supply failure<br>Fan failure (redundant fan)<br>Power supply failure (redundant supply)<br>High temperature warning (>45° C and <50° C) |

**NOTE:** *For overtemperature failure, the position of jumper J26 determines whether the failure is fatal or nonfatal. See Appendix B.*

### 5.3.1 Error Logging and Event Log Entry Format

**The operating system error handlers generate several entry types. 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.

Table 5–4 shows an event structure map for a system uncorrectable PCI target abort error.

**NOTE:** *See Appendix D for the source data Compaq Analyze uses to isolate to the FRUs.*

## Table 5–4 Sample Error Log Event Structure Map (ES45 with 10 PCI Slots)

| OFFSET(hex) | 63 56 | 55 48 | 47 40 | 39 32 | 31 24 | 23 16 | 15 8 | 7 0 |
|---|---|---|---|---|---|---|---|---|
| ech0000<br>ech+nnnn | NEW COMMON OS HEADER | | | | | | | |
| lfh0000<br>lfh+nnnn | STANDARD LOGOUT FRAME HEADER | | | | | | | |
| lfEV680000<br>lfEV68+nnnn | COMMON PAL EV68 SECTION<br>(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]<br>lfett_138[u] | Pchip1 Extended Titan/Typhoon System Packet | | | | | | | |
| eelcb_140 | Pchip 1 PCI Slot 4 Single Device Bus Snapshot Packet | | | | | | | |
| eelcb_190 | Pchip 1 PCI Slot 5 Single Device Bus Snapshot Packet | | | | | | | |
| eelcb_1E0 | Pchip 1 PCI Slot 6 Single Device Bus Snapshot Packet | | | | | | | |
| eelcb_230 | Pchip 1 PCI Slot 7 Single Device Bus Snapshot Packet | | | | | | | |
| eelcb_280 | Pchip 1 PCI Slot 8 Single Device Bus Snapshot Packet | | | | | | | |
| eelcb_2D0 | Pchip 1 PCI Slot 9 Single Device Bus Snapshot Packet | | | | | | | |
| 2D8 | Termination or End Packet | | | | | | | |

## 5.4 Environmental Errors Captured by SRM

**If an environmental error occurs while the SRM console is running, a logout frame similar to Example 5–1 is sent to the console output device. The logout frame is preceded by the message "***unexpected system event through vector 680 on CPU *n*." (usually CPU 0.) For register definitions, see Appendix D.**

### Example 5–1 Console Level Environmental Error Logout Frame

```
P00>>>
*** unexpected system event through vector 680 on CPU 0
os_flags        0000000000000000
cchip_dirx      0004000000000000
tig_smir        0000000000000008
tig_cpuir       000000000000000f
tig_psir        0000000000000003
lm78_isr        0000000000000000
door_open       0000000000000004                ❶
temp_warning    0000000000000000
fan_ctrl_fault  0000000000000000
power_down_code 0000000000000000
reserved_1      0000000000000000
```

❶ This example shows a fan door open event.

```
P00>>>
*** unexpected system event through vector 680 on CPU 0
os_flags        0000000000000000
cchip_dirx      0004000000000000
tig_smir        0000000000000008
tig_cpuir       000000000000000f
tig_psir        0000000000000003
lm78_isr        0000000000000000
door_open       0000000000000040                ❷
temp_warning    0000000000000000
fan_ctrl_fault  0000000000000000
power_down_code 0000000000000000
reserved_1      0000000000000000
```

❷ This example shows a fan door closing event.

# Chapter 6
# System Configuration and Setup

This chapter describes how to configure and set up ES45 systems. The following topics are covered:

- System Consoles
- Displaying the Hardware Configuration
- Setting Environment Variables
- Setting Automatic Booting
- Changing the Default Boot Device
- Setting SRM Security
- Configuring Devices
- Booting Linux

## 6.1 System Consoles

**The SRM console program is located in a flash ROM on the system motherboard. From the console interface, you can set up and boot the operating system, display the system configuration, and run diagnostics. For complete information, see the *ES45 Owner's Guide*.**

### SRM Console

Systems running the *Tru64 UNIX* or *OpenVMS* operating systems are configured from the SRM console, a command-line interface (CLI). From the CLI you can enter commands to configure the system, view the system configuration, boot the system, and run ROM-based diagnostics.

**NOTE:** *The operating systems use different algorithms for system time. If you switch between operating systems (for example, between UNIX and OpenVMS), be sure to reset the time at the operating system level.*

### Linux

The procedure for installing Linux on an Alpha system is described in the Alpha Linux installation document for your Linux distribution. The installation document can be downloaded from the following Web site:

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

### RMC CLI

The remote management console (RMC) provides a command-line interface (CLI) for controlling the system. You can use the CLI either locally or remotely (modem connection) to power the system on and off, halt or reset the system, and monitor the system environment. You can also use the **dump**, **env**, and **status** commands to help diagnose errors. See Chapter 7 for details.

### 6.1.1 Selecting the Display Device

**The SRM console environment variable determines to which display device (VT-type terminal or VGA monitor) the console display is sent.**

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 sends 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 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 (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 (a graphics device) and then resets it to a serial device. After the system initializes, output will be displayed on the serial terminal.

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

### 6.1.2 Setting the Control Panel Message

**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.

You can use the SRM **set ocp_text** command to change this message (see Example 6–1). The message can be up to 16 characters and must be entered in quotation marks.

#### Example 6–1 set ocp_text

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

## 6.2 Displaying the Hardware Configuration

**View the system hardware configuration by entering commands 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.**

Use the following SRM console commands to view the system configuration. See the *Owner's Guide* for details.

**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).

**show memory** Displays configuration of main memory.

## 6.3 Setting Environment Variables

**Environment variables pass configuration information between the console and the operating system. Their settings determine how the system powers up, boots the operating system, and operates.**

- 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*.

### set *envar*

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

**set *envar value***

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

***value*** The new value of the environment variable.

New values for the following environment variables take effect only after you reset the system by pressing the Reset button or issuing the **init** command.

**auto_action**
**console**
**cpu_enabled**
**os_type**
**pk*0_fast**
**pk*0_host_id**
**pk*0_soft_term**
**console_memory_allocation**

### show *envar*

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

**show *envar***

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

Table 6–1 summarizes the SRM environment variables used most often on the ES45 system.

### Table 6–1 SRM Environment Variables

| Variable | Attributes | Description |
|---|---|---|
| **auto_action** | NV,W[1] | Action the console should take following an error halt or power failure. Defined values are:<br>**boot**—Attempt bootstrap.<br>**halt**—Halt, enter console I/O mode.<br>**restart**—Attempt restart. If restart fails, try boot. |
| **bootdef_dev** | NV,W | 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 | Default file name used for the primary bootstrap when no file name is specified by the **boot** command. The default value is **NULL**. |
| **boot_osflags** | NV,W | Default parameters to be passed to system software during booting if none are specified by the **boot** command.<br>**OpenVMS**: Additional parameters are the *root_number* and *boot flags*. The default value is **NULL**.<br>*root_number:* Directory number of the system disk on which OpenVMS files are located.<br>0 (default)—[SYS0.SYSEXE]<br>1—[SYS1.SYSEXE]<br>2—[SYS2.SYSEXE]<br>3—[SYS3.SYSEXE] |

[1] NV—Nonvolatile. The last value saved by system software or set by console commands is preserved across cold bootstraps (when the system goes through a full initialization), and long power outages.

W—Warm nonvolatile. The last value set by system software is preserved across warm bootstraps (UNIX **shutdown -r** command, OpenVMS **REBOOT** command, or a crash and reboot; not all of the SRM initialization is run) and restarts.

## Table 6–1 SRM Environment Variables (Continued)

| Variable | Attributes | Description |
|---|---|---|
| **boot_osflags (continued)** | NV,W | *boot_flags:* The hexadecimal value of the bit number or numbers to set. To specify multiple boot flags, add the flag values (logical OR).<br>1—Bootstrap conversationally (enables you to modify SYSGEN parameters in SYSBOOT).<br>2—Map XDELTA to running system.<br>4—Stop at initial system breakpoint.<br>8—Perform a diagnostic bootstrap.<br>10—Stop at the bootstrap breakpoints.<br>20—Omit header from secondary bootstrap file.<br>80—Prompt for the name of the secondary bootstrap file.<br>100—Halt before secondary bootstrap.<br>10000—Display debug messages during booting.<br>20000—Display user messages during booting.<br>**Tru64 UNIX:** The following parameters are used with this operating system:<br>**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.<br>**s**—Stop in single-user mode. Boots /vmunix to single-user mode and stops at the # (root) prompt.<br>**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. |

*Continued on next page*

## Table 6–1 SRM Environment Variables (Continued)

| Variable | Attributes | Description |
|---|---|---|
| **boot_osflags**<br>(continued) | | **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.<br>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.<br>Baud rate values are 1800, 2000, 2400, 3600, 4800, 7200, 9600, 19200, 38400, 57600. |
| **com2_baud** | NV,W | Sets the baud rate of the COM2 port. The default baud rate is 9600.<br>Baud rate values are 1800, 2000, 2400, 3600, 4800, 7200, 9600, 19200, 38400, 57600. |
| **com1_flow**<br>**com2_flow** | NV,W | The **com1_flow** and **com2_flow** environment variables indicate the flow control on the serial ports. Defined values are:<br>**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.<br>**software**—Use XON/XOFF(default). This is the setting for a standard serial terminal.<br>**hardware**—Use modem signals CTS/RTS. Use this setting if you are connecting a modem to a serial port. |
| **com1_mode** | NV | Specifies the COM1 data flow paths so that data either flows through the RMC or bypasses it. |

### Table 6–1 SRM Environment Variables (Continued)

| Variable | Attributes | Description |
|---|---|---|
| **com1_modem**<br>**com2_modem** | NV,W | Used to tell the operating system whether a modem is present on the COM1 or COM2 ports, respectively.<br>**On**—Modem is present.<br>**Off**—Modem is not present (default value). |
| **console** | NV | Sets the device on which power-up output is displayed.<br>**Graphics**—Sets the power-up output to be displayed at a VGA monitor or device connected to the VGA module.<br>**Serial**—Sets the power-up output to be displayed on the device that is connected to the COM1 (MMJ) port. |
| **console_memory _allocation** | NV | Determines which memory locations the SRM console will allocate for its private use.<br>**Old—**For 1 gigabyte or less, the console carves memory from 0–2 megabytes and at the end of memory, leaving all memory in between available to the operating system. If there is more than 1 gigabyte, the console creates a "memory hole" for the operating system just under 1 gigabyte.<br>**New—**The console takes all needed memory from 0 megabytes to whatever amount is needed. It does not matter how much memory is installed and no holes are ever created. |

*Continued on next page*

### Table 6–1 SRM Environment Variables (Continued)

| Variable | Attributes | Description |
|---|---|---|
| **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. |
| **ei*0_inet_init** or<br>**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). |
| **ei*0_mode** or<br>**ew*0_mode** | NV | Sets the Ethernet controller to the default Ethernet device type.<br>**aui**—Sets the default device to AUI.<br>**bnc**—Sets the default device to ThinWire.<br>**fast**—Sets the default device to fast 100BaseT.<br>**fastfd**—Sets the default device to fast full duplex 100BaseT.<br>**full**—Set the default device to full duplex twisted pair.<br>**Twisted-pair**— Sets the default device to 10BaseT (twisted-pair). |
| **ei*0_protocols** or<br>**ew*0_protocols** | NV | Determines which network protocols are enabled for booting and other functions.<br>**Mop**—Sets the network protocol to MOP for systems using the OpenVMS operating system.<br>**Bootp**—Sets the network protocol to bootp for systems using the Tru64 UNIX operating system.<br>**Bootp,mop**—When the settings are used in a list, the mop protocol is attempted first, followed by bootp. |

### Table 6–1 SRM Environment Variables (Continued)

| Variable | Attributes | Description |
|---|---|---|
| **heap_expand** | NV | Increases the amount of memory available for the SRM console's heap. Valid selections are:<br>NONE (default)<br>64KB<br>128KB<br>256KB<br>512KB<br>1MB<br>2MB<br>3MB<br>4MB |
| **kbd_hardware type** | NV | Sets the keyboard hardware type as either PCXAL or LK411 and enables the system to interpret the terminal keyboard layout correctly. |
| **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). |
| **memory_test** | NV | Specifies the extent to which memory will be tested on Tru64 UNIX. The options are:<br>**Full**—Full memory test will be run. Required for OpenVMS.<br>**Partial**—First 256 MB of memory will be tested.<br>**None**—Only first 32 MB will be tested. |
| **ocp_text** | NV | Overrides the default control panel display text with specified text. |

*Continued on next page*

### Table 6–1 SRM Environment Variables (Continued)

| Variable | Attributes | Description |
|---|---|---|
| **os_type** | NV | Sets the default operating system.<br>**vms or unix**—Sets system to boot the SRM firmware. |
| **password** | NV | Sets a console password. Required for placing the SRM into secure mode. |
| **pci_parity** | NV | Disable or enable parity checking on the PCI bus.<br>**On**—PCI parity enabled (default value)<br>**Off**—PCI parity disabled<br>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.<br>**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.<br>**1**—Sets the default speed for devices on the controller to fast SCSI mode.<br>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. |

## Table 6–1 SRM Environment Variables (Continued)

| Variable | Attribute | Description |
|---|---|---|
| **pk*0_host_id** | NV | Sets the controller host bus node ID to a value between 0 and 7.<br>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.<br>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:<br>**off**—Turns off both low 8 bits and high 8 bits.<br>**Low**—Turns on low 8 bits and turns off high 8 bits.<br>**High**—Turns on high 8 bits and turns off low 8 bits.<br>**On**—Turns on both low 8 bits and high 8 bits. |
| **sys_serial_num** | NV | Sets the system serial number, which is then propagated to all FRUs that have EEPROMs. The serial number can be read by the operating system. |
| **tt_allow_login** | NV | Enables or disables login to the SRM console firmware on alternative console ports.<br>**0**—Disables login on alternative console ports.<br>**1**—Enables login on alternative console ports (default setting).<br>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.<br>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. |

## 6.4 Setting Automatic Booting

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

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 (Linux or *Tru64 UNIX*)

### 6.4.1 Setting the Operating System 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.**

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
```

See the *Owner's Guide* for more information.

## 6.5 Changing the Default Boot Device

**You can change the default boot device with the set bootdef_dev command.**

You can designate a default boot device. You 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 *Owner's Guide* for more information.

## 6.6 Setting SRM Security

**The set password and set secure commands set SRM security. The login command turns off security for the current session. The clear password command returns the system to user mode.**

The SRM console has two modes, user mode and secure mode.

- User mode allows you to use all SRM console commands. User mode is the default mode.
- Secure mode allows you to use only the **boot** and **continue** commands. The **boot** command cannot take command-line parameters when the console is in secure mode. The console boots the operating system using the environment variables stored in NVRAM (**boot_file**, **bootdef_dev**, **boot_flags**).

### Example 6–2 Set Password

```
P00>>> set password                                   ❶
Please enter the password:
Please enter the password again:
P00>>>

P00>>> set password                                   ❷
Please enter the password:
Please enter the password again:
Now enter the old password:
P00>>>

P00>>> set password
Please enter the password:
Password length must be between 15 and 30 characters ❸
P00>>>
```

❶ Setting a password. If a password has not been set and the **set password** command is issued, the console prompts for a password and verification. The password and verification are not echoed.

❷ Changing a password. If a password has been set and the **set password** command is issued, the console prompts for the new password and verification, then prompts for the old password. The password is not changed if the validation password entered does not match the existing password stored in NVRAM.

❸ The password length must be between 15 and 30 alphanumeric characters. Any characters entered after the 30th character are not stored.

### Example 6–3 set secure

```
P00>>> set secure                                ❶
Console is secure. Please login.
P00>>> login                                     ❷
Please enter the password:
P00>>> b dkb0
```

❶ The **set secure** command console puts the console into secure mode. A password must be set before you can issue **set secure**. Once the console is secure, only the **boot** and **continue** commands can be used. The **boot** command cannot take command-line parameters.

❷ Entering the **login** command turns off security features for the current console session. This allows the operator to enter any SRM command—in this case, a **boot** command with command-line parameters.

### Example 6–4 clear password

```
P00>>> clear password
Please enter the password:
Password successfully cleared.
P00>>>
```

Clearing the password returns the system to user mode.

#### If You Forget the Password

If you forget the current password, use the **login** command in conjunction with the control panel Halt button to clear the password, as follows:

1. Enter the **login** command:

   ```
   P00>>> login
   ```

2. When prompted for the password, press the Halt button to the latched position and then press the Return (or Enter) key.
3. Press the Halt button to release the halt. The password is now cleared and the console cannot be put into secure mode unless you set a new password.

## 6.7 Configuring Devices

**Become familiar with the configuration requirements for CPUs and memory before removing or replacing those components. See Chapter 8 for removal and replacement procedures.**

**WARNING: To prevent injury, access is limited to persons who have appropriate technical training and experience. Such persons are expected to understand the hazards of working within this equipment and take measures to minimize danger to themselves or others. These measures include:**

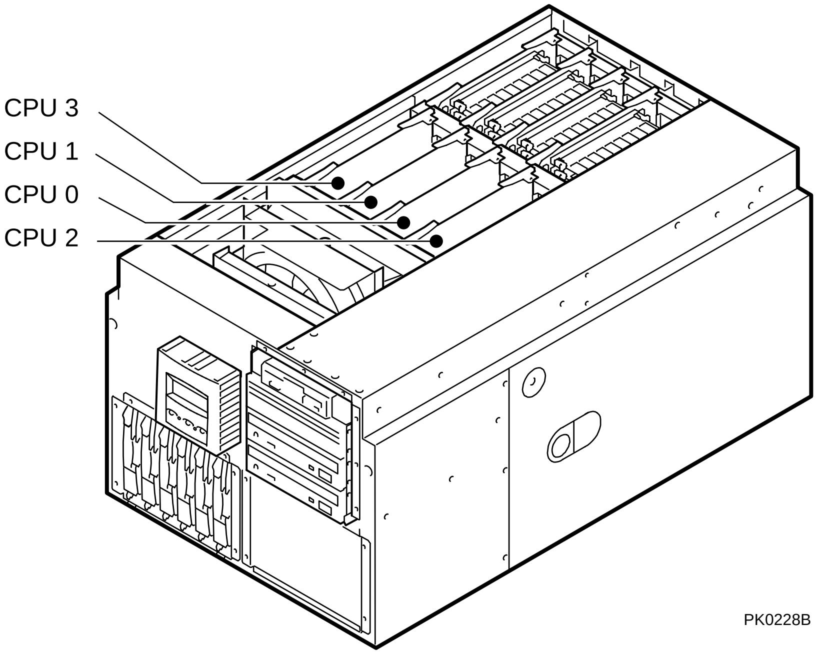
**1. Remove any jewelry that may conduct electricity.**

**2. If accessing the system card cage, power down the system and wait 2 minutes to allow components to cool.**

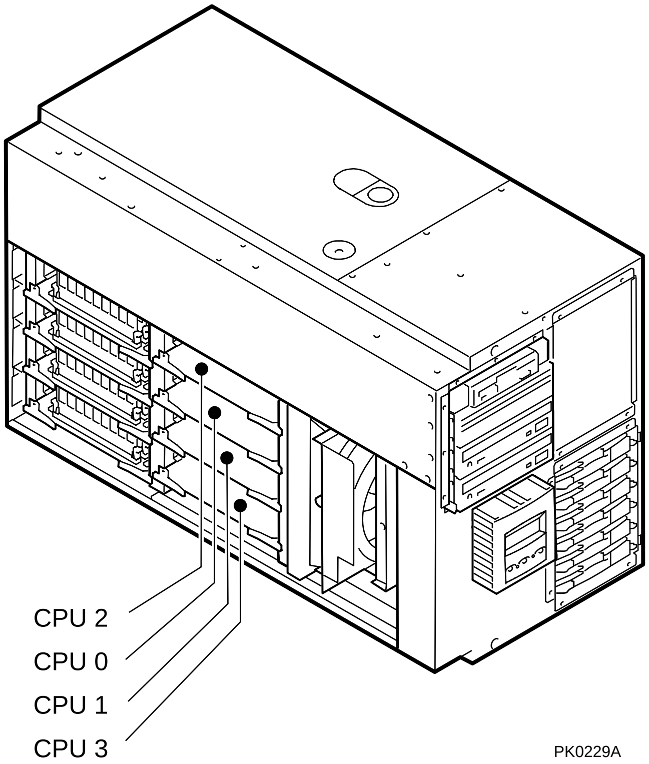
**3. Wear an anti-static wrist strap when handling internal components.**

### 6.7.1 CPU Configuration

**Figure 6–1 CPU Slot Locations (Pedestal/Rack)**



## Figure 6–2 CPU Slot Locations (Tower)



### CPU Configuration Rules

1. A CPU must be installed in slot 0. The system will not power up without a CPU in slot 0.
2. CPU cards must be installed in numerical order, starting at CPU slot 0. See Figure 6–1 and Figure 6–2.
3. CPUs must be identical in speed.

### 6.7.2 Memory Configuration

**Become familiar with the rules for memory configuration before adding DIMMs to the system.**

Refer to Figure 6–4 or Figure 6–5 and observe the following rules for installing DIMMs.

- You can install up to 16 DIMMs or up to 32 DIMMs.
- An option consists of a set of 4 DIMMs. You must install all 4 DIMMs to populate a set.
- Fill sets in numerical order. Populate all 4 slots in Set 0, then populate Set 1, and so on.
- An "array" is one set for systems that support 16 DIMMs and two sets for systems that support 32 DIMMs.
- DIMMs in an array must be the same capacity and type. For example, suppose you have populated Sets 0, 1, 2, and 3. When you populate Set 4, the DIMMs must be the same capacity and type as those installed in Set 0. Similarly, Set 5 must be populated with DIMMs of the same capacity and type as are in Set 1, and so on, as indicated in the following table.

| Array | Systems Supporting 32 DIMMs | Systems Supporting 16 DIMMs |
|---|---|---|
| 0 | Set 0 and Set 4 | Set 0 |
| 1 | Set 1 and Set 5 | Set 1 |
| 2 | Set 2 and Set 6 | Set 2 |
| 3 | Set 3 and Set 7 | Set 3 |

**CAUTION:** *Using different DIMMs may result in loss of data.*

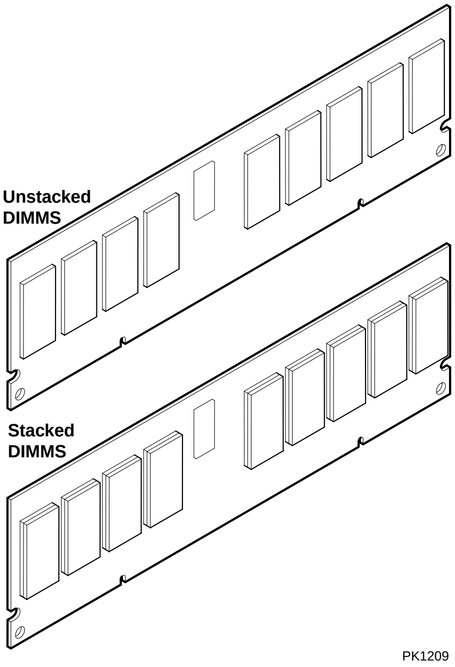
### DIMM Information for Two System Types

DIMMs are manufactured with two types of SRAMs, stacked and unstacked (see Figure 6–3). Stacked DIMMs provide twice the capacity of unstacked DIMMs, and, at the time of shipment, are the highest capacity DIMMs offered by Compaq. The system may have either stacked or unstacked DIMMs.

A memory option consists of a "set" of four DIMMs. The system supports two sets per "array" and four arrays per system. You can mix stacked and unstacked DIMMs within the system, but not within an array. The DIMMs within an array must be of the same capacity and type (stacked or unstacked) because of different memory addressing.

When installing sets 0, 1, 2, and 3, an incorrect mix will not occur. When installing sets 4, 5, 6, or 7, however, you must ensure that the four DIMMs being installed match the capacity and type of DIMMs in the existing array. If necessary, rearrange DIMMs for proper configuration.

### Figure 6–3 Stacked and Unstacked DIMMs



Only the following DIMMs and DIMM options can be used in the ES45 system.

| Density | DIMM | DIMM Option (4 DIMMs per) |
|---|---|---|
| 128 MB | 20-01CBA-09 | MS620-AA (512 MB) |
| 256 MB | 20-01DBA-09 | MS620-BA (1 GB) |
| 512 MB | 20-01EBA-09 | MS620-CA (2 GB) |
| 1 GB | 20-L0FBA-09 | MS620-DA (4 GB)* |

* Toshiba specific DIMM and option.

---

**CAUTION:** *Using different DIMMs may result in loss of data.*

---

### Memory Performance Considerations

Interleaved operations reduce the average latency and increase the memory throughput over non-interleaved operations. With one memory option (4 DIMMs) installed, memory interleaving will not occur. For 2-way interleaving, array 0 & 2 and 1 & 3 must have the same size memory. For 4-way interleaving, array 0 through 3 must have the same size memory.

The output of the **show memory** command provides the memory interleaving status of the system.

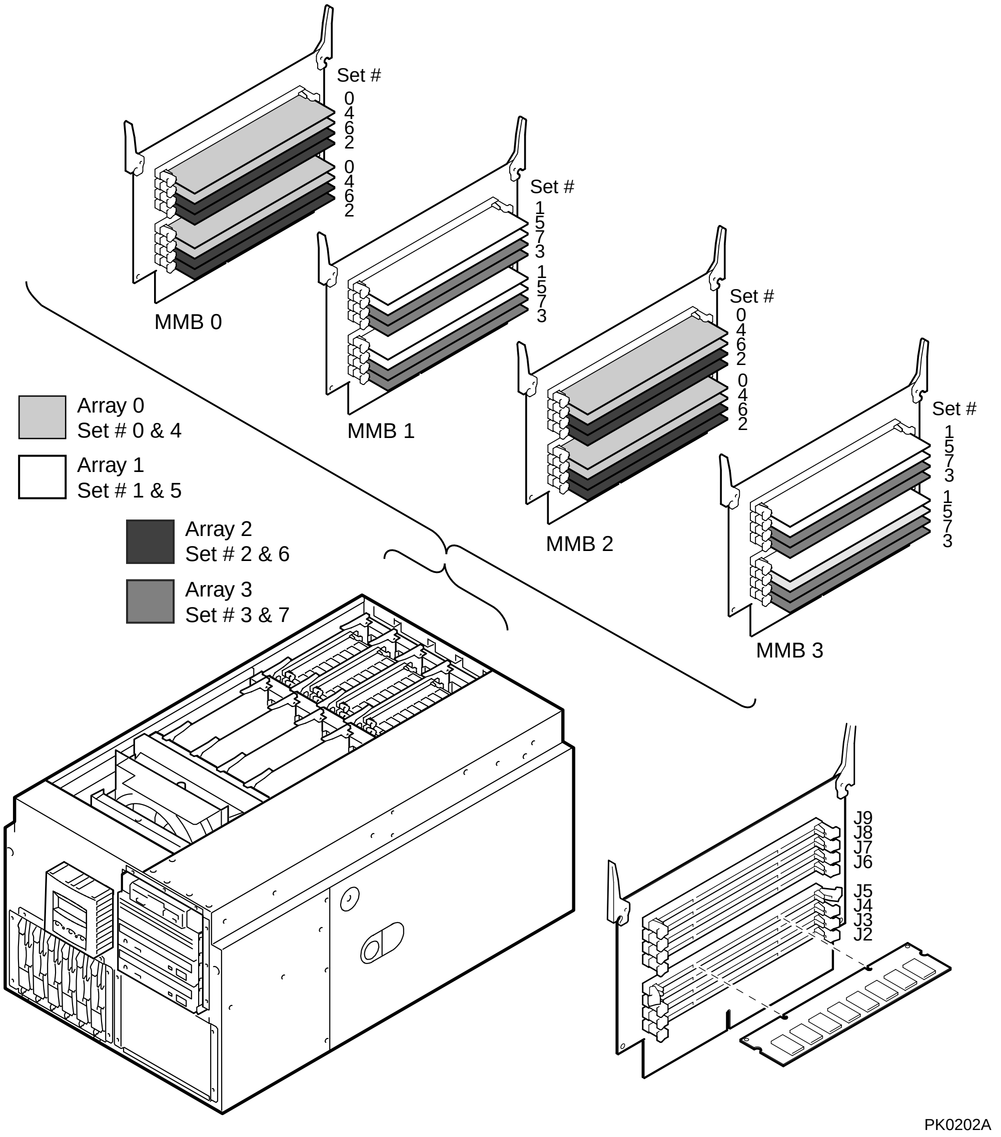
```
P00>>> show memory
 Array       Size       Base Address    Intlv Mode
---------  ----------  ----------------  ----------
   0        4096Mb     0000000000000000    2-Way
   1        1024Mb     0000000200000000    2-Way
   2        4096Mb     0000000100000000    2-Way
   3        1024Mb     0000000240000000    2-Way

       10240 MB of System Memory
```

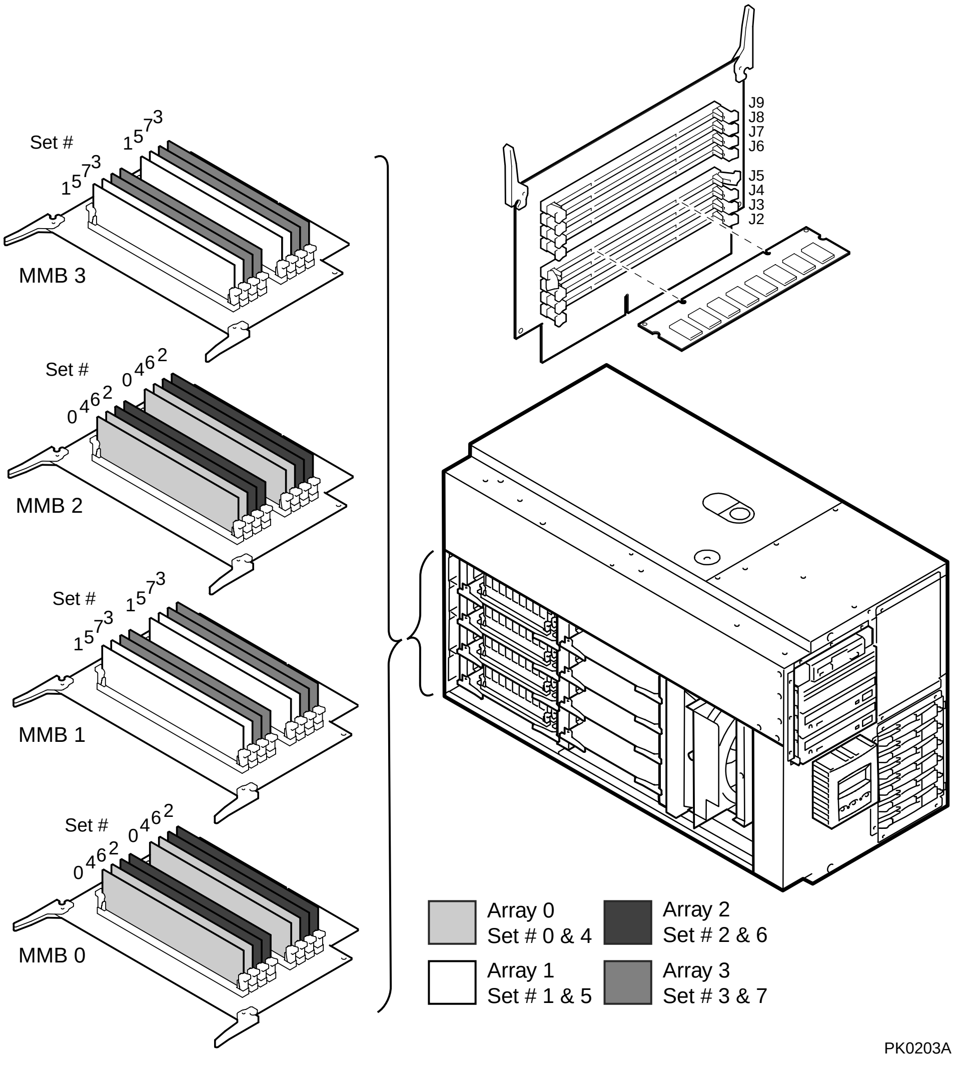
The **show memory** display does not indicate the number of DIMMs or their size. Array 3 could consist of two sets of 128 MB DIMMs (eight DIMMs) or one set of 256 MB DIMMs (four DIMMs). Either combination provides 1024 MB of memory.

For optimum memory utilization and performance, load memory arrays in the following order: 0, 1, 2, 3, 4, 6, 5, and 7. See Figure 6–4 for array locations.

## Figure 6–4 Memory Configuration (Pedestal/Rack)



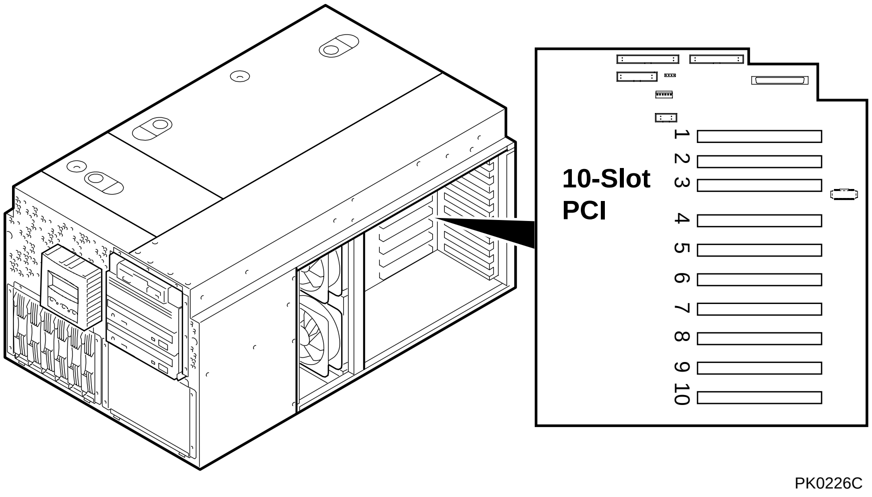
## Figure 6–5 Memory Configuration (Tower)



### 6.7.3 PCI Configuration

PCI modules are either designed for 5.0 volts or 3.3 volt slots, or are universal in design and can plug into either 3.3 or 5.0 volt slots.

**Figure 6–6 PCI Slot Locations (Pedestal/Rack)**



---

**CAUTION:** *Check the keying before you install the PCI module and do not force it in. Plugging a module into a wrong slot can damage it.*

---

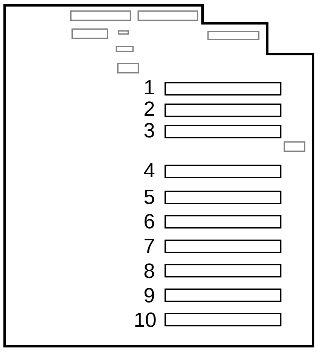
The PCI slots are split across four independent 64-bit PCI buses, three buses at 66 MHz and one bus at 33 MHz. These buses correspond to Hose 0 through Hose 3 in the system logical configuration. The slots on each bus are listed below.

Some PCI options require drivers to be installed and configured. These options come with a floppy or a CD-ROM. Refer to the installation document that came with the option and follow the manufacturer's instructions.

There is no direct correspondence between the physical numbers of the slots on the I/O backplane and the logical slot identification reported with the SRM console **show config** command (described in Chapter 2). The table in Figure 6–7 maps the physical slot numbers to the SRM logical ID numbers for the 10-slot backplane.

## Figure 6–7 PCI Slot Voltages and Hose Numbers

10-Slot PCI I/O Backplane



| | Max<br>Speed | Voltage | Hot-Plug | SRM Console |
|---|---|---|---|---|
| 1 | 66 MHz | 3.3V | No | Hose 2 Slot ID 1 |
| 2 | 66 MHz | 3.3V | No | Hose 2 Slot ID 2 |
| 3 | 33 MHz | 5.0V | No | Hose 0 Slot ID 11 |
| 4 | 66 MHz | 3.3V | Yes | Hose 3 Slot ID 2 |
| 5 | 66 MHz | 3.3V | Yes | Hose 3 Slot ID 1 |
| 6 | 33 MHz | 5.0V | Yes | Hose 0 Slot ID 10 |
| 7 | 66 MHz | 3.3V | Yes | Hose 1 Slot ID 2 |
| 8 | 66 MHz | 3.3V | Yes | Hose 1 Slot ID 1 |
| 9 | 33 MHz | 5.0V | Yes | Hose 0 Slot ID 9 |
| 10 | 33 MHz | 5.0V | Yes | Hose 0 Slot ID 8 |

Quick Reference
SRM Console to
Physical Slot Location

| SRM Console | Physical Slot |
|---|---|
| Hose 0 Slot ID 8 | 10 |
| Slot ID 9 | 9 |
| Slot ID 10 | 6 |
| Slot ID 11 | 3 |
| Hose 1 Slot ID 1 | 8 |
| Slot ID 2 | 7 |
| Hose 2 Slot ID 1 | 1 |
| Slot ID 2 | 2 |
| Hose 3 Slot ID 1 | 5 |
| Slot ID 2 | 4 |

PK0974B

For more information, see http://www.compaq.com/alphaserver/.

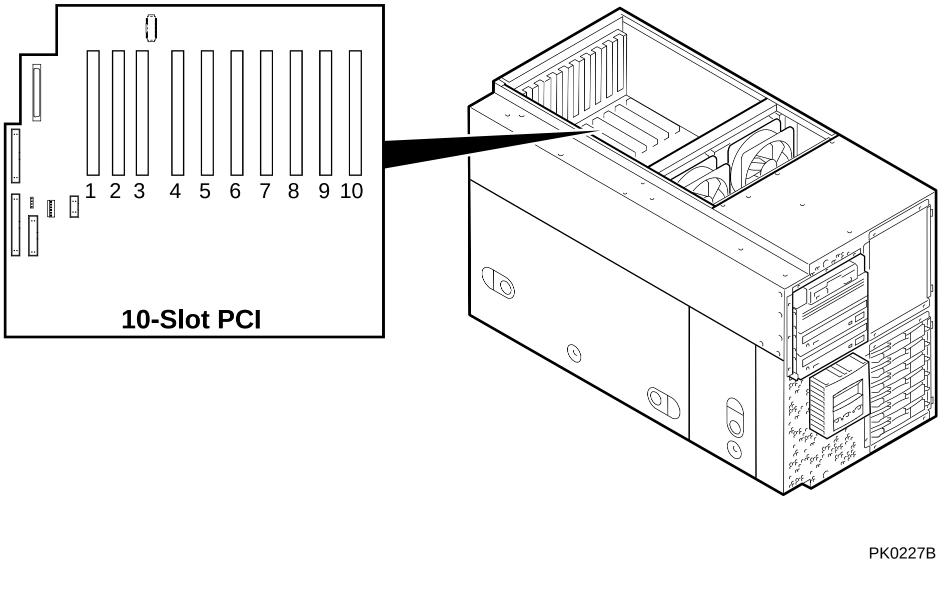
PCI modules are either designed for 5.0 volts or 3.3 volt slots, or are universal in design and can plug into either 3.3 or 5.0 volt slots.

---

**CAUTION:** *Check the keying before you install the PCI module and do not force it in. Plugging a module into a wrong slot can damage it.*

---

### Figure 6–8 PCI Slot Locations (Tower)



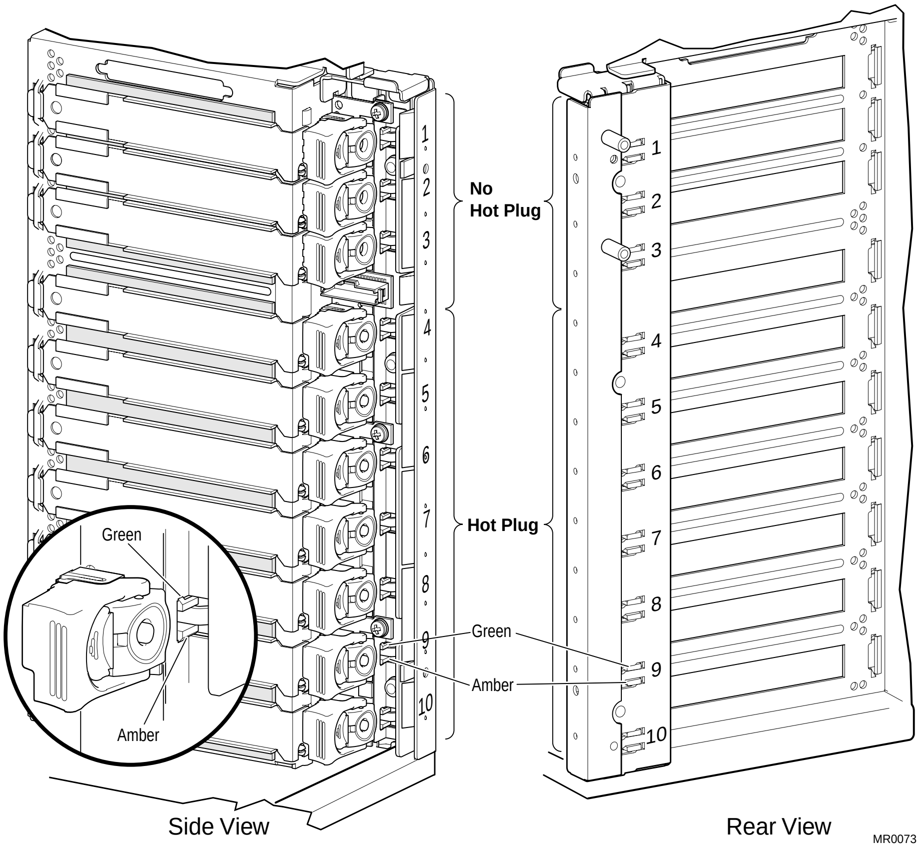
### 6.7.4 PCI Module LEDs

---

**CAUTION:** *Hot plug is not currently supported by the operating systems.*

---

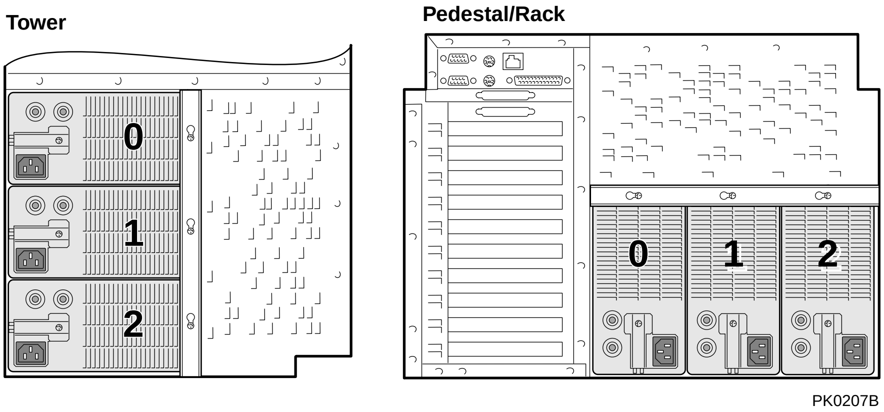
**Figure 6–9 PCI Status LEDs**



| LED | Status |
|---|---|
| Green | Power applied |
| Amber | Power fault |

### 6.7.5 Power Supply Configurations

Figure 6–10 Power Supply Locations



The system can have the following power configurations:

**Two Power supply System (minimum configuration)**

- Two CPUs
- One storage cage
- Four to sixteen DIMMs

**Redundant Power Supply**. If a power supply fails, the redundant supply provides power and the system continues to operate normally. A third power supply adds redundancy for an entry-level system.

**Recommended Installation Order.** Generally, power supply 0 is installed first, power supply 1 second, and power supply 2 third, but the supplies can be installed in any order. See Figure 6–10. The power supply numbering corresponds to the numbering displayed by the SRM **show power** command.

## 6.8 Booting Linux

**Obtain the Linux installation document and install Linux on the system. Then verify the firmware version, boot device, and boot parameters, and issue the boot command.**

The procedure for installing Linux on an Alpha system is described in the Alpha Linux installation document for your Linux distribution. The installation document can be downloaded from the following Web site:

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

You need V5.6-3 or higher of the SRM console to install Linux. If you have a lower version of the firmware, you will need to upgrade. For instructions and the latest firmware images, see the following URL.

http://ftp.digital.com/pub/DEC/Alpha/firmware/

### Linux Boot Procedure

1. Power up the system to the SRM console and enter the **show version** command to verify the firmware version.

```
P00>> show version
version                   V5.6-3 June 15 2001 08:36:11
P00>>
```

2. Enter the **show device** command to determine the unit number of the drive for your boot device, in this case dka0.0.0.17.0.

```
P00>>> sh dev
dka0.0.0.17.0              DKA0              COMPAQ BD018122C9  B016
dka200.2.0.7.1             DKA200            COMPAQ BD018122C9  B016
dqa0.0.0.105.0             DQA0                        CD-224E  9.5B
dva0.0.0.0.0               DVA0
ewa0.0.0.9.0               EWA0              00-00-F8-1B-9C-47
pka0.7.0.7.1               PKA0                  SCSI Bus ID 7
pkb0.7.0.6.0               PKB0                  SCSI Bus ID 7
pkc0.7.0.106.0             PKC0                  SCSI Bus ID 7
P00>>>
```

3. After installing Linux, set **boot** environment variables appropriately for your installation. The typical values indicating booting from dka0 with the first aboot.conf entry are shown in this example.

```
P00>>> set bootdef_dev dka0
P00>>> set boot_file
P00>>> set boot_osflags 0
P00>>> show boot*
boot_dev               dka0.0.0.17.0
boot_file
boot_osflags           0
boot_reset             OFF
bootdef_dev
booted_dev
booted_file
booted_osflags
```

4. From SRM enter the **boot** command. The following example shows abbreviated **boot** output.

### Example 6–5 Linux Boot Output

This example shows messages similar to what you will see when booting Linux. The example is from a RedHat Linux 7.0 boot.

```
>>> boot
(boot dka0.0.0.8.0 -flags 0)
block 0 of dka0.0.0.8.0 is a valid boot block
reading 163 blocks from dka0.0.0.8.0
bootstrap code read in
base = 2d4000, image_start = 0, image_bytes = 14600
initializing HWRPB at 2000
initializing page table at 7fff0000
initializing machine state
setting affinity to the primary CPU
jumping to bootstrap code
aboot: Linux/Alpha SRM bootloader version 0.7
aboot: switching to OSF/1 PALcode version 1.87
aboot: booting from device 'SCSI 0 8 0 0 0 0 0'
aboot: valid disklabel found: 3 partitions.
aboot: loading uncompressed vmlinuz-2.4.3-7privateer2smp...
aboot: loading compressed vmlinuz-2.4.3-7privateer2smp...
aboot: zero-filling 369720 bytes at 0xfffffc0000ce9400
aboot: starting kernel vmlinuz-2.4.3-7privateer2smp with arguments
root=/dev/sda2 console=ttyS0
Linux version 2.4.3-7privateer2smp (root@privateer) (gcc version
2.96 20000731 (Red Hat Linux 7.1 2.96-85)) #1 SMP Thu May 24
11:01:14 EDT 2001
Booting GENERIC on Titan variation Privateer using machine vector
PRIVATEER from SRM
Command line: root=/dev/sda2 console=ttyS0
```

```
memcluster 0, usage 1, start        0, end      362
memcluster 1, usage 0, start      362, end   262135
memcluster 2, usage 1, start   262135, end   262144
freeing pages 362:1024
freeing pages 1700:262135
SMP: 4 CPUs probed -- cpu_present_mask = f
On node 0 totalpages: 262144
zone(0): 262144 pages.
zone(1): 0 pages.
zone(2): 0 pages.
Kernel command line: root=/dev/sda2 console=ttyS0
Using epoch = 1900
Console: colour dummy device 80x25
Calibrating delay loop... 1993.00 BogoMIPS
Memory: 2044536k/2097080k available (2321k kernel code, 41456k
reserved, 2133k data, 432k init)
Dentry-cache hash table entries: 262144 (order: 9, 4194304 bytes)
Buffer-cache hash table entries: 131072 (order: 7, 1048576 bytes)
Page-cache hash table entries: 262144 (order: 9, 4194304 bytes)
Inode-cache hash table entries: 131072 (order: 8, 2097152 bytes)
VFS: Diskquotas version dquot_6.5.0 initialized
POSIX conformance testing by UNIFIX
Using heuristic of 2147483647 cycles.
SMP starting up secondaries.
Calibrating delay loop... 1997.12 BogoMIPS
Calibrating delay loop... 1997.12 BogoMIPS
Calibrating delay loop... 1993.00 BogoMIPS
SMP: Total of 4 processors activated (7987.49 BogoMIPS).
  got res[8000:80ff] for resource 0 of Symbios Logic Inc. (formerly
NCR) 53c895
  got res[8400:843f] for resource 1 of Intel Corporation 82557

       .
       .
       .



autorun ...
... autorun DONE.
NET4: Linux TCP/IP 1.0 for NET4.0
IP Protocols: ICMP, UDP, TCP, IGMP
IP: routing cache hash table of 16384 buckets, 256Kbytes
TCP: Hash tables configured (established 524288 bind 65536)
Linux IP multicast router 0.06 plus PIM-SM
NET4: Unix domain sockets 1.0/SMP for Linux NET4.0.
VFS: Mounted root (ext2 filesystem) readonly.
Freeing unused kernel memory: 432k freed
       .
       .
       .
login:
```

# Chapter 7
# Using the Remote Management Console

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 access to the repository for all error information in the system.

This chapter explains the operation and use of the RMC. Sections are:

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

## 7.1 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). It also provides functionality to read and write configuration and error log information to FRU error log devices.**

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 cover to the system card cage is removed.
    - ➢ 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 the DPR so that this information can be accessed by the system.
- Retrieves information from the DPR and stores it in FRU EEROMs.

The RMC logic is implemented using an 8-bit microprocessor, PIC17C44, as the primary control device. The firmware code is resident within the microprocessor and in flash memory. If the RMC firmware should ever become corrupted or obsolete, you can update it manually using the Loadable Firmware Update Utility. See Chapter 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 supply. You can gain access to the RMC as long as AC power is available to the system (through an AC outlet). Thus, if the system fails, you can still access the RMC and gather error/fault information about the failure.

### DPR Error Repository

The RMC manages an extensive network of FRU $I^2C$ EEPROMs. Information from these EEPROMs is stored in dual-port RAM (DPR)—a shared RAM that facilitates interaction between the RMC and the system—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 DPR. The EEPROM data contains information on configuration and errors. The data is accessible through the TIG chip on the system motherboard.

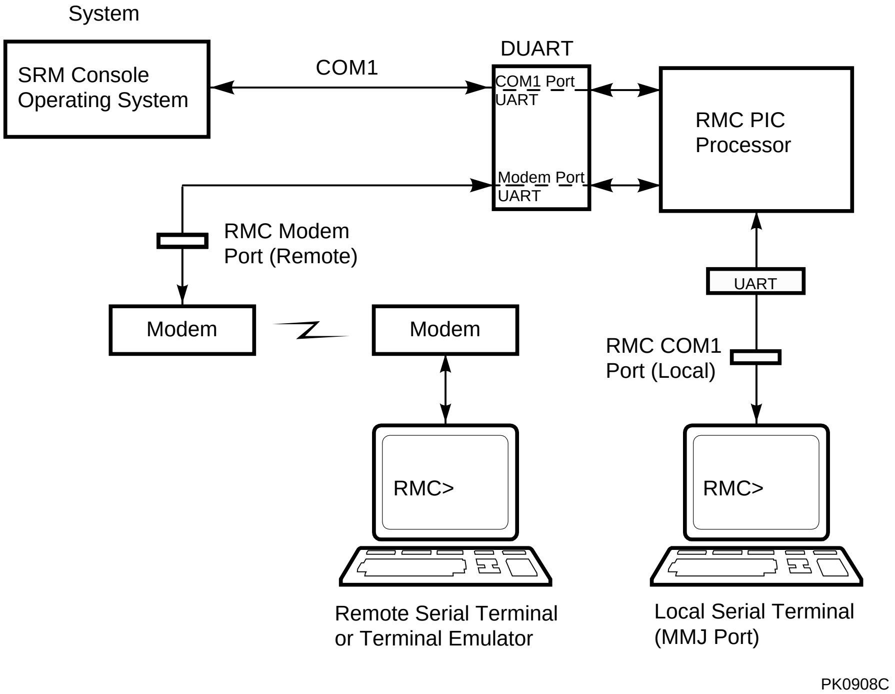
As one of its functions, the TIG provides interfaces for the firmware and the operating system to communicate with the server management logic. The data accessed from DPR provides configuration information to the firmware during start-up. Remote or local applications can read the DPR system error and configuration repository. The error log information is written to the DPR 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 console provides several commands for accessing error information in the DPR. See Section 7.6. Compaq Analyze, described in Chapter 5, can access the FRU EEPROM error logs to provide diagnostic information for system FRUs.

## 7.2 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 7.6.1.**

### Figure 7–1 Data Flow in Through Mode



### 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 connects to the RMC CLI.

Figure 7–1 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 connect to the CLI. Data issued from the CLI is transmitted between the RMC microprocessor and the active port that connects to the RMC CLI.

---

**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.*

---

### Local Mode

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 connect to the RMC CLI from the modem.

### 7.2.1 Bypass Modes

**For modem connection, you can set the operating mode so that data and control signals partially or completely bypass the RMC. The bypass modes are Snoop, Soft Bypass, and Firm Bypass.**

Figure 7–2 Data Flow in Bypass Mode

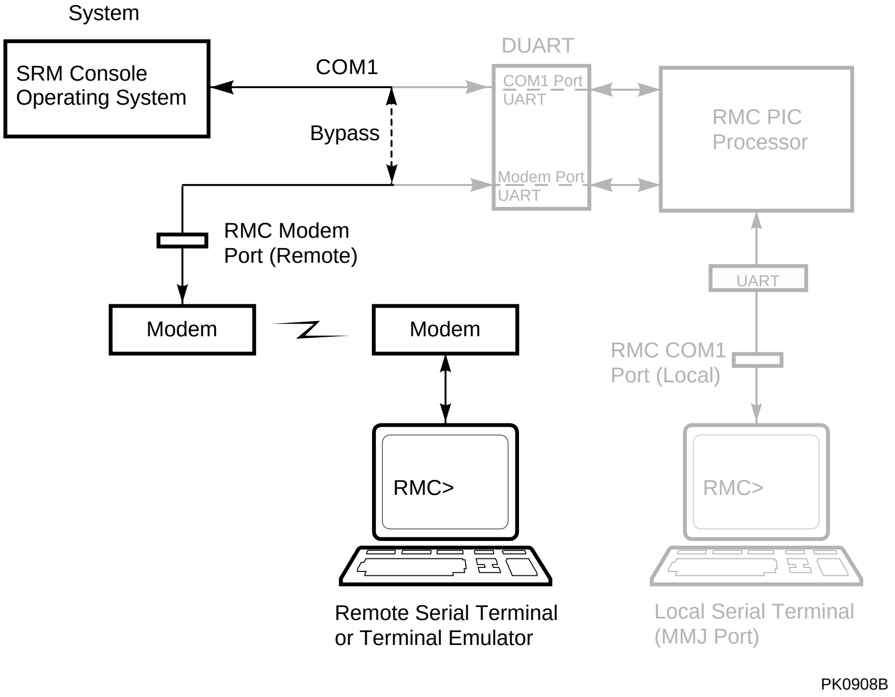


Figure 7–2 shows the data flow in the bypass modes. Note that the internal system COM1 port is connected directly to the modem port.

---

**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 connect to the RMC CLI to switch the COM1 mode if necessary.

### 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 connects to 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

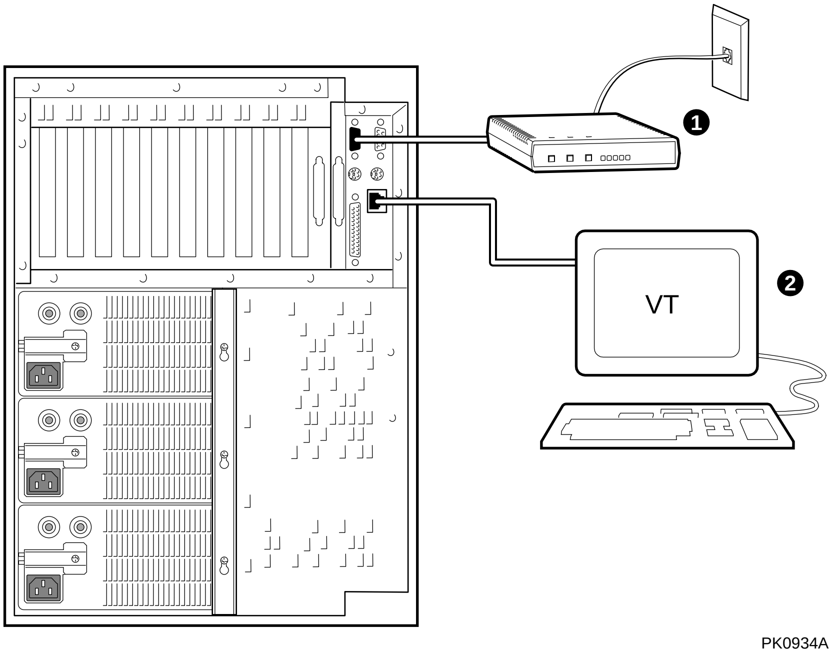
In Firm Bypass mode all data and control signals are routed directly between the system COM1 port and the external modem port. The RMC does not configure or monitor the modem. Firm Bypass mode is useful if you want the system, not the RMC, to fully control the modem port and you want to disable RMC remote 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.

## 7.3 Terminal Setup

**You can use the RMC from a modem hookup or the serial terminal connected to the system. As shown in Figure 7–3, a modem is connected to the dedicated 9-pin modem port ❶ and a terminal is connected to the COM1 serial port/terminal port (MMJ) ❷.**

### Figure 7–3 Terminal Setup for RMC (Tower View)



## 7.4 Connecting to the RMC CLI

**You type an escape sequence to connect to the RMC CLI. You can connect to the CLI from any of the following: a modem, the local serial console terminal, the local VGA monitor, or the system. The "system" includes the operating system, SRM, or an application.**

- You can connect to the RMC CLI from the local terminal regardless of the current operating mode.
- You can connect to the RMC CLI 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 CLI session can be active at a time.*

### Connecting from a Serial Terminal

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

```
^[^[ 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 invoked the RMC CLI. In the following example, the **quit** command returns you to the system COM1 port.

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

### Connecting from the Local VGA Monitor

To connect to the RMC CLI from the local VGA monitor, the **console** environment variable must be set to **graphics** and the SRM console must be running.

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 you enter the escape sequence, the system connects to the CLI and the RMC> prompt is displayed.

When the RMC CLI session is completed, reset the system with the Reset button on the operator control panel or issue the RMC **reset** command.

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

## 7.5 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.

## 7.6 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}**
**dep**
**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**

The commands for setting up and using the RMC are described in the following sections. The **dep** command is reserved. For an RMC commands reference, see the *Owner's 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 ***
  ```

- Use the Backspace key to erase input.

### 7.6.1 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 7–1 set com1_mode

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

### 7.6.2 Displaying the System Status

**The RMC status command displays the current RMC settings. Table 7–1 explains the status fields.**

#### Example 7–2 status

```
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
RMC 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: CPU door opened
Logout Timer: 20 minutes
User String:
```

### Table 7–1 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.<br>OFF = System is off. |
| System Halt: | Asserted = System has been halted.<br>Deasserted = Halt has been released. |
| RMC Power Control: | ON= System has powered on from RMC.<br>OFF = System has powered off from RMC. |
| Escape Sequence: | Current escape sequence for access to RMC console. |
| Remote Access: | Enabled = Modem for remote access is enabled.<br>Disabled = Modem for remote access is disabled. |
| RMC Password: | Set = Password set for modem access.<br>Not set = No password set for modem access. |
| Alert Enable: | Enabled = Dial-out enabled for sending alerts.<br>Disabled = Dial-out disabled for sending alerts. |
| Alert Pending: | YES = Alert has been triggered.<br>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, power supply 1 failed). |
| 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. |

### 7.6.3 Displaying the System Environment

**The RMC env command provides a snapshot of the system environment.**

#### Example 7–3 env

```
RMC> env
       System Hardware Monitor

Temperature (warnings at 48.00C, power-off at 53.00C)               ❶

    CPU0: 27.00C    CPU1: 28.00C    CPU2: 27.00C    CPU3: 28.00C
    Zone0: 26.00C    Zone1: 28.00C    Zone2: 26.00C
Fan RPM                                                             ❷

    Fan1: 2149    Fan2: 2177    Fan3: 2136                          ❸
    Fan4: 2163    Fan5: OFF     Fan6: 2033
Power Supply(OK, FAIL, OFF, '----' means not present)               ❹


    PS0 : OK      PS1 : OK      PS2 : OK
    CPU0: OK      CPU1: OK      CPU2: OK      CPU3: OK
CPU CORE voltage
    CPU0: +1.640V    CPU1: +1.640V    CPU2: +1.640V    CPU3: +1.630V ❺

CPU IO voltage
    CPU0: +1.640V    CPU1: +1.640V    CPU2: +1.640V    CPU3: +1.630V
CPU CACHE voltage
    CPU0: +2.444V    CPU1: +2.405V    CPU2: +2.431V    CPU3: +2.418V
Bulk voltage
    +3.3V Bulk: +3.213V    +5V Bulk: +4.888V   +12V Bulk: +11.907V ❻
         Vterm: +1.580V       Cterm: +1.580V   -12V Bulk: -11.466V
    +2.5V Bulk: +2.457V  +1.5V Bulk: ----

RMC>
```

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

❷ Temperature of PCI backplane: Zone 0 includes PCI slots 1–3, Zone 1 includes PCI slots 7–10, and Zone 2 includes PCI slots 4–6.

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

❹ The normal power supply status 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 core voltage for all CPUs should be the same, and the I/O voltage for all CPUs should be the same.

❻ Bulk power supply voltage. The Vterm and Cterm voltage regulators are located on the system motherboard.

## 7.6.4 Dumping DPR Data

**The dump command dumps unformatted data from DPR locations 0–3FFF hex. The information might be useful for system troubleshooting. Use the DPR address table in Appendix C to analyze the data.**

### Example 7–4 dump

```
RMC> dump
Address: 10          ❶
Count: ee            ❷
   ❸
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>
```

❶ DPR address

❷ Number of bytes dumped (in hex). In the example the **dump** command dumps EF bytes from address 10.

❸ Bytes 10:15 are the time stamp. See Appendix C for the meaning of other locations.

The **dump** command allows you to dump data from the DPR. You can use this command locally or remotely if you are not able to access the SRM console because of a system crash.

The **dump** command accepts two arguments:

**Address:** Prompts for the starting address

**Count:** Prompts for the number of following consecutive bytes. If no count is specified, the count defaults to 0.

## 7.6.5 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.

### Example 7–5 power on/off

```
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.

### Example 7–6 halt in/out

```
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.

### Example 7–7 reset

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

### 7.6.6 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. Connect to the RMC CLI from either the local serial terminal or the local VGA monitor to set up the parameters.

#### Example 7–8 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
.
.
.
```

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>
```

1. At the RMC> prompt, enter commands to monitor and control the remote system.
2. When you have finished a modem session, enter the **hangup** command to cleanly terminate the session and disconnect from the server.

## 7.6.7 Configuring Dial-Out Alert

**When you are not monitoring the system from a modem connection, you can use the RMC dial-out alert feature to remain informed of system status. If 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 7.6.6.

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

### Example 7–9 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 connect to the RMC CLI, 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 7–2. 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 7–2 Elements of Dial String and Alert String

| Dial String | |
|---|---|
| | 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) |
| 9, | The number for an outside line (in this example, 9). Enter the number for an outside line if your system requires it. |
| | , = Pause for 2 seconds. |
| 15085553333 | Phone number of the paging service. |
| **Alert String** | |
| ,,,,,, | Each comma (,) provides a 2-second delay. In this example, a delay of 12 seconds is set to allow the paging service to answer. |
| 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 entire string. |

### 7.6.8 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.

#### Example 7–10 set escape

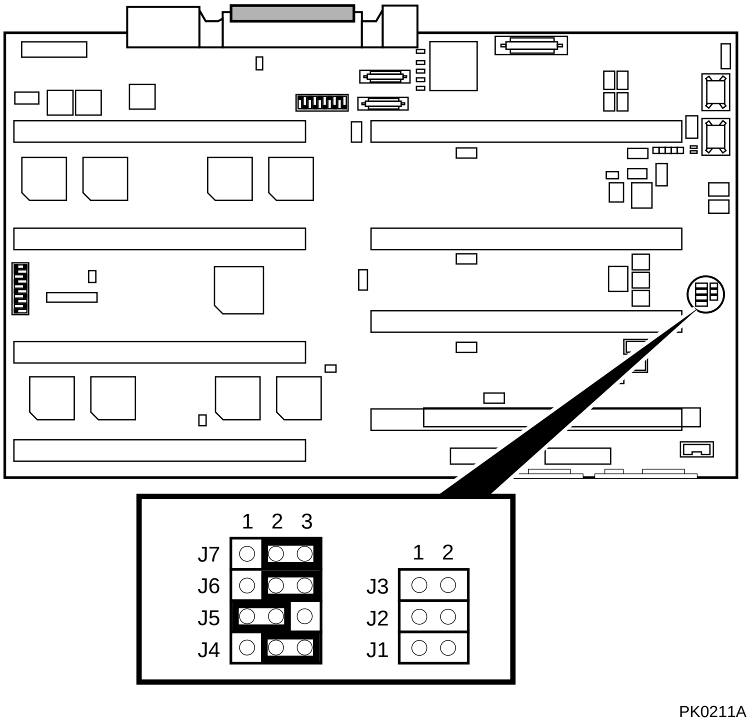
```
RMC> set escape
Escape Sequence: un
RMC> status
.
.
.
Escape Sequence: ^[^[FUN
```

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

## 7.7 Resetting the RMC to Factory Defaults

**If the non-default RMC escape sequence has been lost or forgotten, RMC must be reset to factory settings to restore the default escape sequence.**

**Figure 7–4 RMC Jumpers (Default Positions)**



**NOTE:** *J1, J2, and J3 are reserved.*

The following procedure restores the default settings:

1. Shut down the operating system and press the Power button on the operator control panel to the OFF position.
2. Unplug the power cord from each power supply. Wait until the +5V Aux LEDs on the power supplies go off before proceeding.
3. Remove enclosure panels as described in Chapter 8.
4. Remove the system card cage cover and fan cover from the system chassis, as described in Chapter 8.
5. Remove CPU 1 as described in Chapter 8.
6. On the system motherboard, install jumper J6 over pins 1 and 2. See Figure 7–4. (The default jumper positions are shown.)
7. Plug a power cord into one power supply and wait for the control panel to display the message “System is down.”
8. Unplug the power cord. Wait until the +5V Aux LED on the power supply goes off before proceeding.
9. Move J6 from pins 1 and 2 and install it over pins 2 and 3.
10. Reinstall CPU 1, the card cage cover and fan cover, and the enclosure panels.
11. Plug the power cord into each of 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 7.6.6.*

## 7.8 Troubleshooting Tips

**Table 7–3 lists possible causes and suggested solutions for symptoms you might see.**

Table 7–3 RMC Troubleshooting

| Symptom | Possible Cause | Suggested Solution |
|---|---|---|
| You cannot connect to the RMC CLI 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 terminal cannot communicate with the RMC correctly. | System and terminal baud rates do not match. | Set the baud rate for the terminal to be the same as for the system. 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. |

*Continued on next page*

Table 7–3 RMC Troubleshooting (Continued)

| Symptom | Possible Cause | Suggested Solution |
|---|---|---|
| RMC will not answer when modem is called. (continued from previous page) | On AC 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, you see 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 you enter 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. |

# Chapter 8
# FRU Removal and Replacement

This chapter describes the procedures for removing and replacing FRUs on ES45 systems.

Unless otherwise specified, install a FRU by reversing the steps shown in the removal procedures.

⚠ **WARNING: To prevent injury, access is limited to persons who have appropriate technical training and experience. Such persons are expected to understand the hazards of working within this equipment and take measures to minimize danger to themselves or others. These measures include:**
**1. Remove any jewelry that may conduct electricity.**
**2. If accessing the system card cage, power down the system and wait 2 minutes to allow components to cool.**
**3. Wear an anti-static wrist strap when handling internal components.**

**NOTE:** *If you are installing or replacing CPU cards, memory DIMMs, or PCI cards, become familiar with the location of the card slots and configuration rules. See Chapter 6.*

**CAUTION:** *Static electricity can damage integrated circuits. Always use a grounded wrist strap (29-26246) and grounded work surface when working with internal parts of a computer system.*

*Remove jewelry before working on internal parts of the system.*

**IMPORTANT! After you have replaced FRUs and determined that the system has been restored to its normal operating condition, you must clear the system error information repository (error information logged to the DPR). Use the clear_error all command to clear all errors logged in the FRU EEPROMs and to initialize the central error repository. See Section 4.4 for details on clear_error.**

## 8.1 FRUs

**Table 8–1 lists the FRUs by part number and description. Figure 8–1 shows the location of FRUs in the pedestal/rack systems, and Figure 8–2 shows the location of FRUs in the tower system.**

### Table 8–1 FRU List

| Part # | Description | |
|---|---|---|
| **Cables** | | |
| 17-04787-01 | Power and signal harness assembly | |
| 17-04785-01 | Fan harness assembly | |
| 17-04786-01 | Sensor cable harness assembly | |
| 17-03971-07 | OCP cable assembly | |
| 17-04867-01 | 68-conductor SCSI cable (six drive cage) | |
| 17-04009-02 | 68-pin to 50-pin adapter cable (SCSI removable media) | |
| 17-03970-04 | Floppy cable assembly | |
| 17-04400-06 | Junk I/O connector cable | |
| 17-04705-03 | SCSI removable media device to PCI card SCSI controller | |
| 17-03971-11 | 10-pin storage subsystem management cable (30 inch) | |
| 17-05042-01 | PCI hot swap module to PCI backplane | |
| 17-05021-01 | IDE cable (CD ROM) | |
| **Fans** | | |
| 70-40074-01 | Fan assembly, 172 MM | Fan 6 |
| 70-40073-01 | Fan assembly, 120 MM | Fans 1 and 2 |
| 70-40073-02 | Fan assembly, 120 MM | Fan 5 |
| 70-40072-01 | Fan assembly, 120 MM | Fan 3 |
| 70-40071-01 | Fan assembly, 120 MM | Fan 4 |

Table 8–1 FRU List (Continued)

| Part # | Description | |
|---|---|---|
| **Fans** | | |
| 70-40074-01 | Fan assembly, 172 MM | Fan 6 |
| 70-40073-01 | Fan assembly, 120 MM | Fans 1 and 2 |
| 70-40073-02 | Fan assembly, 120 MM | Fan 5 |
| 70-40072-01 | Fan assembly, 120 MM | Fan 3 |
| 70-40071-01 | Fan assembly, 120 MM | Fan 4 |
| **CPU Module** | | |
| 54-30466-03 | EV68 CB LGA CPU 1 GHz with 8 MB L2 Cache | |
| **Memory DIMMs** | | |
| 20-01CBA-09 | 128 MB Mono 200 pin Sync DIMM 133 MHz | |
| 20-01DBA-09 | 256 MB Mono 200 pin Sync DIMM 133 MHz | |
| 20-01EBA-09 | 512 MB Mono 200 pin Sync DIMM 133 MHz | |
| 20-L0FBA-09 | 1 GB Stacked 200 pin Sync DIMM 133 MHz | |

### Table 8–1 FRU List (Continued)

| Part # | Description |
|---|---|
| **Other Modules and Components** | |
| 70-33894-02 | OCP |
| 54-30414-02 | PCI Hot swap module |
| 54-30348-02 | 8-slot MMB for 200-pin DIMMs |
| 54-30348-03 | 4-slot MMB for 200-pin DIMMs |
| 70-31349-01 | Speaker assembly |
| 30-50802-01 | Hard drive cage assembly, 6 slot, 1-in. universal drives |
| 54-30292-02 | System motherboard |
| 54-25575-02 | I/O connector module |
| 54-30418-01 | PCI backplane, 10-slot |
| 54-30414-02 | Switch / LED HP PCI lever (PCI hot swap module) |
| 3R-A1629-AA | SCSI environmental module (NILE) |
| 30-49448-01 | Power supply, 720 Watts |
| SN-LKQ46-Ax | Keyboard, OpenVMS |
| SN-LKQ47-Ax | Keyboard, Tru64 UNIX |
| SN-PBQWS-WA | Mouse, 3-button |
| 12-37977-02 | Key for doors |
| 3R-A2503-AA | CD-ROM drive, 40x half-height |
| 3R-A2753-AA | Floppy drive |

### 8.1.1 Power Cords

**Tower enclosures ordered in North America include a 220 V power cord. Non-North American orders require one country-specific power cord. Pedestal systems ordered in North American include two 220 V power cords. Non-North American orders require two country-specific power cords.**

Table 8–2 lists the country-specific power cords for tower and pedestal systems.

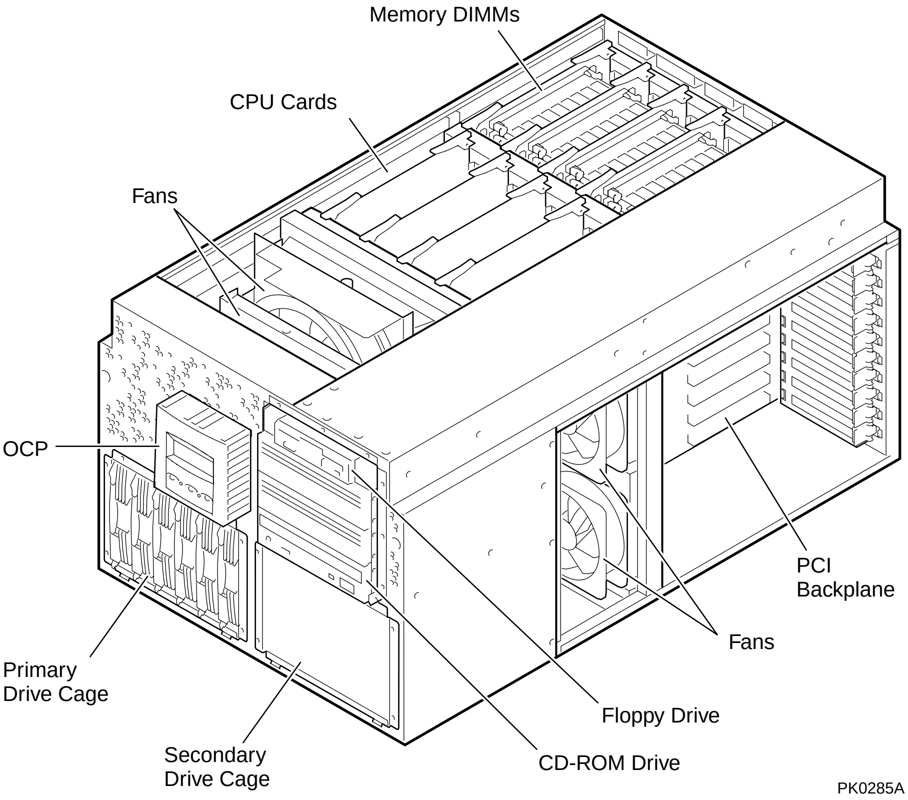
**Table 8–2 Country-Specific Power Cords**

| Power Cord | Country | Length |
|---|---|---|
| BN26J-1K | North American 220 V | 75 in. |
| 3X-BN46F-02 | Japan | 2.5 m |
| BN19H-2E | Australia, New Zealand | 2.5 m |
| BN19C-2E | Central Europe | 2.5 m |
| BN19A-2E | UK, Ireland | 2.5 m |
| BN19E-2E | Switzerland | 2.5 m |
| BN19K-2E | Denmark | 2.5 m |
| BN19M-2E | Italy | 2.5 m |
| BN19S-2E | Egypt, India, South Africa | 2.5 m |

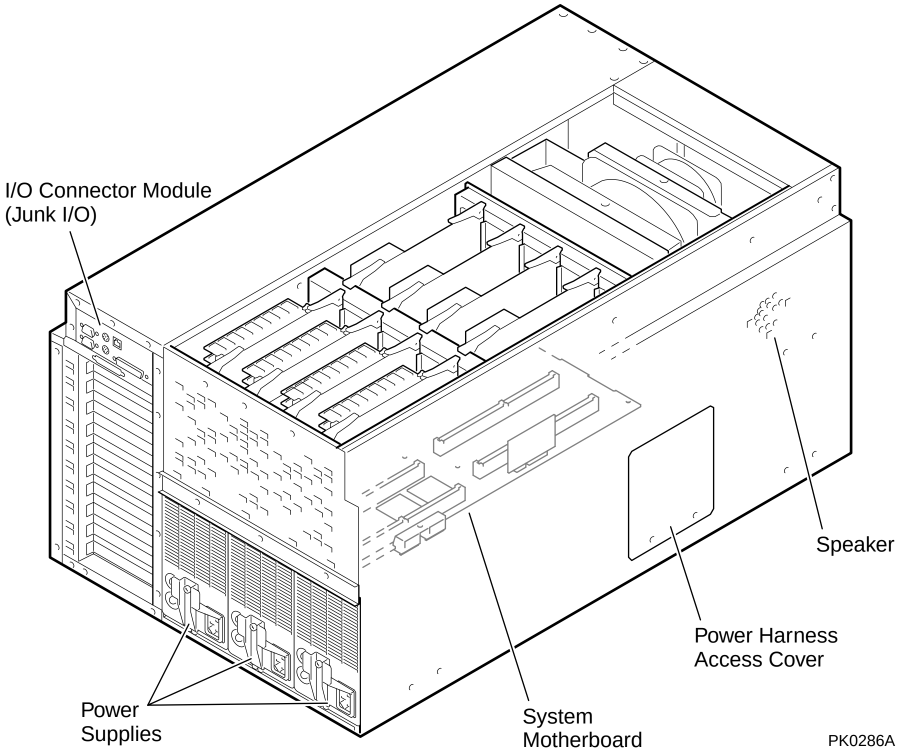
## 8.1.2 FRU Locations

**Figure 8–1 and Figure 8–2 show the location of FRUs in the pedestal and rackmount configurations.**

### Figure 8–1 FRUs — Front/Top (Pedestal/Rack View)



### Figure 8–2 FRUs — Rear (Pedestal/Rack View)



### 8.1.3 Important Information Before Replacing FRUs

**The system must be shut down before you replace most FRUs. The exceptions are power supplies, individual fans, universal hard drives, and PCI cards in slots 4 – 10 (when the operating system supports this function). After replacing FRUs you must clear the system error information repository with the SRM clear_error all command.**

#### Tools

You need the following tools to remove or replace FRUs.

- Phillips #1 (10-inches) and #2 screwdrivers (magnetic screwdrivers are recommended)
- Allen wrench (3 mm)
- Anti-static wrist strap

#### Hot-Plug FRUs

The following are hot-plug FRUs. You can replace them while the system is operating.

- Power supplies
- Individual fans

### Before Replacing Non Hot-Plug FRUs

Follow the procedure below before replacing non hot-plug FRUs. For universal disk drives, you must shut down the operating system, but you do not need to turn off system power.

1. Shut down the operating system.
2. Shut down power to external options, where appropriate.
3. Turn off power to the system.
4. Unplug the power cord from each power supply.

**WARNING: To prevent injury, unplug the power cord from each power supply before installing components.**

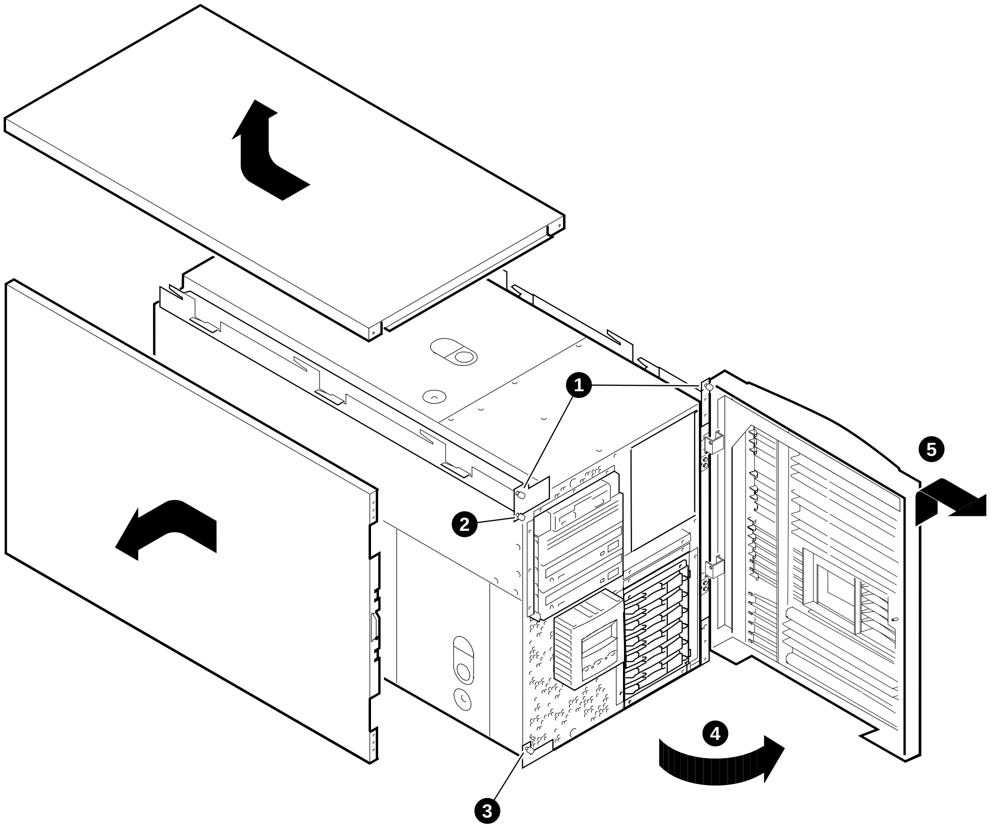
### After Replacing FRUs

**IMPORTANT!** After you have replaced FRUs and have determined that the system has been restored to its normal operating condition, you must clear the system error information repository (error information logged to the DPR).

Use the **clear_error all** command to clear all errors and initialize the central error repository. See Section 4.4 for details.

## 8.2 Removing Enclosure Panels

### Figure 8–3 Enclosure Panel Removal (Tower)

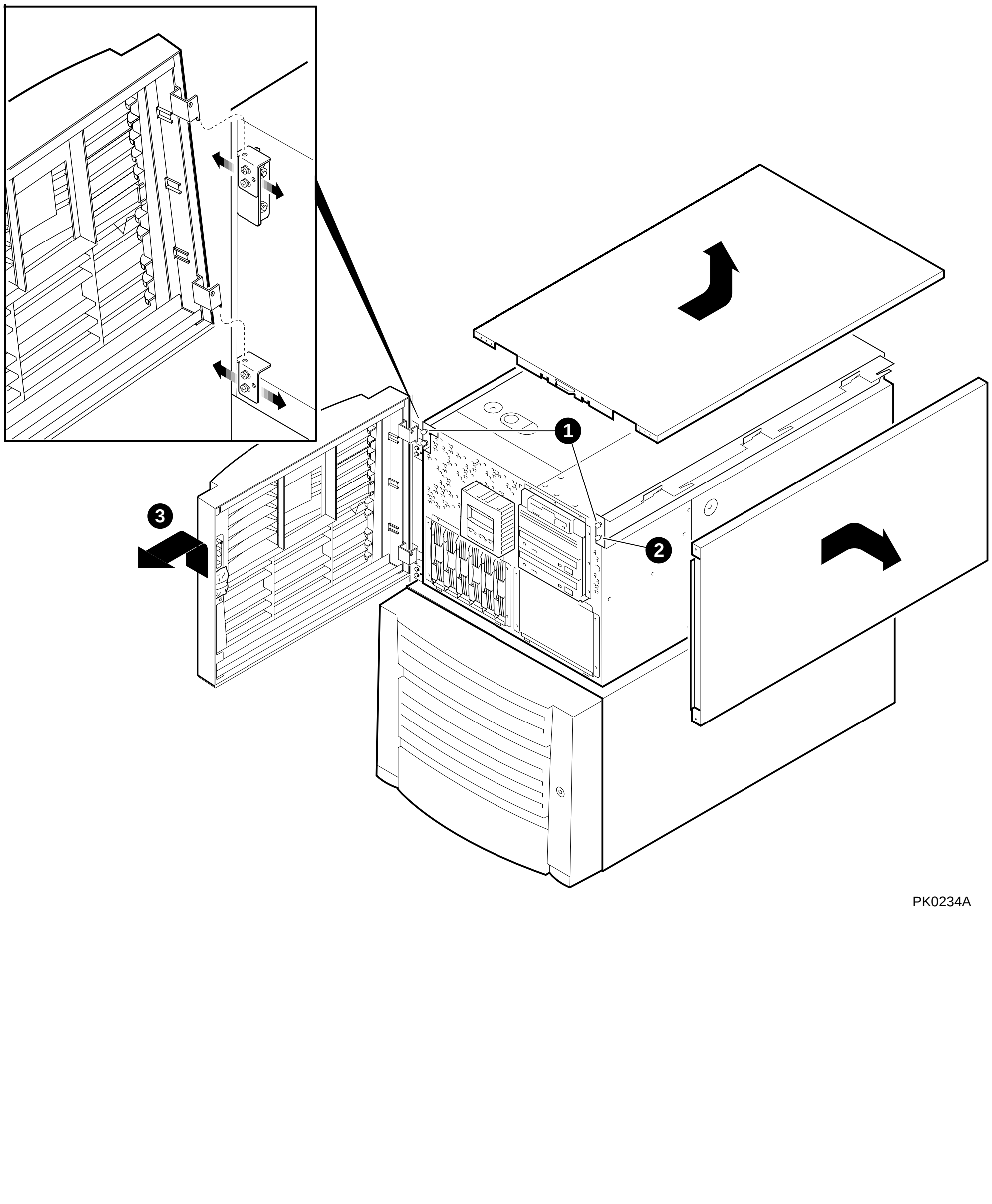


### To Remove Enclosure Panels from a Tower

The enclosure panels are secured by captive screws.

1. From the open position ➍, lift up and away to remove the front door ➎.
2. To remove the top panel, loosen the top left and top right screws ➊. Slide the top panel back and lift it off the system.
3. To remove the left panel, loosen the screw ➋ at the top and the screw ➌ at the bottom. Slide the panel back and then tip it outward. Lift it off the system.

**Figure 8–4 Enclosure Panel Removal (Pedestal)**



### To Remove Enclosure Panels from a Pedestal

The enclosure panels are secured by captive screws.

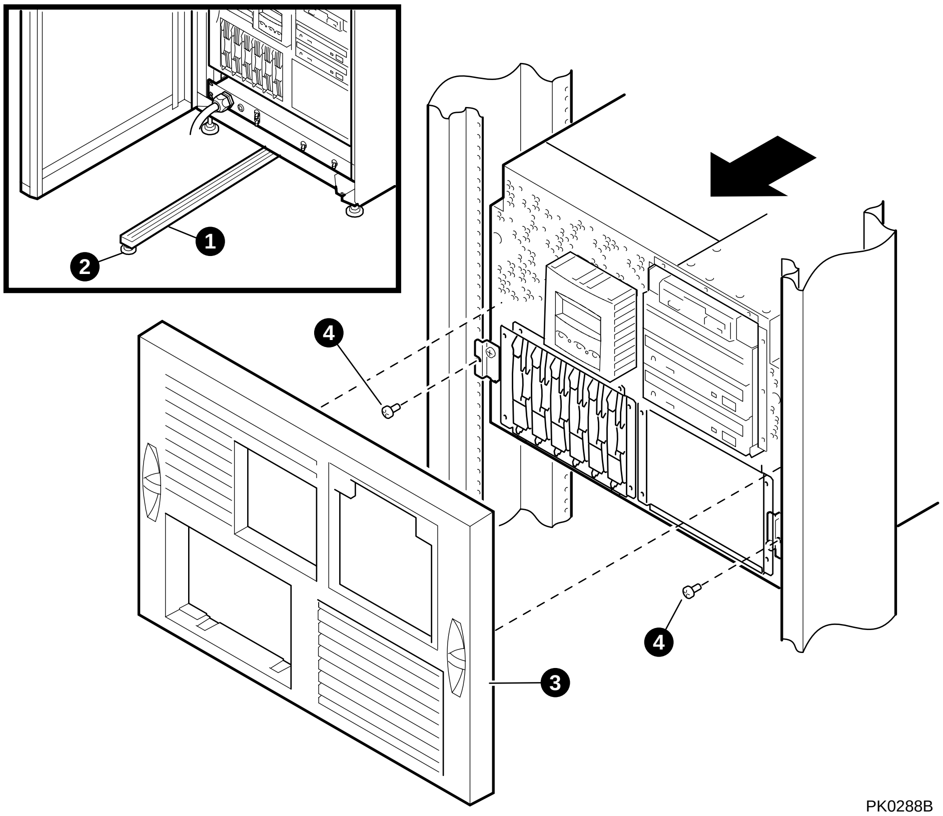
1. From the open position, lift up and away ❸ to remove the front door (the bottom door is removed in the same way).
2. Remove the top enclosure panel by loosening the captive screws shown in ❶. Slide the top panel back and lift it off the system.
3. To remove the right enclosure panel, loosen the captive screw shown in ❷. Slide the panel back and then tip it outward. Lift the panel from the three tabs.

## 8.3 Accessing the System Chassis in a Cabinet

**In a rackmount system, the system chassis is mounted to slides.**

⚠ **WARNING: Pull out the stabilizer bar and extend the leveler foot to the floor before you pull out the system. This precaution prevents the cabinet from tipping over.**

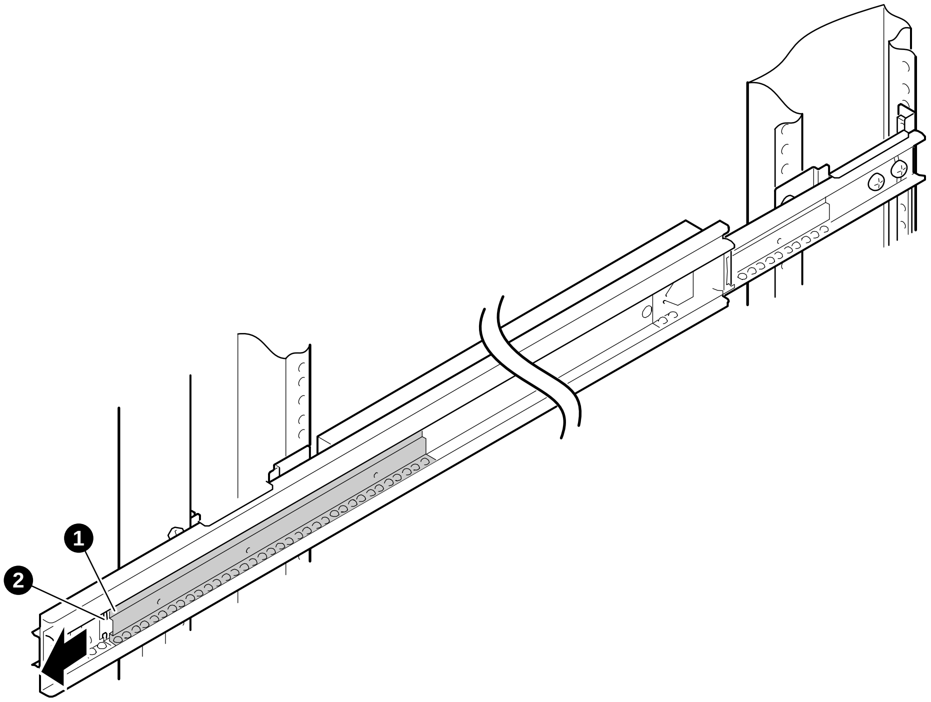
Figure 8–5 Accessing the Chassis in a Cab



**WARNING:**
**1. Make sure that all other hardware in the cabinet is pushed in and attached.**
**2. The system is very heavy. Do not attempt to lift it manually. Use a material lift or other mechanical device.**
**3. The inner race must be moved forward prior to installing the system. Failure to do so may cause bodily harm.**

1. Move the inner race ❶ all the way forward so that it is touching the tabs ❷ on both rails as shown in Figure 8–6.

### Figure 8–6 Moving the Inner Race Forward



## 8.4 Removing Covers from the System Chassis

**The system chassis has three covers: the fan cover, the system card cage cover, and the PCI card cage cover. Remove a cover by loosening the quarter-turn captive screw, pulling up on the ring, and sliding the cover from the system chassis.**

⚠ === V @ >240VA

**WARNING: High current area. Currents exceeding 240 VA can cause burns or eye injury. Avoid contact with parts or remove power prior to access.**

⚠

**WARNING: Contact with moving fan can cause severe injury to fingers. Avoid contact or remove power prior to access.**

### To Gain Access to the System Chassis

1. Open the front door of the cabinet.
2. Pull out the stabilizer bar ❶ at the bottom of the cabinet until it stops.
3. Extend the leveler foot ❷ at the end of the stabilizer bar to the floor.
4. Snap out the front bezel ❸.
5. Remove and set aside the two screws ❹ (one per side), if present, that secure the system to the cabinet.
6. Pull the system out until it locks.

Figure 8–7 and Figure 8–8 show the location and removal of covers on the tower and pedestal/rackmount systems, respectively. The numbers in the illustrations correspond to the following:

❶ 3mm Allen captive quarter-turn screw that secures each cover.

❷ Spring-loaded ring that releases cover. Each cover has a ring.

❸ Fan area cover. This area contains the 6.75-in main system fan and a redundant fan.

❹ System card cage cover. This area contains CPUs, memory DIMMs, MMBs, and system motherboard. To remove the system card cage cover, you must first remove the fan area cover ❸. An interlock switch shuts the system down when you remove the system card cage cover.

❺ PCI card cage cover. This area contains PCI cards, the PCI backplane, I/O connector assembly and four fans.

## Figure 8–7 Covers on the System Chassis (Tower)

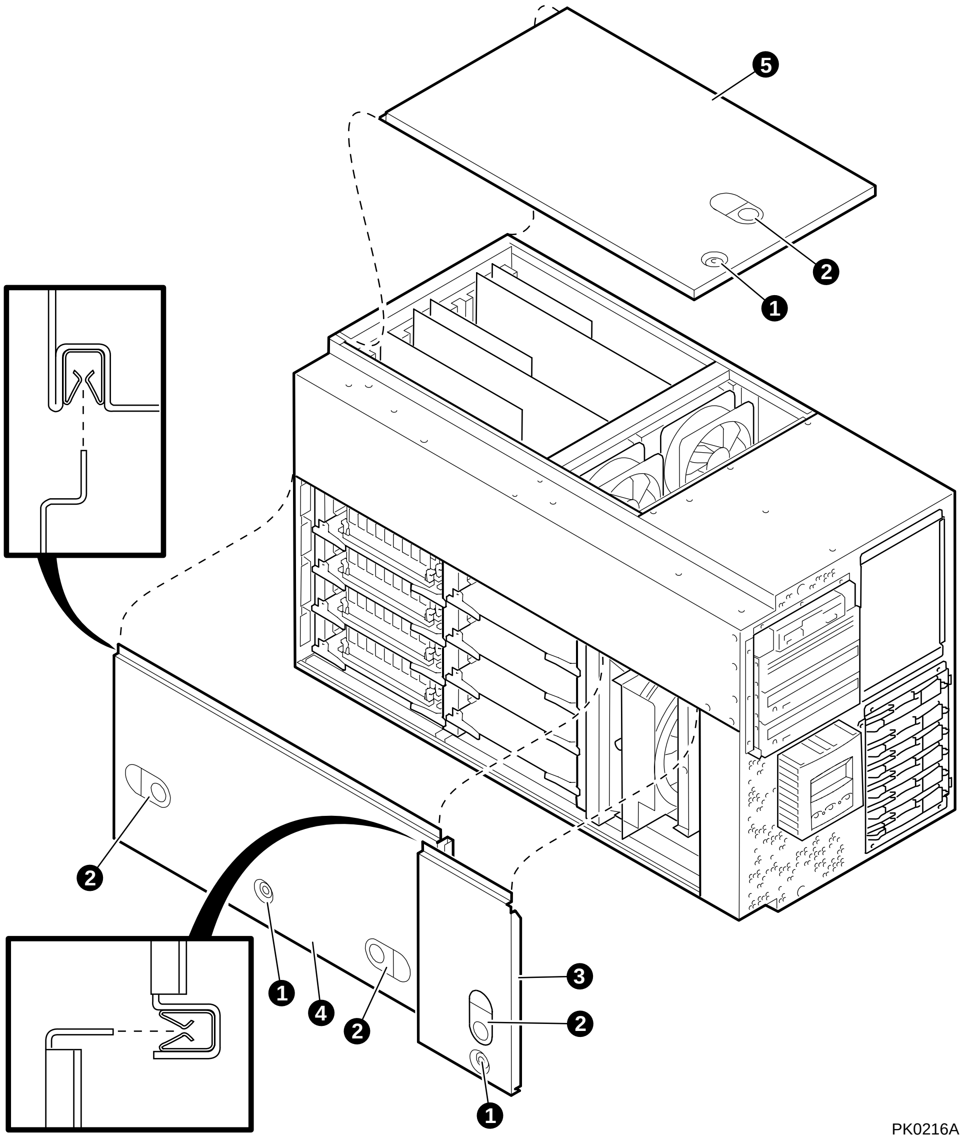
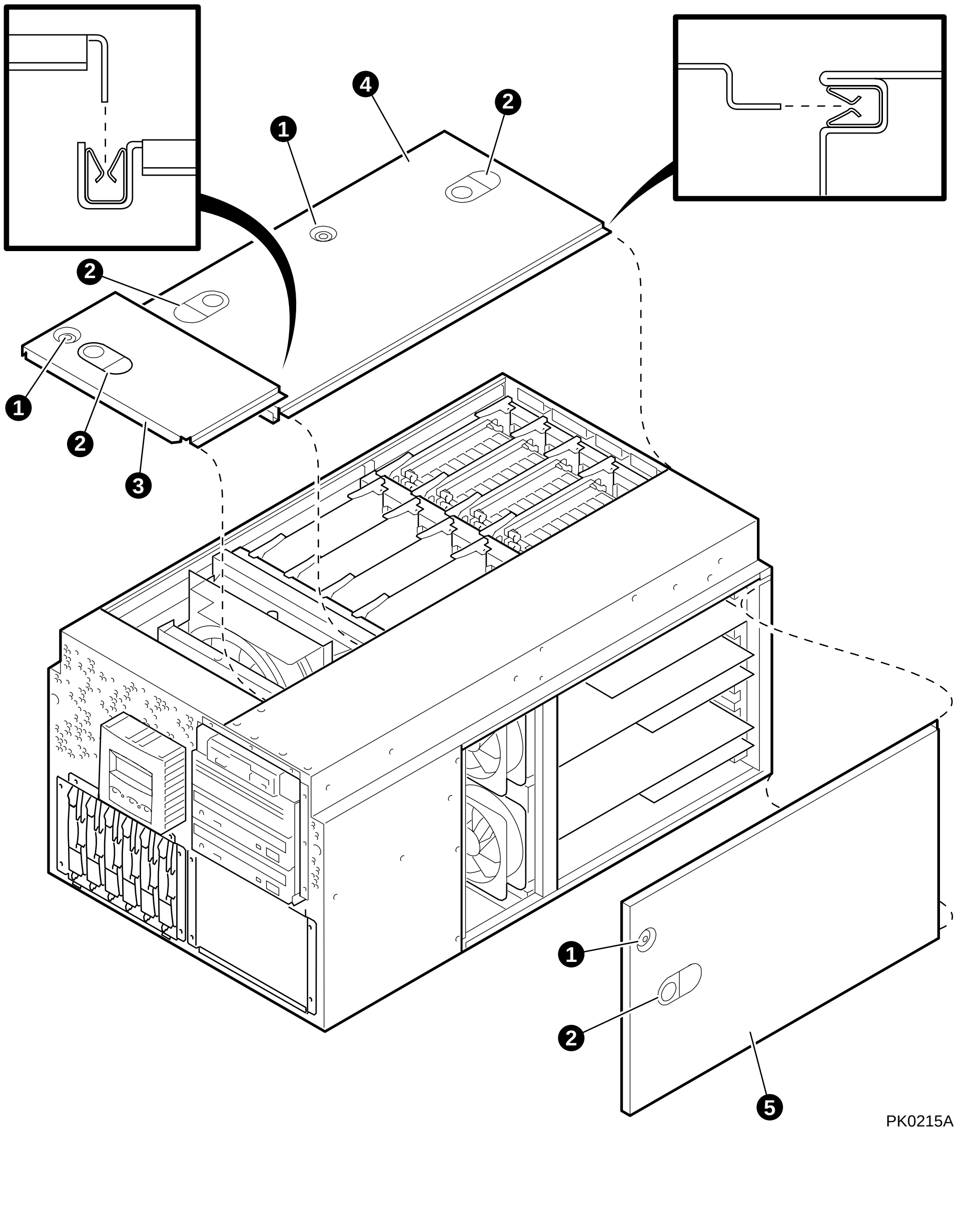
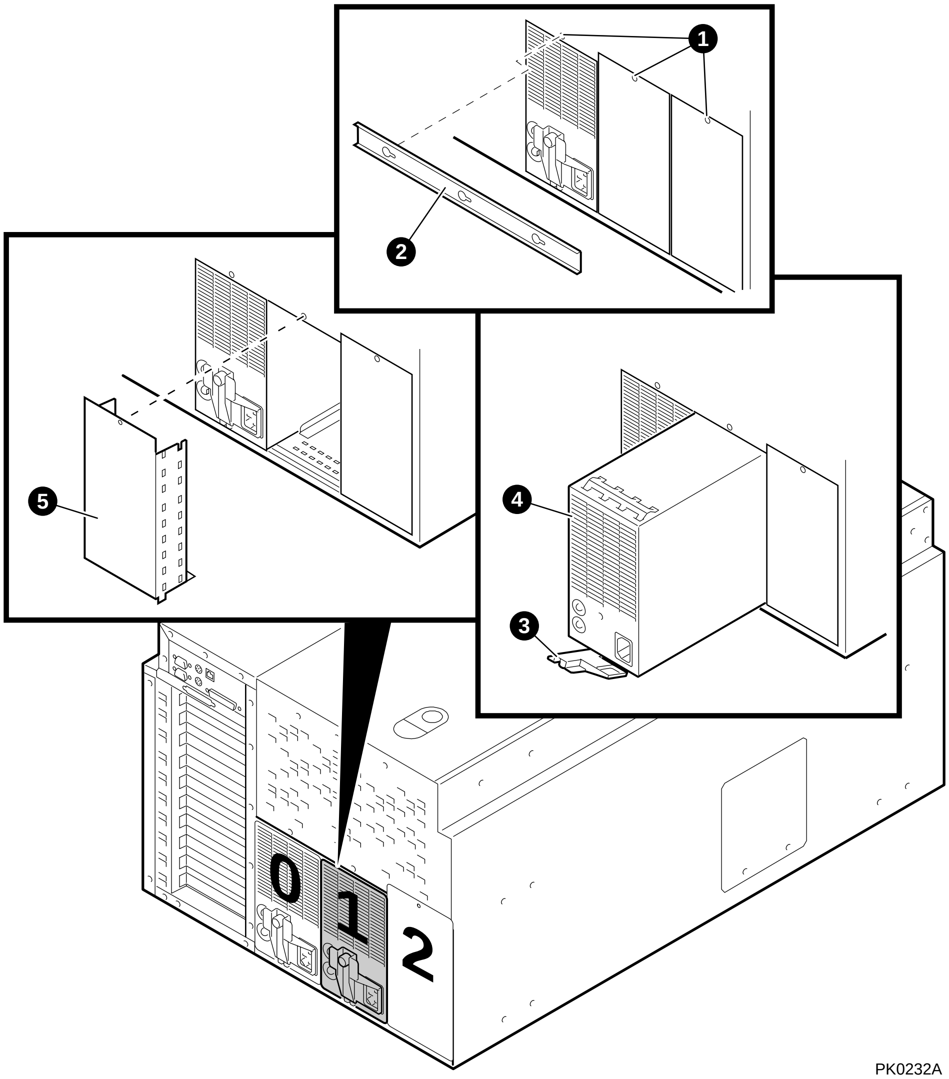


**Figure 8–8 Covers on the System Chassis (Pedestal/Rack)**



## 8.5 Power Supply

### Figure 8–9 Replacing or Adding a Power Supply



⚠ **WARNING: Hazardous voltages are contained within the power supply. Do not attempt to service. Return to factory for service.**

The power supply is a hot-plug component. As long as the system has a redundant supply, you can replace a supply while the system is running.

### Replacing a Power Supply

1. Unplug the AC power cord.
2. Loosen the three Phillips screws ❶ that secure the power supply bracket. (Do not remove the screws.) Remove the bracket ❷.
3. Loosen the captive screw on the latch ❸ and swing the latch to unlock the power supply.
4. Pull the power supply ❹ out of the system.

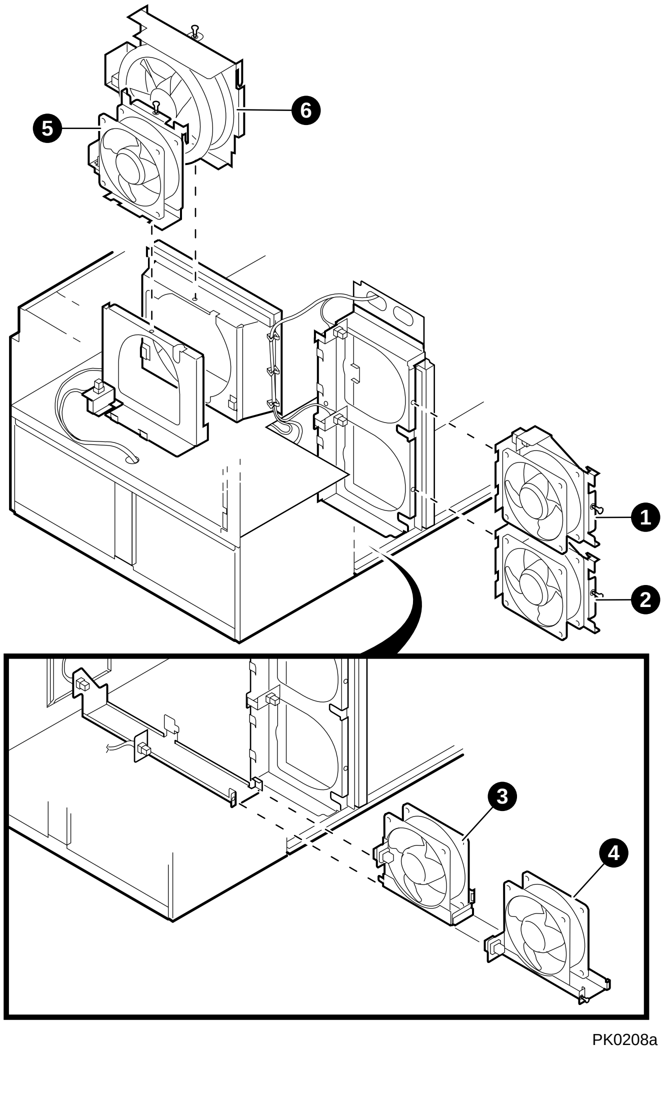
**NOTE:** *When installing an additional supply, remove the screw and blank cover ❺ on the slot into which you are installing the supply.*

### Verification

1. Plug the AC power cord into the supply. Wait a few seconds for the POK LED to light.
2. Check that both power supply LEDs are lit.

## 8.6 Fans

### Figure 8–10 Replacing Fans



The fans are hot-plug components. You can replace individual fans while the system is running.

**WARNING: Contact with moving fan can cause severe injury to fingers. Avoid contact or remove power prior to access.**

=== V @ >240VA

**WARNING: High current area. Currents exceeding 240 VA can cause burns or eye injury. Avoid contact with parts or remove power prior to access.**

### Replacing Fans

Remove the cover from the fan area (fans ❺ and ❻) or the PCI card cage (fans ❶,❷,❸, and ❹.

1. Pull the pop-up latch to unlock it, and lift the fan out of the system. Fan ❸ has no pop-up latch. It is held in place by fan ❹.
2. Install the new fan, taking care to align it as it slides in. Press the pop-up latch to lock the fan in place.
3. Replace the cover to the fan area or the PCI card cage.

### Verification — RMC

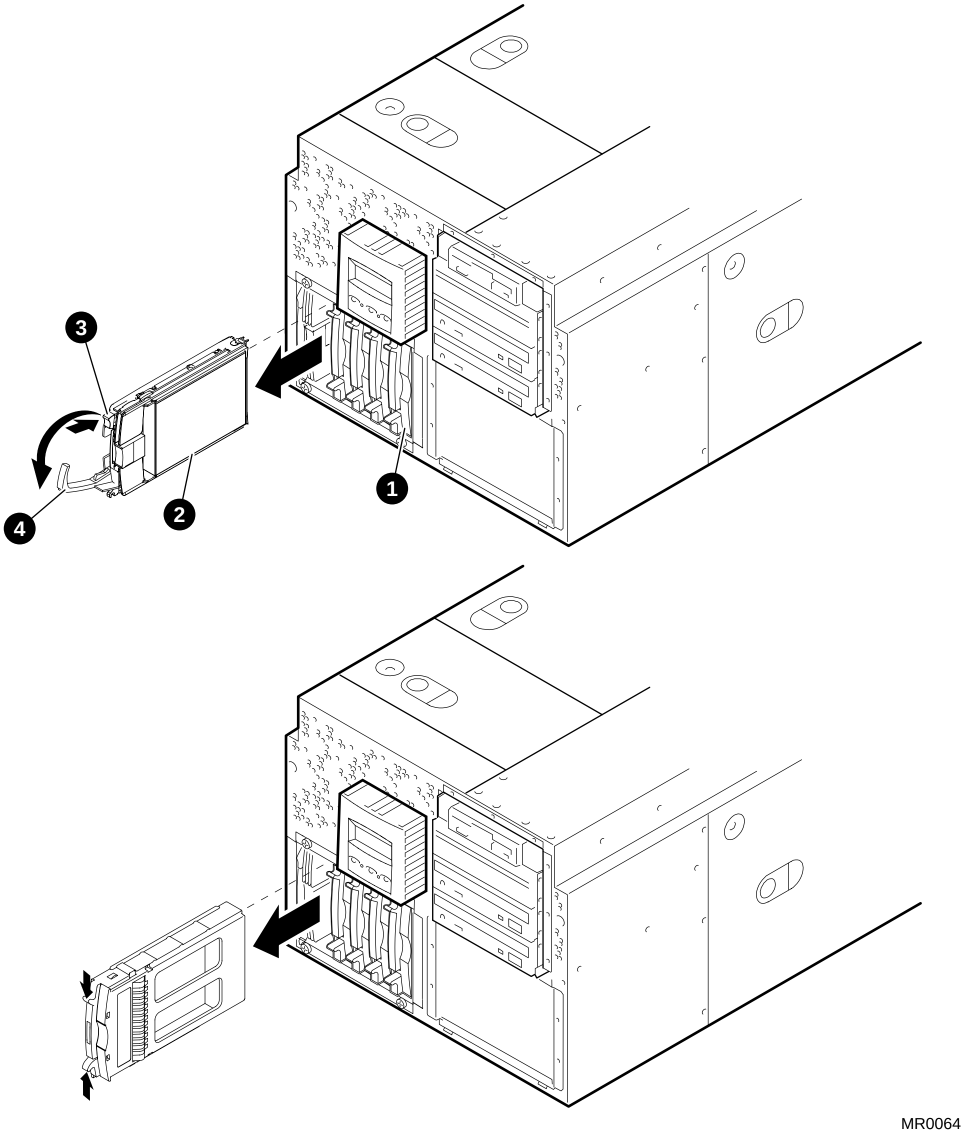
1. Invoke the remote management console.
2. Enter the **env** command to verify the fan status.

## 8.7 Universal Hard Disk Drives

**The system uses hot-pluggable universal hard disk drives. Hot-pluggable drives can be replaced without removing power from the system or interrupting the transfer of data over the SCSI bus.**

**WARNING: To prevent injury, access is limited to persons who have appropriate technical training and experience. Such persons are expected to understand the hazards of working within this equipment and take measures to minimize danger to themselves or others.**

## Figure 8–11 Replacing or Adding a Hard Drive



### Installing a Drive

1. Access the storage drive area and remove the drive blank ❶ for the next available slot (Drives are installed left to right, SCSI ID 0 – 5).
2. Insert the new drive ❷ into the cage and push it in while pivoting the release lever in toward the drive.
3. Push the release lever ❸ in until it engages the ejector button ❹.

### Replacing a Drive

1. Press the ejector button ❹ in and pivot the release lever ❸ to the open position.
2. Pull out on the drive until it is disconnected from the backplane connector.

---

**CAUTION:** *Do not remove the drive while the disk is spinning.*

---

3. When you are sure that the disk is no longer spinning, remove the drive from the enclosure.
4. Insert the replacement drive in until it is against the backplane connector. Continue to push it while pivoting the release lever ❸ to the full upright position.
5. Push the release lever in until it engages the ejector button ❹.

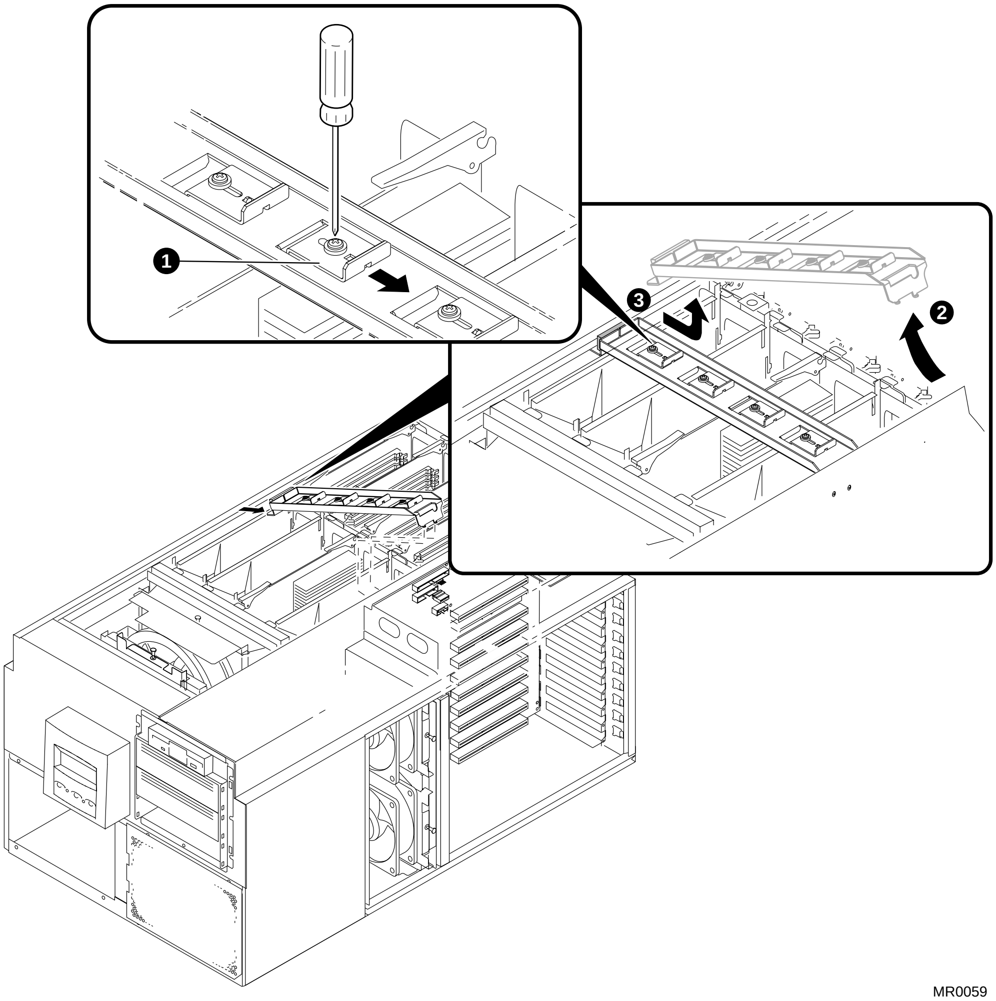
Observe the drive status LEDs to ensure that the new drive or replacement drive is functioning properly.

The SRM console polls for SCSI devices every 30 seconds. If the device does not appear to be working, access the SRM console and enter the **show device** command to view a list of the bootable devices.

## 8.8 Removing the Shipping Bracket

**The shipping bracket provides protection and stabilization for CPU modules during shipment.**

Figure 8–12 Removing the Shipping Bracket



Complete the following procedure to remove the shipping bracket:

Unscrew and loosen the slide ❶ that holds the CPUs.

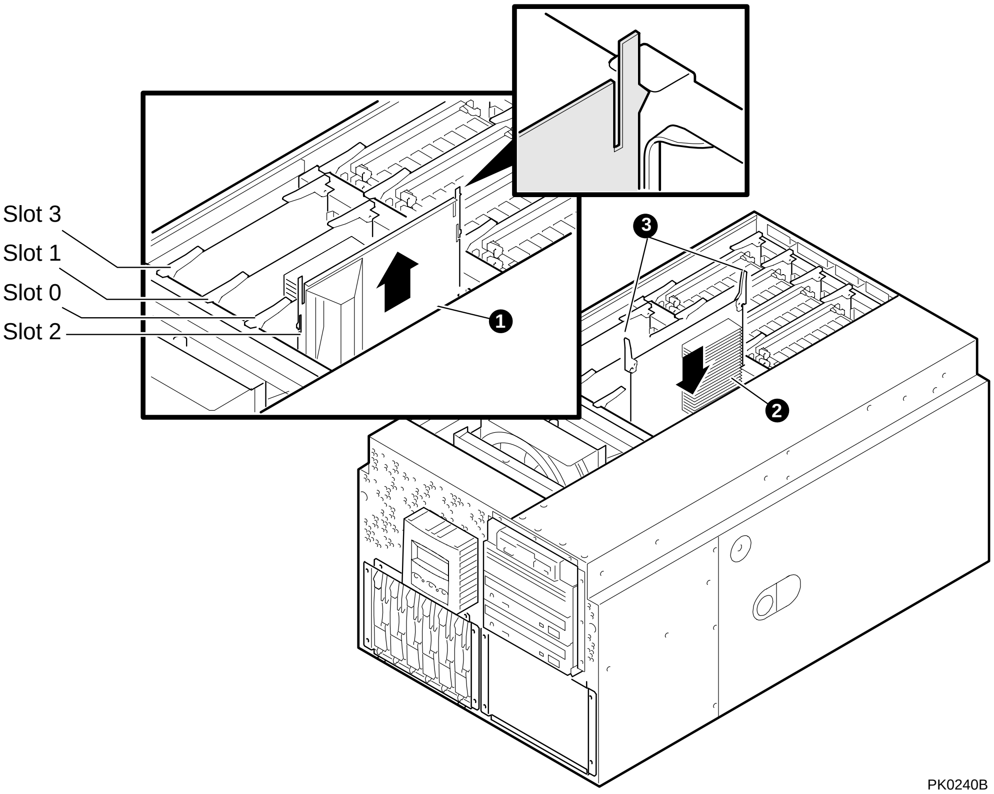
Push the bracket toward point ❸ to release it and then pull up at ❷. Save the shipping bracket for possible future shipment of the server.

**NOTE:** *The shipping bracket is only needed when shipping the server. You do not need to reinstall it, but save the strap for possible future use. If you ship the server in the future, reinstall the strap by reversing the removing procedure.*

## 8.9 CPUs

**Shut the system down before adding or replacing a CPU.**

Figure 8–13 Adding or Replacing CPU Cards



**WARNING: To prevent injury, access is limited to persons who have appropriate technical training and experience. Such persons are expected to understand the hazards of working within this equipment and take measures to minimize danger to themselves or others.**

**WARNING: Do not remove CPUs or memory modules until the green LEDs are off (approximately 20 seconds after a power-down.**

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

**WARNING: To prevent injury, unplug the power cord from each power supply before installing components.**

1. Shut down the operating system and turn off power to the system. Unplug the power cord from each power supply.
2. Access the system chassis by following the instructions in Section 8.2 or 8.3.
3. Remove the covers from the fan area and the system card cage as explained in Section 8.4.
4. When adding a CPU module, install it in the next lowest numbered slot available (See Figure 8–13).
5. When adding a CPU module, remove and discard the airflow deflector plate ❶ from the CPU slot.
6. Insert the CPU card ❷ into the connector and push down on both latches ❸ simultaneously.
7. Replace the system card cage cover, fan cover, and enclosure covers.
8. Reconnect the power cords.

### Verification

1. Turn on power to the system.
2. During power-up, observe the screen display. The newly installed CPU should appear in the display.
3. Issue the **show config** command to display the status of the new CPU.

## 8.10 Memory DIMMs

DIMMs are manufactured with two types of SRAMs, stacked and unstacked. Stacked DIMMs provide twice the capacity of unstacked DIMMs and, at the time of shipment, are the highest capacity DIMMs offered by Compaq. Your system may have either stacked or unstacked DIMMs.

A memory option consists of a “set” of four DIMMs. The system supports two sets per “array” and four arrays per system. You can mix stacked and unstacked DIMMs within the system, but not within an array. The DIMMs within an array must be of the same capacity and type (stacked or unstacked) because of different memory addressing.

When installing sets 0, 1, 2, and 3, an incorrect mix will not occur. When installing sets 4, 5, 6, or 7, however, you must ensure that the four DIMMs being installed match the type of DIMMs in the existing array. If necessary, rearrange DIMMs for proper configuration.

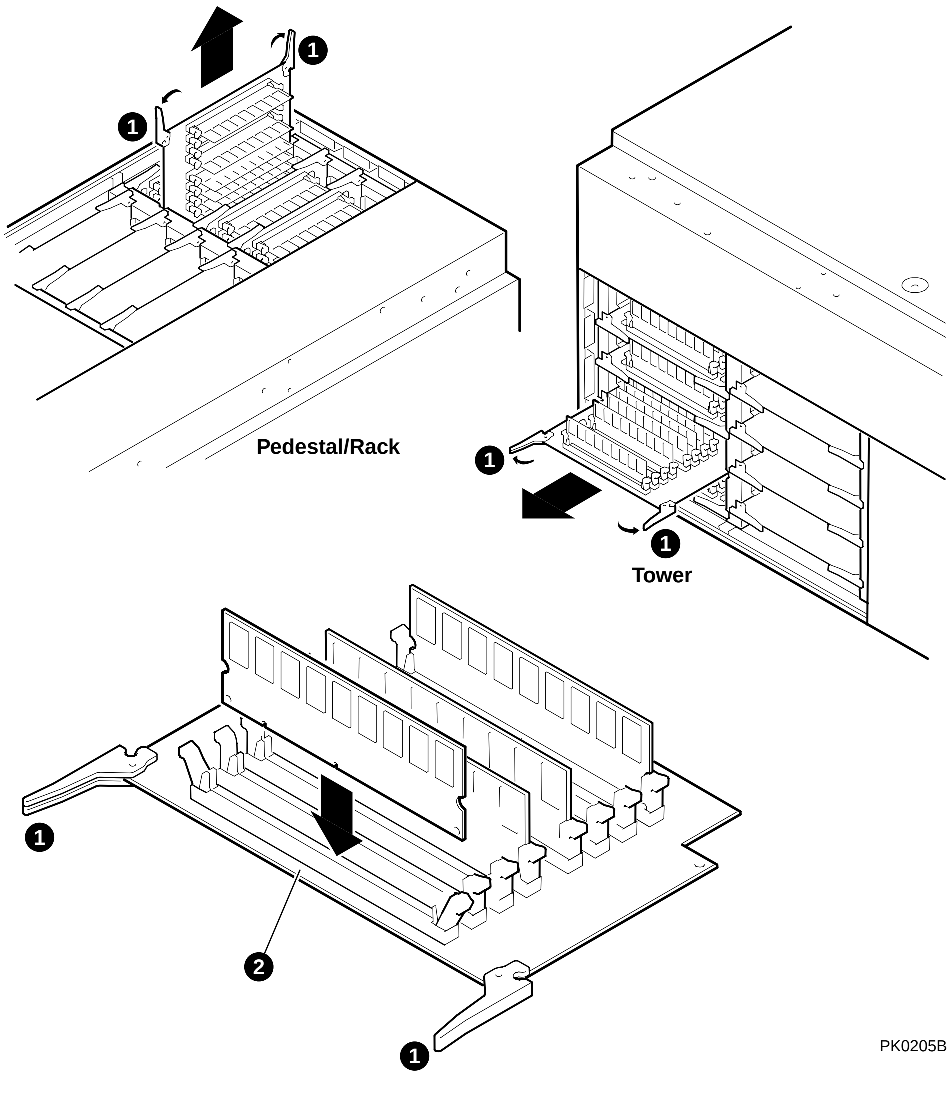
Only the following DIMMs and DIMM options can be used in the ES45 system.

| **Density** | **DIMM** | **DIMM Option (4 DIMMs per)** |
|---|---|---|
| 128 MB | 20-01CBA-09 | MS620-AA (512 MB) |
| 256 MB | 20-01DBA-09 | MS620-BA (1 GB) |
| 512 MB | 20-01EBA-09 | MS620-CA (2 GB) |
| 1 GB | 20-L0FBA-09 | MS620-DA (4 GB)* |

* Toshiba specific DIMM and option.

**CAUTION:** *Using different DIMMs may result in loss of data.*

## Figure 8–14 Installing and Removing MMBs and DIMMs



**WARNING: To prevent injury, access is limited to persons who have appropriate technical training and experience. Such persons are expected to understand the hazards of working within this equipment and take measures to minimize danger to themselves or others.**

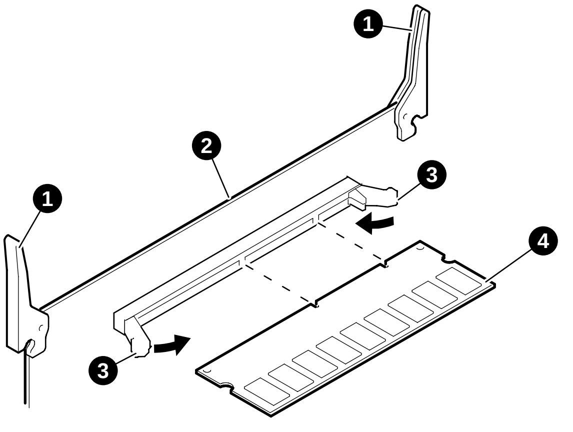
**WARNING: Modules have parts that operate at high temperatures. Wait 2 minutes after power is removed before touching any module.**

**WARNING: To prevent injury, unplug the power cord from each power supply before installing components.**

1. Shut down the operating system and turn off power to the system. Unplug the power cord from each power supply.
2. Access the system chassis by following the instructions in Section 8.2, Removing Enclosure Panels.
3. Remove the fan cover and the system card cage cover.
4. Use Figure 8–16 or Figure 8–17 to determine where sets of memory DIMMs should be installed. Begin with the next available lowest numbered set.
5. Release the clips ❶ securing the appropriate MMB ❷ and slide out the MMB.

6. Release the clips ❸ (Figure 8–15) on the MMB slot where you will install the DIMM ❹.
7. Install the DIMM and align the notches on the gold fingers with the connector keys.
8. Secure the DIMM with the clips ❸ on the MMB slot.

## Figure 8–15 Aligning DIMM in MMB



9. Reinstall the MMB.
10. Replace the system card cage cover and enclosure covers.
11. Reconnect the power cords.

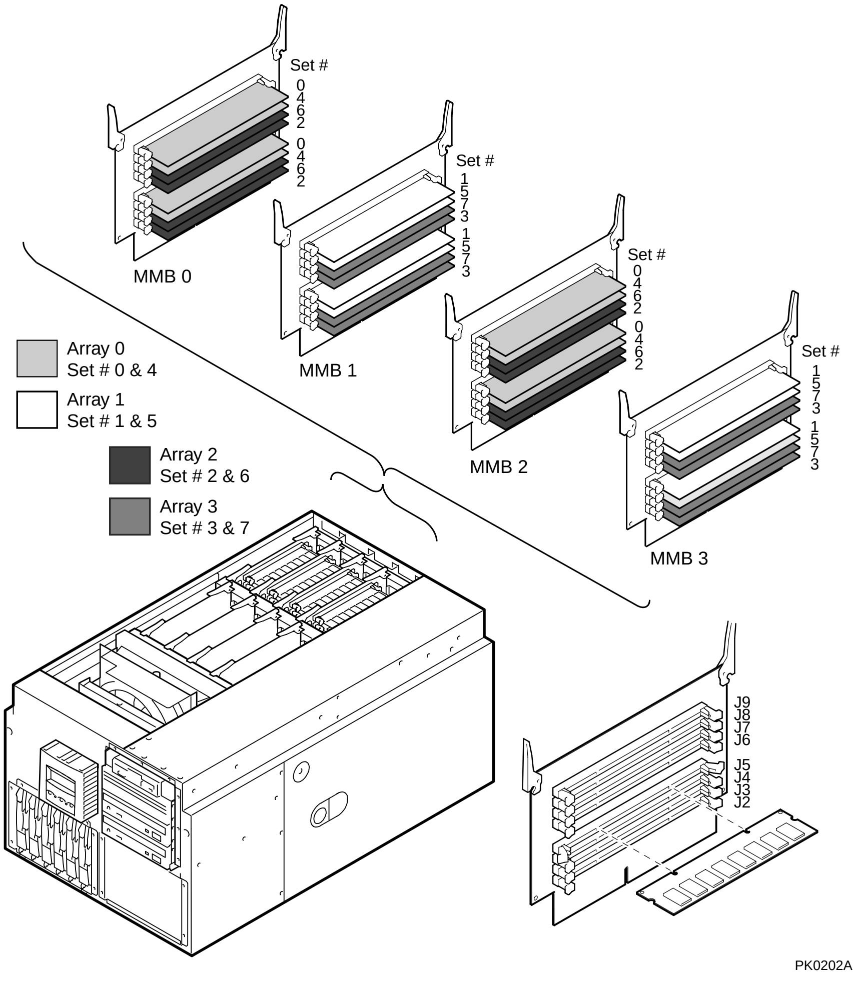
### Verification

1. Turn on power to the system.
2. During power-up, observe the screen display for memory. The display shows how much memory is in each array.
3. Issue the **show memory** command to display the total amount of memory in the system.

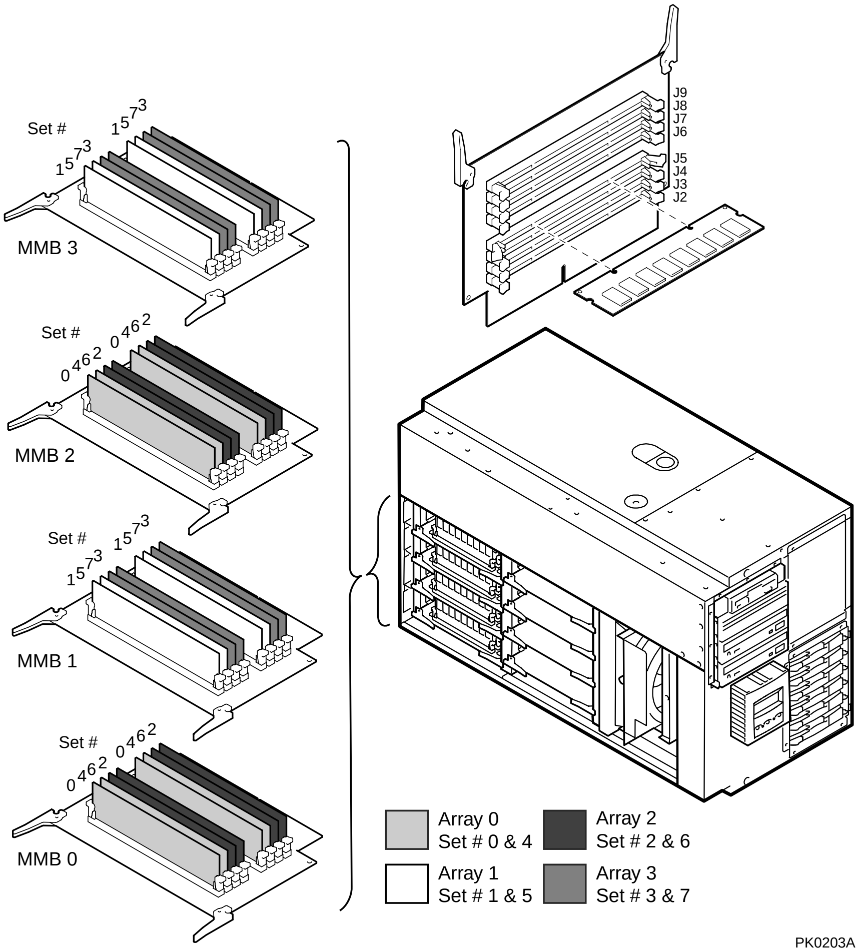
## 8.10.1 Determining Memory Configuration

For optimum memory utilization and performance load memory DIMMs into arrays in the following order: 0, 1, 2, 3, 4, 6, 5, and 7.

### Figure 8–16 Pedestal/Rack Memory Configuration



## Figure 8–17 Tower Memory Configuration



## 8.11 PCI Cards

**Some PCI options require drivers to be installed and configured. These options come with a floppy or a CD-ROM. Refer to the installation document that came with the option and follow the manufacturer's instructions.**

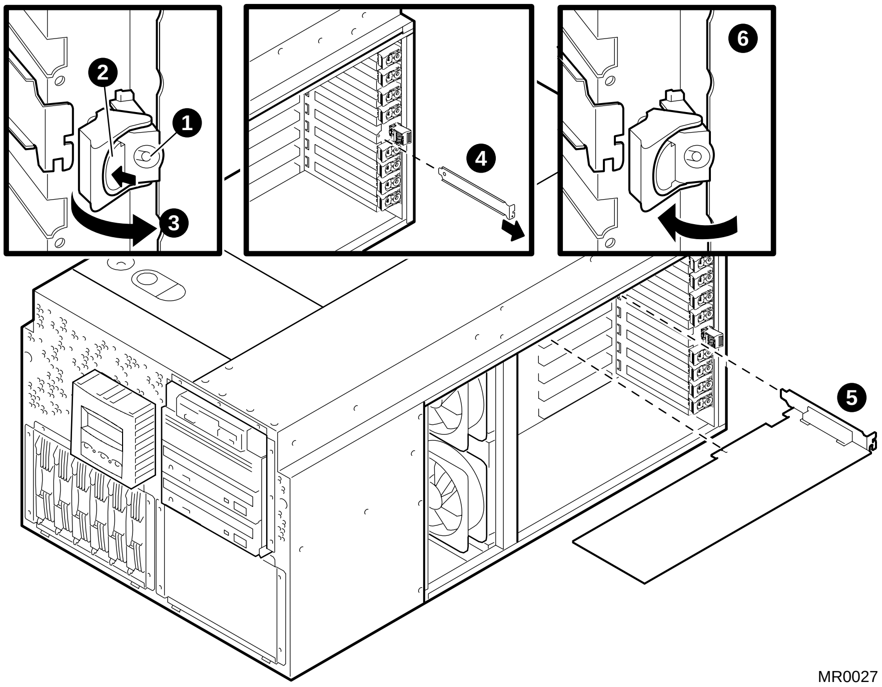
⚠ **WARNING: To prevent injury, access is limited to persons who have appropriate technical training and experience. Such persons are expected to understand the hazards of working within this equipment and take measures to minimize danger to themselves or others.**

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

⚠ ⎓ V @ >240V **WARNING: High current area. Currents exceeding 240 VA can cause burns or eye injury. Avoid contact with parts or remove power prior to access.**

⚠ **WARNING: The I/O area houses parts that operate at high temperatures. Avoid contact with components to prevent a possible burn.**

Figure 8–18 Installing or Replacing a PCI Card



### Adding or Replacing a PCI Card

**CAUTION:** *Hot plug is not currently supported on the operating system. Do not press switches ➊ or ➋ on the hot swap board. Pressing switches can result in loss of data.*

Complete the following procedure to add or remove a PCI option card.

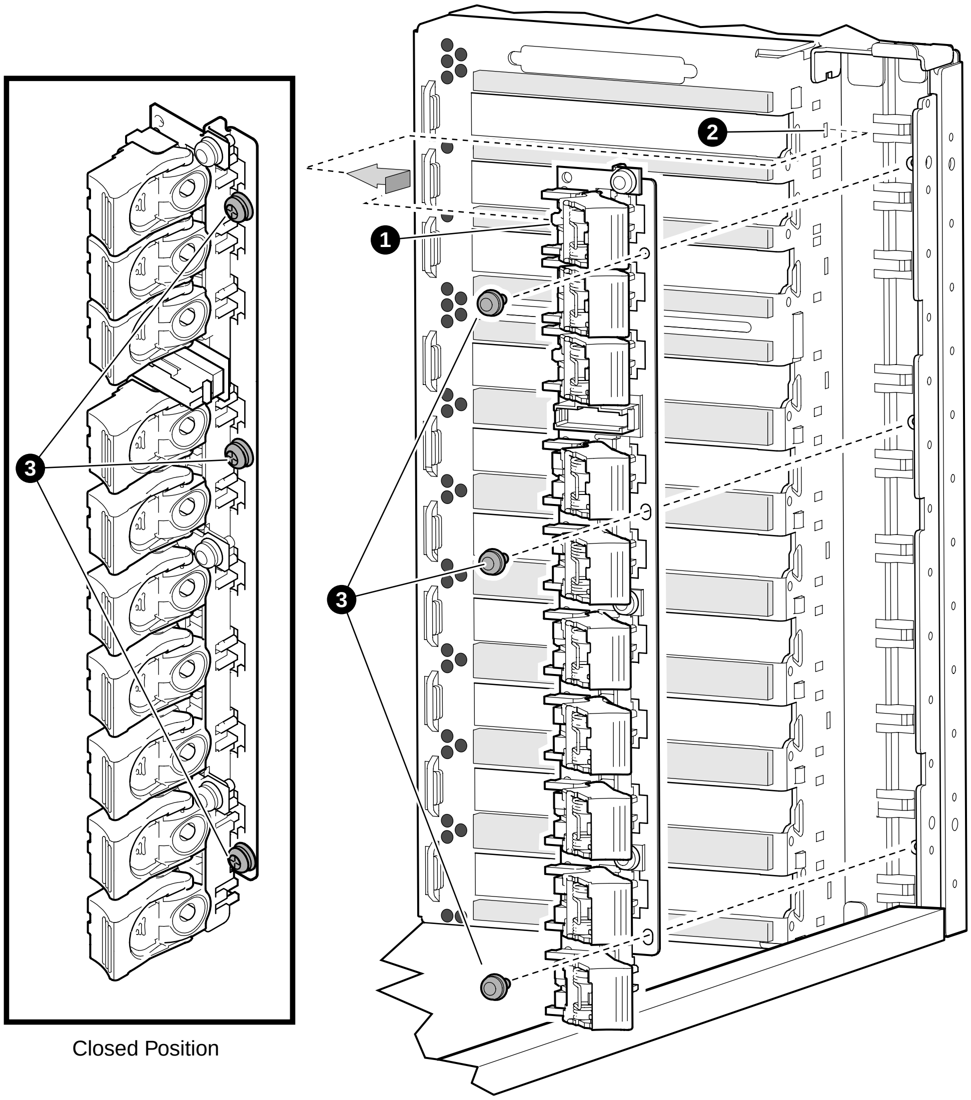
1. Turn off the system power.
2. Press in latch button ➋ and open ➌ the latch.
3. When replacing a PCI module, remove the bad module by pulling straight out.
4. When adding a PCI option card into an unused slot, remove the blank bulkhead ➍.
5. Install the new PCI option card ➎.
6. Close latch ➏.

**NOTE:** *Some full-length PCI cards may have extender brackets for installing into ISA/EISA-style card cages. Remove the extender brackets before installing such a card.*

### Verification

1. Turn on power to the system.
2. During power-up, observe the screen display for PCI information. The new option should be listed in the display.
3. Issue the SRM **show config** command. Examine the PCI bus information in the display to make sure that the new option is listed.
4. Enter the SRM **show device** command to display the device name of the new option.

## Figure 8–19 PCI Module Hot Swap Assembly



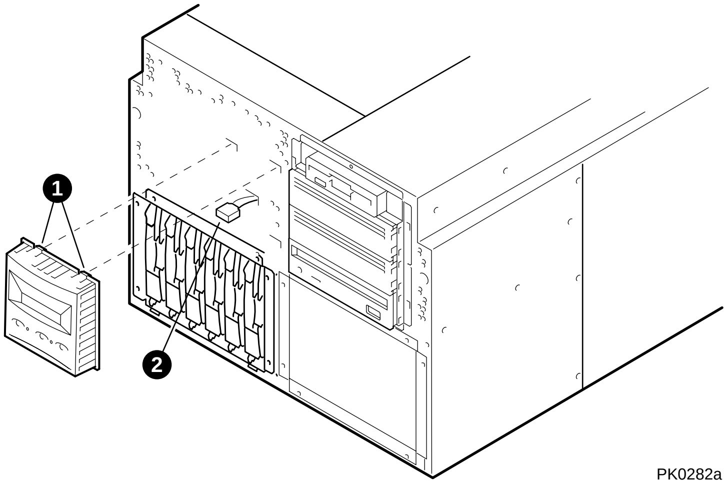
### 8.11.1 Replacing the PCI Hot Swap Module

Shut the system down before adding or replacing the PCI hot swap module.

1. Halt all applications and power down the system.
2. Unscrew and remove the three ❸ M3x 6mm screws and attached washers. (Note: make sure you remove the three lower screws that hold the switch board in place.)
3. Pull the module straight back removing the six tabs in the module ❶ from the slots ❷ in the metal.
4. To assemble a new module, reverse the procedure.

## 8.12 OCP Assembly

### Figure 8–20 Replacing the OCP Assembly



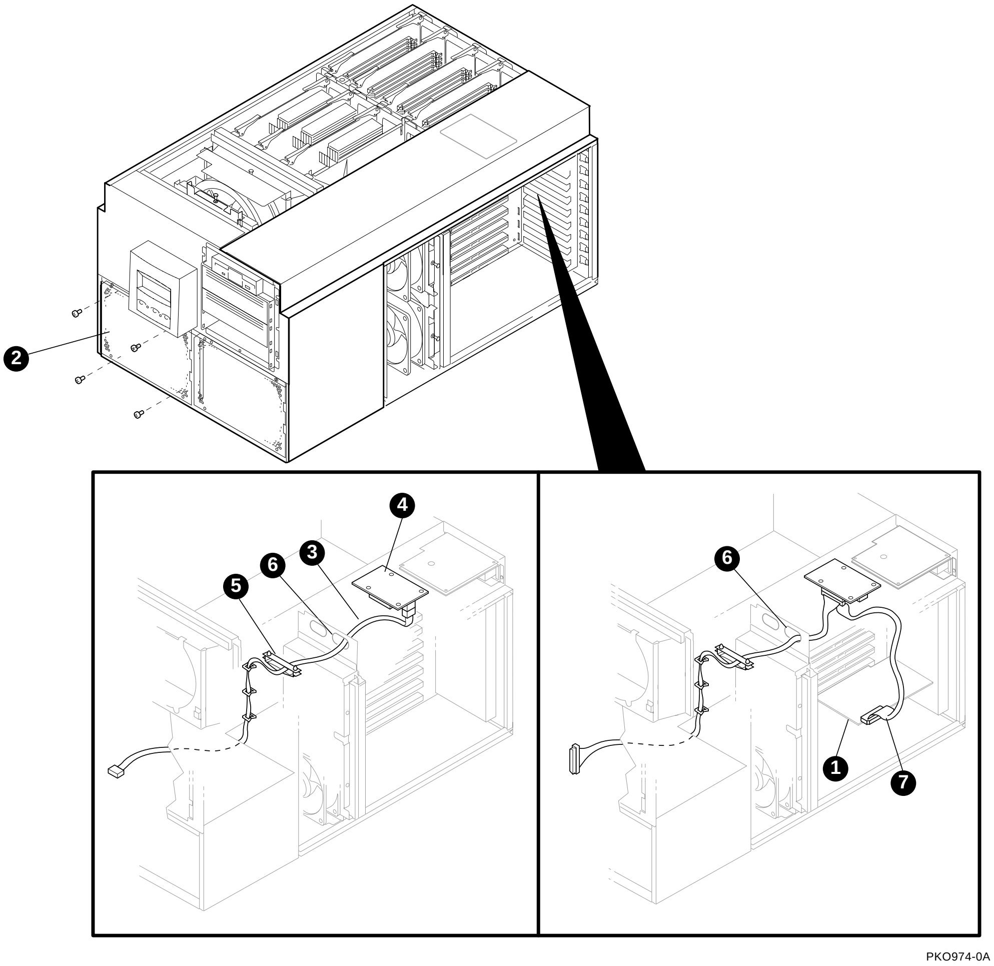
### Replacing the OCP Assembly

Shut the system down before removing the OCP assembly.

1. Press the two tabs ❶ on the top of the OCP assembly to release it.
2. Rotate the assembly toward you and lift it out of the two bottom tabs.
3. Disconnect the OCP cable ❷.
4. Reverse steps 1 through 3 to replace the OCP assembly.

## 8.13 Installing Disk Cages

### Figure 8–21 Cabling and Preparation for Installing Disk Cages



**WARNING: To prevent injury, access is limited to persons who have appropriate technical training and experience. Such persons are expected to understand the hazards of working within this equipment and take measures to minimize danger to themselves or others.**

**WARNING: To prevent injury, unplug the power cord from each power supply before installing components.**

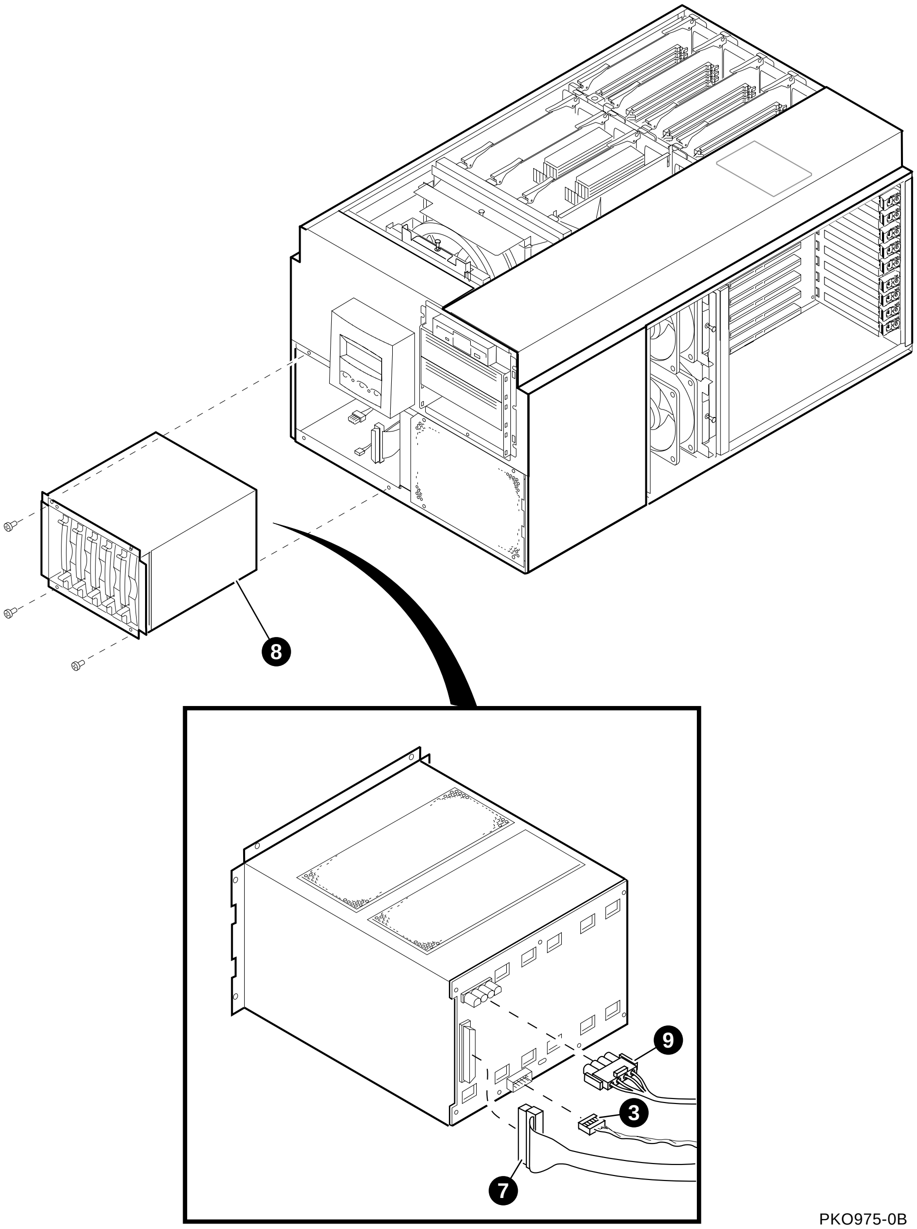
### Installing the Right Cage (or Top Cage)

1. Shut down the operating system and turn off power to the system. Unplug the power cord from each power supply.
2. Remove enclosure panels and remove the cover from the PCI card cage.

**NOTE:** *When installing a drive cage into the left cage area, remove fans 3 and 4.*

3. Install a SCSI controller ❶ in the PCI backplane.
4. Unscrew the four screws securing the drive cage filler plate ❷ and set them aside. Remove and discard the filler plate.
5. Attach the 10-pin cable (17-03971-11) ❸ to the environmental card ❹.
6. Snap the environmental card onto the rearmost set of four pop inserts in the top of the PCI card cage.
7. Snap open the cable management clip ❺. Thread the 10-pin cable through the opening ❻ and the clip and route as shown.
8. Plug the shorter end of the 68-pin cable ❼(17-04867-01) into the SCSI controller. Plug the middle connector into the environmental card, route as shown, and close the clip.

**Figure 8–22 Disk Cage Installation**



**CAUTION:** *Always plug the cables into the cage that is on the same side as the cage that was removed.*

9. Partially slide the drive cage ❽ into the system chassis.
10. Connect the power source ❾ (located inside enclosure) to the drive cage.
11. Attach the 10-pin cable ❸ and 68-pin cable ❼ to the drive cage.
12. Slide the cage in the rest of the way and attach it with the four screws set aside previously.
13. Replace fans 3 and 4 (if removed previously), PCI card cage cover, and enclosure panels.
14. Install disk drives.

**CAUTION:** *Disk drives must be installed from left to right. Otherwise, the system will not find the system disk.*

13. Plug in the power cords.

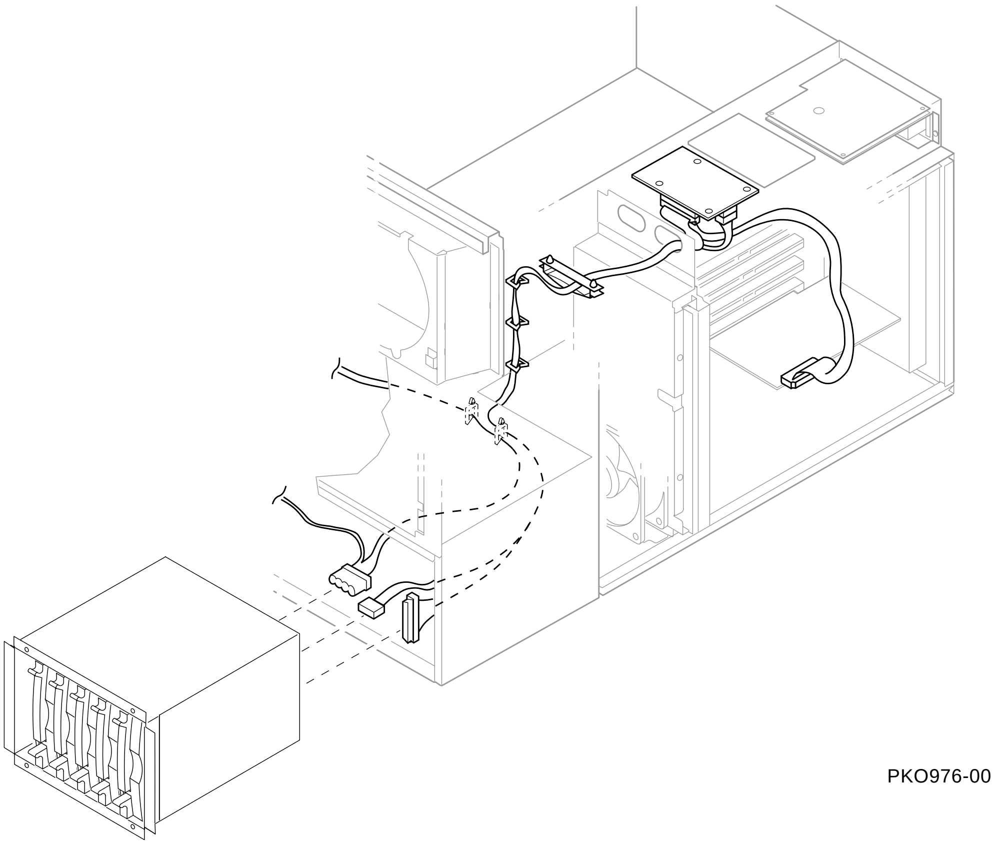
### Verification

1. Turn on power to the system.
2. At the P00>>> prompt, enter the SRM **show device** command to display the devices and controllers in the system. The list should include the SCSI controller and disk drives that you installed.

### 8.13.1 Cabling a Second Disk Drive Cage

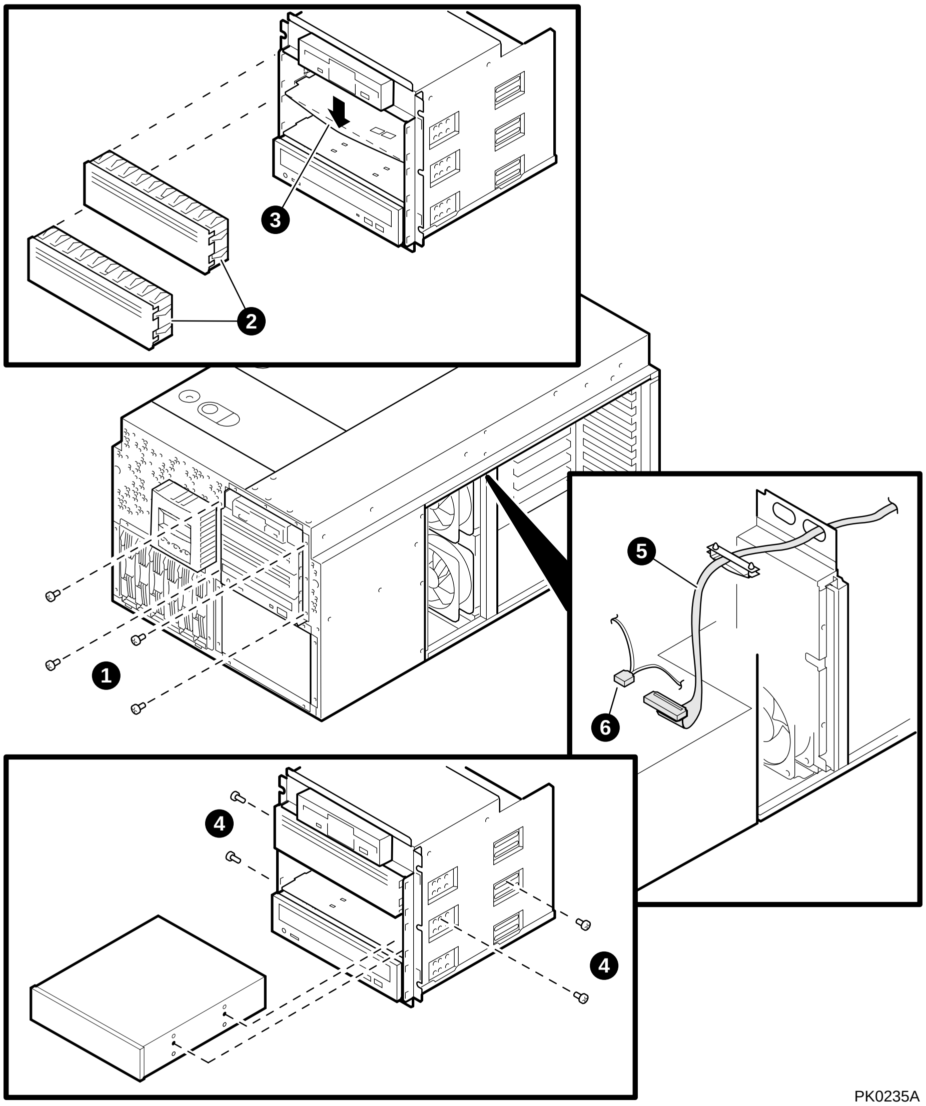
**If you are installing a second drive cage, refer to the following illustration for cable routing.**

Figure 8–23 Cabling a Second Disk Cage



## 8.14 Adding or Replacing Removable Media

### Figure 8–24 Adding a 5.25-Inch Device



**WARNING: To prevent injury, access is limited to persons who have appropriate technical training and experience. Such persons are expected to understand the hazards of working within this equipment and take measures to minimize danger to themselves or others.**

**WARNING: To prevent injury, unplug the power cord from each power supply before installing components.**

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

1. Shut down the operating system and turn off power to the system. Unplug the power cord from each power supply.
2. Remove the cover to the PCI card cage area.
3. Unplug the signal and power cables to the CD.
4. Remove and set aside the four screws ❶ securing the removable media cage.

**CAUTION:** *Be careful not to tangle the wires to the CD-ROM and floppy.*

5. Slide the cage out far enough to gain access to the floppy cables. Unplug the cables and remove the cage.
6. Remove a blank storage panel ❷ for the desired storage slot by pushing from behind the panel. If you are installing a full-height device, remove two panels.
7. Remove the divider plate❸ between the two slots by pressing the center of the plate and bending it sufficiently to free it from the slots.
8. Set the SCSI ID on the device as desired.

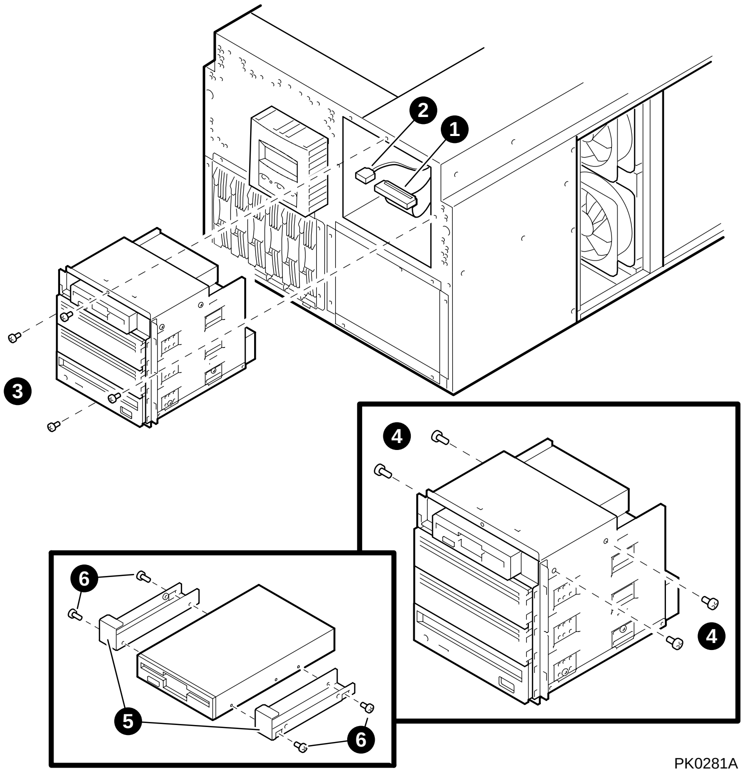
9. Slide the storage device into the desired storage slot and secure the device to the unit with four of the screws ❹ provided inside the removable media drive cage.
10. Pull the floppy cables back in.
11. Slide the removable media cage back in and replace the four screws set aside previously.
12. Plug in the signal cable ❺, route it into the PCI cage, and attach it to the appropriate controller.
13. Plug the power cable (4-conductor) ❻ into the storage device.
14. Plug the signal and power cables back into the CD.
15. Replace the PCI card cage cover and enclosure covers.
16. Reconnect the power cords.

### Verification

1. Turn on power to the system.
2. When the system powers up to the P00>>> prompt, enter the SRM **show device** command to determine the device name. For example, look for dq, dk, ew, and so on.

## 8.15 Floppy Drive

### Figure 8–25 Replacing the Floppy Drive



**WARNING: To prevent injury, unplug the power cord from each power supply before installing components.**

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

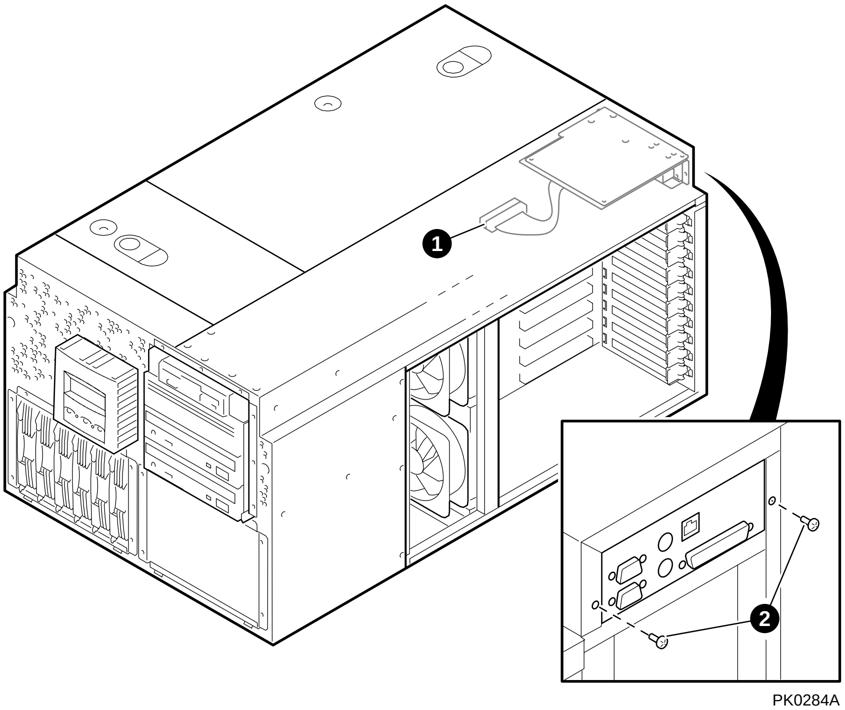
### Replacing the Floppy Drive

Shut the system down before removing the floppy drive.

1. Remove the cover to the PCI card cage.
2. Unplug the signal cable ❶ and power cable ❷ from all devices except the floppy.
3. Remove and set aside the four screws ❸ that secure the removable media cage.
4. Slide the cage out far enough to gain access to the floppy cables. Unplug the cables and remove the cage.
5. Remove the cage.
6. Remove the four screws ❹ that secure the floppy drive, and slide the floppy out.
7. Remove the mounting brackets ❺ (two screws ❻ in each bracket) from the floppy.

## 8.16 I/O Connector Assembly

### Figure 8–26 Replacing the I/O Connector Assembly



**WARNING: To prevent injury, unplug the power cord from each power supply before installing components.**

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

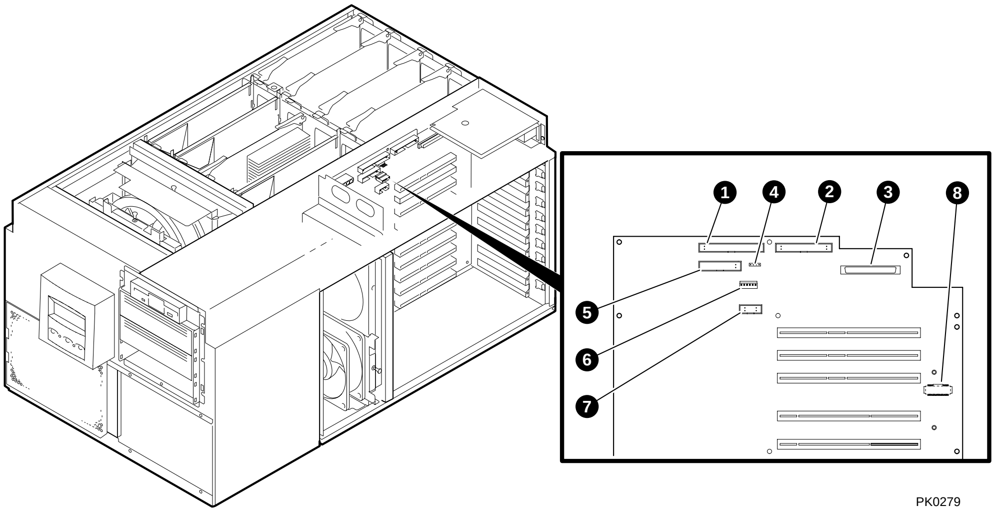
### Replacing the I/O Connector Assembly

Shut the system down before removing the I/O connector assembly.

1. Unplug all I/O connectors from the rear of the unit.
2. Remove the cover from the PCI card cage.
3. Remove PCI cards as needed for access.
4. Unplug the 68-pin signal cable ❶.
5. Remove the two screws ❷ that secure the assembly to the back of the unit.
6. Pull the assembly out through the PCI area.

## 8.17 PCI Backplane

### Figure 8–27 Cables Connected to PCI Backplane



|  | Cable | Connects To: |
|---|---|---|
| ❶ | 17-05021-01 | CD-ROM |
| ❷ | 17-03970-04 | Floppy |
| ❸ | 17-04400-06 | I/O controller module |
| ❹ | 70-31349-01 | Speaker |
| ❺ | 17-04785-01 | Fans |
| ❻ | 17-04786-01 | Cover sensors |
| ❼ | 17-03971-07 | OCP |
| ❽ | 17-05042-01 | Hot swap module |

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

**WARNING: To prevent injury, unplug the power cord from each power supply before installing components.**

### Disconnecting the Cables

Shut the system down before accessing the PCI area.

1. Remove the cover to the PCI card cage.
2. Record the location of installed PCI cards.
3. Remove all external cables from the PCI bulkheads in the rear of the unit. Remove internal cables from PCI cards.
4. Unlatch and remove the cards from the card cage.
5. Remove the separators that are between the PCI slots by pressing on the tab ❶ (see Figure 8–28) and sliding the separator to the right so that the holes (❸ and ❺) slide out from connection tabs (❷ and ❹).
6. Disconnect cables connected to the PCI backplane. See Figure 8–27.
7. Remove the top fan (pedestal/rack orientation) or left fan (tower orientation). This permits access to an ejector lever needed for removing the PCI backplane.

**Figure 8–28 Removing the Separators**

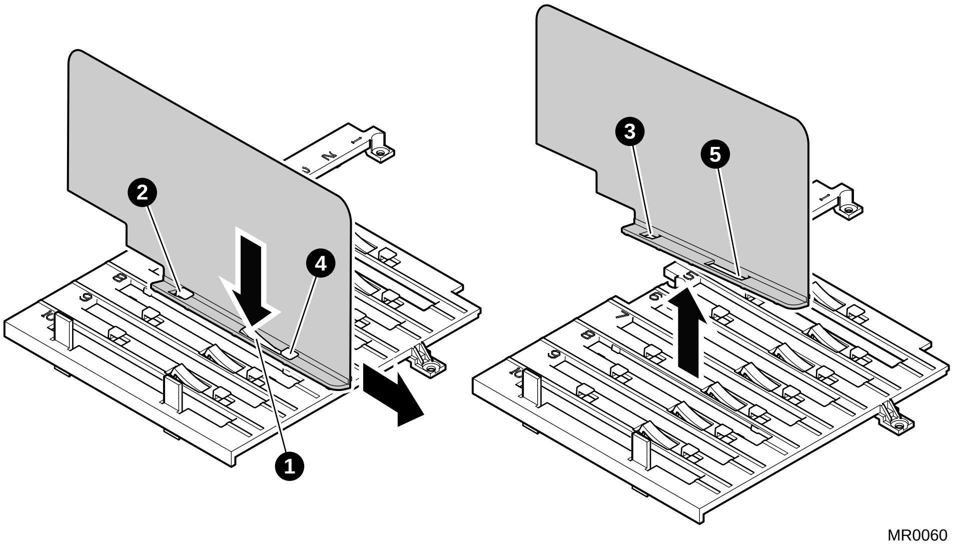
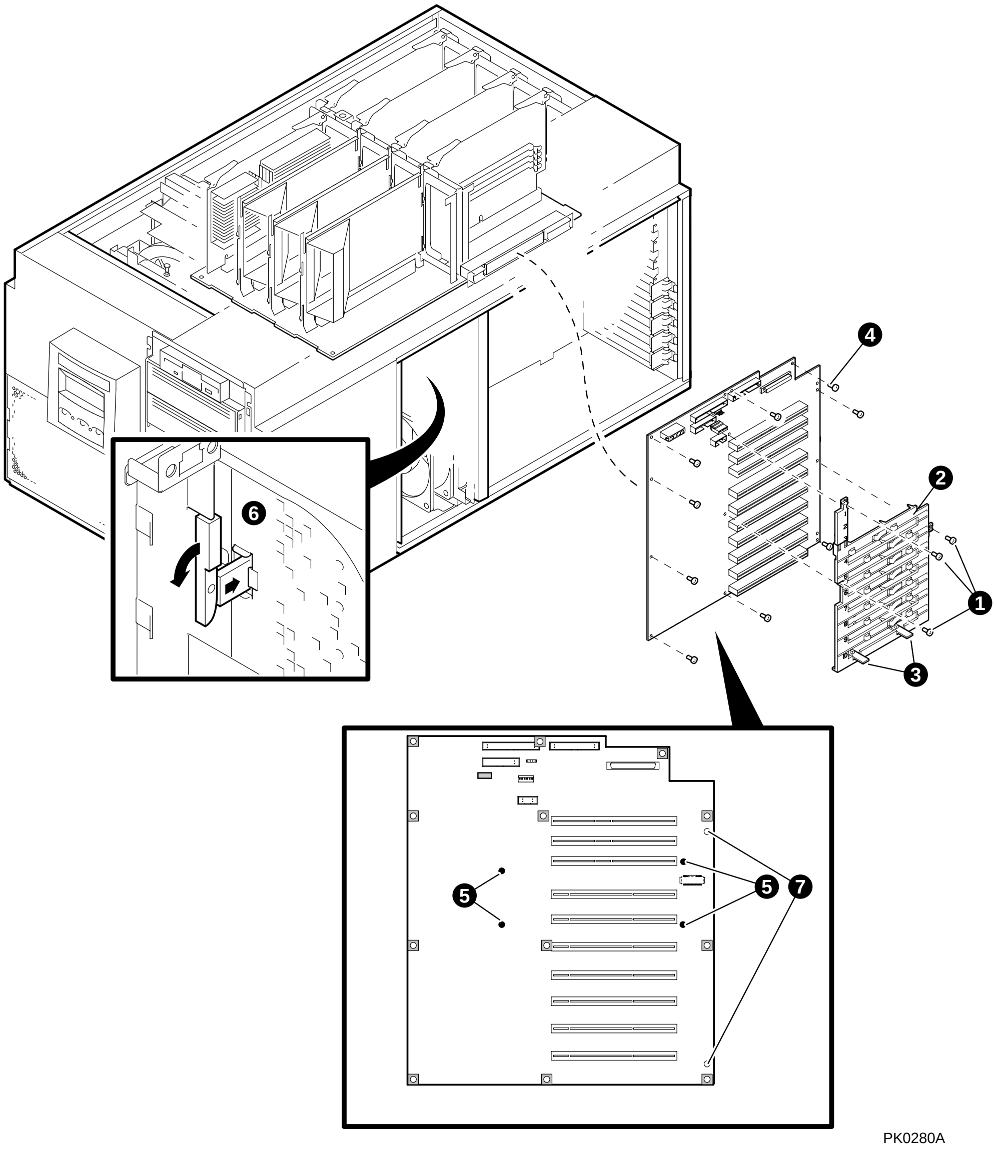


**Figure 8–29 Replacing the PCI Backplane**



### Replacing the PCI Backplane

**CAUTION:** *When removing the PCI backplane, be careful not to flex the board. Flexing the board may damage the BGA component connections.*

1. Remove the three screws ❶ that secure the base ❷ holding the PCI dividers.
2. Press up the two tabs❸ on the base and remove the base.
3. Remove the other nine screws ❹ that secure the PCI backplane to the chassis.

   **CAUTION:** *Do not remove the four additional nonwashered screws ❺. Removing them inactivates the built-in mechanism for extracting the PCI backplane from the system.*

4. Use the ejector lever ❻ in the fan area to separate the PCI backplane from the system motherboard; then lift the backplane out of the chassis.

**NOTE:** *When installing a new PCI backplane, align the backplane on the guide pins ❼, and press the board firmly until it is seated. Seating the PCI backplane requires considerable pressure. When seating the PCI backplane in a cabinet, a second person should brace the chassis to ensure that no excessive stress is placed on the rails.*

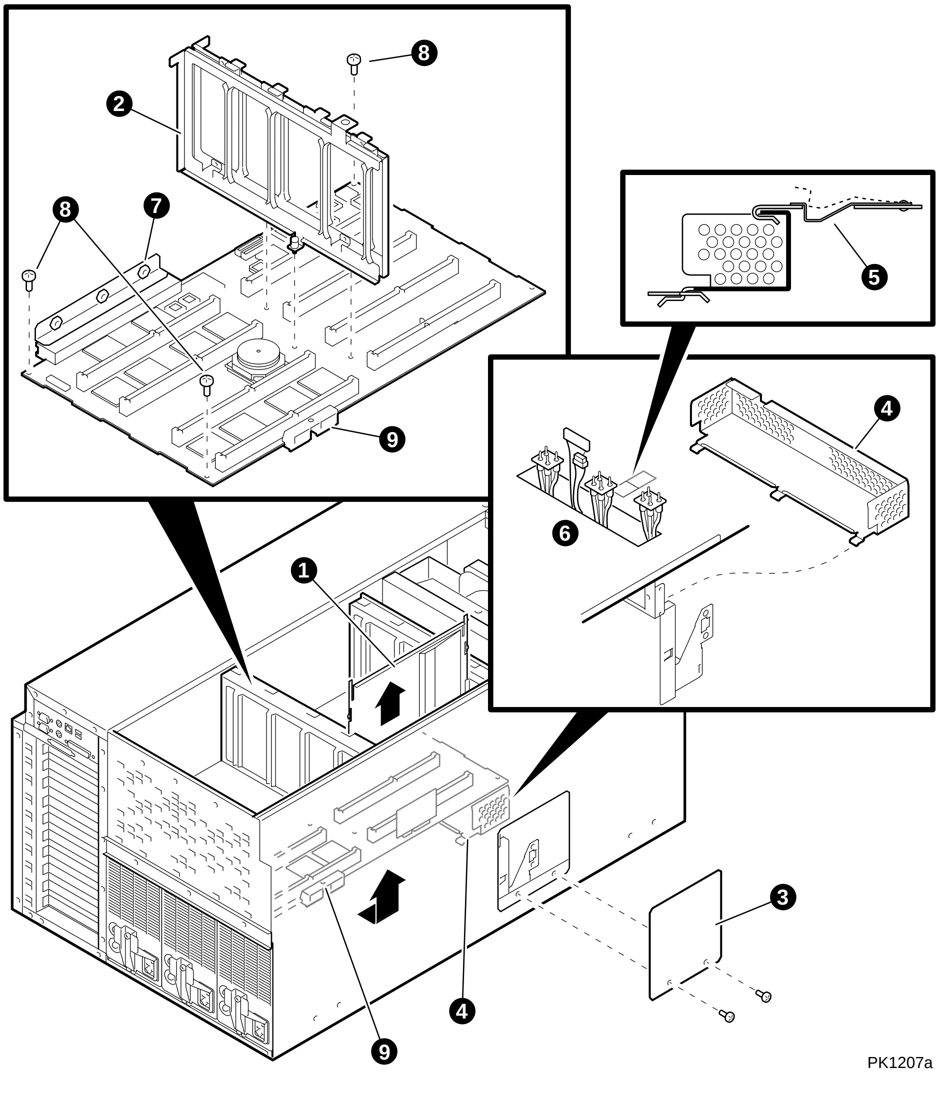
### Reinstalling the Separators

Reference Figure 8–29 for this procedure.

1. Align the left hole ❸ with the corresponding connection tab ❷ in the divider. Note that the right corresponding hole ❺ and connection tab ❹ are aligned.
2. Push the separator ❶ in and slide it to the left.

## 8.18 System Motherboard

### Figure 8–30 Replacing the System Motherboard



**WARNING: To prevent injury, unplug the power cord from each power supply before installing components.**

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

**CAUTION:** *When removing the system motherboard, be careful not to flex the board. Flexing the board may damage the BGA component connections.*

**NOTE:** *Replacing the system motherboard requires the removal of other FRUs. Review the removal procedures for the fans, MMBs, CPUs, and drive cage before beginning the system motherboard removal procedure.*

1. Remove the three covers from the system chassis.
2. Remove fans 3 and 4 in the PCI area (the inner fans).
3. Record the positions of the MMBs and CPUs, and remove the MMBs and CPUs.
4. Remove the CPU airflow deflectors ❶, if present.
5. Loosen the three captive Phillips screws holding the middle support bracket ❷. The screws pop up when sufficiently loosened. Pull the bracket straight out.
6. Remove the drive cage (left cage in pedestal/rack, bottom cage in tower), if installed, or the blank panel.
7. Remove the two Phillips flat-head screws that secure the small cover ❸ to the left side (pedestal/rack) or bottom (tower) of the system and remove the cover. Set aside the screws. (The cover provides access to the power harness.)

8. Remove the power harness bracket ❽ as follows: Push up on the spring latch ❺ to release the bracket, slide the bracket forward, and remove it.
9. Unplug the five connectors ❻ on the bottom of the system motherboard.
10. Using a nut driver, loosen the three nuts ❼ (7 mm) on the flange over the intermodule connector so that it can move freely. Move the flange up from the connector and tighten one of the flange nuts to keep the flange out of the way.

---

**NOTE:** *After replacing the motherboard, loosen the flange nut and push the flange down to the intermodule connector. Retighten the nuts on the flange. While tightening the nuts, put pressure on the flange to compress it into the connector.*

---

11. Remove the three Phillips screws ❽ that secure the system motherboard.
12. A white plastic ejector ❾ and two holes in the sheet metal under it are used to help disengage the motherboard. Insert a screwdriver through the hole in the ejector into the closest hole and pry the system motherboard away from the PCI backplane. Insert the screwdriver into the second hole that is now exposed and pry again to fully disengage the system motherboard connector from the PCI backplane.
13. Extract the system motherboard.

### After installing a new motherboard:

1. Power up to the P00>>> prompt.
2. Enter the **clear_error all** command.
3. Enter the **set sys_serial_num** command to set the system serial number. (The serial number is on a label on the back of the system.) For example:

```
P00>>> set sys_serial_num NI900100022
```

---

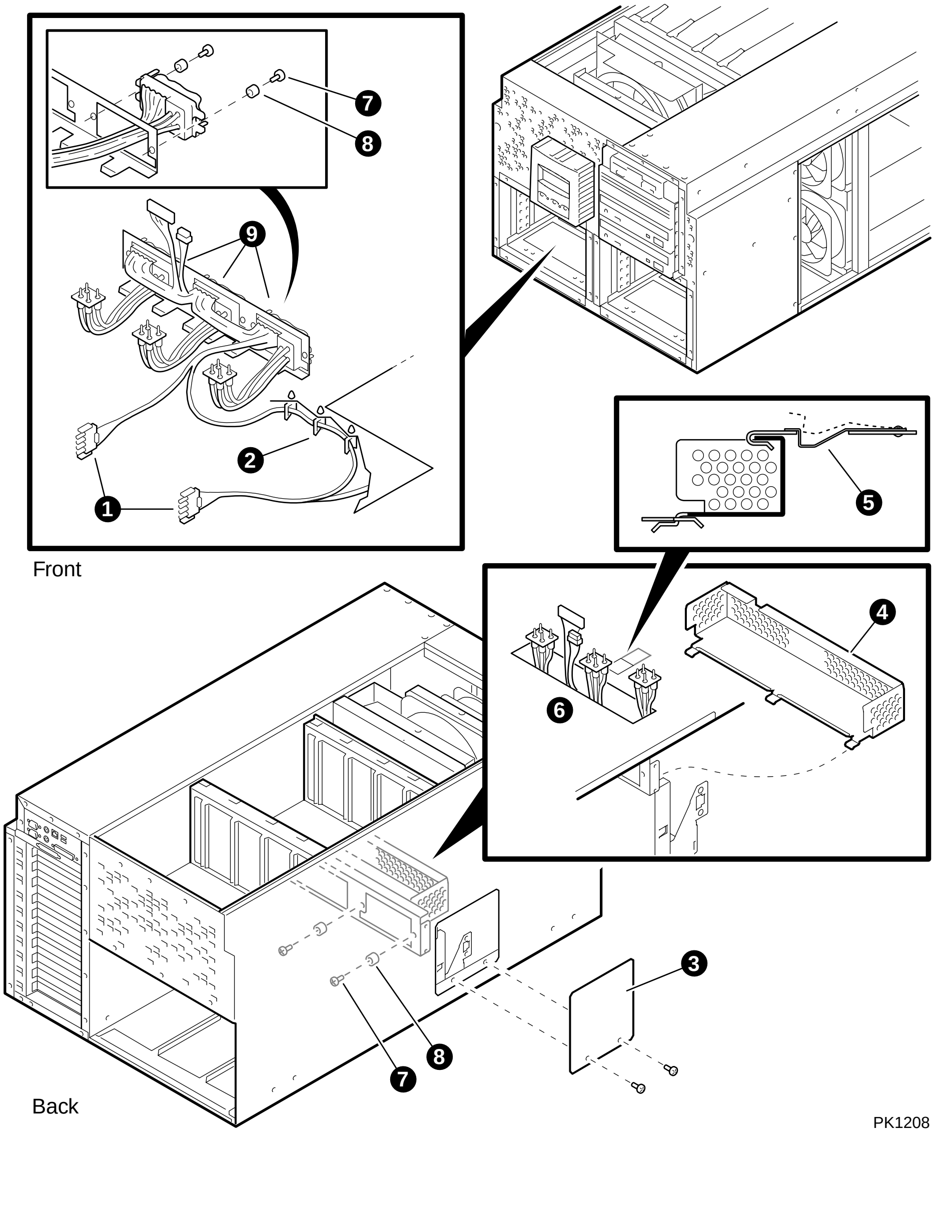
**IMPORTANT:** *The system serial number must be set correctly. Compaq Analyze will not work with an incorrect serial number.*

---

The serial number propagates to all FRU devices that have EEPROMs.

## 8.19 Power Harness

### Figure 8–31 Replacing the Power Harness



**WARNING: To prevent injury, unplug the power cord from each power supply before installing components.**

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

**NOTE:** *Replacing the power harness requires the removal of other system FRUs. Review the removal procedures for the power supplies, fans, and drive cage before beginning the harness removal procedure.*

1. Remove the power supplies and any blank power supply panels.
2. Remove the cover to the PCI card cage.
3. Remove fans 4 and 3 (the inner fans).
4. Unplug the connectors to each removable media device (except the floppy).
5. Remove the four screws that secure the removable media cage. Slide out the cage to access the floppy power connector. Disconnect the floppy power connector and slide the cage back in.
6. Unplug the power connector ❶ to the drive cage or cages.
7. Remove the harness from the cable clamps ❷.
8. Remove the drive cage (left cage in pedestal/rack, bottom cage in tower), if installed, or the blank panel.
9. Remove the two Phillips flat-head screws that secure the small cover ❸ to the left side (pedestal/rack) or bottom (tower) of the system and remove the panel. Set aside the screws. (Removing the small cover provides better access to the power harness bracket.)
10. Remove the power harness bracket ❹ as follows: Push up on the spring latch ❺ to release the bracket, slide the bracket forward, and remove it .
11. Unplug the five connectors ❻ on the bottom of the system motherboard.

12. Remove the two screws ❼ and two plastic bushings ❽ on each of the three power supply connectors ❾. The screws are located deep inside the power supply cavity. Set aside the screws and bushings for reinstallation.
13. Starting with the left connector (as viewed from the rear of the system), pull the connector to the right and angle it so that you can push the left end out through the opening.
14. Remove the power harness.

# Appendix A
# SRM Console Commands

This appendix lists the SRM console commands that are most frequently used with the ES4x family of systems.

### Table A–1 SRM Commands Used on ES45 Systems

| Command | Function |
|---|---|
| **boot** | Loads and starts the operating system. |
| **buildfru** | Initializes $I^2$Cbus EEPROM data structures for the named FRU. |
| **cat el** | Displays the console event log. Same as **more el**, but scrolls rapidly. The most recent errors are at the end of the event log and are visible on the terminal screen. |
| **clear_error** | Clears errors logged in the FRU EEPROMs as reported by the **show error** command. |
| **clear_error all** | Clears all errors. |
| **continue** | Resumes program execution on the specified processor or on the primary processor if none is specified. |
| **crash** | Forces a crash dump at the operating system level. |
| **deposit** | Writes data to the specified address of a memory location, register, or device. |
| **edit** | Invokes the console line editor on a RAM file or on the user power-up script, “nvram,” which is always invoked during the power-up sequence. |
| **examine** | Displays the contents of a memory location, register, or device. |

*Continued on next page*

### Table A–1 SRM Commands Used on ES45 Systems (Continued)

| Command | Function |
|---|---|
| **exer** | Exercises one or more devices by performing specified read, write, and compare operations. |
| **floppy_write** | Runs a write test on the floppy drive to determine whether you can write on the diskette. |
| **galaxy** | Same as **lpinit**. |
| **grep** | Searches for “regular expressions”—specific strings of characters—and prints any lines containing occurrences of the strings. |
| **hd** | Dumps the contents of a file (byte stream) in hexadecimal and ASCII. |
| **help** ***command*** | Displays information about the specified console command. |
| **info** | Displays registers and data structures. |
| **init** | Resets the SRM console and reinitializes the hardware. |
| **kill** | Terminates a specified process. |
| **kill_diags** | Terminates all executing diagnostics. |
| **lpinit** | Used in an OpenVMS Galaxy environment. Initializes the hardware resources into zero, one, or two partitions. |
| **man** | Displays information about the specified console command. |
| **memexer** | Runs a requested number of memory tests in the background. |
| **memtest** | Tests a specified section of memory. |
| **more el** | Same as **cat el**, but displays the console event log one screen at a time. |
| **net -ic** | Initialize 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. |

Table A–1 SRM Commands Used on ES45 Systems (Continued)

| Command | Function |
|---|---|
| **rmc** | Invokes the remote management console from the local VGA monitor. |
| **set** ***envar*** | Sets or modifies the value of an environment variable. |
| **set sys_serial_num** | Sets the system serial number. |
| **show** ***envar*** | Displays the state of the specified environment variable. |
| **show config** | Displays the logical configuration at the last system initialization. |
| **show device** | Displays a list of controllers and bootable devices in the system. |
| **show error** | Reports errors logged in the FRU EEPROMs. |
| **show fru** | Displays information about field replaceable units (FRUs). |
| **show memory** | Displays information about system memory. |
| **show pal** | Displays the versions of *Tru64 UNIX* and *OpenVMS* PALcode. |
| **show power** | Displays information about system environmental characteristics, including power supplies, system fans, CPU fans, and temperature. |
| **show_status** | Displays the progress of diagnostic tests. Reports one line of information for each executing diagnostic. |
| **show version** | Displays the version of the SRM console program installed on the system. |
| **sys_exer** | Exercises the devices displayed with the show config command |
| **sys_exer -lb** | Runs console loopback tests for the COM2 serial port and the parallel port during the **sys_exer** test sequence. |
| **test** | Verifies the configuration of the devices in the system. |
| **test -lb** | Runs loopback tests for the COM2 serial port and the parallel port in addition to verifying the configuration of devices. |

# Appendix B
# Jumpers and Switches

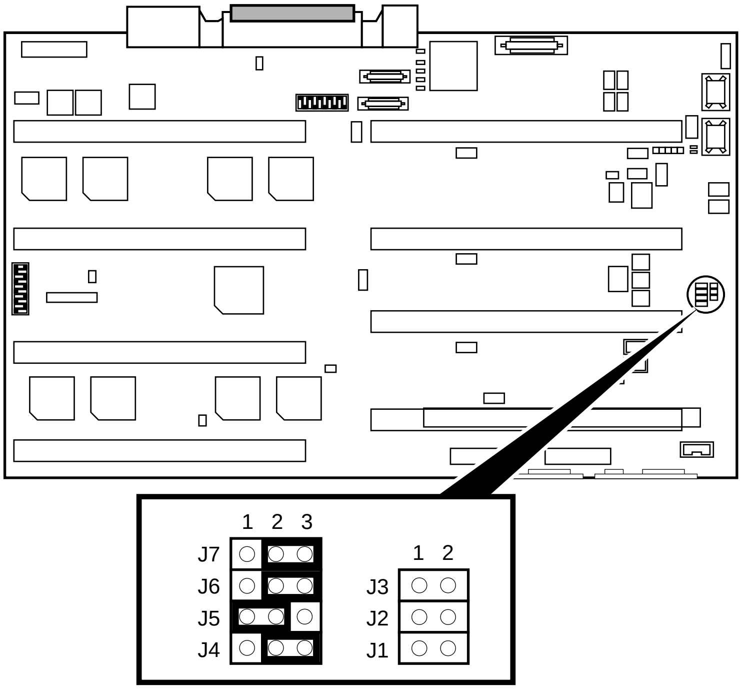
This chapter lists and describes the configuration jumpers and switches on the system motherboard and PCI board. Sections are as follows:

- RMC and SPC Jumpers on System Motherboard
- TIG/SROM Jumpers on System Motherboard
- Clock Generator Switch Settings
- Jumper on PCI Board
- Setting Jumpers

## B.1 RMC and SPC Jumpers on System Motherboard

**The RMC jumpers can be used to override the RMC defaults. For example, if a high-speed modem is connected to COM1, you can disable J4 to prevent RMC from receiving characters that might cause interference. The SPC jumpers are reserved.**

Figure B–1 RMC and SPC Jumpers



PK0211A

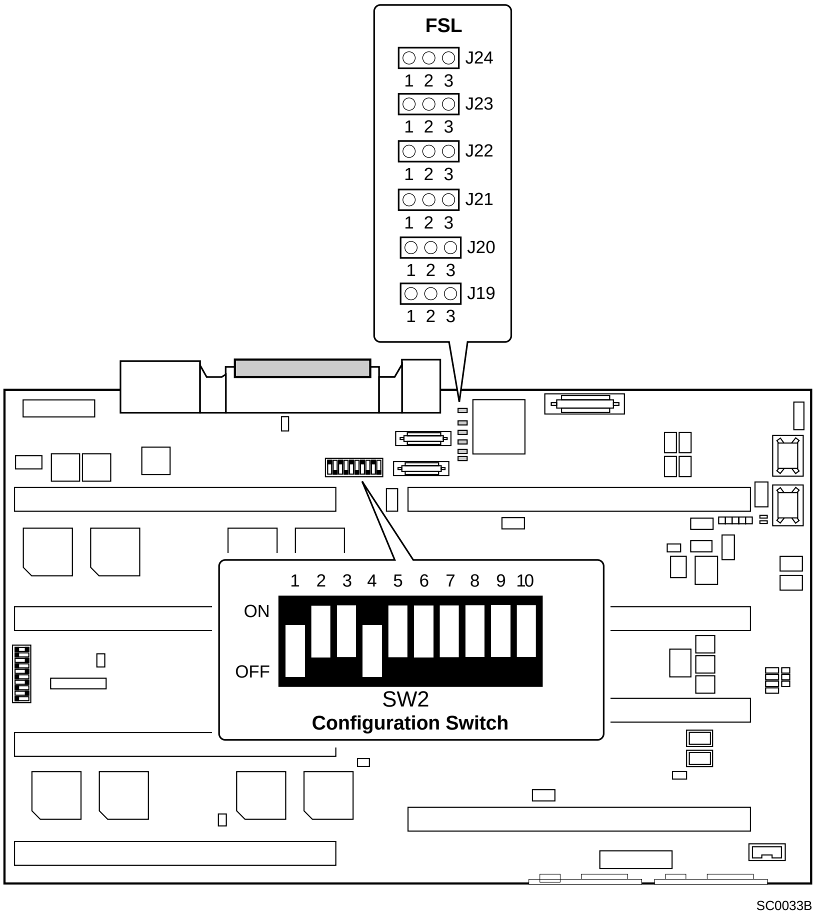
### Table B–1 RMC/SPC Jumper Settings

| Jumper | Description |
|---|---|
| J7 | 1–2: Disables RMC flash update<br>2–3: Enables RMC flash update (default)<br>Disabling RMC flash update prevents other operators from erasing or updating the RMC. |
| J6 | 1–2: Sets RMC back to defaults<br>2–3: Normal RMC operating mode (default)<br>If the RMC escape sequence is set to something other than the default, and you have forgotten the sequence, RMC must be reset to factory settings to restore the default escape sequence. See Chapter 8 for the reset procedure. |
| J5 | 1–2: Causes system to shut down if overtemperature limit is reached (default)<br>2–3: Permits system to continue running at overtemperature. |
| J4 | 1–2: Disables COM1 bypass<br>2–3: Allows RMC to control COM1 bypass (default)<br>No jumper installed: Forces COM1 bypass |
| J1 | Not installed (default). When installed, bypasses power-up checks of processors by system power controller. |
| J2 | Reserved (not installed). |
| J3 | Reserved (not installed). |

## B.2 TIG/FSL Jumpers on System Motherboard

**TIG/SROM jumpers allow you to load the TIG if flash RAM is corrupted or load the fail-safe loader (FSL) if SRM firmware is corrupted.**

### Figure B–2 TIG/FSL Jumpers



**NOTE:** *See Chapter 3 for instructions on activating the FSL.*

### Table B–2 TIG/FSL Jumper Descriptions

| Jumper | Description |
|---|---|
| J24 | 1–2: Load TIG from flash ROM (default)<br>2–3: Load TIG from serial ROM. This setting allows you to load the TIG if the flash ROM is corrupted. |
| J21 | Jumper for enabling fail-safe loader (FSL)<br>FIR_FUNC1 (bit 1)<br>1–2= 0, 2–3= 1 |
| J22 | Must be in default positions over pins 1 and 2 to enable FSL.<br>1–2 = 0, 2–3 = 1 |
| J23 | Must be in default positions over pins 1 and 2 to enable FSL.<br>FIR_FUNC2 (bit 2)<br>1–2= 0, 2–3 = 1 |
| J20 | Allows writes to flash ROM (normal).<br>1 – 2= 1, 2 – 3= 0 |
| J19 | Allows writes to flash ROM (normal).<br>1 – 2= 1, 2 – 3= 0 |

### Table B–3 Firmware Function Table (FIR_FUNC)

| Bits 210 | Meaning |
|---|---|
| 000 | Normal |
| 001 | Prevent flash loads. Load from SROM. |
| 010 | Load from floppy |
| 111 | Lock console. Prevents the writing of flash from CPUs. |

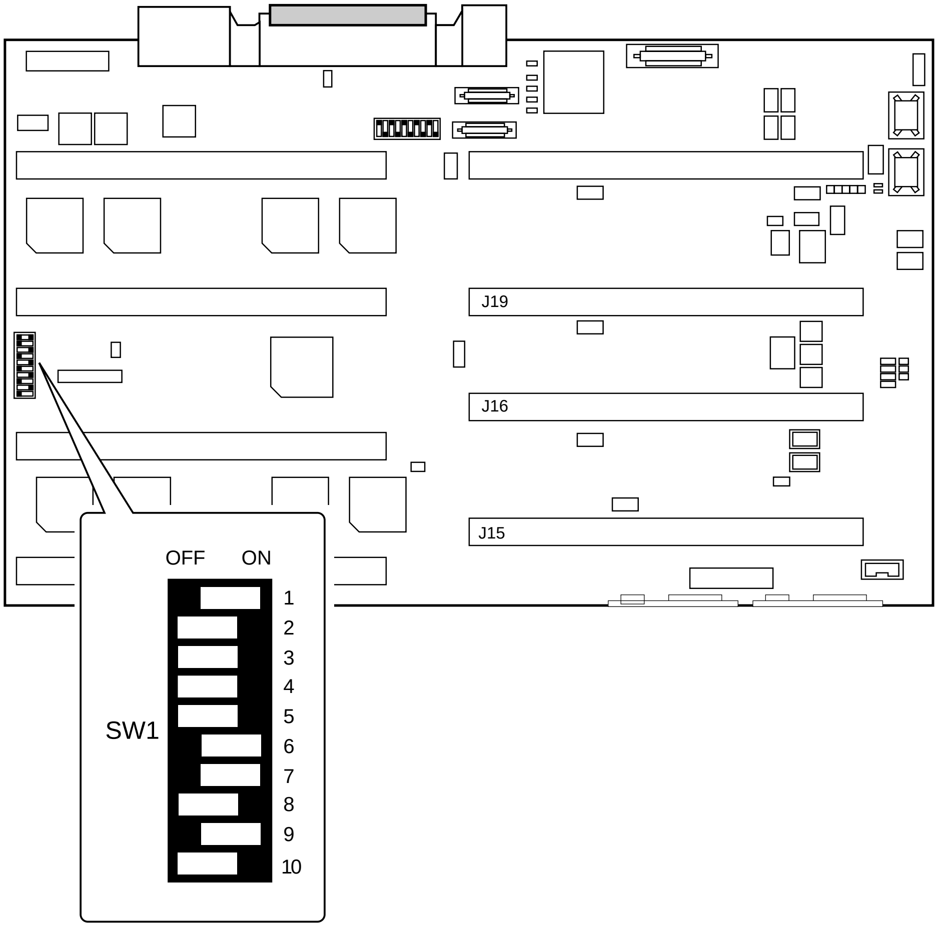
The configuration switchpack (SW2) sets the clock speed for the system motherboard. The settings should not be changed.

| | |
|---|---|
| SW1 | SYS_EXT_DELAY1 (off) |
| SW2 | SYS_EXT_DELAY0 (on) |
| SW3 | SYS_FILL_DELAY (on) |
| SW4 | CPU_CFWD_PSET (off) |
| SW5 | Reserved |
| SW6 | Reserved |
| SW7 | Y_DIV3 (on) |
| SW8 | Y_DIV2 (on) |
| SW9 | Y_DIV1 (on) |
| SW10 | Y_DIV0 (on) |

## B.3 Clock Generator Switch Settings

**Switchpack E16 on the system motherboard sets the frequency of the main clock on the system motherboard. The settings should not be changed.**

### Figure B–3 CSB Switchpack E16



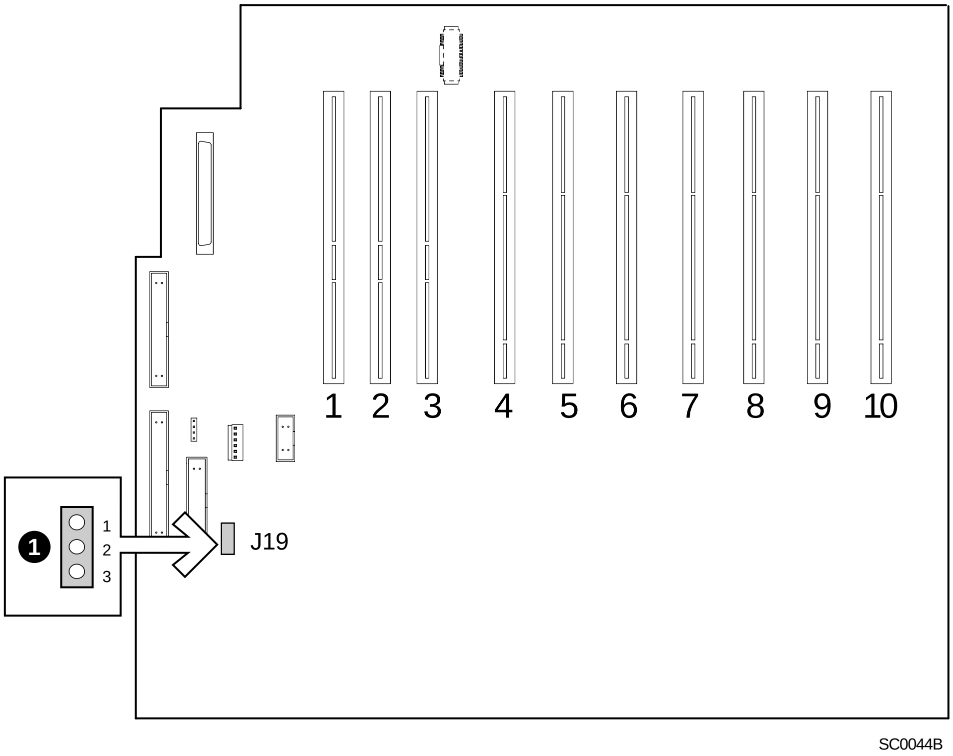
### Table B–4 Clock Generator Settings

| | |
|---|---|
| SW1 | M0 (on) |
| SW2 | M1 (off) |
| SW3 | M2 (off) |
| SW4 | M3 (off) |
| SW5 | M4 (off) |
| SW6 | M5 (on) |
| SW7 | M6 (on) |
| SW8 | N0 (off) |
| SW9 | N1 (on) |
| SW10 | XTAL_SEL (off) |

## B.4 Jumper on PCI Board

**You can set J19 on the PCI board to force DTR so that a modem will not be disconnected if the system is power cycled.**

### Figure B–4 PCI Board Jumper



### Table B–5 PCI Board Jumper Description

| | Jumper | Description |
|---|---|---|
| ❶ | J19 | 1–2: Do not force COM1 DTR<br>2–3: Force COM1 DTR (default)<br>This jumper allows you to force DTR. The default position prevents disconnection of the modem on a power cycle. |

## B.5 Setting Jumpers

**Review the material in the previous sections of this appendix before setting any system jumpers. First, shut down the system and remove the power cord from each power supply.**

**CAUTION:** *Static electricity can damage integrated circuits. Always use a grounded wrist strap (29-26246) and grounded work surface when working with internal parts of a computer system.*

*Remove jewelry before working on internal parts of the system.*

### Setting Jumpers

1. Shut down the operating system.
2. Shut down power on all external options connected to the system.
3. Turn off power to the system.
4. Unplug the power cord from each power supply and wait for all LEDs to turn off.
5. Remove enclosure panels and chassis covers to gain access to the system motherboard or PCI board.
   - If you are setting RMC jumpers, remove CPU 1 to gain access to the jumpers.
   - If you are setting TIG/SROM jumpers, remove MMB 1 to gain access to the jumpers.
   - If you are setting PCI jumpers, you typically do not need to remove any PCI cards. However, if you have a full-length card in slot 10, remove it.
6. Locate the jumper you need to set. Refer to the illustrations in this appendix. Set the jumpers as needed.
7. Reinstall any modules you removed.
8. Reinstall the chassis covers and enclosure panels.
9. Plug the power cords into the supplies.

# Appendix C
# DPR Address Layout

This appendix shows the address layout of the dual-port RAM (DPR). Use the SRM **examine dpr:*address*** command (where *address* is the offset from the base of the DPR) or use the RMC **dump** command to view locations in the DPR. See Appendix D for definitions of locations written when environmental error events occur.

### Table C–1 DPR Address Layout

| Location (Hex) | Logical Indicator | Written By | Used For |
|---|---|---|---|
| 0 | 0 | SROM | EV6 BIST status 1=good 0=bad |
| 1 | 1 | SROM | Bit[7]=Master Bits[0,1]=CPU_ID |
| 2 | 2 | SROM | Test STR status 1=good 0=bad |
| 3 | 3 | SROM | Test CSC status 1=good 0=bad |
| 4 | 4 | SROM | Test Pchip 0 PCTL status 1=good 0=bad |
| 5 | 5 | SROM | Test Pchip 1 PCTL status 1=good 0=bad |
| 6 | 6 | SROM | Test DIMx status 1=good 0=bad |
| 7 | 7 | SROM | Test TIG bus status |
| 8 | 8 | SROM | Dual-Port RAM test DD= started |
| 9 | 9 | SROM | Status of DPR test 1=good 0=bad |
| A | A | SROM | Status of CPU speed function FF=good 0=bad |
| B | B | SROM | Lower byte of CPU speed in MHz |
| C | C | SROM | Upper byte of CPU speed in MHz |
| D:F | - | - | Reserved |
| 10:15 | | SROM | Power On Time Stamp for CPU 0—written as BCD<br>Byte 10 = Hours (0-23)<br>Byte 11 = Minutes (0-59)<br>Byte 12 = Seconds (0-59)<br>Byte 13 = Day of Month (1-31)<br>Byte 14 = Month (1-12)<br>Byte 15 = Year (0-99) |

## Table C–1 DPR Address Layout (Continued)

| Location (Hex) | Logical Indicator | Written By | Used For |
|---|---|---|---|
| 16 | | SROM | SROM Power On Error Indication for CPU is “alive.” For example; 0 = no error, 2 = Secondary time-out Error, 3 = Bcache Error |
| 17:1D | | | Unused |
| 1E | | SROM | Last “sync state” reached; 80=Finished GOOD |
| 1F | | SROM | Size of Bcache in MB |
| 20:3F | 20 | | Repeat for CPU1 of CPU0 0-1F |
| 40:5F | 20 | | Repeat for CPU2 of CPU0 0-1F |
| 60:7F | 20 | | Repeat for CPU3 of CPU0 0-1F |
| 80 | 80 | SROM | Array 0 (AAR 0) Configuration<br>Bits<7:4><br>4 = non split - lower set only<br>5 = split - lower set only<br>9 = split - upper set only<br>D = split - 8 DIMMs<br>F = Twice split - 8 DIMMs<br>Bits<3:0><br>0 = Configured - Lowest array<br>1 = Configured - Next lowest array<br>2 = Configured - Second highest array<br>3 = Configured - Highest array<br>4 = Misconfigured - Missing DIMM(s)<br>8 = Misconfigured - Illegal DIMM(s)<br>C = Misconfigured - Incompatible DIMM(s) |

*Continued on next page*

### Table C–1 DPR Address Layout (Continued)

| Location (Hex) | Logical Indicator | Written By | Used For |
|---|---|---|---|
| 81 | 81 | SROM | Array 0 (AAR 0)Size (x64 Mbytes)<br>0 = no good memory<br>1 = 64 Mbyte<br>2 = 128 Mbyte<br>4 = 256 Mbyte<br>8 = 512 Mbyte<br>10 = 1 Gbyte<br>20 = 2 Gbyte<br>40 = 4 Gbyte<br>80 = 8 Gbyte |
| 82 | 82 | SROM | Array 1 (AAR 1) Configuration |
| 83 | 83 | SROM | Array 1 (AAR 1) Size (x64 Mbytes) |
| 84 | 84 | SROM | Array 2 (AAR 2) Configuration |
| 85 | 85 | SROM | Array 2 (AAR 2) Size (x64 Mbytes) |
| 86 | 86 | SROM | Array 3 (AAR 3) Configuration |
| 87 | 87 | SROM | Array 3 (AAR 3) Size (x64 Mbytes) |
| 88:8B | | SROM | Byte to define failed DIMMs for MMBs<br>88 - MMB 0<br>89 - MMB 1<br>8A - MMB 2<br>8B - MMB 3<br>Bit set indicates failure.<br>Bit definitions ( bit 0 = DIMM 1, bit 1 = DIMM2, bit 2 = DIMM 3, bit 7 = DIMM 8) |
| 8C:8F | 8C-8F | SROM | Byte to define misconfigured DIMMs for MMBs<br>8C – MMB 0<br>8D – MMB 1<br>8E – MMB 2<br>8F – MMB 3<br>Bit definitions ( bit 0 = DIMM 1, bit 1 = DIMM2, bit 2 = DIMM 3, bit 7 = DIMM 8) |
| 90 | 90 | RMC | Power Supply/VTERM present |
| 91 | 91 | RMC | Power Supply PS_POK bits |
| 92 | 92 | RMC | AC input value from Power Supply |

## Table C–1 DPR Address Layout (Continued)

| Location | Logical | Written | |
|---|---|---|---|
| (Hex) | Indicator | By | Used For |
| 93:96 | 93 | RMC | Temperature from CPU(x) in BCD |
| 97:99 | 97 | RMC | Temperature Zone(x) from 3 PCI temp sensors |
| 9A:9F | 9A | RMC | Fan Status; Raw Fan speed value |
| A0:A9 | A0 | RMC | Failure registers used as part of the 680 machine check logout frame. See Appendix D. |
| AA | | RMC | Fan status (bit 0 = fan 1, bit 1 = fan 2,<br>1- indicates good; 0 indicates fan failure |
| AB | | RMC | Status of RMC to read I$^2$C bus of MMB0 DIMMs<br>Definition:<br>Bit 7 - DIMM 8 0=OK 1=Fail<br>Bit 6 - DIMM 7<br>Bit 5 - DIMM 6<br>……………..<br>Bit 0 - DIMM 1 |
| AC | | RMC | Status of RMC to read I$^2$C bus of MMB1 DIMMs |
| AD | | RMC | Status of RMC to read I$^2$C bus of MMB2 DIMMs |
| AE | | RMC | Status of RMC to read I$^2$C bus of MMB3 DIMMs |
| AF | | RMC | Status of RMC to read MMB and CPU I$^2$C buses<br>Definition:<br>Bit 7 - MMB3 0=OK 1=Fail<br>Bit 6 - MMB2<br>Bit 5 - MMB1<br>Bit 4 - MMB0<br>Bit 3 - CPU3<br>Bit 2 - CPU2<br>Bit 1 - CPU1<br>Bit 0 - CPU0 |
| B0 | | RMC | Status of RMC to read CPB (PCI backplane) I$^2$C EEROM0=OK 1 = fail |
| B1 | | RMC | Status of RMC to read CSB (motherboard) I$^2$C EEROM<br>0=OK 1 = fail |

*Continued on next page*

## Table C–1 DPR Address Layout (Continued)

| Location (Hex) | Logical Indicator | Written By | Used For |
|---|---|---|---|
| B2 | | RMC | Status of RMC to read SCSI backplane<br>Definition:<br>Bit 0 — SCSI backplane 0<br>Bit 1 — SCSI backplane 1<br>Bit 4 — Power supply 0<br>Bit 5 — Power supply 1<br>Bit 6 — Power supply 2 |
| B3:B9 | | Unused | Unused |
| BA | | RMC | I$^2$C done, BA = finished |
| BB | | RMC | RMC Power on Error indicates error during power-up (1=Flash Corrupted) |
| BC | | RMC | RMC flash update error status |
| BD | | RMC | Copy of PS input Value. See Appendix D. |
| BE | | RMC | Copy of the byte from the I/O expanders on the SPC loaded by the RMC on fatal errors. See Appendix D. |
| BF | | RMC | Reason for system failure. See Appendix D. |
| C0 | | | Reason for system failure. |
| C1:D8 | | | Unused |
| D9 | | RMC | Baud rate |
| DA | | TIG | Indicates TIG finished loading its code (0xAA indicates done) |
| DB:E3 | | RMC | Fan/Temp info from PS1 |
| E4:EC | | RMC | Fan/Temp info from PS2 |
| ED:F5 | | RMC | Fan/Temp info from PS3 |
| F6:F8 | | Unused | Unused |
| F9 | | Firmware | Buffer Size (0-0xFF) or 1 to 256 bytes |
| FA:FB | FA | Firmware | Command address qualifier<br>FA = lower byte, FB = upper byte |

## Table C–1 DPR Address Layout (Continued)

| Location (Hex) | Logical Indicator | Written By | Used For |
|---|---|---|---|
| FC | FC | RMC | Command status associated with the RMC response to a request from the firmware<br>0 = successful completion<br>80 = unsuccessful completion<br>81 = invalid command code<br>82 = invalid command qualifier |
| FD | FD | RMC | Command ID associated with the RMC response to a request from the firmware |
| FE | FE | Firmware | Command Code associated with a “command” sent to the RMC<br>1 = update I$^2$C EEROM<br>2 = update baud rate<br>3 = display to OCP<br>F0 = update RMC flash |
| FF | FF | Firmware | Command ID associated with a “command” sent to the RMC |
| 100:1FF | 100 | RMC | Copy of EEROM on MMB0 J4 DIMM 1, initially read on I$^2$C bus by RMC when 5 volts supply turned on. Written by Compaq Analyze after error diagnosed to particular FRU |
| 200:2FF | 200 | RMC | Copy of EEROM on MMB0 J8 |
| 300:3FF | 300 | RMC | Copy of EEROM on MMB0 J5 |
| 400:4FF | 400 | RMC | Copy of EEROM on MMB0 J9 |
| 500:5FF | 500 | RMC | Copy of EEROM on MMB0 J2 |
| 600:7FF | 600 | RMC | Copy of EEROM on MMB0 J6 |
| 700:7FF | 700 | RMC | Copy of EEROM on MMB0 J3 |
| 800:8FF | 800 | RMC | Copy of EEROM on MMB0 J7 |
| 900:9FF | 900 | RMC | Copy of EEROM on MMB1 J4 |
| A00:AFF | A00 | RMC | Copy of EEROM on MMB1 J8 |
| B00:BFF | B00 | RMC | Copy of EEROM on MMB1 J5 |
| C00:CFF | C00 | RMC | Copy of EEROM on MMB1 J9 |
| D00:DFF | D00 | RMC | Copy of EEROM on MMB1 J2 |
| E00:EFF | E00 | RMC | Copy of EEROM on MMB1 J6 |
| F00:FFF | F00 | RMC | Copy of EEROM on MMB1 J3 |

*Continued on next page*

### Table C–1 DPR Address Layout (Continued)

| Location (Hex) | Logical Indicator | Written By | Used For |
|---|---|---|---|
| 1000:10FF | 1000 | RMC | Copy of EEROM on MMB1 J7 |
| 1100:11FF | 1100 | RMC | Copy of EEROM on MMB2 J4 |
| 1200:12FF | 1200 | RMC | Copy of EEROM on MMB2 J8 |
| 1300:13FF | 1300 | RMC | Copy of EEROM on MMB2 J5 |
| 1400:14FF | 1400 | RMC | Copy of EEROM on MMB2 J9 |
| 1500:15FF | 1500 | RMC | Copy of EEROM on MMB2 J2 |
| 1600:16FF | 1600 | RMC | Copy of EEROM on MMB2 J6 |
| 1700:17FF | 1700 | RMC | Copy of EEROM on MMB2 J3 |
| 1800:18FF | 1800 | RMC | Copy of EEROM on MMB2 J7 |
| 1900:19FF | 1900 | RMC | Copy of EEROM on MMB3 J4 |
| 1A00:1AFF | 1A00 | RMC | Copy of EEROM on MMB3 J8 |
| 1B00:1BFF | 1B00 | RMC | Copy of EEROM on MMB3 J5 |
| 1C00:1CFF | 1C00 | RMC | Copy of EEROM on MMB3 J9 |
| 1D00:1DFF | 1D00 | RMC | Copy of EEROM on MMB3 J2 |
| 1E00:1EFF | 1E00 | RMC | Copy of EEROM on MMB3 J6 |
| 1F00:1FFF | 1F00 | RMC | Copy of EEROM on MMB3 J3 |
| 2000:20FF | 2000 | RMC | Copy of EEROM on MMB3 J7 |
| 2100:21FF | 2100 | RMC | Copy of EEROM from CPU0 |
| 2200:22FF | 2200 | RMC | Copy of EEROM from CPU1 |
| 2300:23FF | 2300 | RMC | Copy of EEROM from CPU2 |
| 2400:24FF | 2400 | RMC | Copy of EEROM from CPU3 |
| 2500:25FF | 2500 | RMC | Copy of MMB 0 J5 FRU EEROM |
| 2600:26FF | 2600 | RMC | Copy of MMB 1 J7 FRU EEROM |
| 2700:27FF | 2700 | RMC | Copy of MMB 2 J6 FRU EEROM |
| 2800:28FF | 2800 | RMC | Copy of MMB 3 J8 FRU EEROM |
| 2900:29FF | 2900 | RMC | Copy of EEROM on CPB (PCI backplane) |
| 2A00:2AFF | 2A00 | RMC | Copy of EEROM on CSB (motherboard) |
| 2B00:2BFF | 2B00 | RMC | Last EV68 Correctable Error—ASCII character string that indicates correctable error occurred, type, FRU, and so on. Backed up in CSB (motherboard) EEROM. Written by Compaq Analyze |

### Table C–1 DPR Address Layout (Continued)

| Location (Hex) | Logical Indicator | Written By | Used For |
|---|---|---|---|
| 2C00:2CFF | 2C00 | RMC | Last Redundant Failure—ASCII character string that indicates redundant failure occurred, type, FRU, and so on. Backed up in system CSB (motherboard) EEROM. Written by Compaq Analyze |
| 2D00:2DFF | 2D00 | RMC | Last System Failure—ASCII character string that indicates system failure occurred, type, FRU, and so on. Backed up in CSB (motherboard) EEROM. Written by Compaq Analyze. |
| 2E00:2FFF | 2E00 | RMC | Uncorrectable machine logout frame (512 bytes) |
| 3000:3008 | | SROM | SROM Version (ASCII string) |
| 3009:300B | | RMC | Rev Level of RMC first byte is letter Rev [x/t/v] second 2 bytes are major/minor. This is the rev level of the RMC on-chip code. |
| 300C:300E | | RMC | Rev Level of RMC first byte is letter Rev [x/t/v] second 2 bytes are major/minor. This is the rev level of the RMC flash code. |
| 300F:3010 | 300F | RMC | Revision Field of the DPR Structure |
| 3011:30FF | | Unused | Unused |
| 3100:31FF | | RMC | Copy of PS0 EEROM (first 256 bytes) |
| 3200:32FF | | RMC | Copy of PS1 EEROM (first 256 bytes) |
| 3300:33FF | | RMC | Copy of PS2 EEROM (first 256 bytes) |
| 3400 | | SROM | Size of Bcache in MB |
| 3401 | | SROM | Flash SROM is valid flag; 8 = valid, 0 = invalid |
| 3402 | | SROM | System's errors determined by SROM |
| 3403:340F | | SROM/SRM | Reserved for future SROM/SRM communication |
| 3410:3417 | | SROM/SRM | Jump to address for CPU0 |

*Continued on next page*

## Table C–1 DPR Address Layout (Continued)

| Location (Hex) | Logical Indicator | Written By | Used For |
|---|---|---|---|
| 3418 | | SROM/SRM | Waiting to jump to flag for CPU0 |
| 3419 | | SROM | Shadow of value written to EV6 DC_CTL register. |
| 341A:341E | | SROM | Shadow of most recent writes to EV6 CBOX “Write-many” chain. |
| 341F | | SROM/SRM | Reserved for future SROM/SRM communication |
| 3420:342F | | SROM/SRM | Repeat for CPU1 of CPU0 3410-341F |
| 3430:343F | | SROM/SRM | Repeat for CPU2 of CPU0 3410-341F |
| 3440:344F | | SROM/SRM | Repeat for CPU3 of CPU0 3410-341F |
| 3450:349F | | SROM/ RMC | Reserved for SROM mini-console via RMC communication area. Future design. |
| 34A0:34A7 | | SROM | Array 0 to DIMM ID translation<br>Bits<7:5><br>0 = Exists, No Error<br>1 = Expected Missing<br>2 = Error - Missing DIMM(s)<br>4 = Error - Illegal DIMM(s)<br>6 = Error - Incompatible DIMM(s)<br>Bits<4:0><br>Bits <2:0> = DIMM + 1 (1-8)<br>Bits <4:3> = MMB (0-3) |
| 34A8:34AF | | SROM | Repeat for Array 1 of Array 0 34A0:34A7 |
| 34B0:34B7 | | SROM | Repeat for Array 2 of Array 0 34A0:34A7 |
| 34B8:34CF | | SROM | Repeat for Array 3 of Array 0 34A0:34A7 |
| 34C0:34FF | 34C0 | SROM | Used as scratch area for SROM |

### Table C–1 DPR Address Layout (Continued)

| Location (Hex) | Logical Indicator | Written By | Used For |
|---|---|---|---|
| 3500:35FF | | Firmware | Used as the dedicated buffer in which SRM writes OCP or FRU EEROM data. Firmware will write this data, RMC will only read this data. |
| 3600:36FF | 3600 | SRM | Reserved |
| 3700:37FF | | SRM | Reserved |
| 3800:3AFF | | RMC | RMC scratch space |
| 3B00:3BFF | | RMC | First SCSI backplane EEROM |
| 3C00:3CFF | | RMC | Second SCSI backplane EEROM |
| 3D00:3DFF | | RMC | PS0 second 256 bytes |
| 3E00:3EFF | | RMC | PS1 second 256 bytes |
| 3F00:3FFF | | RMC | PS2 second 256 bytes |

# Appendix D
# Registers

This appendix describes 21264 (EV68) internal processor registers; 21274 (Titan) system support chipset registers; and dual-port RAM (DPR) registers that are related to general logout frame errors. It also provides CPU and system uncorrectable and correctable machine logout frames and error state bit definitions of all the platform logout frame registers.

### 21264 (EV68) Registers

Ibox Status Register (I_STAT)
Memory Management Status Register (MM_STAT)
Dcache Status Register (DC_STAT)
Cbox Read Register
Exception Address Register (EXC_ADDR)
Interrupt Enable and Current Processor Mode Register (IER_CM)
Interrupt Summary Register (ISUM)
PAL Base Register (PAL_BASE)
Ibox Control Register (I_CTL)
Process Context Register (PCTX)

### 21274 (Titan) System Registers

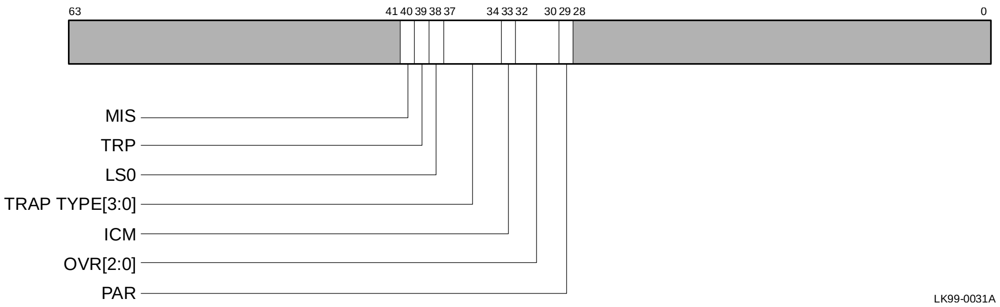
21274 Cchip Miscellaneous Register (MISC)
21274 Device Interrupt Request Register (DIR*n*, *n*=0,1,2,3)
21274 Pchip Error Register (PERROR)
21274 Pchip System Error Register (SERROR)
21274 Array Address Registers

### DPR Registers

DPR Registers (for 680 correctable error state capture)
DPR Registers (for $I^2C$ bus)
DPR Registers (power supply status from $I^2C$ bus)
DPR 680 Fatal Registers (for 680 uncorrectable error state capture)

## D.1 Ibox Status Register (I_STAT)

**The Ibox Status Register (I_STAT) is read/write-1-to-clear register that contains Ibox status information.**



### Table D–1 Ibox Status Register Fields

| Name | Bits | Type | Description |
|---|---|---|---|
| Reserved | <63:41> | RO | Reserved for Compaq. |
| MIS | <40> | RO | ROProfileMeMispredictTrap.<br>If the I_STAT<TRP> bit is set, this bit indicates that the profiled instruction caused a mispredict trap. JSR/JMP/RET/COR or HW_JSR/<br>HW_JMP/HW_RET/HW_COR mispredicts do not set this bit but can be recognized by the presence of one of these instructions at the PMPC location with the I_STAT<TRP> bit set. This identification is exact in all cases except error condition traps. Hardware corrected Icache parity or Dcache ECC errors, and machine check traps can occur on any instruction in the pipeline. |
| TRP | <39> | RO | ProfileMe Trap.<br>This bit indicates that the profiled instruction caused a trap. The trap type field, PMPC register, and instruction at the PMPC location are needed to distinguish all trap types. |

### Table D–1 Ibox Status Register Fields (Continued)

| Name | Bits | Type | Description |
|---|---|---|---|
| LS0 | <38> | RO | ProfileMe Load-Store Order Trap.<br>LS0 <38> RO ProfileMe Load-Store Order Trap.<br>If the profiled instruction caused a replay trap, this bit indicates that the precise trap cause was an Mbox load-store order replay trap. If clear, this bit indicates that the replay trap was any one of the following:<br>Mbox load-load order<br>Mbox load queue full<br>Mbox store queue full<br>Mbox wrong size trap (such as, STL • •LDQ)<br>Mbox Bcache alias (2 physical addresses map to same Bcache line)<br>Mbox Dcache alias (2 physical addresses map to same Dcache line)<br>Icache parity error<br>Dcache ECC error |
| TRAP TYPE<3:0> | <37:34> | RO | RO ProfileMe Trap Types.<br>If the profiled instruction caused a trap (indicated by I_STAT<TRP>), this field indicates the trap type as listed here:<br>**Value Trap Type**<br>0 Replay<br>1 Invalid (unused)<br>2 DTB Double miss (3 level page tables)<br>3 DTB Double miss (4 level page tables)<br>4 Floating point disabled<br>5 Unaligned Load/Store<br>6 DTB Single miss<br>7 Dstream Fault<br>8 OPCDEC<br>9 Invalid (use PMPC, described below)<br>10 Machine Check<br>11 Invalid (use PMPC, described below)<br>12 Arithmetic<br>13 Invalid (use PMPC, described below)<br>14 MT_FPCR<br>15 Reset<br>Traps due to ITB miss, Istream access violation, or interrupts are not reported in the trap type field because they do not cause pipeline aborts. |

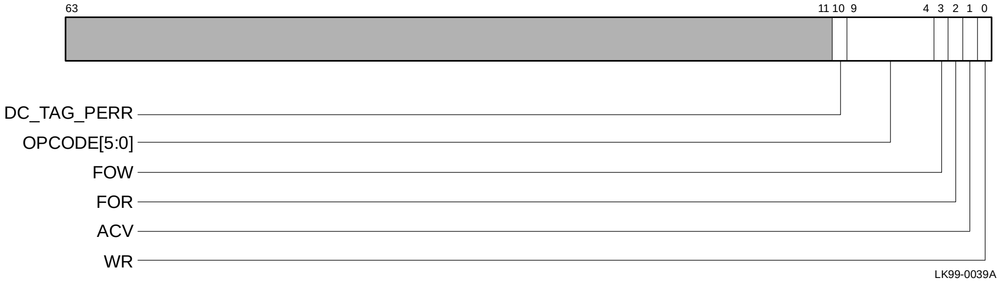
*Continued on next page*

## Table D–1 Ibox Status Register Fields (Continued)

| Name | Bits | Type | Description |
|---|---|---|---|
| | | | Instead, these traps cause pipeline redirection and can be distinguished by examining the PMPC value for the presence of the corresponding PAL-code entry offset addresses indicated below. In these cases, the ProfileMe interrupt will normally be delivered when exiting the trap PALcode flow and the EXC_ADDR register will contain the original PC that encountered the redirect trap.<br>**PMPC<14:0>** **Trap**<br>0581 ITB miss<br>0481 Istream Access Violation<br>0681 Interrupt |
| ICM | <33> | RO | ProfileMe Icache Miss.<br>This bit indicates that the profiled instruction was contained in an aligned 4-instruction Icache fetch block that requested a new Icache fill stream. |
| OVR<2:0> | <32:30> | RO | ProfileMe Counter 0 Overcount.<br>This bit indicates a value (0-7) that must be subtracted from the counter 0 result to obtain an accurate count of the number of instructions retired in the interval beginning three cycles after the profiled instruction reaches pipeline stage 2 and ending four cycles after the profiled instruction is retired. |
| PAR | <29> | WIC | Icache Parity Error.<br>This bit indicates that the Icache encountered a parity error on instruction fetch. When a parity error is detected, the Icache is flushed, a replay trap back to the address of the error instruction is generated, and a correctable read interrupt is requested. See also I_STAT<LAM>. |
| Reserved | <28:0> | RO | Reserved for COMPAQ |

## D.2 Memory Management Status Register (MM_STAT)

**The Memory Management Status Register (MM_STAT) is a read-only register. When a Dstream TB miss or fault occurs, information about the error latched in MM_STAT.MM_STAT is not updated when a LD_VPTE gets a DTB miss instruction.**

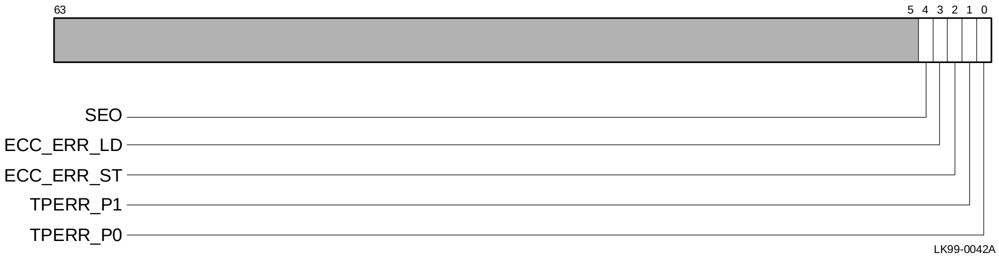


### Table D–2 Memory Management Status Register Fields

| Name | Bits | Type | Description |
|---|---|---|---|
| Reserved | <63:11> | | Reserved for Compaq. |
| DC_TAG_ PERR | <10> | RO | This bit is set when a D-cache tag parity error occurs during the initial tag probe of a load or store instruction. The error created a synchronous fault to the D_FAULT PALcode entry point and is correctable. The virtual address associated with the error is available in the VA register. |
| OPCODE | <9:4> | RO | Opcode of the instruction that caused the error. HW_LD is displayed as 3 and HW_ST is displayed as 7. |
| FOW | <3> | RO | Set when a fault-on-write error occurs during a write transaction and PTE<FOW> was set. |
| FOR | <2> | RO | Set when a fault-on-read error occurs during a read transaction and PTE<FOR> was set. |
| ACV | <1> | RO | Set when an access violation occurs during a transaction. Access violations include a bad virtual address. |
| WR | <0> | RO | Set when an error occurs during a write transaction. |

## D.3 Dcache Status Register (DC_STAT)

**The Dcache Status Register (DC_STAT) is a read-write register. If a Dcache tag parity error or data ECC error occurs, information about the error is latched in this register.**



### Table D–3 Dcache Status Register Fields

| Name | Bits | Type | Description |
|---|---|---|---|
| Reserved | <63:5> | | Reserved for Compaq. |
| SEO | <4> | W1C | Second error occurred. When set, indicates that a second D-cache store ECC error occurred within 6 cycles of the previous D-cache store ECC error. |
| ECC_ERR_LD | <3> | W1C | ECC error on load. When set, indicates that a single-bit ECC error occurred while processing a load from the D-cache or any fill. |
| ECC_ERR_ST | <2> | W1C | ECC error on store. When set, indicates that an ECC error occurred while processing a store. |
| TPERR_P1 | <1> | W1C | Tag parity error—pipe 1. When set, indicates that a D-cache tag probe from pipe 1 resulted in a tag parity error. The error is uncorrectable and results in a machine check. |
| TPERR_P0 | <0> | W1C | Tag parity error—pipe 0. When set, this bit indicates that a D-cache tag probe from pipe 1 resulted in a tag parity error. The error is uncorrectable and results in a machine check. |

## D.4 Cbox Read Register

**The Cbox Read Register is read only by PAL code and is an element in the CPU or system uncorrectable and correctable machine check error logout frame.**

### Table D–4 Cbox Read Register Fields

| Name | Description | |
|---|---|---|
| C_SYNDROME_1<7:0> | Syndrome for the upper QW in the OW of victim that was scrubbed. See Appendix E. | |
| C_SYNDROME_0<7:0> | Syndrome for the lower QW in the OW of victim that was scrubbed. See Appendix E. | |
| C_STAT<4:0> | **Bits** | **Error Status** |
| | 00000 | Either no error, or error on a speculative load, of a B-cache victim read due to a D-cache/B-cache miss. |
| | 00001 | BC_PERR (B-cache tag parity error) |
| | 00010 | DC_PERR (duplicate tag parity error) |
| | 00011 | DSTREAM_MEM_ERR |
| | 00100 | DSTREAM_BC_ERR |
| | 00101 | DSTREAM_DC_ERR |
| | 0011X | PROBE_BC_ERR |
| | 01000 | Reserved |
| | 01001 | Reserved |
| | 01010 | Reserved |
| | 01011 | ISTREAM_MEM_ERR |

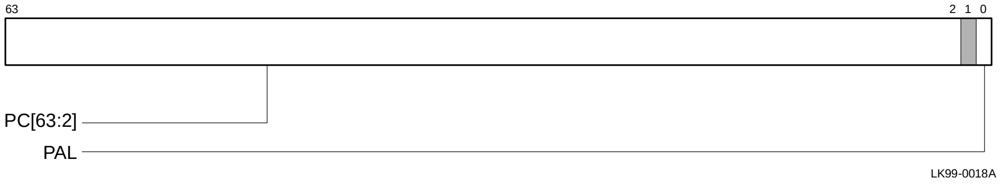
*Continued on next page*

### Table D–4 Cbox Read Register Fields (Continued)

| Name | Description | |
|---|---|---|
| C_STAT<4:0><br>(continued) | **Bits** | **Error Status** |
| | 01100 | ISTREAM_BC_ERR |
| | 01101 | Reserved |
| | 0111X | Reserved |
| | 10011 | DSTREAM_MEM_DBL |
| | 10100 | DSTREAM_BC_DBL |
| | 11011 | ISTREAM_MEM_DBL |
| | 11100 | ISTREAM_BC_DBL |
| C_STS<3:0> | If C_STAT equals *xxx*_MEM_ERR or *xxx*_BC_ERR, then C_STAT contains the status of the block as follows; otherwise, the value of C_STAT is X. | |
| | **Bit Value** | **Status of Block** |
| | 7–4 | Reserved |
| | 3 | Parity |
| | 2 | Valid |
| | 1 | Dirty |
| | 0 | Shared |
| C_ADDR<6:42> | Address of the last reported ECC or parity error. If C_STAT value is DSTREAM_DC_ERR, only bits <6:19> are valid. | |

## D.5 Exception Address Register (EXC_ADDR)

**The Exception Address Register (EXC_ADDR) is a read-only register that is updated by hardware when it encounters an exception or interrupt bit.**



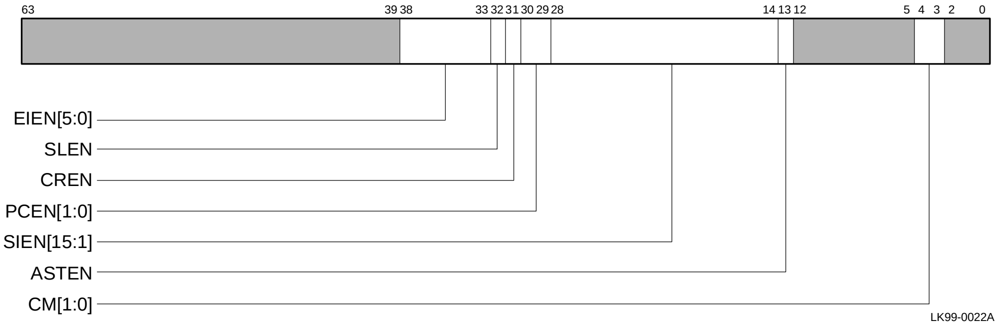
EXC_ADDR<0> is set if the associated exception occurred in PAL mode. The exception actions are:

- If the exception was a fault or a synchronous trap, EXC_ADDR contains the PC of the instruction that triggered the fault or trap.
- If the exception was an interrupt, EXC_ADDR contains the PC of the next instruction that would have executed if the interrupt had not occurred.

## D.6 Interrupt Enable and Current Processor Mode Register (IER_CM)

**The Interrupt Enable and Current Processor Mode Register (IER_CM) contains the interrupt enable and current processor mode bit fields.**

These bit fields can be written either individually or together with a single HW_MTPR instruction. When bits <7:2> of the IPR index field of a HW_MTPR instruction contain the value 000010, this register is selected. Bits <1:0> of the IPR index indicate which bit fields are to be written: bit<1> corresponds to the IER field and bit<0> corresponds to the processor mode field. A HW_MFPR instruction to this register returns the values in both fields.



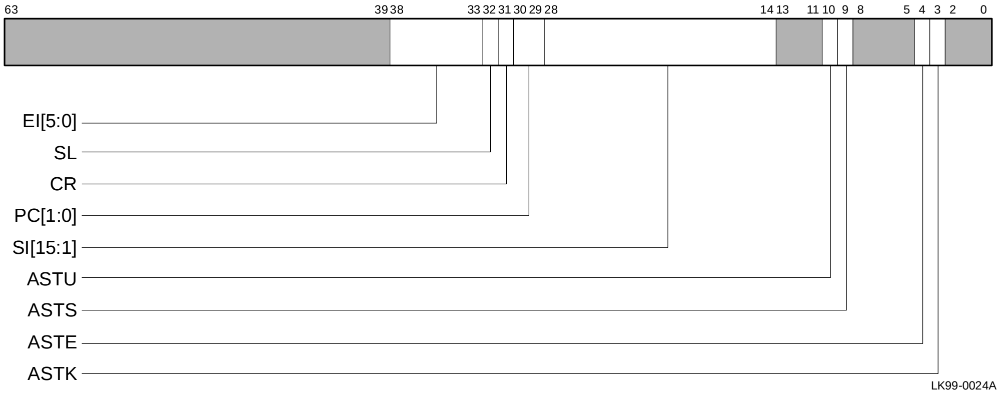
### Table D–5 IER_CM Register Fields

| Name | Extent | Type | Description |
|---|---|---|---|
| Reserved | <63:39> | | |
| EIEN<5:0> | <38:33> | RW | External Interrupt Enable |
| SLEN | <32> | RW | Serial Line Interrupt Enable |
| CREN | <31> | RW | Corrected Read Error Interrupt Enable |
| PCEN<1:0> | <30:29> | RW | Performance Counter Interrupt Enables |
| SIEN<15:1> | <28:14> | RW | Software Interrupt Enables |
| ASTEN | <13> | RW | AST Interrupt Enable<br>When set, enables those AST interrupt requests that are also enabled by the value in ASTER. |
| Reserved | <12:5> | | |
| CM<1:0> | <4:3> | RW | Current Mode<br>00 Kernel<br>01 Executive<br>10 Supervisor<br>11 User |
| Reserved | <2:0> | | |

## D.7 Interrupt Summary Register (ISUM)

**The Interrupt Summary Register (ISUM) is a read-only register that records all pending hardware, software, and AST interrupt requests that have their corresponding enable bit set.**

If a new interrupt (hardware, serial line, crd, or performance counters) occurs simultaneously with an ISUM read, the ISUM read returns zeros. That condition is normally assumed to be a passive release condition. The interrupt is signaled again when the PALcode returns to native mode. The effects of this condition can be minimized by reading ISUM twice and ORing the results.

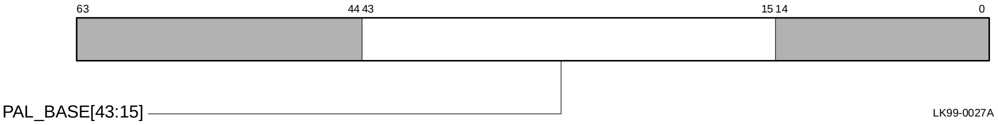


### Table D–6 ISUM Register Fields

| Name | Extent | Type | Description |
|---|---|---|---|
| Reserved | <63:39> | | |
| EI<5:0> | <38:33> | RO | External Interrupts |
| SL | <32> | RO | Serial Line Interrupt |
| CR | <31> | RO | Corrected Read Error Interrupts |
| PC<1:0> | <30:29> | RO | Performance Counter Interrupts<br>PC0 when PC<0> is set.<br>PC1 when PC<1> is set. |
| SI<15:1> | <28:14> | RO | Software Interrupts |
| Reserved | <13:11> | | |
| ASTU, ASTS | <10>,<9> | RO | AST Interrupts<br>For each processor mode, the bit is set if an associated AST interrupt is pending. This includes the mode's ASTER and ASTRR bits and whether the processor mode value held in the IER_CM register is greater than or equal to the value for the mode. |
| Reserved | <8:5> | | |
| ASTE, ASTK | <4>,<3> | RO | AST Interrupts<br>For each processor mode, the bit is set if an associated AST interrupt is pending. This includes the mode's ASTER and ASTRR bits and whether the processor mode value held in the IER_CM register is greater than or equal to the value for the mode. |
| Reserved | <2:0> | | |

## D.8 PAL Base Register (PAL_BASE)

**The PAL Base Register (PAL_BASE) is a read-write register that contains the base physical address for PALcode. Its contents are cleared by a chip reset but are not cleared after waking up from sleep mode or from fault reset.**

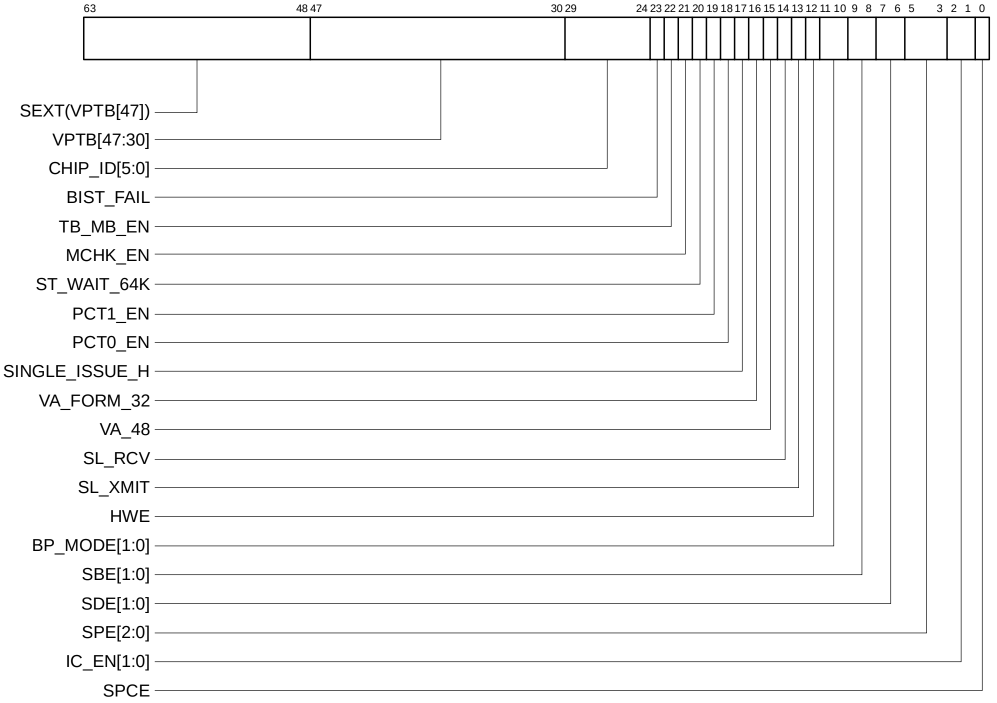


### Table D–7 PAL_BASE Register Fields

| Name | Extent | Type | Description |
|---|---|---|---|
| Reserved | <63:44> | RO, 0 | Reserved for COMPAQ. |
| PAL_BASE | <43:15> | RW | Base physical address for PALcode. |
| Reserved | <14:0> | RO, 0 | Reserved for COMPAQ. |

## D.9 Ibox Control Register (I_CTL)

**The Ibox Control Register (I_CTL) is a read-write register that controls various Ibox functions. Its contents are cleared by a chip reset.**



LK99-0029A

### Table D–8 I_CTL Register Fields

| Name | Extent | Type | Description |
|---|---|---|---|
| SEXT(VPTB<47>) | <63:48> | RW,0 | Sign extended VPTB<47>. |
| VPTB<47:30> | <47:30> | RW,0 | Virtual Page Table Base. |
| CHIP_ID<5:0> | <29:24> | RO | This is a read-only field that supplies the revision ID number for the EV68CB/EV68DC part. EV68CB/EV68DC pass 2.3 ID is 010111. |
| BIST_FAIL | <23> | RO,0 | Indicates the status of BIST (clear = pass, set = fail). |
| TB_MB_EN | <22> | RW,0 | When set, the hardware ensures that the virtual-mode loads in DTB and ITB fill flows that access the page table and the subsequent virtual mode load or store that is being retried are “ordered” relative to another processor’s stores. This must be set for multiprocessor systems in which no MB instruction is present in the TB fill flow, unless there are other mechanisms present that ensure coherency. |
| MCHK_EN | <21> | RW,0 | Machine check enable — set to enable machine checks. |
| CALL_PAL_R23 | <20> | RW,0 | CALL_PAL linkage register. If this bit is one, the CALL_PAL linkage register is R23; when zero, it is R27. Coordinate setting this bit with SDE<1:0> to ensure that the shadow register is used as the linkage register. |
| PCT1_EN | <19> | RW,0 | Enable performance counter #1. If this bit is one, the performance counter will count if either the system (SPCE) or process (PPCE) performance counter enable is asserted. |

### Table D–8 I_CTL Register Fields (Continued)

| Name | Extent | Type | Description |
|---|---|---|---|
| PCT0_EN | <18> | RW,0 | Enable performance counter #0. If this bit is one, the performance counter will count if EITHER the system (SPCE) or process (PPCE) performance counter enable is set. |
| SINGLE_ISSUE_H | <17> | RW,0 | When set, this bit forces instructions to issue only from the bottom-most entries of the IQ and FQ. |
| VA_FORM_32 | <16> | RW,0 | This bit controls address formatting on a read of the IVA_FORM register. |
| VA_48 | <15> | RW,0 | This bit controls the format applied to effective virtual addresses by the IVA_FORM register and the Ibox virtual address sign extension checkers. When VA_48 is clear, 43-bit virtual address format is used, and when VA_48 is set, 48-bit virtual address format is used. The effect of this bit on the IVA_FORM register is identical to the effect of VA_CTL<VA_48> on the VA_FORM register.<br>When VA_48 is set, the sign extension checkers generate an ACV if va<63:0> ≠ SEXT(va<47:0>). When VA_48 is clear, the sign extension checkers generate an ACV if va<63:0> ≠ SEXT(va<42:0>).<br>This bit also affects DTB_DOUBLE Traps. If set, the DTB double miss traps vector to the DTB_DOUBLE_4 entry point.<br>DTB_DOUBLE PALcode flow selection is not affected by VA_CTL<VA_48>. |

*Continued on next page*

## Table D–8 I_CTL Register Fields (Continued)

| Name | Extent | Type | Description |
|---|---|---|---|
| SL_RCV | <14> | RO | When in native mode, any transition on SL_RCV, driven from the SromData_H pin, results in a trap to the PALcode interrupt handler. When in PALmode, all interrupts are blocked. The interrupt routine then begins sampling SL_RCV under a software timing loop to input as much data as needed, using the chosen serial line protocol. |
| SL_XMIT | <13> | WO | When set, drives a value on SromClk_H. |
| HWE | <12> | RW,0 | If set, allow PALRES intructions to be executed in kernel mode. Note that modification of the ITB while in kernel mode/native mode may cause UNPREDICTABLE behavior. |
| BP_MODE<1:0> | <11:10> | RW,0 | Branch Prediction Mode Selection.<br>BP_MODE<1>, if set, forces all branches to be predicted to fall through. If clear, the dynamic branch predictor is chosen.<br>BP_MODE<0>. If set, the dynamic branch predictor chooses local history prediction. If clear, the dynamic branch predictor chooses local or global prediction based on the state of the chooser. |
| SBE<1:0> | <9:8> | RW,0 | Stream Buffer Enable.<br>The value in this bit field specifies the number of Istream buffer prefetches (besides the demand-fill) that are launched after an Icache miss. If the value is zero, only demand requests are launched. |
| SDE<1:0> | <7:6> | RW,0 | PALshadow Register Enable.<br>Enables access to the PALshadow registers. If SDE<1> is set, R4-R7 and R20-R23 are used as PALshadow registers. SDE<0> does not affect 21264 operation. |

## Table D–8 I_CTL Register Fields (Continued)

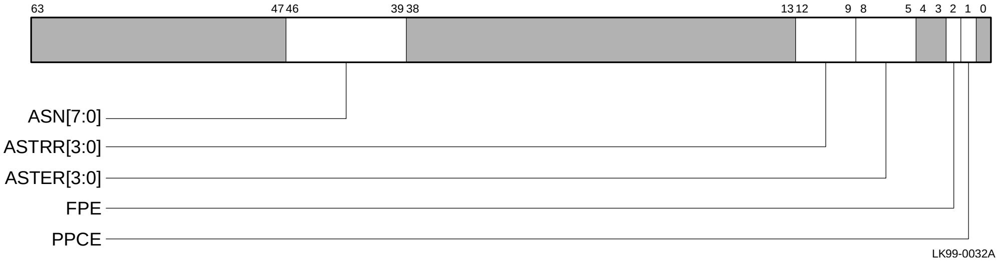
| Name | Extent | Type | Description |
|---|---|---|---|
| SPE<2:0> | <5:3> | RW,0 | Super Page Mode Enable.<br>Identical to the SPE bits in the Mbox M_CTL SPE<2:0>. |
| IC_EN<1:0> | <2:1> | RW,3 | Icache Set Enable.<br>At least one set must be enabled. The entire cache may be enabled by setting both bits. Zero, one, or two Icache sets can be enabled. This bit does not clear the Icache, but only disables fills to the affected set. |
| SPCE | <0> | RW,0 | System Performance Counting Enable.<br>Enables performance counting for the entire system if individual counters (PCTR0 or PCTR1) are enabled by setting PCT0_EN or PCT1_EN, respectively.<br>Performance counting for individual processes can be enabled by setting PCTX<PPCE>. |

*Continued on next page*

## D.10 Process Context Register (PCTX)

**The process context register (PCTX) contains information associated with the context of a process.**

The process context register (PCTX) contains information associated with the context of a process. Any combination of the bit fields within this register may be written with a single HW_MTPR instruction. When bits <7:6> of the IPR index field of a HW_MTPR instruction contain the value $01_2$, this register is selected. Bits <4:0> of the IPR index indicate which bit fields are to be written.



The following table lists the correspondence between IPR index bits and register fields.

| IPR Index Bit | Register Field |
|---|---|
| 0 | ASN |
| 1 | ASTER |
| 2 | ASTRR |
| 3 | PPCE |
| 4 | FPE |

Table D–9 lists the PXTX register fields.

## Table D–9 PCTX Register Fields

| Name | Extent | Type | Description |
|---|---|---|---|
| Reserved | <63:47> | | |
| ASN<7:0> | <46:39> | RW | Address space number. |
| Reserved | <38:13> | | |
| ASTRR<3:0> | <12:9> | RW | AST request register—used to request AST interrupts in each of the four processor modes.<br>To generate a particular AST interrupt, its corresponding bits in ASTRR and ASTER must be set, along with the ASTE bit in IER.<br>Further, the value of the current mode bits in the PS register must be equal to or higher than the value of the mode associated with the AST request.<br>The bit order with this field is:<br>User Mode 12 12<br>Supervior Mode 11 11<br>Executive Mode 10 10<br>Kernel Mode 9 9 |
| ASTER<3:0> | <8:5> | RW | AST enable register—used to individually enable each of the four AST interrupt requests.<br>The bit order with this field is:<br>User Mode 8 8<br>Supervisor Mode 7 7<br>Executive Mode 6 6<br>Kernel Mode 5 5 |
| Reserved | <4:3> | | |

*Continued on next page*

## Table D–9 PCTX Register Fields (Continued)

| Name | Extent | Type | Description |
|---|---|---|---|
| FPE | <2> | RW,1 | Floating-point enable—if clear, floating-point instructions generate FEN exceptions. This bit is set by hardware on reset. |
| PPCE | <1> | RW | Process performance counting enable.<br>Enables performance counting for an individual process with counters PCTR0 or PCTR1, which are enabled by setting PCT0_EN or PCT1_EN, respectively.<br>Performance counting for the entire system can be enabled by setting I_CTL<SPCE>. |

## D.11 21274 Cchip Miscellaneous Register (MISC)

**This register is designed so that there are no read side effects, and that writing a 0 to any bit has no effect. Therefore, when software wants to write a 1 to any bit in the register, it need not be concerned with read-modify-write or the status of any other bits in the register. Once NXM is set, the NXS field is locked so that initial NXM error information is not overwritten by subsequent errors. It is unlocked when the software clears the NXM CPU; however, writing it locks out the other CPU. Writing a 1 to ACL (arbitration clear) clears both ABW bits and both ABT (arbitration try) bits and unlocks the ABW field.**

| | |
|---|---|
| **Address** | 801 A000 0080 |
| **Access** | RW |

### Table D–10 21274 Cchip Miscellaneous Register Fields

| Name | Bits | Type | Initial State | Description |
|---|---|---|---|---|
| RES | <63:44> | MBZ, RAZ | 0 | Reserved. |
| DEVSUP | <43:40> | WO | 0 | |
| REV | <39:32> | RO | 1 | Cchip revision reads as 16 |
| NXS | <31:29> | RO | 0 | NXM source—Device that caused the NXM. Unpredictable if NXM not set.<br>0 = CPU0 1 = CPU1<br>2 = CPU2 3 = CPU3<br>4 = P-chip 0 5 = P-chip 1<br>6, 7 = Reserved |
| NXM | <28> | R, W1C | 0 | Nonexistent memory address detected. Sets DRIR<63> and locks the NXS field until it is cleared. |
| RES | <27:25> | MBZ, RW | 0 | Reserved. |
| ACL | <24> | WO | 0 | Arbitration clear—writing a 1 to this bit clears the ABT and ABW fields. |
| ABT | <23:20> | R, W1S | 0 | Arbitration try—writing a 1 to these bits sets them. |
| ABW | <19:16> | R, W1S | 0 | Arbitration won—writing a 1 to these bits sets them unless one is already set, in which case the write is ignored. |
| IPREQ | <15:12> | WO | 0 | Interprocessor interrupt request—write a 1 to the bit corresponding to the CPU you want to interrupt. Writing a 1 here sets the corresponding bit in the IPINTR. |

### Table D–10 21274 Cchip Miscellaneous Register Fields (Continued)

| Name | Bits | Type | Initial<br>State | Description |
|---|---|---|---|---|
| IPINTR | <11:8> | R, W1C | 0 | Interprocessor interrupt pending—one bit per CPU. |
| ITINTR | <7:4> | R, W1C | 0 | Interval timer interrupt pending—one bit per CPU. |
| RES | <3:2> | MBZ, RW | 0 | Reserved. |
| CPUID | <1:0> | RO | - | ID of the CPU performing the read. |

## D.12 21274 Cchip CPU Device Interrupt Request Register (DIRn, n=0,1,2,3)

**Register *n* applies to CP*Un*. These registers indicate which interrupts are pending to the CPUs. If a raw request bit is set and the corresponding mask bit is set, then the corresponding bit in this register will be set and the appropriate CPU will be interrupted.**

If a raw request bit is set and the corresponding mask bit is set, then the corresponding bit in this register will be set and the appropriate CPU will be interrupted.

| | |
|---|---|
| **Address** | 801 A000 0280 CPU0<br>801 A000 02C0CPU1<br>801 A000 0680 CPU2<br>801 A000 06C0 CPU3 |
| **Access** | RO |

### Table D–11 21274 Device Interrupt Request Register Fields

| | | | Initial | |
|---|---|---|---|---|
| **Name** | **Bits** | **Type** | **State** | **Description** |
| ERR | <63:58> | RO | 0 | IRQ0 error interrupts<br><63> Cchip detected MISC <NXM><br><62> Recommended hookup to Pchip0 error<br><61> Recommended hookup to Pchip1 error<br><60> Recommended hookup to Pchip0 soft error<br><59> Recommended hookup to Pchip1 soft error |
| RES | <57:56> | RO | 0 | Reserved |
| DEV | <55:12> | RO | 0 | IRQ1 PCI interrupts pending to the CPU |
| Hot Plug | <11:9> | RO | 0 | Hot plug controller interrupt |
| DEV | <8:0> | RO | 0 | IRQ 1 PCI controller interrupt |

## D.13 21274 Array Address Registers (AAR0–AAR3)

**The Array Address Registers define the base address and size for each memory array.**

Table D–12 21274 Array Address Register (AAR)

| Field | Bits | Type | Init | Description |
|---|---|---|---|---|
| RES | <63:35> | MBZ,RAZ | 0 | Reserved. |
| ADDR | <34:24> | RW | 0 | Base address – Bits <34:24> of the physical byte address of the first byte in the array. |
| RES | <23:17> | MBZ,RAZ | 0 | Reserved. |
| DBG | 16 | RW | 0 | Enables this memory port to be used as a debug interface. |
| ASIZ | <15:12> | RW | 0 | Array size. This field must be non-zero for AAR0.<br>**Value** **Size**<br>0000 0 (bank disabled)<br>0001 16MB<br>0100 32MB<br>0011 64MB<br>0100 128MB<br>0101 256MB<br>0110 512MB<br>0111 1GB<br>1000 2GB<br>1001 4GB<br>1010 8GB<br>1011 1111 Reserved |

*Continued on next page*

### Table D–12 21274 Array Address Register (AAR) (Continued)

<table>
<tr><th>Field</th><th>Bits</th><th>Type</th><th>Init</th><th>Description</th></tr>
<tr><td>RES</td><td>&lt;11:10&gt;</td><td>MBZ,RAZ</td><td>0</td><td>Reserved.</td></tr>
<tr><td>DSA</td><td>&lt;9&gt;</td><td>RW</td><td>0</td><td>Double (Twice)-split array</td></tr>
<tr><td>SA</td><td>&lt;8&gt;</td><td>RW</td><td>0</td><td>Split array.</td></tr>
<tr><td>RES</td><td>&lt;7:4&gt;</td><td>MBZ,RAZ</td><td>0</td><td>Reserved.</td></tr>
<tr><td>ROWS</td><td>&lt;3:2&gt;</td><td>RW</td><td>0</td><td>Number of row bits in the SDRAMs.
<table>
<tr><th>Value</th><th>Number of Bits</th></tr>
<tr><td>0</td><td>11</td></tr>
<tr><td>1</td><td>12</td></tr>
<tr><td>2</td><td>13</td></tr>
<tr><td>3</td><td>Reserved</td></tr>
</table></td></tr>
<tr><td>BNKS</td><td>&lt;1:0&gt;</td><td>RW</td><td>0</td><td>Number of bank bits in the SDRAMs
<table>
<tr><th>Value</th><th>Number of Bits</th></tr>
<tr><td>0</td><td>1</td></tr>
<tr><td>1</td><td>2</td></tr>
<tr><td>2</td><td>3</td></tr>
<tr><td>3</td><td>Reserved</td></tr>
</table></td></tr>
</table>

## D.14 Pchip System Error Register (SERROR)

**This register is used for logging system errors.**

When system error bits <4, 2:0> are set, this entire register is frozen. Only the NXIO bit and the LOST bit can be set after that. All other values will be held until bits <2:1> are clear. When an error is detected and one of bits <2:1> is set, the associated quadword address, CAP bus command, and syndrome is captured in bits <63:16> of this register. The NXIO bit does not log any address information, and does not lock the register.

### Table D–13 Pchip System Error Register

| Field | Bits | Type | Init | Description |
|---|---|---|---|---|
| SYN | <63:56> | RO | 0 | ECC syndrome of error |
| CMD | <55:54> | RO | 0 | Transaction type |
| | | | | **Value Command** |
| | | | | 00 DMA read |
| | | | | 01 DMA RMW |
| | | | | 10 SGTE read |
| | | | | 11 Reserved |
| SOURCE | <53:52> | RO | 0 | **Source bus** |
| | | | | **Value Command** |
| | | | | 00 GPCI |
| | | | | 01 APCI |
| | | | | 10 AGP HP |
| | | | | 11 AGP LP |

*Continued on next page*

## Table D–13 Pchip System Error Register (Continued)

| Field | Bits | Type | Init | Description |
|---|---|---|---|---|
| RES | <51:47> | RAZ | 0 | **Reserved** |
| ADDR | <46:15> | RO | 0 | Address of the erroneous quadword |
| RES | <14:5> | RAZ | 0 | Reserved |
| LOST_CRE | <4> | R,W1C | 0 | Lost a correctable ECC error because it was detected after this register was locked. c_err is asserted as long as this bit is set. |
| NXIO | <3> | R,W1C | 0 | Nonexistent IO error. Indicates that a reserved IO space was addressed by the CPU.<br>Logged if SERREN<NXIO> is set.<br>No address is logged, and the setting of this bit does not affect the state of the Lost bit.<br>**h_err** is asserted as long as this bit is set. |
| CRE | <2> | R,W1C | 0 | Correctable ECC error. Logged if SER-REN< CRE> is set. **c_err** is asserted as long as this bit is set. |
| UECC | <1> | R,W1C | 0 | Uncorrectable ECC error. Logged if SER-REN< UECC> is set. **h_err** is asserted as long as this bit is set. |
| LOST_UECC | <0> | R,W1C | 0 | Lost an Uncorrectable ECC error because it was detected after this register was locked. **h_err** is asserted as long as this bit is set. |

## D.15 Pchip A/G PCI Error Register (GPERROR, APERROR)

**This register is used for logging PCI errors on the GPCI or APCI buses respectively.**

The GPCI and APCI registers are identical. If any of bits <11:2> are set, then this entire register is frozen and the Pchip output signal **h_err** is asserted. Only bits <1:0> can be set after that. All other values will be held until bits <11:2, 0> are clear. When an error is detected and one of bits <10:2> is set, the associated cache block address, PCI command, and syndrome is captured in bits <55:52, 48:14> of this register. A monster window address has PCI address bit <40> set, and this is reflected in A/G PERROR bit <48>. Likewise, a non-MWIN dual address cycle (DAC) has PCI address bit <39> set, which shows up in A/G PERROR bit <47>. Bits <46:14> of A/G PERROR contain the longword PCI address bits <34:02> (a DAC could have bits set in PCI address bits <34:32>). Bits <11:1> of this register are only set if the corresponding enable bits are set in the PERREN register.

**NOTE:** *Software must not perform back-to-back writes to this register. A write to this register must be followed by a read from any PA-chip CSR, or a write to any other CSR (except for the corresponding A/G PERRSET register). Back-to-back writes will yield unpredictable results.*

### Table D–14 Pchip Error Register

<table>
<tr><th>Field</th><th>Bits</th><th>Type</th><th>Init</th><th>Description</th></tr>
<tr><td>RES</td><td>&lt;63:56&gt;</td><td>RAZ</td><td>0</td><td>Reserved</td></tr>
<tr><td>CMD</td><td>&lt;55:52&gt;</td><td>RO</td><td>0</td><td>PCI command
<table>
<tr><th>Value</th><th>Command</th></tr>
<tr><td>0000</td><td>Interrupt Ackn</td></tr>
<tr><td>0001</td><td>Special Cycle</td></tr>
<tr><td>0010</td><td>I/O Read</td></tr>
<tr><td>0011</td><td>I/O Write</td></tr>
<tr><td>0100</td><td>Reserved</td></tr>
<tr><td>0101</td><td>Reserved</td></tr>
<tr><td>0110</td><td>Memory Read</td></tr>
<tr><td>0111</td><td>Memory Write</td></tr>
<tr><td>1000</td><td>Reserved</td></tr>
<tr><td>1001</td><td>Reserved</td></tr>
<tr><td>1010</td><td>Configuration Read</td></tr>
<tr><td>1011</td><td>Configuration Write</td></tr>
<tr><td>1100</td><td>Mem Read Multiple</td></tr>
<tr><td>1101</td><td>Dual Address Cycle</td></tr>
<tr><td>1110</td><td>Memory Read Line</td></tr>
<tr><td>1111</td><td>Memory Write and Invalidate</td></tr>
</table></td></tr>
<tr><td>RES</td><td>&lt;51:49&gt;</td><td>RAZ</td><td>0</td><td>Reserved</td></tr>
<tr><td>MWIN</td><td>&lt;48&gt;</td><td>RO</td><td>0</td><td>Indicates that the erroneous access was to the Monster Window (PCI address bit &lt;40&gt;).</td></tr>
<tr><td>DAC</td><td>&lt;47&gt;</td><td>RO</td><td>0</td><td>Indicates that the erroneous access was a DAC (PCI address bit &lt;39&gt;).</td></tr>
</table>

### Table D–14 Pchip Error Register (Continued)

| Field | Bits | Type | Init | Description |
|---|---|---|---|---|
| ADDR | <46:14> | RO | 0 | Contains PCI address bits <34:02> |
| RES | <13:11> | RAZ | 0 | Reserved |
| IPTPW | <10> | R,W1C | 0 | Invalid peer-to-peer read. |
| IPTPR | <9> | R,W1C | 0 | Invalid peer-to-peer write. |
| NDS | <8> | R,W1C | 0 | No **devsel** received as a PCI master. |
| DPE | <7> | R,W1C | 0 | Pchip detected a parity error on data it received from another PCI device. If the command logged in bits <55:52> of this register is a read, the PCI transaction that encountered the error was a PIO Read or a PTP read on the destination bus, and the Pchip was the master. If the command logged is a write, the transaction that encountered the error was a DMA Write or a PTP write on the source bus (will only be logged in the GPERROR for Pchip Rev 0) and the Pchip was a target. |
| TA | <6> | R,W1C | 0 | Target abort received as PCI master. |
| APE | <5> | R,W1C | 0 | Address parity error detected as target. |
| SGE | <4> | R,W1C | 0 | Scatter Gather Error, invalid page table entry. |

*Continued on next page*

### Table D–14 Pchip Error Register (Continued)

| Field | Bits | Type | Init | Description |
|---|---|---|---|---|
| DCRTO | <3> | R,W1C | 0 | Delayed completion Retry timeout as PCI target. |
| PERR | <2> | R,W1C | 0 | Pchip received a **p_err_l** assertion on data it sent to the PCI. If the command logged in bits <55:52> of this register is a read, the PCI transaction that encountered the error was a DMA Read or a PTP read on the source bus and the Pchip was the target. If the command logged is a write, the transaction that encountered the error was a PIO Write or a PTP write on the destination bus. |
| SERR | <1> | R,W1C | 0 | Set when **serr_l** assertion is detected on the PCI. Does not log any other information in this register. Does not affect the logging of the Lost bit. |
| LOST | <0> | R,W1C | 0 | Lost an error because it was detected after this register was locked. |

## D.16 Pchip AGP Error Register (AGPERROR )

**The register is used for logging AGP errors.**

If any of bits <6:4, 0> are set, then this entire register is frozen and the Pchip output signal **h_err** is asserted. Only bit <0> can be set after that. All other values will be held until bits <6:4> are clear. When an error is detected and one of bits <6:4> is set, the associated cache block address and AGP bus command are captured in bits <63:16> of this register.

### Table D–15 Pchip AGP Error Register

| Field | Bits | Type | Init | Description |
|---|---|---|---|---|
| RES | <63:60> | MBZ | 0 | Reserved |
| FENCE | <59> | RO | 0 | Fence bit, used only if command code indicates LP transaction. |
| LENGTH | <58:53> | RO | 0 | AGJP transaction length in quadwords |
| CMD | <52:50> | RO | 0 | AGP Command<br>**Value** **Command**<br>000 Read (low priority)<br>001 Read (high priority)<br>010 Write (low priority)<br>011 Write (high priority)<br>100,101 Reserved<br>110 Flush<br>111 Fence |

*Continued on next page*

### Table D–15 Pchip AGP Error Register (Continued)

| Field | Bits | Type | Init | Description |
|---|---|---|---|---|
| MWIN | <49> | RO | 0 | Monster Window hit |
| DAC | <48> | RO | 0 | DAC |
| RES | <47> | RAZ | 0 | Reserved |
| ADDR | <46:15> | RO | 0 | AGP address <34:3> corresponding to the erroneous quadword. |
| RES | <14.7> | RAZ | 0 | Reserved |
| NOWINDOW | <6> | R,W1C | 0 | An incoming AGP address did not match the Window registers. |
| PTP | <5> | R,W1C | 0 | An incoming scatter-gather address had the PTP bit enabled. |
| IPTE | <4> | R,WIC | 0 | Invalid page table entry. |
| RESCMD | <3> | R,W1C | 0 | Reserved command received on the PCI. Logged if AGPERREN<RESCMD> is set. No other information is logged. Does not affect the setting of the Lost bit |
| HPQFULL | <2> | R,W1C | 0 | Reserved Command received on the PCI. Logged if AGPERREN<RESCMD> is set. No other information is logged; Does not affect the setting of the Lost bit |
| LPQFULL | <1> | R,WIC | 0 | The AGP Request Queue is full and a subsequent LP transaction was received. Logged if AGPER-REN< LPQFULL> is set. No other information is logged; does not affect the setting of the Lost bit. |
| LOST | <0> | R,WIC | 0 | Lost an error because it was detected after this register was locked. |

## D.17 DPR Registers for 680 Correctable Machine Check Logout Frames

**DPR Locations A0:A9 represent the information that the console will read when a 680 machine check logout frame is loaded. They provide the interrupt information obtained by the RMC through the LM78 sensors. When an error occurs, the RMC writes the bits and delivers an IRQ to the SRM console. The SRM reads the bits and clears them. On the next 680 error, the RMC writes the error into the A0:A9 locations.**

### Table D–16 DPR Locations A0:A9

| DPR Location | Description |
|---|---|
| A0 | If bit is set, the associated fault is active. |
| | Bit 0 +3.3v out of tolerance |
| | +5 v out of tolerance |
| | +12 v out of tolerance |
| | Vterm out of tolerance |
| | PCI backplane Zone 0 temp sensor is overtemp |
| | BTI (overtemp signals from all CPU and LM78 sensors) |
| | Fan 1 fault (below the minimum RPM) |
| | Fan 2 fault (below the minimum RPM) |
| A1 | Bit 0 CTERM out of tolerance<br>2 –12 v out of tolerance |

## Table D–16 DPR Locations A0:A9 (Continued)

| DPR<br>Location | Description |
|---|---|
| A2 | If bit is set the associated fault is active. |
| | Bit 0 CPU0_VCORE out of tolerance |
| | 1 CPU0_VIO out of tolerance |
| | 2 CPU1_VCORE out of tolerance |
| | 3 CPU1_VIO out of tolerance |
| | 4 PCI backplane LM78 1 is overtemp |
| | 5 Not used |
| | 6 Fan 4 fault |
| | 7 Fan 5 fault |
| A3 | Bit 0 +1.5 volt out of tolerance |
| | 1 CPU0_VCACHE out of tolerance |
| | 2 CPU1_VCACHE out of tolerance |
| | 4 +2.5 volt out of tolerance |
| | 5 CPU2_VCACHE out of tolerance |
| | 6 CPU3_VCACHE out of tolerance |
| A4 | If bit is set the associated fault is active. |
| | Bit 0 CPU2_VCORE out of tolerance |
| | 1 CPU2_VIO out of tolerance |
| | 2 CPU3_VCORE out of tolerance |
| | 3 CPU3_VIO out of tolerance |
| | 4 PCI backplane LM78 2 is overtemp |
| | 5 Not used |
| | 6 Fan 3 fault |
| | 7 Fan 6 fault |
| A5 | Bit 7 AC_input value high limit |
| | Bit 6 AC_input value low limit |
| | Bit 5 Monitor the temperature |
| | Bit 4 Current from +12 volt rail is out of tolerance |
| | Bit 3 Current from 5.5 volt rail is out of tolerance |
| | Bit 2 Current from 3.3 volt rail is out of tolerance |
| | Bit 1-0 Failing power supply number (0,1,2 are valid) |

## Table D–16 DPR Locations A0:A9 (Continued)

| DPR Location | Description |
|---|---|
| A6 | These bits indicate a door has been opened.<br>Bit 0 unused<br>1 CPU door is open<br>2 Fan door is open<br>3 PCI door is open<br>5 System CPU door is open<br>6 System fan door is open<br>7 System PCI door is open |
| A7 | Temperature Warning Mask<br>Bit 0 CPU0 temp warning<br>1 CPU1 temp warning<br>2 CPU2 temp warning<br>3 CPU3 temp warning<br>4 Temp Zone 0 (LM78 0 on PCI backplane)<br>5 Temp Zone 1 (LM78 1 on PCI backplane)<br>6 Temp Zone 2 (LM78 2 on PCI backplane) |
| A8 | Fan Controller Fault. This indicates a fan is not responding to a different RPM range as set by the RMC. (It is used to indicate that the fan failed to reach its maximum RPM at power-up.)<br>Bit 0 Fan 1<br>1 Fan 2<br>3 Fan 3<br>4 Fan 4<br>5 Fan 5<br>6 Fan 6 |
| A9 | These bits indicate which temperature zone the rise or fall in temperature occurred in.<br>Bit 0 CPU fans spin at the maximum speed<br>Bit 1 CPU fans reduce the speed from the maximum speed<br>Bit 2 PCI fans spin at the maximum speed<br>Bit 3 PCI fans reduce the speed from the maximum speed |

## D.18 DPR Power Supply Status Registers

**The RMC reads nine bytes of information from each of the three power supplies. The first byte is read from an I/O expander port, the second four bytes and the last four bytes are read from the A–D converter.**

### Table D–17 Nine Bytes Read from Power Supply

| DPR Location | Definition |
|---|---|
| DB/E4/ED | Reads I/O expander on power supply 0, 1, 2<br>Bit 0 PS_ID0_L<br>1 PS_ID1_L<br>2 Reserved (Pulled up so bit is always enabled)<br>3 Thermal_Shutdown_H<br>4:7 Tied to High within PS |
| DC/E5/EE | 3.3V_current. Each step equals 0.255 (0xFF x 0.33203 = 85A) |
| DD/E6/EF | 5 V_current. Each step equals 0.255 (0xFF x 0.33203 = 85A) |
| DE/E7/F0 | 12 V_current. Each step equals 0.033 (0xFF x 0.07813 = 20A) |
| DF/E8/F1 | Fan_Speed (0x8B = 7 V) |
| E0/E9/F2 | AC_INPUT value in hex. Each step equals 1.07422VAC (0xFF x 1.07422 = 275VAC) |
| E1/EA/F3 | Power_supply_internal_temperature (hot)<br>Byte represents a temp value<br>1 bit = 0.756° C |
| E2/EB/F4 | Power_supply_inlet_temperature<br>1 bit = 0.266° C |
| E3/EC/F5 | Spare |

**NOTE:** *The DPR locations refer to power supplies. For example, DB/E4/ED = power supply 0/1/2. The same is true for all locations listed in the table.*

## D.19 DPR 680 Fatal Registers

**The RMC is powered by an auxiliary 5V supply that is independent from the system power subsystem. When any catastrophic failures (such as overtemperature failure) occur, this error state is captured as shown in Table D–18. The information is used to populate the console data log uncorrectable error frame in Environ_QW_8.**

### Table D–18 DPR 680 Fatal Registers

| DPR Location | Definition |
|---|---|
| BD | Copy of the power supply AC input value<br>Bit 0 PS0 1 indicates AC input is valid; 0 indicates invalid<br>Bit 1 PS1<br>Bit 2 PS2 |
| BE | Snapshot of the fault I/O expander, which indicates PS, VTERM, CPU regulator fault if bit is set.<br>Bit 0 PS0<br>Bit 1 PS1<br>Bit 2 PS2<br>Bit 3 VTERM<br>Bit 4 CPU0<br>Bit 5 CPU1<br>Bit 6 CPU2<br>Bit 7 CPU3 |
| BF | RMC shutdown code<br>Bit 0 Unused<br>Bit 1 No CPU in CPU slot 0<br>Bit 2 Invalid CPU SROM voltage setting or checksum<br>Bit 3 TIG load initialization or sequence fail<br>Bit 4 Overtemperature failure<br>Bit 5 CPU door open<br>Bit 6 CPU fans 5 and 6 failed<br>Bit 7 CTERM failure |

## D.20 CPU and System Uncorrectable Machine Check Logout Frame

**The SRM console builds the uncorrectable machine check logout frames and passes them to the OS error handlers. The OS error handlers further process and subsequently log the formatted error event into the system binary error log.**

### Table D–19 CPU and System Uncorrectable Machine Check Logout Frame

| 63 56 | 55 48 | 47 40 | 39 32 | 31 24 | 23 16 | 15 8 | 7 0 | Offset(Hex) |
|---|---|---|---|---|---|---|---|---|
| Retryable/Second Error Flags | | | | Frame Size(00C8) | | | | 00000000 |
| System Area Offet(00A0) | | | | EV68 Area Offset(0018) | | | | 00000008 |
| Machine Check Frame Revision(1) | | | | Machine Check Code | | | | 00000010 |
| EV68 Ibox Status (I_STAT<31:29>) | | | | | | | | 00000018 |
| EV68 Dcache Status (DC_STAT<4:0>) | | | | | | | | 00000020 |
| EV68 Cbox (C_ADDR<43:6>) | | | | | | | | 00000028 |
| EV68 Cbox (C_SYNDROME_1<7:0>) | | | | | | | | 00000030 |
| EV68 Cbox (C_SYNDROME_0<7:0>) | | | | | | | | 00000038 |
| EV68 Cbox (C_STAT<4:0>) | | | | | | | | 00000040 |
| EV68 Cbox (C_STS<3:0>) | | | | | | | | 00000048 |
| EV68 TB Miss or Fault Status(MM_STAT<10:0>) | | | | | | | | 00000050 |
| EV68 Exception Address (EXC_ADDR) | | | | | | | | 00000058 |
| EV68 Interrupt Enablement and Current Processor Mode (IER_CM) | | | | | | | | 00000060 |
| EV68 Interrupt Summary Register (ISUM) | | | | | | | | 00000068 |
| EV68 Reserved 0 | | | | | | | | 00000070 |
| EV68 PAL Base Address (PAL_BASE) | | | | | | | | 00000078 |
| EV68 Ibox Control (I_CTL) | | | | | | | | 00000080 |
| EV68 Ibox Process Context (PCTX) | | | | | | | | 00000088 |
| EV68 Reserved 1 | | | | | | | | 00000090 |
| EV68 Reserved 2 | | | | | | | | 00000098 |
| Software Error Summary Flags | | | | | | | | 000000A0 |
| Cchip CPUx Device Interrupt Request Register (DIRx System Primary CPU Fault Watcher) | | | | | | | | 000000A8 |
| Cchip Miscellaneous Register (MISC) | | | | | | | | 000000B0 |
| Pchip 0 Error Register (P0_PERROR) | | | | | | | | 000000B8 |
| Pchip 1 Error Register (P1_PERROR) | | | | | | | | 000000C0 |

**NOTE:** *For CPU uncorrectable offsets B0–B8 are zeroed and system uncorrectable offsets 18–98 are zeroed.*

## D.21 Console Data Log Event Environmental Error Logout Frame (680 Uncorrectable)

**Compaq Analyze uses the logout frame in Table D–20 for its decomposition of all 680 system environmental uncorrectable error frames.**

Table D–20 Console Data Log Event Environmental Error Logout Frame (680 Uncorrectable)

| 63 56 | 55 48 | 47 40 | 39 32 | 31 24 | 23 16 | 15 8 | 7 0 | Offset (Hex) |
|---|---|---|---|---|---|---|---|---|
| Revision (1) | | Type (3) | | Class (12) | | Length (80) | | 00000000 |
| Processor WHAMI | | | | | | | | 00000008 |
| Retryable/Second Error Flags | | | | Frame Size 0070) | | | | 00000010 |
| System Area Offet(0020) | | | | EV68 Area Offset(0020[1]) | | | | 00000018 |
| Machine Check Frame Revision | | | | Machine Check Code (206) | | | | 00000020 |
| Software Error Summary Flags | | | | | | | | 00000028 |
| Cchip CPUx Device Interrupt Request Register (DIRx System Primary CPU Fault Watcher) | | | | | | | | 00000030 |
| Environ_QW_1 (TIG System Management Information Register (SMIR)) | | | | | | | | 00000038 |
| Environ_QW_2 (TIG CPU Information Register (CPUIR)) | | | | | | | | 00000040 |
| Environ_QW_3 (TIG Power Supply Information Register (PSIR)) | | | | | | | | 00000048 |
| Environ_QW_4 (System_PS/Temp/Fan_Fault - LM78_ISR ) | | | | | | | | 00000050 |
| Environ_QW_5 (System_Doors) | | | | | | | | 00000058 |
| Environ_QW_6(System_Temperature_Warning) | | | | | | | | 00000060 |
| Environ_QW_7(System_Fan_Control_Fault) | | | | | | | | 00000068 |
| Environ_QW_8(Fatal_Power_Down_Codes) | | | | | | | | 00000070 |
| Environ_QW_9(Environmental Reserved 1) | | | | | | | | 00000078 |

**NOTE:** *Only Environ_QW_8 contains valid error state capture. All other Environ_QW_1-7, 9 will be zeroed.*

[1] Per Alpha SRM requirement.

## D.22 CPU and System Correctable Machine Check Logout Frame

**The SRM console builds the correctable machine check logout frames and passes them to the OS error handlers. The OS error handlers further process and subsequently log the formatted error event into the system binary error log. The operating systems contain built-in throttling mechanisms to handle high-volume bursting of these correctable error conditions.**

### Table D–21 CPU and System Correctable Machine Check Logout Frame

<table>
<tr><th>63 56</th><th>55 48</th><th>47 40</th><th>39 32</th><th>31 24</th><th>23 16</th><th>15 8</th><th>7 0</th><th>Offset (Hex)</th></tr>
<tr><td colspan="4">Retryable / Second Error Flags</td><td colspan="4">Frame Size(0080)</td><td>00000000</td></tr>
<tr><td colspan="4">System Area Offet(0058)</td><td colspan="4">EV68 Area Offset(0018)</td><td>00000008</td></tr>
<tr><td colspan="4">Machine Check Frame Revision(1)</td><td colspan="4">Machine Check Code</td><td>00000010</td></tr>
<tr><td colspan="8">EV68 Ibox Status (I_STAT<31:29>)</td><td>00000018</td></tr>
<tr><td colspan="8">EV68 Dcache Status (DC_STAT<4:0>)</td><td>00000020</td></tr>
<tr><td colspan="8">EV68 Cbox (C_ADDR<43:6>)</td><td>00000028</td></tr>
<tr><td colspan="8">EV68 Cbox (C_SYNDROME_1<7:0>)</td><td>00000030</td></tr>
<tr><td colspan="8">EV68 Cbox (C_SYNDROME_0<7:0>)</td><td>00000038</td></tr>
<tr><td colspan="8">EV68 Cbox (C_STAT<4:0>)</td><td>00000040</td></tr>
<tr><td colspan="8">EV68 Cbox (C_STS<3:0>)</td><td>00000048</td></tr>
<tr><td colspan="8">EV68 TB Miss or Fault Status(MM_STAT<10:0>)</td><td>00000050</td></tr>
<tr><td colspan="8">Software Error Summary Flags (See section 1.4.2)</td><td>00000058</td></tr>
<tr><td colspan="8">Cchip CPUx Device Interrupt Request Register (DIRx System Primary CPU Fault Watcher)</td><td>00000060</td></tr>
<tr><td colspan="8">Cchip Miscellaneous Register (MISC)</td><td>00000068</td></tr>
<tr><td colspan="8">Pchip 0 Error Register (P0-PERROR)</td><td>00000070</td></tr>
<tr><td colspan="8">Pchip 1 Error Register (P1-PERROR )</td><td>00000078</td></tr>
</table>

**NOTE:** *For CPU correctable offsets 68–78 will be zeroed and system uncorrectable offsets 18–50 will be zeroed.*

## D.23 Environmental Error Logout Frame (680 Correctable)

**Table D–22 shows Environ_QW_1:7 and Environ_QW_8 error state capture information from DPR locations A0:A9 and BD:BF, respectively.**

### Table D–22 Environmental Error Logout Frame

| 63 56 | 55 48 | 47 40 | 39 32 | 31 24 | 23 16 | 15 8 | 7 0 | Offset (Hex) |
|---|---|---|---|---|---|---|---|---|
| Retryable/Second Error Flags | | | | Frame Size 0070) | | | | 00000000 |
| System Area Offet(0018) | | | | EV68 Area Offset(0018[1]) | | | | 00000008 |
| Machine Check Frame Revision(1) | | | | Machine Check Code (206) | | | | 00000010 |
| Software Error Summary Flags | | | | | | | | 00000018 |
| Cchip CPUx Device Interrupt Request Register (DIRx System Primary CPU Fault Watcher) | | | | | | | | 00000020 |
| Environ_QW_1 (TIG System Management Information Register (SMIR)) | | | | | | | | 00000028 |
| Environ_QW_2 (TIG CPU Information Register (CPUIR)) | | | | | | | | 00000030 |
| Environ_QW_3 (TIG Power Supply Information Register (PSIR)) | | | | | | | | 00000038 |
| Environ_QW_4 (System_PS/Temp/Fan_Fault - LM78_ISR ) | | | | | | | | 00000040 |
| Environ_QW_5 (System_Doors) | | | | | | | | 00000048 |
| Environ_QW_6(System_Temperature_Warning) | | | | | | | | 00000050 |
| Environ_QW_7(System_Fan_Control_Fault) | | | | | | | | 00000058 |
| Environ_QW_8(Fatal_Power_Down_Codes) | | | | | | | | 00000060 |
| Environ_QW_9(Environmental Reserved 1) | | | | | | | | 00000068 |

**NOTE:** *Only Environ_QW_1–7 contain valid error state capture. All other Environ_QW_8,9 will be zeroed.*

[1] Per Alpha SRM requirement.

## D.24 Platform Logout Frame Register Translation

**Compaq Analyze uses information from all logout frames for its decomposition of all error events. The error state bit definitions of all platform logout frame registers is shown in Table D–23.**

### Table D–23 Bit Definition of Logout Frame Registers

| Register Identification | Bit Field | Text Translation Description |
|---|---|---|
| C_SYNDROME_0 | <7:0> | Syndrome for lower quadword in octaword of victim that was scrubbed as follows : |

| <7:0>(Hex) | Data Bit | <7:0>(Hex) | Data Bit |
|---|---|---|---|
| CE | 00 | 4F | 32 |
| CB | 01 | 4A | 33 |
| D3 | 02 | 52 | 34 |
| D5 | 03 | 54 | 35 |
| D6 | 04 | 57 | 36 |
| D9 | 05 | 58 | 37 |
| DA | 06 | 5B | 38 |
| DC | 07 | 5D | 39 |
| 23 | 08 | A2 | 40 |
| 25 | 09 | A4 | 41 |
| 26 | 10 | A7 | 42 |
| 29 | 11 | A8 | 43 |
| 2A | 12 | AB | 44 |
| 2C | 13 | AD | 45 |
| 31 | 14 | B0 | 46 |
| 34 | 15 | B5 | 47 |
| 0E | 16 | 8F | 48 |
| 0B | 17 | 8A | 49 |
| 13 | 18 | 92 | 50 |
| 15 | 19 | 94 | 51 |
| 16 | 20 | 97 | 52 |
| 19 | 21 | 98 | 53 |
| 1A | 22 | 9B | 54 |
| 1C | 23 | 9D | 55 |
| E3 | 24 | 62 | 56 |
| E5 | 25 | 64 | 57 |
| E6 | 26 | 67 | 58 |
| E9 | 27 | 68 | 59 |

### Table D–23 Bit Definition of Logout Frame Registers (Continued)

| Register Identification | Bit Field | Text Translation Description | | | |
|---|---|---|---|---|---|
| | | EA | 28 | 6B | 60 |
| | | EC | 29 | 6D | 61 |
| C_SYNDROME_0 (continued) | | <7:0>(Hex) | Data Bit | <7:0>(Hex)70 | Data Bit |
| | | F1 | 30 | 75 | 62 |
| | | F4 | 31 | 10 | 63 |
| | | 01 | CB0 | 20 | CB4 |
| | | 02 | CB1 | 40 | CB5 |
| | | 04 | CB2 | 80 | CB6 |
| | | 08 | CB3 | | CB7 |
| C_SYNDROME_1 | <7:0> | Syndrome for upper quadword in octaword of victim that was scrubbed (same as specified above) | | | |
| C_STAT | <4:0> | <4:0>(Hex) | Detected Error [1] | | |
| | | 00 | No Error unless DC_STAT<3> = 1 indicating bcache/dcache victim read ECC error.<br>SNGL_BC_TAG_PERR | | |
| | | 01 | SNGL_DC_DUPLICATE_TAG_PERR | | |
| | | 02 | SNGL_DSTREAM_MEM_ECC_ERROR | | |
| | | 03 | SNGL_DSTREAM_BC_ECC_ERR | | |
| | | 04 | SNGL_DSTREAM _DC_ECC_ERR | | |
| | | 05 | SNGL_BC_PROBE _HIT_ERR | | |
| | | 06 or 07 | SNGL_ISTREAM_MEM_ECC _ERR | | |
| | | 0B | SNGL_ISTREAM_BC _ECC_ERR | | |
| | | 0C | DBL_DSTREAM_MEM_ECC_ERR | | |
| | | 13 | DBL_DSTREAM_BC_ECC_ERR | | |
| | | 14 | DBL_ISTREAM_MEM_ECC_ERR | | |
| | | 1B | DBL_ISTREAM_BC_ECC_ERR | | |
| | | 1C | | | |
| C_STS | <7:4> | Reserved | | | |
| | <3:0> | Captured status of the Bcache in INIT mode (<3>= Parity, <2> = Valid, <1> = Dirty, <0> = Shared). | | | |

*Continued on next page*

[1] SNGL: Single-bit error leading to correctable error; DBL: double-bit error leading to uncorrectable error.

### Table D–23 Bit Definition of Logout Frame Registers (Continued)

| Register Identification | Bit Field | Text Translation Description |
|---|---|---|
| C_ADDR | <42:6> | Address of last reported ECC or parity error. If C_STAT<4:0> = 05(Hex) then only C_ADDR<19:6> are valid. |
| I_STAT | <63:41> | Reserved |
| | <40> | ProfileMe Mispredict Trap |
| | <39> | ProfileMe Trap |
| | <38> | ProfileMe Load-Store Order Trap |
| | <37:34> | ProfileMe Trap Types |
| | <33> | ProfileMe Icache Miss |
| | <32:30> | ProfileMe Counter 0 Overcount |
| | <29> | Set = icache encountered a parity error on instruction fetch and a reply trap is performed which generates a correctable read interrupt. |
| | <28:0> | Reserved |
| DC_STAT | <4:0> | 00001(Bin) = Dcache tag probe pipeline 0 error;<br>00010(Bin) = Dcache tag probe pipeline 1 error;<br>00100(Bin) = Dcache data ECC error during store;<br>01000(Bin) = Dcache, Bcache or System fill data ECC error during load;<br>10000(Bin) = Dcache data store ECC error occurred within 6 cycles of the previous Dcache store ECC error. |
| MM_STAT | <3:0> | 0001(Bin)= Write reference triggered error;<br>0010(Bin) = Reference caused an access violation;<br>0100(Bin) = PTE<FOR> bit set during read reference error;<br>1000(Bin) = PTE<FOW> bit set during write reference error. |
| | <10> | Set = Dcache tag parity correctable error during initial tag probe of load/store instruction. |
| | <9:4> | Opcode of instruction which triggered error. |

## Table D–23 Bit Definition of Logout Frame Registers (Continued)

| Register Identification | Bit Field | Text Translation Description |
|---|---|---|
| EXC_ADDR | <0> | Set = exception or interrupt occurred in PAL mode |
| | <63:2> | Contains the PC address of the instruction that would have executed if the error interrupt did not occur. |
| IER_CM | <4:3> | 00(Bin) = Kernel Mode, 01(Bin) = Executive Mode, 10(Bin) = Supervisor Mode, 11(Bin) = User Mode |
| | <13> | Set = enables those AST interrupt requests by ASTER |
| | <28:14> | Software interrupt enables |
| | <30:29> | Performance counter interrupt enables |
| | <31> | Set = Correctable read error interrupt enabled |
| | <32> | Set = Serial Line Interrupt Enabled |
| | <38:33> | External IRQ<5:0> enable |
| I_SUM | <4:3> | AST Kernel and Executive Interrupts pending ;<br><3> Set = Kernel Mode AST interrupt pending,<br><4> Set =Executive Mode AST interrupt pending |
| | <10:9> | AST Supervisor and User Interrupts pending ;<br><9> Set =Supervisor Mode AST interrupt pending,<br><10> Set =User Mode AST interrupt pending |
| | <28:14> | Software interrupts pending |
| | <32> | Serial line interrupt pending |
| | <31> | Set = Corrected read interrupt pending |
| | <30:29> | Performance counter interrupts pending |
| | <38:33> | External interrupts pending |
| PAL_BASE | <43:15> | Contains the physical base address for PALcode |

*Continued on next page*

### Table D–23 Bit Definition of Logout Frame Registers (Continued)

| Register Identification | Bit Field | Text Translation Description |
|---|---|---|
| I_CTL | <2:1> | 01(Bin) and 10(Bin) for Icache set 1 or 2 enabled, respectively |
| | <7:6> | 01(Bin) and 10(Bin) for R8-R11 & R24-R27 and R4-R7 & R20-R23 are used for PAL shadow registers, respectively |
| | <13> | Set = forces bad Icache tag parity |
| | <14> | Set = forces bad Icache data parity |
| | <15> | Clear and set for 43 bit or 48 bit virtual address format, respectively |
| | <20> | Clear or set for R23 or R27 used as CALL_PAL linkage register, respectively |
| | <21> | Set to enable machine check processing |
| | <29:24> | Revision ID number for EV68 Chip as follows: 01(Hex) = Pass 1.0; 02(Hex) = Pass 2.2; 03(Hex) = Pass 2.3; 0x04 (Hex) = Pass 3.0. |
| | <47:30> | Virtual page table base address |
| PCTX | | Ibox process context register as follows : |
| | <0> | Reserved/RAZ |
| | <1> | If set, both performance counters are enabled |
| | <2> | If clear , floating-point instructions generate FEN exceptions |
| | <4:3> | Reserved/RAZ |
| | <8:5> | Enable AST U,S,E,K interrupt requests |
| | <12:9> | Request AST U,S,E.K interrupts |
| | <38:13> | Reserved/RAZ |
| | <46:39> | Address Space Number |
| | <63:47> | Reserved/RAZ |
| Software Error Summary Flags | | PAL and OS Error handler signaling software flags |
| | <0> | Set = Pchip0 P_Error<9:0> error has occurred. |
| | <1> | Set = Pchip1 P_Error<9:0> error has occurred. |
| | <2> | Set = Pchip0 or Pchip1 P_Error <11/10> |
| | <63:3> | Uncorrectable/correctable error, or CPU correctable error, or CPU uncorrectable error has occurred. |
| | | Unused |

### Table D–23 Bit Definition of Logout Frame Registers (Continued)

| ID | Bit Field | Text Translation Description |
|---|---|---|
| MISC | <43:40> | Suppress IRQ1 interrupts to 1(Hex) for CPU0, 2(Hex) for CPU1, 4(Hex) for CPU2, and 8(Hex) for CPU3 Cchip |
| | <39:32> | Cchip Revision Level : 00-07(Hex) for C2, 08-0F(Hex) for C4 |
| | <31:29> | 0(Hex) for CPU0, 1(Hex) for CPU1, 2(Hex) for CPU2, 3(Hex) for CPU3, 4(Hex) for Pchip0, 5(Hex) for Pchip1, as device (source) which caused the NXM |
| | <28> | Set = NXM address detected, <31:29> are locked, DRIR <63> is set |
| | <24> | Write 1 = Arbitration Clear |
| | <23:20> | =1(Hex) for CPU0, 2(Hex) for CPU1, 4(Hex) for CPU2, and 8(Hex) for CPU3 Arbitration Trying |
| | <19:16> | =1(Hex) for CPU0, 2(Hex) for CPU1, 4(Hex) for CPU2, and 8(Hex) for CPU3 Arbitration Won |
| | <15:12> | =1(Hex) for CPU0, 2(Hex) for CPU1, 4(Hex) for CPU2, and 8(Hex) for CPU3 to set interprocessor interrupt request. |
| | <11:8> | =1(Hex) for CPU0, 2(Hex) for CPU1, 4(Hex) for CPU2, and 8(Hex) for CPU3 interprocessor interrupt (IRQ<3>) pending |
| | <7:4> | =1(Hex) for CPU0, 2(Hex) for CPU1, 4(Hex) for CPU2, and 8(Hex) for CPU3 interval timer interrupt (IRQ<2>) pending |
| | <1:0> | =00(Bin) for CPU0, 01(Bin) for CPU1, 10(Bin) for CPU2, 11(Bin) for CPU3 ID performing the read. |

*Continued on next page*

### Table D–23 Bit Definition of Logout Frame Registers (Continued)

| ID | Bit Field | Text Translation Description |
|---|---|---|
| DIRx | <63> | Internal Cchip asynchronous error <i.e.NXM> (IRQ0) |
| | <62> | P0_Pchip error (IRQ0) |
| | <61> | P1_Pchip error (IRQ0)) |
| | <60> | P2_Pchip error (future designs) (IRQ0) |
| | <59> | P3_Pchip error (future designs) (IRQ0) |
| | <58> | OCP or RMC Halt(IRQ0) |
| | <57:56> | Unused |
| | <55> | INTR -PCI_ISA Device Interrupt error(IRQ1) |
| | <54> | SMI- System Mgmt Interrupt error(IRQ1) |
| | <53> | NMI - Non-Maskable Interrupt-fatal error (IRQ1) |
| | <52> | Unused |
| | <51> | Unused |
| | <50> | Environmental Temp,Doors,Fans errors (IRQ1) |
| | <49> | Unused |
| | <48> | Unused |
| | <47:44> | Pchip1_SLOT5<3:0>-System PCI Slot 9 INTa,b,c,d (IRQ1) |
| | <43:40> | Pchip1_SLOT4<3:0>-System PCI Slot 8 INTa,b,c,d (IRQ1) |
| | <39:36> | Pchip1_SLOT3<3:0>-System PCI Slot 7 INTa,b,c,d (IRQ1) |
| | <35:32> | Pchip1_SLOT2<3:0>-System PCI Slot 6 INTa,b,c,d (IRQ1) |
| | <31:28> | Pchip1_SLOT1<3:0>-System PCI Slot 5 INTa,b,c,d (IRQ1) |
| | <27:24> | Pchip1_SLOT0<3:0>-System PCI Slot 4 INTa,b,c,d (IRQ1) |
| | <23:20> | Pchip0_SLOT4<3:0>-System PCI Slot 3 INTa,b,c,d (IRQ1) |
| | <19:16> | Pchip0_SLOT3<3:0>-System PCI Slot 2 INTa,b,c,d (IRQ1) |
| | <15:12> | Pchip0_SLOT2<3:0>-System PCI Slot 1 INTa,b,c,d (IRQ1) |
| | <11:8> | Pchip0_SLOT1<3:0>-System PCI Slot 0 INTa,b,c,d (IRQ1) |
| | | ***Note:Pchip0_SLOT0 = PCI/ISA Cypress/Acer Bridge*** |
| | <7:0> | Unused |

### Table D–23 Bit Definition of Logout Frame Registers (Continued)

| Register Identification | Bit Field | Text Translation Description |
|---|---|---|
| P0 & 1_ERROR | <63:56> | ECC Syndrome of CRE or UECC error - Same as EV68. |
| | <55:52> | When CRE or UECC failing transaction: 0000(Bin) = DMA Read; 0001(Bin) = DMA RMW; 0011(Bin) = S/G Read.<br>PCI command of transaction when error not CRE or UECC : 0000(Bin) = PCI IACKCycle ; 0001(Bin) = PCI Special Cycle ; 0010(Bin) = PCI I/O Read; 0011(Bin) = PCI I/O Write; 0100(Bin) = Reserved ; 0101(Bin) = PCI PTP Write ; 0110(Bin) = PCI Memory Read ; 0111(Bin) = PCI Memory Write from CPUx; 1000(Bin) = PCI CSR Read; |
| | <51> | If clear = valid <63:56>,<55:52>, and <50:16> error information if any <11:0> bits are set, otherwise invalid. |
| | <50:16> | If <11> or <10> =set and <51> =clear, <50:19> = System address <34:3> of erred quadword and <18:16> = 000(Bin); else if any one of <9:0> =set and <51> = clear, <50:48> = 000(Bin),<47:18> = starting PCI address <31:2> of erred transaction, <17:16> = 00(Bin) if not DAC; 01(Bin) if DAC SG Windows 3; 1x(Bin) if Monster Window |
| | <15:12> | MBZ, RAZ |
| | <11> | Set = Correctable ECC Error (M or T[2]) |
| | <10> | Set = Uncorrectable ECC Error (M or T) |
| | <9> | Reserved – MBZ/RAZ |
| | <8> | Set = No device select as PCI (M) error |
| | <7> | Set = PCI read data parity error as PCI (M) |
| | <6> | Set = Target abort error detected as PCI (M) |
| | <5> | Set = Address parity error detected as potential PCI |
| | <4> | Set = Invalid S/G page table entry detected as PCI |
| | <3> | Set = Delayed completion retry time-out error as PCI |
| | <2> | Set = PERR# error as PCI (M) |
| | <1> | Set = SERR# error as PCI (M or T) |
| | <0> | Set = Error occurred / lost after this register locked |

*Continued on next page*

[2] M refers to PCI Master; T refers to PCI Target

### Table D–23 Bit Definition of Logout Frame Registers (Continued)

| Register Identification | Bit Field | Text Translation Description |
|---|---|---|
| SMIR | <7> | Inverted Sys_Rst = System is being reset |
| (Environ_QW_1) | <6> | Inverted PCI_Rst1 = PCI Bus #1 is in reset |
| | <5> | Inverted PCI_Rst0 = PCI Bus #0 is in reset |
| | <4> | Set = System temperature over 50 degrees C failure |
| | <3> | unused |
| | <2> | Set = Sys_DC_Notok failure detected |
| | <1> | Inverted OCP_RMC_Halt = OCP or RMC halt detected |
| | <0> | Set = System Power Supply failure detected |
| CPUIR | <7> | Set = CPU3 regulator or configuration sequence fail |
| (Environ_QW_2) | <6> | Set = CPU2 regulator or configuration sequence fail |
| | <5> | Set = CPU1 regulator or configuration sequence fail |
| | <4> | Set = CPU0 regulator or configuration sequence fail |
| | <3> | Set = CPU3 regulator is enabled |
| | <2> | Set = CPU2 regulator is enabled |
| | <1> | Set = CPU1 regulator is enabled |
| | <0> | Set = CPU0 regulator is enabled |
| PSIR | <7> | Not Used |
| (Environ_QW_3) | <6> | Set = Power Supply 2 failed and was enabled |
| | <5> | Set = Power Supply 1 failed and was enabled |
| | <4> | Set = Power Supply 0 failed and was enabled |
| | <3> | Not Used |
| | <2> | Set = Power Supply 2 is enabled |
| | <1> | Set = Power Supply 1 is enabled |
| | <0> | Set = Power Supply 0 is enabled |

**Table D–23 Bit Definition of Logout Frame Registers (Continued)**

| Register Identification | Bit Field | Text Translation Description |
|---|---|---|
| System_PS/Temp/<br>Fan_Fault_<br>LM78_ISR<br>(Environ_QW_4) | <0> | Set = PS +3.3V out of tolerance |
| | <1> | Set = PS +5V out of tolerance |
| | <2> | Set = PS +12V out of tolerance |
| | <3> | Set = VTERM out of tolerance |
| | <4> | Set = Temperature zone 0 (PCI Backplane slots 1-3 area) over limit failure |
| | <5> | Set = LM75 CPU0-3 Temperature over limit failure (OLF) |
| | <6> | Set = System Fan 1 failure |
| | <7> | Set = System Fan 2 failure |
| | <8> | Set = CTERM out of tolerance |
| | <9> | Unused |
| | <10> | Set = -12V out of tolerance |
| | <15:11 | Unused |
| | <16> | Set = CPU0_VCORE +2V out of tolerance |
| | <17> | Set = CPU0_VIO +1.5V out of tolerance |
| | <18> | Set = CPU1_VCORE +2V out of tolerance |
| | <19> | Set = CPU1_VIO +1.5V out of tolerance |
| | <20> | Set = Temperature zone 1 (PCI Backplane slots 7-10) (OLF) |
| | <21> | Unused |
| | <22> | Set = System Fan 4 failure |
| | <23> | Set = System Fan 5 failure |
| | <31:24> | Unused |
| | <32> | Set = CPU2_VCORE +2V out of tolerance |
| | <33> | Set = CPU2_VIO +1.5V out of tolerance |
| | <34> | Set = CPU3_VCORE +2V out of tolerance |
| | <35> | Set = CPU3_VIO +1.5V out of tolerance |
| | <36> | Set = Temperature zone 2 (PCI Backplane slots 4-6) (OLF) |
| | <37> | Unused |
| | <38> | Set = System Fan 3 failure |
| | <39> | Set = System Fan 6 failure |
| | <41:40> | 00(Bin) = Power supply 0; 01 (Bin) = power supply 1; 10 (Bin) = power supply 2; 11(Bin) = Reserved that has caused the <42:47> warning condition. |
| | <42> | Set = Power supply 3.3V rail above high amperage warning |
| | <43> | Set = Power supply 5.0V rail above high amperage warning |
| | <44> | Set = Power supply 12V rail above high amperage warning |
| | <45> | Set = Power supply high temperature warning |
| | <46> | Set = Power supply AC input low limit warning |

## Table D–23 Bit Definition of Logout Frame Registers (Continued)

| Register Identification | Bit Field | Text Translation Description |
|---|---|---|
| | <47> | Set = Power supply AC input high limit warning |
| | <63:48> | Unused |
| System_Doors | <0> | Unused |
| (Environ_QW_5) | <1> | Set = System CPU door is open |
| | <2> | Set = System Fan door is open |
| | <3> | Set = System PCI door is open |
| | <4> | Unused |
| | <5> | Set = System CPU door is closed |
| | <6> | Set = System Fan door is closed |
| | <7> | Set = System PCI door is closed |
| | <63:8> | Unused |
| System_Temperature_Warning | <0> | Set = CPU0 temperature warning fault has occurred |
| | <1> | Set = CPU1 temperature warning fault has occurred |
| (Environ_QW_6) | <2> | Set = CPU2 temperature warning fault has occurred |
| | <3> | Set = CPU3 temperature warning fault has occurred |
| | <4> | Set = System temperature zone 0 warning fault has occurred |
| | <5> | Set = System temperature zone 1 warning fault has occurred |
| | <6> | Set = System temperature zone 2 warning fault has occurred |
| | <63:7> | Unused |
| System_Fan_Control_Fault | <0> | Set = System Fan 1 is not responding to RMC Commands |
| | <1> | Set = System Fan 2 is not responding to RMC Commands |
| (Environ_QW_7) | <2> | Set = System Fan 3 is not responding to RMC Commands |
| | <3> | Set = System Fan 4 is not responding to RMC Commands |
| | <4> | Set = System Fan 5 is not responding to RMC Commands |
| | <5> | Set = System Fan 6 is not responding to RMC Commands |
| | <7:6> | Unused |
| | <8> | Set = CPU fans 5/6 at maximum speed |
| | <9> | Set = CPU fans 5/6 reduced speed from maximum |
| | <10> | Set = PCI fans 1-4 at maximum speed |
| | <11> | Set = PCI fans 1-4 reduced speed from maximum. |

### Table D–23 Bit Definition of Logout Frame Registers

| Register Identification | Bit Field | Text Translation Description |
|---|---|---|
| Fatal_Power_Down_Codes | <0> | Set = Power Supply 0 AC input fail |
| (Environ_QW_8) | <1> | Set = Power Supply 1 AC input fail |
| | <2> | Set = Power Supply 2 AC input fail |
| | <3:7> | Unused |
| | <8> | Set = Power Supply 0 DC fail |
| | <9> | Set = Power Supply 1 DC fail |
| | <10> | Set = Power Supply 2 DC fail |
| | <11> | Set = Vterm fail |
| | <12> | Set = CPU0 Regulator fail |
| | <13> | Set = CPU1 Regulator fail |
| | <14> | Set = CPU2 Regulator fail |
| | <15> | Set = CPU3 Regulator fail |
| | <16> | Unused |
| | <17> | Set = No CPU in system motherboard CPU slot 0 |
| | <18> | Set = Invalid CPU SROM voltage setting or checksum |
| | <19> | Set = TIG load initialization or sequence fail |
| | <20> | Set = Overtemperature fail |
| | <21> | Set = CPU door open fail |
| | <22> | Set = System fan 5 (CPU backup fan) fail |
| | <23> | Set = Cterm fail |
| | <63:24> | Unused |

# Appendix E
# Isolating Failing DIMMs

This appendix explains how to manually isolate a failing DIMM from the failing address and failing data bits. It also covers how to isolate single-bit errors. The following topics are covered:

- Information for Isolating Failures
- DIMM Isolation Procedure
- EV68 Single-Bit Errors

## E.1 Information for Isolating Failures

**Table E–1 lists the information needed to isolate the failure. See Appendix D for the register table for the Array Address Registers (AARs). The failing address and failing data can come from a variety of different locations such as the SROM serial line, SRM screen displays, the SRM event log, and errors detected by the 21264 (EV68) chip.**

Convert the address to data bits if the address is not on a 256-bit alignment (address ends in a value less than 20 or address *xxxxx*20 or address *xxxxxnn*, where *nn* is 1 through 1F). For example, using failing address 0x1004 and failing data bit 8(dec), first multiply the failing address 4 by 8 = 32. Then add 32 to the failing data bit to yield the actual failing data bit 40. This conversion yields the new failing information to be failing address 0x1000 and failing data bit = 40(dec).

### Table E–1 Information Needed to Isolate Failing DIMMs

| **Failing Address** | |
|---|---|
| **Failing Data/Check bits** | |
| **Array Address Registers** | **Memory Addresses** |
| CSC | 801.A000.0000 |
| AAR0 | 801.A000.0100 |
| AAR1 | 801.A000.0140 |
| AAR2 | 801.A000.0180 |
| AAR3 | 801.A000.01C0 |
| **DPR Locations** | **Memory Addresses** |
| DPR:80 | 801.1000.2000 |
| DPR:82 | 801.1000.2080 |
| DPR:84 | 801.1000.2100 |
| DPR:86 | 801.1000.2180 |

## E.2 DIMM Isolation Procedure

**Use the procedure in this section to isolate the failing DIMM.**

1. Find the failing array by using the failing address and the Array Address Registers (AARs—see Appendix D). Use the AAR base address and size to create an Address range for comparing the failing address.

   For example, if AAR1 base address was 40000000 (1 GB) and its size was 10000000 (256 MB), the address range would be 40000000–4FFFFFFF (4–4.25 GB). This range would be used to compare against the failing address.

2. Determine if the Address XORing is enabled.

   - If Address XORING is enabled, use Table E–2 to find the real array on which the failure occurred for 4-way interleaving, or **Error! Reference source not found.** for 2-way interleaving.
   - If Bit 51 of the CSC register is set to 1, XORing is disabled.

### Table E–2 Determining the Real Failed Array for 4-Way Interleaving

| Failing Address <8:7> | Original Array 0 | Original Array 1 | Original Array 2 | Original Array 3 |
|---|---|---|---|---|
| 00 | Real Array 0 | Real Array 1 | Real Array 2 | Real Array 3 |
| 01 | Real Array 1 | Real Array 0 | Real Array 3 | Real Array 2 |
| 10 | Real Array 2 | Real Array 3 | Real Array 0 | Real Array 1 |
| 11 | Real Array 3 | Real Array 2 | Real Array 1 | Real Array 0 |

### Table E–3 Determining the Real Failed Array for 2-Way Interleaving

| Failing Address <8> | Original Array 0 | Original Array 1 | Original Array 2 | Original Array 3 |
|---|---|---|---|---|
| 0 | Real Array 0 | Real Array 1 | Real Array 2 | Real Array 3 |
| 1 | Real Array 2 | Real Array 3 | Real Array 0 | Real Array 1 |

3. After finding the real array, determine whether it is the lower array set or the upper array set. Use DPR locations 80, 82, 84, and 86 listed in Table E–1. Table E–4 shows the description of these locations.

**Table E–4 Description of DPR Locations 80, 82, 84, and 86**

| DPR Location | Description | |
|---|---|---|
| 80 | Array 0 (AAR 0) Configuration | |
| | Bits<7:4> | Bits<3:0> |
| | 4 = non split—lower set only<br>5 = split—lower set only<br>9 = split—upper set only<br>D = split—8 DIMMs<br>F = Twice split—8 DIMMs | 0 = Configured—Lowest array<br>1 = Configured—Next lowest array<br>2 = Configured—Second highest array<br>3 = Configured—Highest array<br>4 = Misconfigured—Missing DIMM(s)<br>8 = Miconfigured—Illegal DIMM(s)<br>C = Misconfigured—Incompatible DIMM(s) |
| 82 | Array 1 (AAR 1) configuration | |
| 84 | Array 2 (AAR 2) configuration | |
| 86 | Array 3 (AAR 3) configuration | |

4. Use the following table to determine the proper set. Bits<27,28,29,30,31,32> are from the failing address.

| Array Size | Configuration Type Bits <7:4> from DPR | | |
|---|---|---|---|
| | 4 & 5 | 9 | D & F |
| 256MB | Lower Set | Upper Set | Bit <27> == 0 – Lower Set, 1– Upper Set |
| 512MB | Lower Set | Upper Set | Bit <28> == 0 – Lower Set, 1– Upper Set |
| 1GB | Lower Set | Upper Set | Bit <29> == 0 – Lower Set, 1– Upper Set |
| 2GB | Lower Set | Upper Set | Bit <30> == 0 – Lower Set, 1– Upper Set |
| 4GB | Lower Set | Upper Set | Bit <31> == 0 – Lower Set, 1– Upper Set |
| 8GB | Lower Set | Upper Set | Bit <32> == 0 – Lower Set, 1– Upper Set |

5. Now that you have the real array, the failing Data/Check bits, and the correct set, use Table E–5 to find the failing DIMM or DIMMs.

The table shows data bits 0–255 and check bits 0–31. These data bits indicate a single-bit error. An SROM compare error would yield address and data bits from 0–63. When you convert the address to be in the correct range, the failing data would be somewhere between 0 and 255.

### Table E–5 Failing DIMM Lookup Table

| | Array 0 | | | | Array 1 | | | | Array 2 | | | | Array 3 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Data Bits | Lower Set | | Upper Set | | Lower Set | | Upper Set | | Lower Set | | Upper Set | | Lower Set | | Upper Set | |
| | MMB | J# | MMB | J# | MMB | J# | MMB | J# | MMB | J# | MMB | J# | MMB | J# | MMB | J# |
| 0 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 1 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 2 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 3 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 4 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 5 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 6 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 7 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 8 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 9 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 10 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 11 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 12 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 13 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 14 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 15 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 16 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 17 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 18 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 19 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 20 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 21 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 22 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 23 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |

## Table E–5 Failing DIMM Lookup Table (Continued)

| | Array 0 | | | | Array 1 | | | | Array 2 | | | | Array 3 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Data Bits | Lower Set | | Upper Set | | Lower Set | | Upper Set | | Lower Set | | Upper Set | | Lower Set | | Upper Set | |
| | MMB | J# | MMB | J# | MMB | J# | MMB | J# | MMB | J# | MMB | J# | MMB | J# | MMB | J# |
| 24 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 25 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 26 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 27 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 28 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 29 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 30 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 31 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 32 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 33 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 34 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 35 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 36 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 37 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 38 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 39 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 40 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 41 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 42 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 43 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 44 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 45 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 46 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 47 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |

*Continued on next page*

## Table E–5 Failing DIMM Lookup Table (Continued)

| | Array 0 | | | | Array 1 | | | | Array 2 | | | | Array 3 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Data Bits | Lower Set | | Upper Set | | Lower Set | | Upper Set | | Lower Set | | Upper Set | | Lower Set | | Upper Set | |
| | MMB | J# | MMB | J# | MMB | J# | MMB | J# | MMB | J# | MMB | J# | MMB | J# | MMB | J# |
| 48 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 49 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 50 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 51 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 52 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 53 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 54 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 55 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 56 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 57 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 58 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 59 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 60 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 61 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 62 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 63 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 64 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 65 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 66 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 67 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 68 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 69 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 70 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 71 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |

### Table E–5 Failing DIMM Lookup Table (Continued)

| | Array 0 | | | | Array 1 | | | | Array 2 | | | | Array 3 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Data Bits | Lower Set | | Upper Set | | Lower Set | | Upper Set | | Lower Set | | Upper Set | | Lower Set | | Upper Set | |
| | MMB | J# | MMB | J# | MMB | J# | MMB | J# | MMB | J# | MMB | J# | MMB | J# | MMB | J# |
| 72 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 73 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 74 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 75 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 76 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 77 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 78 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 79 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 80 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 81 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 82 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 83 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 84 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 85 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 86 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 87 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 88 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 89 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 90 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 91 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 92 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 93 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 94 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 95 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |

*Continued on next page*

### Table E–5 Failing DIMM Lookup Table (Continued)

| | Array 0 | | | | Array 1 | | | | Array 2 | | | | Array 3 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Data Bits | Lower Set | | Upper Set | | Lower Set | | Upper Set | | Lower Set | | Upper Set | | Lower Set | | Upper Set | |
| | MMB | J# | MMB | J# | MMB | J# | MMB | J# | MMB | J# | MMB | J# | MMB | J# | MMB | J# |
| 96 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 97 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 98 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 99 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 100 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 101 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 102 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 103 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 104 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 105 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 106 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 107 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 108 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 109 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 110 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 111 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 112 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 113 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 114 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 115 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 116 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 117 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 118 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 119 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |

## Table E–5 Failing DIMM Lookup Table (Continued)

| | Array 0 | | | | Array 1 | | | | Array 2 | | | | Array 3 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Data Bits | Lower Set | | Upper Set | | Lower Set | | Upper Set | | Lower Set | | Upper Set | | Lower Set | | Upper Set | |
| | MMB | J# | MMB | J# | MMB | J# | MMB | J# | MMB | J# | MMB | J# | MMB | J# | MMB | J# |
| 120 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 121 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 122 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 123 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 124 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 125 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 126 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 127 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 128 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 129 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 130 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 131 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 132 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 133 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 134 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 135 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 136 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 137 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 138 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 139 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 140 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 141 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 142 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 143 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |

*Continued on next page*

## Table E–5 Failing DIMM Lookup Table (Continued)

| | Array 0 | | | | Array 1 | | | | Array 2 | | | | Array 3 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Data Bits | Lower Set | | Upper Set | | Lower Set | | Upper Set | | Lower Set | | Upper Set | | Lower Set | | Upper Set | |
| | MMB | J# | MMB | J# | MMB | J# | MMB | J# | MMB | J# | MMB | J# | MMB | J# | MMB | J# |
| 144 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 145 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 146 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 147 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 148 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 149 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 150 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 151 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 152 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 153 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 154 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 155 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 156 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 157 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 158 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 159 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 160 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 161 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 162 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 163 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 164 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 165 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 166 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 167 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |

### Table E–5 Failing DIMM Lookup Table (Continued)

| | Array 0 | | | | Array 1 | | | | Array 2 | | | | Array 3 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Data Bits | Lower Set | | Upper Set | | Lower Set | | Upper Set | | Lower Set | | Upper Set | | Lower Set | | Upper Set | |
| | MMB | J# | MMB | J# | MMB | J# | MMB | J# | MMB | J# | MMB | J# | MMB | J# | MMB | J# |
| 168 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 169 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 170 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 171 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 172 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 173 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 174 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 175 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 176 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 177 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 178 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 179 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 180 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 181 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 182 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 183 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 184 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 185 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 186 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 187 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 188 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 189 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 190 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |

*Continued on next page*

## Table E–5 Failing DIMM Lookup Table (Continued)

| | Array 0 | | | | Array 1 | | | | Array 2 | | | | Array 3 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Data Bits | Lower Set | | Upper Set | | Lower Set | | Upper Set | | Lower Set | | Upper Set | | Lower Set | | Upper Set | |
| | M<br>M<br>B | J<br># | M<br>M<br>B | J<br># | M<br>M<br>B | J<br># | M<br>M<br>B | J<br># | M<br>M<br>B | J<br># | M<br>M<br>B | J<br># | M<br>M<br>B | J<br># | M<br>M<br>B | J<br># |
| 191 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 192 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 193 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 194 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 195 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 196 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 197 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 198 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 199 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 200 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 201 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 202 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 203 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 204 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 205 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 206 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 207 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 208 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 209 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 210 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 211 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 212 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 213 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 214 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 215 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |

## Table E–5 Failing DIMM Lookup Table (Continued)

| | Array 0 | | | | Array 1 | | | | Array 2 | | | | Array 3 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Data Bits | Lower Set | | Upper Set | | Lower Set | | Upper Set | | Lower Set | | Upper Set | | Lower Set | | Upper Set | |
| | MMB | J# | MMB | J# | MMB | J# | MMB | J# | MMB | J# | MMB | J# | MMB | J# | MMB | J# |
| 216 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 217 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 218 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 219 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 220 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 221 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 222 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 223 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 224 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 225 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 226 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 227 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 228 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 229 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 230 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 231 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 232 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 233 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 234 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 235 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 236 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 237 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 238 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 239 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |

*Continued on next page*

## Table E–5 Failing DIMM Lookup Table (Continued)

| | Array 0 | | | | Array 1 | | | | Array 2 | | | | Array 3 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Data Bits | Lower Set | | Upper Set | | Lower Set | | Upper Set | | Lower Set | | Upper Set | | Lower Set | | Upper Set | |
| | MMB | J# | MMB | J# | MMB | J# | MMB | J# | MMB | J# | MMB | J# | MMB | J# | MMB | J# |
| 240 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 241 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 242 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 243 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 244 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 245 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 246 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 247 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 248 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 249 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 250 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 251 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 252 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 253 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 254 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 255 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |

## Table E–5 Failing DIMM Lookup Table (Continued)

| | Array 0 | | | | Array 1 | | | | Array 2 | | | | Array 3 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Check Bits | Lower Set | | Upper Set | | Lower Set | | Upper Set | | Lower Set | | Upper Set | | Lower Set | | Upper Set | |
| | M<br>M<br>B | J<br># | M<br>M<br>B | J<br># | M<br>M<br>B | J<br># | M<br>M<br>B | J<br># | M<br>M<br>B | J<br># | M<br>M<br>B | J<br># | M<br>M<br>B | J<br># | M<br>M<br>B | J<br># |
| 0 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 1 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 2 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 3 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 4 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 5 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 6 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 7 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 8 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 9 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 10 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 11 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 12 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 13 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 14 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 15 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 16 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 17 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 18 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 19 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 20 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 21 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 22 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 23 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |

*Continued on next page*

## Table E–5 Failing DIMM Lookup Table (Continued)

| | Array 0 | | | | Array 1 | | | | Array 2 | | | | Array 3 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Check Bits | Lower Set | | Upper Set | | Lower Set | | Upper Set | | Lower Set | | Upper Set | | Lower Set | | Upper Set | |
| | MMB | J# | MMB | J# | MMB | J# | MMB | J# | MMB | J# | MMB | J# | MMB | J# | MMB | J# |
| 24 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 25 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 26 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 27 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |
| 28 | 0 | 5 | 0 | 4 | 1 | 5 | 1 | 4 | 0 | 2 | 0 | 3 | 1 | 2 | 1 | 3 |
| 29 | 2 | 5 | 2 | 4 | 3 | 5 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 |
| 30 | 0 | 9 | 0 | 8 | 1 | 9 | 1 | 8 | 0 | 6 | 0 | 7 | 1 | 6 | 1 | 7 |
| 31 | 2 | 9 | 2 | 8 | 3 | 9 | 3 | 8 | 2 | 6 | 2 | 7 | 3 | 6 | 3 | 7 |

## E.3 EV68 Single-Bit Errors

**The procedure for detection down to the set of DIMMs for a single-bit error is very similar to the procedure described in the previous sections. However, you cannot isolate down to a specific data or check bit.**

The 21264 (EV68) chip detects and reports a C_ADDR<42:6> failing address that is accurate to the cache block (64 bytes). The syndrome registers (Table E–6) detect data syndrome information, providing isolation down to the low or high quadword of the target octaword that the fault has been detected within. Each of the syndrome registers is able to report 64 data bits (the quadword) and 8 check bits (memory data bus ECC bits).

Table E–6 shows the syndrome hexadecimal to physical data or check bit decoding. For example, if you have an EV68 single-bit C_Syndrome_0 hexadecimal error value equal to 23, the second column indicates the decoded physical data or check bit for this encoding. Use these physical data bits in conjunction with the previously described isolation procedure to isolate the failing DIMMs.

### Table E–6 Syndrome to Data Check Bits Table

| Syndrome | C_Syndrome 0 | C_Syndrome 1 |
|---|---|---|
| CE | Data Bit 0 or 128 | Data Bit 64 or 192 |
| CB | Data Bit 1 or 129 | Data Bit 65 or 193 |
| D3 | Data Bit 2 or 130 | Data Bit 66 or 194 |
| D5 | Data Bit 3 or 131 | Data Bit 67 or 195 |
| D6 | Data Bit 4 or 132 | Data Bit 68 or 196 |
| D9 | Data Bit 5 or 133 | Data Bit 69 or 197 |
| DA | Data Bit 6 or 134 | Data Bit 70 or 198 |
| DC | Data Bit 7 or 135 | Data Bit 71 or 199 |
| 23 | Data Bit 8 or 136 | Data Bit 72 or 200 |
| 25 | Data Bit 9 or 137 | Data Bit 73 or 201 |
| 26 | Data Bit 10 or 138 | Data Bit 74 or 202 |
| 29 | Data Bit 11 or 139 | Data Bit 75 or 203 |
| 2A | Data Bit 12 or 140 | Data Bit 76 or 204 |
| 2C | Data Bit 13 or 141 | Data Bit 77 or 205 |

### Table E–6 Syndrome to Data Check Bits Table (Continued)

| Syndrome | C_Syndrome 0 | C_Syndrome 1 |
|---|---|---|
| 31 | Data Bit 14 or 142 | Data Bit 78 or 206 |
| 34 | Data Bit 15 or 143 | Data Bit 79 or 207 |
| 0E | Data Bit 16 or 144 | Data Bit 80 or 208 |
| 0B | Data Bit 17 or 145 | Data Bit 81 or 209 |
| 13 | Data Bit 18 or 146 | Data Bit 82 or 210 |
| 15 | Data Bit 19 or 147 | Data Bit 83 or 211 |
| 16 | Data Bit 20 or 148 | Data Bit 84 or 212 |
| 19 | Data Bit 21 or 149 | Data Bit 85 or 213 |
| 1A | Data Bit 22 or 150 | Data Bit 86 or 214 |
| 1C | Data Bit 23 or 151 | Data Bit 87 or 215 |
| E3 | Data Bit 24 or 152 | Data Bit 88 or 216 |
| E5 | Data Bit 25 or 153 | Data Bit 89 or 217 |
| E6 | Data Bit 26 or 154 | Data Bit 90 or 218 |
| E9 | Data Bit 27 or 155 | Data Bit 91 or 219 |
| EA | Data Bit 28 or 156 | Data Bit 92 or 220 |
| EC | Data Bit 29 or 157 | Data Bit 93 or 221 |
| F1 | Data Bit 30 or 158 | Data Bit 94 or 222 |
| F4 | Data Bit 31 or 159 | Data Bit 95 or 223 |
| 4F | Data Bit 32 or 160 | Data Bit 96 or 224 |
| 4A | Data Bit 33 or 161 | Data Bit 97 or 225 |
| 52 | Data Bit 34 or 162 | Data Bit 98 or 226 |
| 54 | Data Bit 35 or 163 | Data Bit 99 or 227 |
| 57 | Data Bit 36 or 164 | Data Bit 100 or 228 |
| 58 | Data Bit 37 or 165 | Data Bit 101 or 229 |
| 5B | Data Bit 38 or 166 | Data Bit 102 or 230 |
| 5D | Data Bit 39 or 167 | Data Bit 103 or 231 |
| A2 | Data Bit 40 or 168 | Data Bit 104 or 232 |
| A4 | Data Bit 41 or 169 | Data Bit 105 or 233 |
| A7 | Data Bit 42 or 170 | Data Bit 106 or 234 |
| A8 | Data Bit 43 or 171 | Data Bit 107 or 235 |
| AB | Data Bit 44 or 172 | Data Bit 108 or 236 |
| AD | Data Bit 45 or 173 | Data Bit 109 or 237 |

### Table E–6 Syndrome to Data Check Bits Table (Continued)

| Syndrome | C_Syndrome 0 | C_Syndrome 1 |
|---|---|---|
| B0 | Data Bit 46 or 174 | Data Bit 110 or 238 |
| B5 | Data Bit 47 or 175 | Data Bit 111 or 239 |
| 8F | Data Bit 48 or 176 | Data Bit 112 or 240 |
| 8A | Data Bit 49 or 177 | Data Bit 113 or 241 |
| 92 | Data Bit 50 or 178 | Data Bit 114 or 242 |
| 94 | Data Bit 51 or 179 | Data Bit 115 or 243 |
| 97 | Data Bit 52 or 180 | Data Bit 116 or 244 |
| 98 | Data Bit 53 or 181 | Data Bit 117 or 245 |
| 9B | Data Bit 54 or 182 | Data Bit 118 or 246 |
| 9D | Data Bit 55 or 183 | Data Bit 119 or 247 |
| 62 | Data Bit 56 or 184 | Data Bit 120 or 248 |
| 64 | Data Bit 57 or 185 | Data Bit 121 or 249 |
| 67 | Data Bit 58 or 186 | Data Bit 122 or 250 |
| 68 | Data Bit 59 or 187 | Data Bit 123 or 251 |
| 6B | Data Bit 60 or 188 | Data Bit 124 or 252 |
| 6D | Data Bit 61 or 189 | Data Bit 125 or 253 |
| 70 | Data Bit 62 or 190 | Data Bit 126 or 254 |
| 75 | Data Bit 63 or 191 | Data Bit 127 or 255 |
| 01 | Check Bit 0 or 16 | Check Bit 8 or 24 |
| 02 | Check Bit 1 or 17 | Check Bit 9 or 25 |
| 04 | Check Bit 2 or 18 | Check Bit 10 or 26 |
| 08 | Check Bit 3 or 19 | Check Bit 11 or 27 |
| 10 | Check Bit 4 or 20 | Check Bit 12 or 28 |
| 20 | Check Bit 5 or 21 | Check Bit 13 or 29 |
| 40 | Check Bit 6 or 22 | Check Bit 14 or 30 |
| 80 | Check Bit 7 or 23 | Check Bit 15 or 31 |

# Index