

# DECmpp 12000/Sx Model 100

---

## Parallel Disk Array Reference Manual

Part Number: EK-DECAB-RM. B01

**September 1992**

This document provides a complete description of the DECmpp 12000/Sx Parallel Disk Array hardware, its installation and service procedures.

**Revision/Update Information:** This document has been revised for DECmpp Version 1.1.

**Operating System and Version:** ULTRIX Version 4.2A.  
Future releases may require higher versions.

**Software Version:** DECmpp 12000/Sx Version 1.1.

**Digital Equipment Corporation  
Maynard, Massachusetts**

---

**First Printing, January 1992**  
**Revised, September 1992**

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may be used or copied only in accordance with the terms of such license.

**NOTICE—Class A Computing Device:**

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 computing device pursuant to Subpart J or Part 15 of FCC rules, which are designed to provide reasonable protection against such radio frequency interference when operated in a commercial environment. Operation of this equipment in a residential area may cause interference; in which case, measures taken to correct the interference are at the user's expense.

Restricted Rights: Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013.

© Digital Equipment Corporation 1992.

All Rights Reserved.

The postpaid Reader's Comments forms at the end of this document request your critical evaluation to assist in preparing future documentation.

The following are trademarks of Digital Equipment Corporation: DECnet, DECstation, DECsupport, DECsystem, DECwindows, Rdb/VMS, ThinWire, TURBOchannel, ULTRIX, VAX, VAX DOCUMENT, VMS, and the DIGITAL logo.

The following are registered trademarks of the MasPar Computer Corporation: MasPar and the MasPar logo. The following are trademarks of the MasPar Computer Corporation: MasPar Data Display Library (MPDDL), MasPar FORTRAN (MPF), MasPar Input/Output Channel, MasPar Parallel Application Language (MPL), MasPar Parallel Disk Array (MPDA), and MasPar Programming Environment (MPPE). UNIX is a registered trademark of UNIX System Laboratories, Inc.

This document was prepared using VAX DOCUMENT, Version 2.0.

---

# Contents

|   |      |
|---|------|
| <b>Preface</b> .....                                | ix   |
| <b>1 Parallel Disk Array Overview</b>               |      |
| 1.1 Parallel Disk Array Components .....            | 1-5  |
| 1.1.1 Disk Array Controller .....                   | 1-6  |
| 1.1.2 DA Controller Backplane .....                 | 1-8  |
| 1.1.3 DA Backplane .....                            | 1-9  |
| 1.1.3.1 DA Controller Interconnect .....            | 1-11 |
| 1.1.3.2 Disk Canister Interconnect .....            | 1-11 |
| 1.1.3.3 Multiplexer Interconnect .....              | 1-11 |
| 1.1.3.4 Lightpipe Interconnect .....                | 1-11 |
| 1.1.3.5 Power Interconnect .....                    | 1-11 |
| 1.2 Disk Chassis .....                              | 1-12 |
| 1.2.1 Disk Canisters .....                          | 1-12 |
| 1.2.2 MUX Printed Circuit Board .....               | 1-13 |
| 1.2.2.1 Data Multiplexing .....                     | 1-13 |
| 1.2.2.2 Write Data .....                            | 1-13 |
| 1.2.2.3 Sector Pulse Monitor .....                  | 1-13 |
| 1.3 DPU-PDA Interface Cable .....                   | 1-13 |
| 1.4 Power/Fan Tray .....                            | 1-13 |
| 1.4.1 Power System .....                            | 1-15 |
| 1.4.2 Power/Fan Tray Indicators .....               | 1-16 |
| 1.4.3 Exhaust Fan .....                             | 1-17 |
| <b>2 Parallel Disk Array Installation</b>           |      |
| 2.1 Parallel Disk Array Site Preparation .....      | 2-1  |
| 2.1.1 Physical Characteristics .....                | 2-1  |
| 2.1.1.1 Size and Service Clearance .....            | 2-1  |
| 2.1.1.2 Weight .....                                | 2-1  |
| 2.1.2 Power and Cooling Requirements .....          | 2-1  |
| 2.1.3 Power and Environmental Specifications .....  | 2-3  |
| 2.2 PDA Installation .....                          | 2-4  |
| 2.2.1 Unpacking and Positioning the PDA .....       | 2-4  |
| 2.2.1.1 Assembling the Enclosure Skirts .....       | 2-6  |
| 2.2.2 Electrical Installation .....                 | 2-7  |
| 2.2.2.1 Printed Circuit Boards .....                | 2-7  |
| 2.2.2.2 Installing Disk Array Controller PCBs ..... | 2-8  |
| 2.2.3 Installing MUX PCBs .....                     | 2-8  |
| 2.3 Installing Disk Canisters .....                 | 2-8  |
| 2.3.1 Jumpers and Terminators .....                 | 2-8  |
| 2.4 Connecting DPU and PDA .....                    | 2-11 |
| 2.4.1 Installing the DPU PCBs .....                 | 2-11 |

|     |                                      |      |
|-----|--------------------------------------|------|
| 2.5 | Setting DPU Jumpers . . . . .        | 2-15 |
| 2.6 | Turning On the DPU and PDA . . . . . | 2-17 |

### 3 Parallel Disk Array System Administration

|       |   |     |
|-------|---|-----|
| 3.1   | Establishing PDA Communications . . . . . | 3-2 |
| 3.2   | Controller Configuration . . . . .        | 3-5 |
| 3.3   | Setting Up File Systems . . . . .         | 3-7 |
| 3.3.1 | Default Partitions . . . . .              | 3-7 |
| 3.4   | Upgrade Procedures . . . . .              | 3-8 |
| 3.4.1 | Upgrading to 8 Disks . . . . .            | 3-8 |
| 3.4.2 | Upgrading to 16 Disks . . . . .           | 3-8 |

### 4 Disk Formatting

|         |  |     |
|---------|--|-----|
| 4.1     | The FORMAT Command . . . . .   | 4-1 |
| 4.1.1   | Operation Examples . . . . .   | 4-3 |
| 4.1.1.1 | Bring Up a New Bank . . . . .  | 4-3 |
| 4.1.1.2 | Bringing Up a New Bank with Flaw Data on Host . . . . .                  | 4-3 |
| 4.1.1.3 | Formatting Newly Installed Disks Within an Online/Mounted Bank . . . . . | 4-3 |
| 4.1.1.4 | Reformatting Online/Mounted Banks with Growth Errors Logged . . . . .    | 4-4 |
| 4.2     | The MOUNTBANK Command . . . . .  | 4-4 |
| 4.3     | The RECONSTRUCT Command . . . . .  | 4-5 |
| 4.3.1   | Disabling and Replacing a Drive . . . . .                                | 4-6 |
| 4.3.2   | Assigning a Standby Drive . . . . .                                      | 4-6 |
| 4.3.3   | Reconstructing a Drive . . . . .   | 4-6 |
| 4.4     | The SCRUB Command . . . . .  | 4-8 |
| 4.5     | The SPECIFYFLAW Command . . . . .  | 4-9 |

### 5 Removal and Replacement Procedures

|       |  |      |
|-------|--|------|
| 5.1   | Extending the DA Controller Chassis . . . . .        | 5-3  |
| 5.2   | Replacing DA Controller PCBs . . . . .               | 5-5  |
| 5.3   | Replacing Disk Canisters . . . . .                   | 5-6  |
| 5.4   | Replacing MUX PCBs . . . . .                         | 5-7  |
| 5.5   | Replacing the Lightpipe PCB . . . . .                | 5-8  |
| 5.6   | Replacing Power/Fan Tray . . . . .                   | 5-9  |
| 5.7   | Replacing Power Cables to the DA Backplane . . . . . | 5-12 |
| 5.8   | Replacing Clock PCB . . . . .                        | 5-14 |
| 5.9   | Replacing DA Controller Backplane Cabling . . . . .  | 5-15 |
| 5.9.1 | Replacing PDA-DPU Interface Cable . . . . .          | 5-15 |
| 5.9.2 | Replacing Backplane Disk Cables . . . . .            | 5-17 |
| 5.10  | Replacing the Lightpipe Signal Cable . . . . .       | 5-18 |
| 5.11  | Replacing Lightpipe Power Cable . . . . .            | 5-19 |
| 5.12  | Replacing Temperature Sensor Cable . . . . .         | 5-20 |

## 6 Parallel Disk Array Diagnostics

|       |   |      |
|-------|---|------|
| 6.1   | Lightpipe Indicator Status . . . . .                | 6-1  |
| 6.2   | Diagnostic Procedures . . . . .                     | 6-3  |
| 6.3   | Diagnostic Environment . . . . .                    | 6-3  |
| 6.3.1 | Automatic Mode . . . . .                            | 6-4  |
| 6.4   | VME Adapter Tests . . . . .                         | 6-4  |
| 6.5   | PDA Test Programs . . . . .                         | 6-4  |
| 6.5.1 | Internal DA Controller Test (da_intern) . . . . .   | 6-4  |
| 6.5.2 | Read/Write Configuration Test (da_config) . . . . . | 6-4  |
| 6.5.3 | Read/Write Buffer Test (da_buff) . . . . .          | 6-4  |
| 6.5.4 | Mount Bank Test (da_mount) . . . . .                | 6-4  |
| 6.5.5 | Read/Seek Test (da_seek) . . . . .                  | 6-5  |
| 6.5.6 | Read/Write Disk Test (da_disk) . . . . .            | 6-5  |
| 6.6   | DA Controller Tests . . . . .                       | 6-5  |
| 6.7   | Status Operations . . . . .                         | 6-5  |
| 6.7.1 | Disk Error Logging . . . . .                        | 6-5  |
| 6.7.2 | Error Information . . . . .                         | 6-7  |
| 6.7.3 | Extended Transfer Error Status . . . . .            | 6-9  |
| 6.7.4 | Command/Status Summary . . . . .                    | 6-10 |

## 7 Upgrading the Parallel Disk Array

|     |  |     |
|-----|--|-----|
| 7.1 | Upgrading to Eight Disk Drives . . . . .   | 7-2 |
| 7.2 | Upgrading to Sixteen Disk Drives . . . . . | 7-3 |

## A Configuration Data

|       |   |      |
|-------|---|------|
| A.1   | The READCONFIG Command . . . . .                    | A-1  |
| A.2   | The WRITECONFIG Command . . . . .                   | A-2  |
| A.3   | System Configuration Table . . . . .                | A-3  |
| A.3.1 | Controller Configuration . . . . .                  | A-4  |
| A.3.2 | Bank Control/Status Configuration . . . . .         | A-10 |
| A.3.3 | Disk Parameter Table Overview . . . . .             | A-15 |
| A.3.4 | Bad Block Table . . . . .                           | A-18 |
| A.3.5 | Bank X, Disk X, System Area Configuration . . . . . | A-19 |
| A.3.6 | System Area Home Page . . . . .                     | A-20 |
| A.3.7 | System Area RFT and GET Pages . . . . .             | A-21 |

## B Parallel Disk Array System Block Diagrams

|     |   |     |
|-----|---|-----|
| B.1 | Parallel Disk Array System Block Diagrams . . . . . | B-1 |
|-----|---|-----|

## C Recommended Spares List

## D Disk Array Variations and Options

## E Parallel Disk Array Reference Pages

|                    |      |
|--------------------|------|
| dabinit(8)         | E-2  |
| dabstat(8)         | E-3  |
| dadisk(8)          | E-4  |
| dadsbl(8)          | E-5  |
| daflaw(8)          | E-6  |
| daformat(8)        | E-8  |
| dainit(8)          | E-10 |
| damapchs(8)        | E-11 |
| damount(8)         | E-12 |
| darecon(8)         | E-13 |
| dascrub(8)         | E-14 |
| daspecflaw(8)      | E-16 |
| dastat(8)          | E-17 |
| dasubs(8)          | E-18 |
| daxformat(8)       | E-19 |
| daxrdc(8)          | E-20 |
| daxscrub(8)        | E-21 |
| daxwrc(8)          | E-22 |
| da_error_log(8)    | E-23 |
| da_error_report(8) | E-24 |

## Index

## Figures

|      |  |      |
|------|--|------|
| 1-1  | DECmpp 12000/Sx Parallel Disk Array      | 1-2  |
| 1-2  | Parallel Disk Array, Model SMP11-AA      | 1-3  |
| 1-3  | Parallel Disk Array, Model SMP11-AB      | 1-4  |
| 1-4  | Parallel Disk Array, Model SMP11-AC      | 1-5  |
| 1-5  | DA Controller Front Panel                | 1-6  |
| 1-6  | DA Controller Slot Assignments           | 1-7  |
| 1-7  | DA Controller Backplane                  | 1-8  |
| 1-8  | DA Drive Backplane (front)               | 1-9  |
| 1-9  | DA Drive Backplane (rear)                | 1-10 |
| 1-10 | Disk Drive Canister                      | 1-12 |
| 1-11 | Power/Fan Tray                           | 1-14 |
| 1-12 | Power Bridge and Cables                  | 1-16 |
| 1-13 | Power/Fan Tray Service Indicators        | 1-16 |
| 2-1  | Parallel Disk Array Enclosure Dimensions | 2-2  |
| 2-2  | Removing Enclosure Shipping Bracket      | 2-4  |
| 2-3  | PDA Enclosure Levelers                   | 2-5  |
| 2-4  | Front and Side Skirts                    | 2-6  |
| 2-5  | Rear Skirt                               | 2-7  |
| 2-6  | Disk Drive PCB (Component Side)          | 2-10 |
| 2-7  | Inserting Disk Canister                  | 2-11 |

|      |   |      |
|------|---|------|
| 2-8  | DPU Card Cage . . . . .                       | 2-12 |
| 2-9  | DA Interface PCB Cable Connections . . . . .  | 2-13 |
| 2-10 | Data Parallel Unit Card Cage Slots . . . . .  | 2-14 |
| 2-11 | DPU Backplane Jumper Locations . . . . .      | 2-16 |
| 5-1  | PDA Enclosure (rear) . . . . .                | 5-2  |
| 5-2  | Extending the Disk Array Controller . . . . . | 5-4  |
| 5-3  | Replacing the DA Controller PCB . . . . .     | 5-5  |
| 5-4  | Power/Fan Tray (rear) . . . . .               | 5-10 |
| 5-5  | Replacing the Power/Fan Tray . . . . .        | 5-11 |
| 5-6  | Power Bridge and Cables . . . . .             | 5-13 |
| 5-7  | Replacing the Clock PCB . . . . .             | 5-14 |
| 6-1  | PDA Front Indicators . . . . .                | 6-2  |
| B-1  | Full System Block Diagram . . . . .           | B-1  |
| B-2  | Disk Array Controller Block Diagram . . . . . | B-2  |
| B-3  | Disk Array Backplane Block Diagram . . . . .  | B-2  |

## Tables

|      |  |      |
|------|--|------|
| 1    | Related Documents . . . . .                                | x    |
| 1-1  | PDA Configurations . . . . .                               | 1-2  |
| 1-2  | DA Backplane Connectors . . . . .                          | 1-10 |
| 1-3  | Power System Capabilities . . . . .                        | 1-15 |
| 1-4  | Backplane Power Requirements . . . . .                     | 1-15 |
| 1-5  | Power/Fan Tray Service Indicators . . . . .                | 1-17 |
| 2-1  | PDA Specifications . . . . .                               | 2-3  |
| 4-1  | Bank Configurations . . . . .                              | 4-7  |
| 6-1  | Diagnostic Programs . . . . .                              | 6-3  |
| 6-2  | Status Bit Definitions . . . . .                           | 6-7  |
| 6-3  | Command-Related Error Codes . . . . .                      | 6-10 |
| A-1  | READCONFIG Command Status . . . . .                        | A-2  |
| A-2  | System Configuration Byte Organization . . . . .           | A-3  |
| A-3  | Controller Configuration Block Byte Organization . . . . . | A-4  |
| A-4  | Controller Configuration Block Bit Description . . . . .   | A-5  |
| A-5  | Disk Error Log Select Bit Descriptions . . . . .           | A-7  |
| A-6  | 32-Byte Bank Control/Status Block . . . . .                | A-10 |
| A-7  | #2 Byte Bank Control/Status Block Description . . . . .    | A-11 |
| A-8  | Error Checking and Correcting Bit Descriptions . . . . .   | A-13 |
| A-9  | Bit Status and Conditions . . . . .                        | A-14 |
| A-10 | Disk Parameter Table Byte Organization . . . . .           | A-15 |
| A-11 | Drive Type and Flag Bit Definitions . . . . .              | A-16 |
| A-12 | Values for Disk Parameter Table . . . . .                  | A-17 |
| A-13 | Bad Block Table Byte Organization . . . . .                | A-18 |
| A-14 | Home Page Byte Descriptions . . . . .                      | A-20 |
| A-15 | Home Page Byte Group Descriptions . . . . .                | A-20 |
| C-1  | Parallel Disk Array RSL . . . . .                          | C-1  |
| D-1  | DECmpp Disk Array Variations . . . . .                     | D-1  |





---

# Preface

The SMP11-Ax Parallel Disk Array (PDA) is a random access, hard disk storage device designed to complement the DECmpp 12000/Sx and DECmpp 12000-LC/Sx Series Massively Parallel Processing systems.

This document provides a description of the Parallel Disk Array hardware and how to install and service it. Anyone who services the PDA and needs to replace any components in it should read this manual and be familiar with the procedures.

## Intended Audience

This guide is for use by Digital Services personnel and by self-maintenance customers who will be servicing the PDA system. It may also be used by PDA system administrators.

## Document Structure

This manual contains seven chapters and five appendixes:

- Chapter 1 is a general overview of the PDA configurations and components.
- Chapter 2 contains installation procedures, including connecting the PDA to the DECmpp 12000/Sx system data parallel unit (DPU) and powering up the PDA.
- Chapter 3 describes system administration procedures for bringing the PDA online.
- Chapter 4 explains the disk formatting instructions.
- Chapter 5 explains how to replace the various PDA field replaceable units.
- Chapter 6 explains the indicator display and the diagnostic procedures and tests.
- Chapter 7 describes procedures for upgrading the PDA to larger storage configurations.
- Appendix A contains PDA configuration data tables.
- Appendix B contains system block diagrams.
- Appendix C lists the PDA recommended spare parts.
- Appendix D lists PDA option variants.
- Appendix E contains PDA-specific reference pages.

## Related Documents

Table 1 is a list of documents that provide additional information about the PDA and the DECmpp 12000/Sx system.

**Table 1 Related Documents**

| <b>Document Title</b>                                | <b>Order Number</b> |
|--|---------------------|
| <i>DECmpp 12000/Sx System Overview Manual</i>        | AA-PMAPB-TE         |
| <i>DECmpp 12000/Sx System Administration Guide</i>   | AA-PKU3C-TE         |
| <i>DECmpp 12000/Sx Parallel VME Reference Manual</i> | EK-DECAB-PM         |
| <i>DECmpp 12000/Sx Hardware Installation Guide</i>   | EK-DECAC-IG         |

## Conventions

The following conventions are used throughout the DECmpp 12000/Sx documentation set.

| Convention                      | Meaning  |
|---------------------------------|--|
| <code>Return</code>             | In examples, a key name shown within a box indicates that you press a key on the keyboard. In text, a key name is not enclosed in a box but is printed with an initial capital letter, like Return.  |
| <code>Ctrl/x</code>             | A key combination, shown with a slash separating the two key names, indicates that you hold down the first key while you press the second key.   |
| <code>MB1, MB2, MB3</code>      | The buttons on a mouse. MB1 is the left button, MB2 is the center button, and MB3 is the right button of a mouse whose button arrangement is right-handed. It is possible to redefine the mouse buttons.   |
| <code>%</code>                  | A percent sign (%) represents the default user prompt for your system.   |
| <code>#</code>                  | A number sign (#) represents the default superuser (root) prompt for your system.  |
| <code>...</code>                | In examples, a horizontal series of dots, or ellipsis, indicates that additional parameters, values, or other information can be entered.  |
| <code>:</code>                  | In examples, a vertical series of dots, or vertical ellipsis, indicates that a portion of the example is intentionally omitted.  |
| <code>[]</code>                 | In syntax descriptions and functional descriptions, brackets indicate optional items.  |
| <code>dpumanager(6)</code>      | Cross-references to the <i>ULTRIX Reference Pages</i> , which include the appropriate section number in parentheses.   |
| <i>italicized text</i>          | In examples, italicized text denotes parameters, values, or other information that will change from either session to session or user to user. In text, italicized words or phrases are used to add emphasis to important words, concepts, or titles of manuals. |
| ULTRIX keywords                 | This typeface is used to indicate system output or the exact name of a command, option, partition, pathname, directory, or file.   |
| Code examples                   | This typeface is used to display program coding examples.  |
| UPPERCASE and lowercase strings | The ULTRIX system differentiates between lowercase and uppercase characters. Literal strings that appear in text, examples, syntax descriptions, and function descriptions must be entered exactly as shown.   |

Three types of notes are used in this manual:

- **Note**  
Gives additional information or information particularly important to the procedure.
- **Caution**  
Indicates potential damage to equipment or data.
- **Warning**  
Indicates potential injury to a person.

---

## Parallel Disk Array Overview

The DECmpp 12000/Sx system includes a massively parallel, data-parallel processor (the data parallel unit, or DPU) and a front-end server (scalar) processor running ULTRIX. In a separate enclosure connected to the DPU, the Parallel Disk Array (PDA), shown in Figure 1-1, provides a parallel, multidisk architecture, consisting of a disk array (DA) controller and an array of disk drives providing the bulk data storage. The disk array comprises 5-1/4 inch ESDI disk drives, arranged in banks of 4, 8, or 16 data disks, with up to 2 associated parity disks. An optional feature is a single hot standby disk, enhancing the PDA fault-tolerant operation.

The disk array is a scalable I/O device for bulk data storage. (Architecturally, it is similar to a conventional BSD 4.2 UNIX file system.) The DA controller provides a high-speed data connection to the disk array. A VME control port allows an external VME agent to perform additional basic control functions. A diagnostic port allows for debugging. Front panel indicators provide status for both DA system power and individual disks.

The PDA provides the DECmpp 12000/Sx with parallel access to large data files and is fault-tolerant, using parity disks and 48-bit error correction code (ECC) on each disk. Failed disks can be replaced during system operation, and the PDA minimizes performance degradation on active disks while reconstructing data on the new disk.

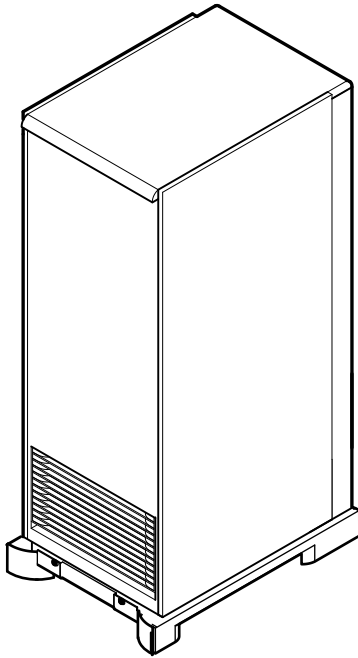
The PDA treats the entire disk array as a single, logical storage device. You can create individual files of up to 2 Gbytes. Many smaller files may be stored across multiple disks. Each formatted data disk provides 1.45 Gbytes of storage, with up to 23 Gbytes of formatted capacity in a 16-disk configuration.

Disk-drive replacement procedures and configuration upgrades require a minimum amount of effort. Each disk drive is in a modular disk canister, which is easily installed or removed. The canisters in one configuration can often be reused in a larger system, and most canisters do not have to be moved from their current positions.

The chassis for expansion to a full complement of disk drive canisters is included in all PDA configurations to permit ease of upgrading storage capacity.

Table 1-1 describes the three PDA configurations. These configurations are shown in Figure 1-2, Figure 1-3, and Figure 1-4.

**Figure 1–1 DECmpp 12000/Sx Parallel Disk Array**

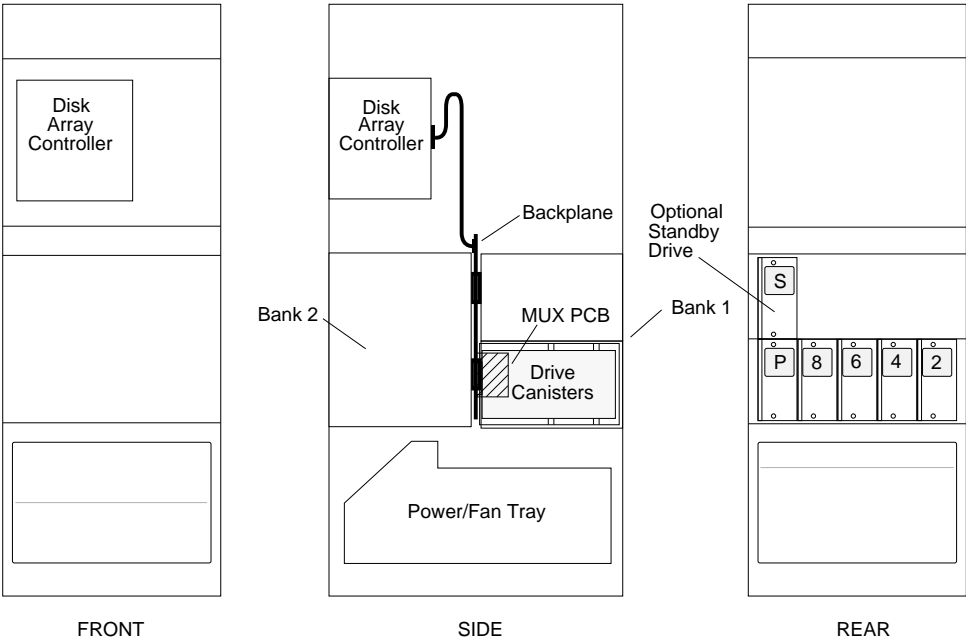


MKV-040000314-82-DG

**Table 1–1 PDA Configurations**

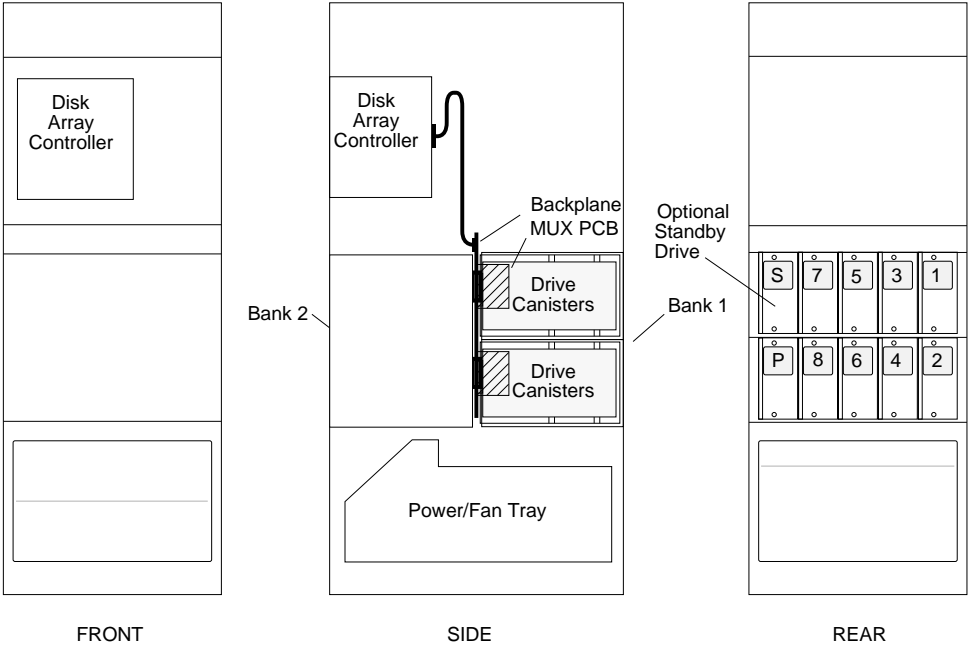
| <b>MODEL</b>               | <b>SMP11-AA</b> | <b>SMP11-AB</b> | <b>SMP11-AC</b> |
|----------------------------|-----------------|-----------------|-----------------|
| Disk Banks                 | 1               | 1               | 2               |
| Data + Parity Disks        | 4 + 1           | 8 + 1           | 16 + 2          |
| Total Disks                | 5               | 9               | 18              |
| Hot Standby (optional)     | 1               | 1               | 1               |
| Total Unformatted Capacity | 6 Gbytes        | 12 Gbytes       | 24 Gbytes       |
| Total Formatted Capacity   | 5.8 Gbytes      | 11.5 Gbytes     | 23 Gbytes       |

**Figure 1–2 Parallel Disk Array, Model SMP11-AA**



MKV-04000314-70-MPS

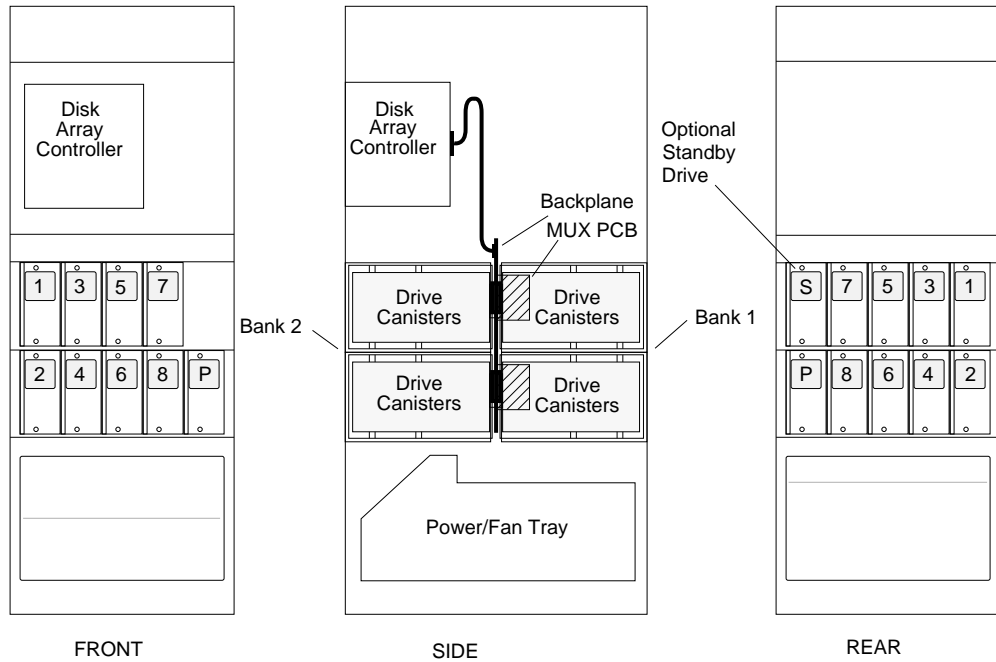
**Figure 1-3 Parallel Disk Array, Model SMP11-AB**



MKV-040000314-71-MPS



**Figure 1–4 Parallel Disk Array, Model SMP11-AC**



MKV-040000314-72-MPS

## 1.1 Parallel Disk Array Components

The PDA has the following major components, which are described individually in more detail in the sections that follow:

- Disk array controller — Contains printed circuit boards (PCBs) controlling data and parity disk drives; provides a link to the DPU backplane through the controller's CPU PCB
- Disk array backplane — Provides power and signal interconnects between disk canisters and DA controller backplane
- Disk chassis — Contains banks of disk canisters
- Power/fan tray — Contains DC power supplies and a cooling fan

The disk array (DA) controller chassis mounts in the upper-front area of the PDA enclosure. The disk drive/canister chassis and DA backplane mounts in the middle of the enclosure. It is 38.86 cm (15.3 inches) high, 48.26 cm (19 inches) wide, and 0.32 cm (0.125-inch) thick.

The power/fan tray at the bottom of the enclosure contains the DC power supply and the cooling fan.

## 1.1 Parallel Disk Array Components

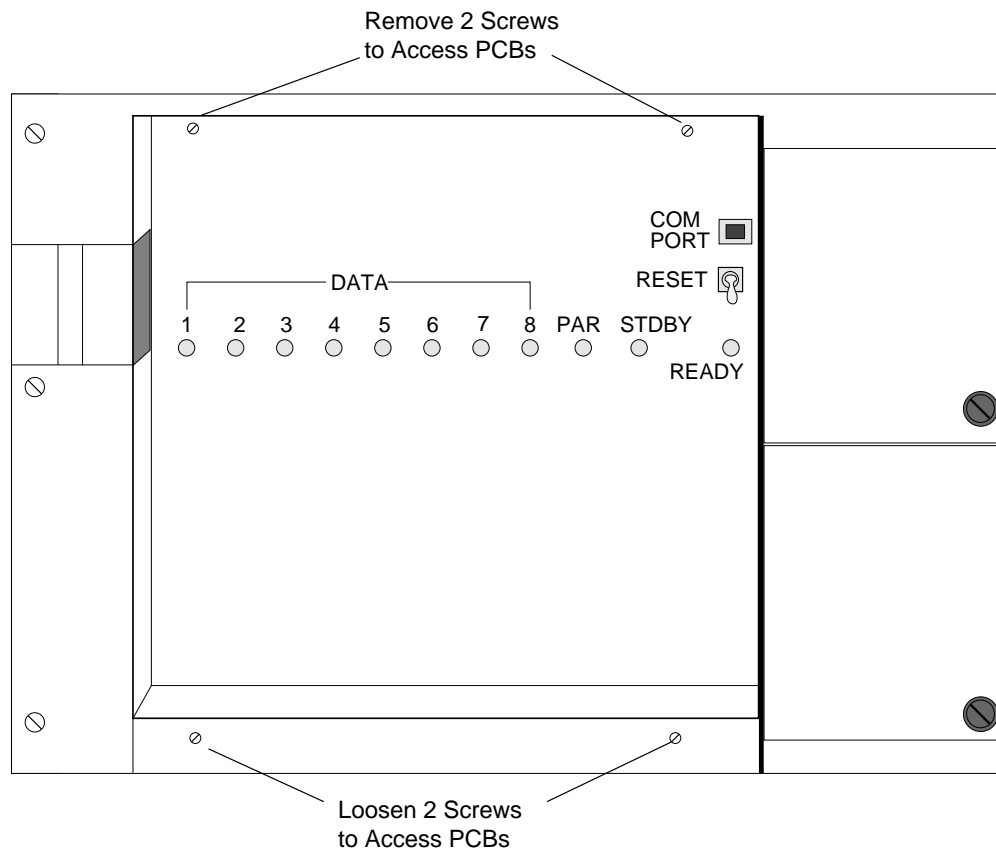
### 1.1.1 Disk Array Controller

The intelligent DA disk controller stores and retrieves data from banks of either four or eight data disks through printed circuit board (PCBs) connected to its embedded backplane. Each data PCB can control up to three data disks. Depending on the configuration, the DA controller can have up to 8 PCBs controlling up to two banks of data disks; a bank can have four or eight data disks. Each bank also has one parity disk to support the data disks.

Figure 1–5 shows the DA controller indicators and connectors:

- 10 disk indicators — red during initialization; green when the disk is on line
- Ready indicator (CPU PCB) — red during initialization; green when the controller is on line
- Reset switch — Forces a powerup system reset (never used during normal operation)
- Com port — RS-232 serial communication port for offline diagnostics, online system configuration, maintenance, and controller status monitoring

Figure 1–5 DA Controller Front Panel



MKV-040000314-100-MPS

## 1.1 Parallel Disk Array Components

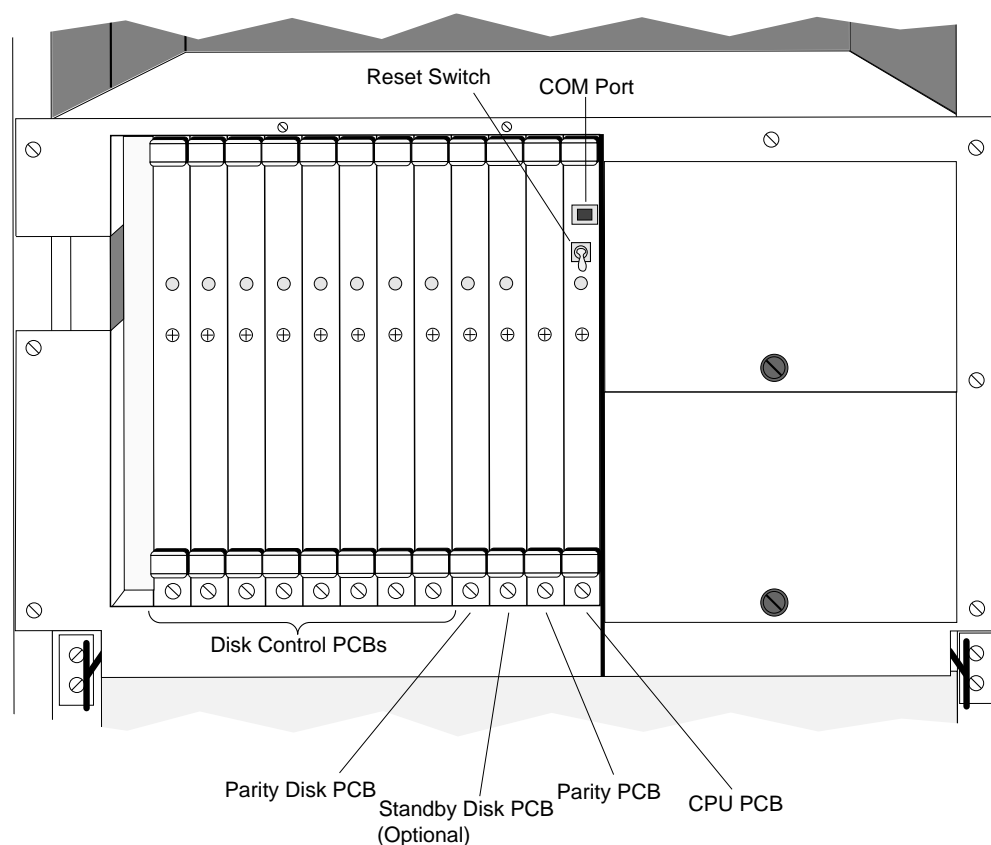
Figure 1–6 shows the PCB slot assignments in the DA controller backplane:

- Slots 1–8: Disk control PCBs
- Slot 9: Parity disk PCB
- Slot 10: Standby disk PCB
- Slot 11: Parity PCB
- Slot 12: Central processing unit (CPU) PCB

The DA controller integrates a master, high-speed interface (HSI) unit on the CPU and Parity PCBs.

The DA controller mounts on the PDA enclosure rails with slide guides. Hinges allow the fully extended DA controller to rotate forward 90 degrees, providing access to its backplane and connectors. The DA controller is 38.1 cm (15 inches) high, 42.55 cm (16.75 inches) wide, and 28.58 cm (11.25 inches) deep. Two 4-1/4 inch fans mounted on top of the DA controller card cage provide cooling for the DA controller PCBs.

**Figure 1–6 DA Controller Slot Assignments**



MKV-04000314-73-MPS

## 1.1 Parallel Disk Array Components

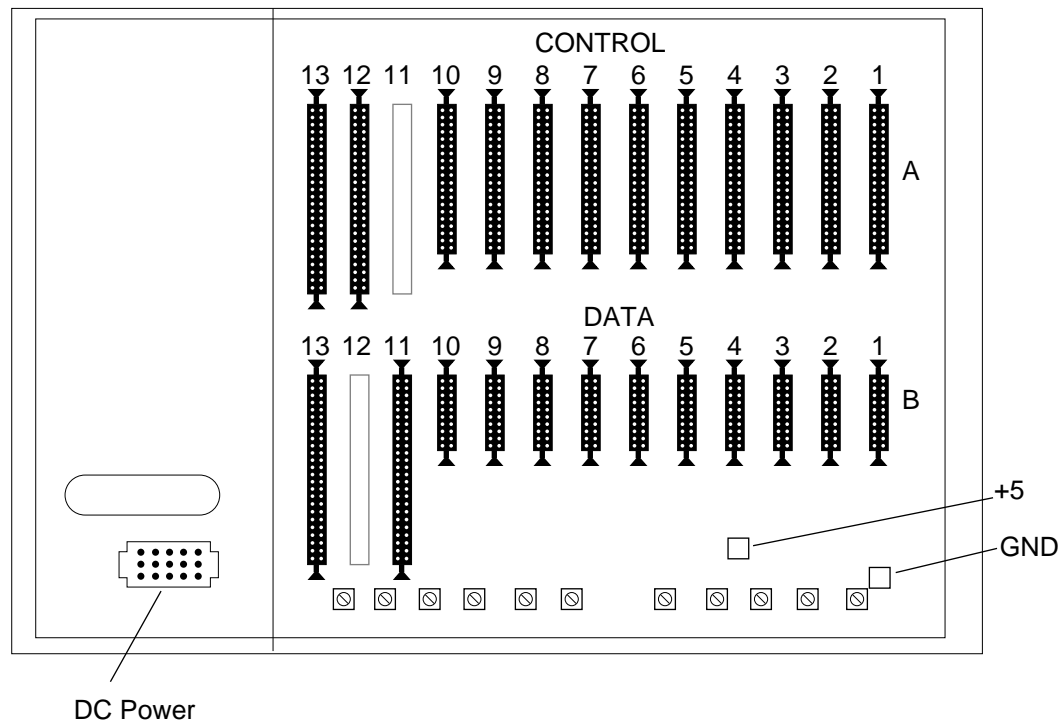
### 1.1.2 DA Controller Backplane

The DA controller backplane (Figure 1–7) has the following connector blocks:

- A1—A10: 34-pin control cable (40-pin connector) to DA backplane connectors *JnP1*
- B1—B10: 20-pin data cable to DA backplane connectors *JnP2*
- A12: 50-pin cable connector conveying control information (command/status) to/from DA interface PCB in DPU
- B11: 50-pin cable connector conveying data to/from DA interface PCB in DPU

Behind the DA controller, two cable clamps and a gravity guide bar hold the cables to the DA backplane in place and support the cable slack necessary to allow the DA controller to rotate forward.

Figure 1–7 DA Controller Backplane

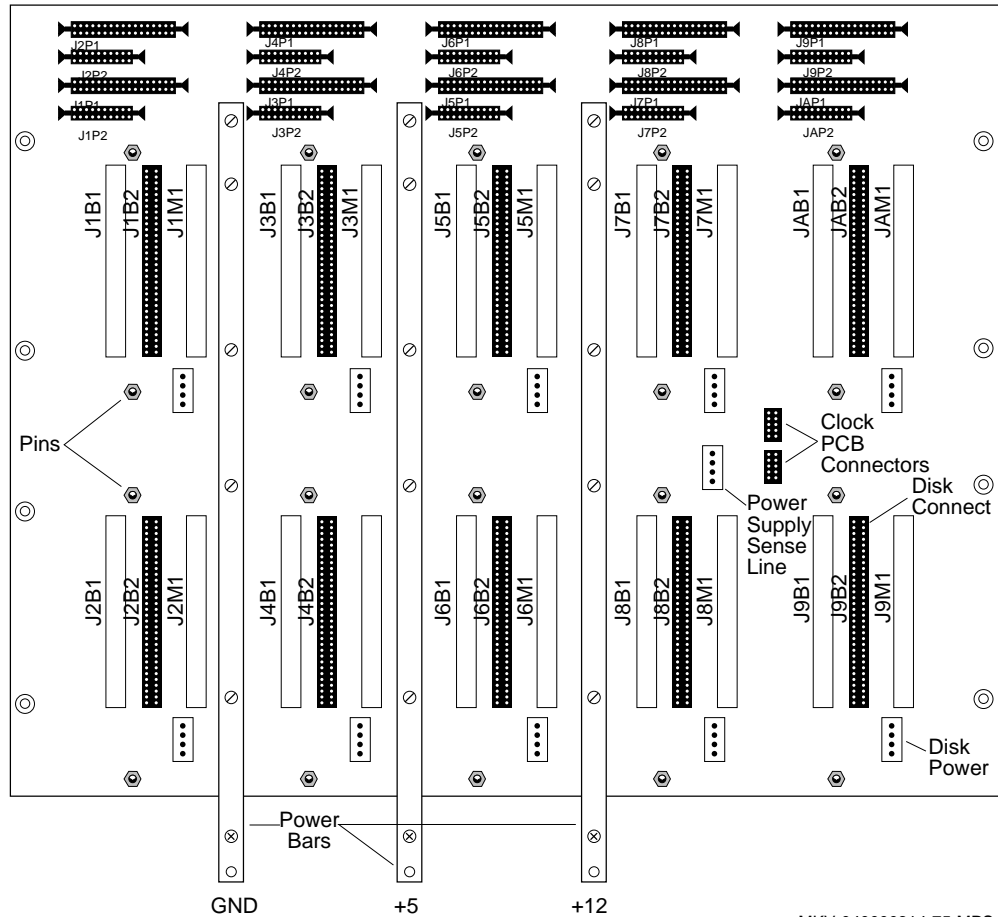


## 1.1 Parallel Disk Array Components

### 1.1.3 DA Backplane

To support the various PDA configurations, the DA backplane has disk signal and power connectors on both the front and rear sides. The multiplexer (MUX) connectors are on the DA backplane rear side. The clock PCB is also on the DA backplane rear side. Figure 1–8 and Figure 1–9 show both sides of the DA backplane.

Figure 1–8 DA Drive Backplane (front)



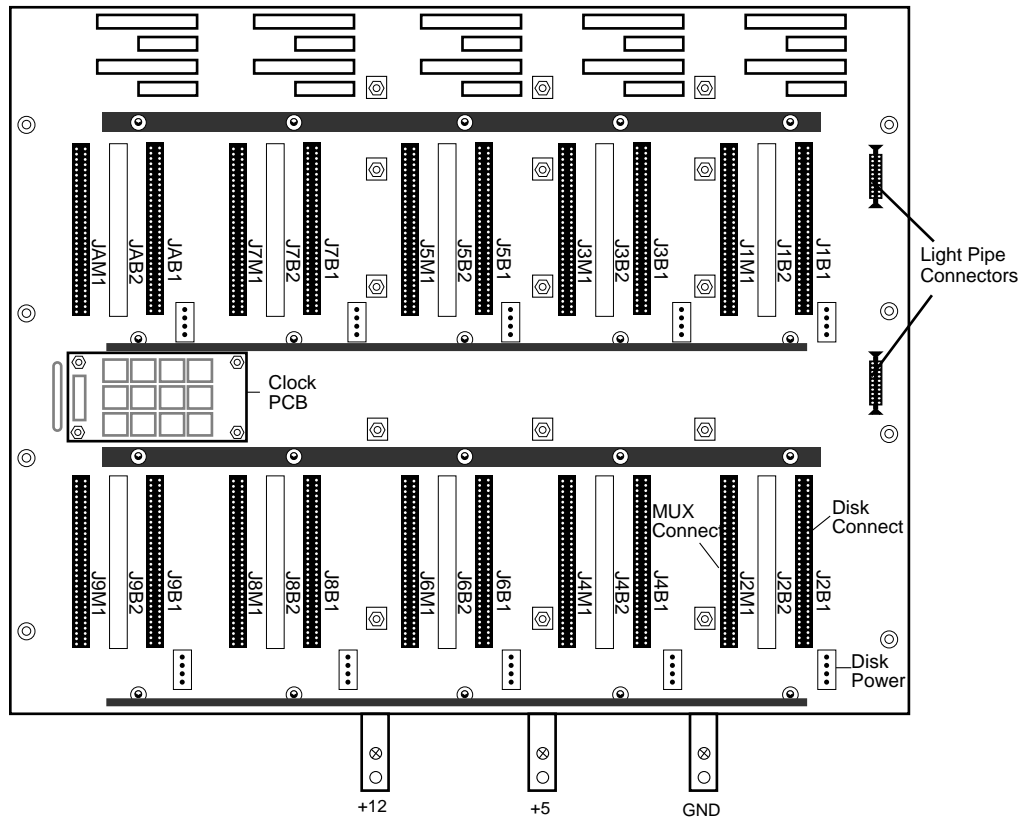
MKV-040000314-75-MPS

The controller assignments for each DA backplane connector correspond to the ESDI J1/P1, J2/P2 pin assignments. The read/write data and clocks, drive-selected, and command-complete signals from the DA controller route through the MUX interconnects. All other signals route directly to the disk canister interconnects.

The DA backplane provides the following system interconnects: array controller, disks, MUX, lightpipe, and power.

## 1.1 Parallel Disk Array Components

Figure 1–9 DA Drive Backplane (rear)



MKV-040000314-76-MPS

Table 1–2 lists the DA backplane connectors.

Table 1–2 DA Backplane Connectors

| Type       | Quantity | Description  |
|------------|----------|--|
| Disk       | 20       | Data signals; 10 on each side of backplane (only 9 are used on the front side) |
| MUX        | 10       | Data and status signals; DA backplane front                                    |
| J1         | 10       | Command cable from DA controller C1; DA backplane front                        |
| J2         | 10       | Data cable from DA controller D1; DA backplane front                           |
| Power      | 20       | Disk power; 10 on each side of backplane (only 9 are used on the front side)   |
| Sense line | 1        | Power sensor from disks to power supply; DA backplane front                    |
| Lightpipe  | 2        | Disk status to front panel indicators; DA backplane front                      |

## 1.1 Parallel Disk Array Components

### 1.1.3.1 DA Controller Interconnect

The front of the DA backplane provides connections for two sets of cables from the DA controller backplane:

- 34-pin command cables
- 20-pin data cables

### 1.1.3.2 Disk Canister Interconnect

Disk canisters plug directly into the 64-pin connectors. (Sixty pins are used.) The 4-pin power connectors supply DC voltage to the disk canisters. The DA backplane interconnects support 20 disk canisters (19 canisters are used) with the following:

- Disk signals to/from backplane
- Disk-to-disk signals (spindle lock)
- Disk power signals

### 1.1.3.3 Multiplexer Interconnect

The MUX PCBs plug directly into the DA backplane 64-pin connectors. Read/write data and clocks, disk-selected, and command-complete signals route here from the DA controller interconnects and from the disk interconnects. Status lines to the lightpipe interconnects originate here.

The DA backplane interconnects provide 10 MUX PCBs with the following:

- Data to/from the disk
- Data to/from the controller
- Drive status signals to the lightpipe and disk-panel indicators

### 1.1.3.4 Lightpipe Interconnect

The DA backplane has two 10-pin lightpipe connectors. The split ribbon cable carrying disk status signals to the lightpipe has two separate 10-pin connectors at one end and a single 50-pin connector on the other, which connects to the PDA front panel lightpipe. Each lightpipe connector is dedicated to signals from one side of the DA backplane. For more detail on the lightpipe displays, refer to Chapter 6.

### 1.1.3.5 Power Interconnect

A system of bus bars supplies the necessary DC voltage to the +12 VDC, +5 VDC, and ground planes within the DA backplane. A 4-pin connector provides power to the disk.

A 4-pin connector provides the sense line connection back to the power supply.

## 1.2 Disk Chassis

### 1.2 Disk Chassis

The rack-mounted, sheet metal disk chassis is similar to a card cage, containing modular disk canisters instead of PCBs. The disk chassis has 10 disk canister slots, arranged five-wide and two-high. The modular disk canisters slide into the disk chassis on standard nylon card guides and connect to the DA backplane behind the disk chassis.

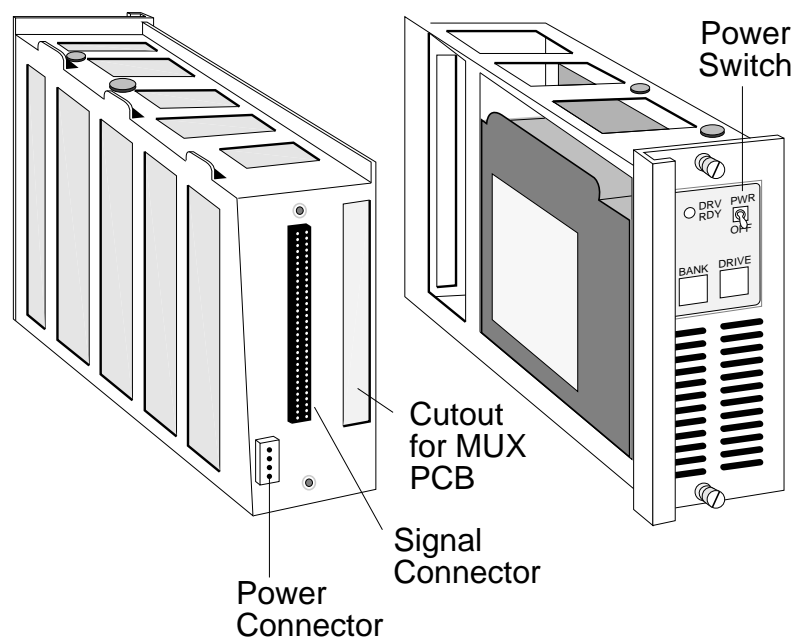
The disk chassis is 38.1 cm (15 inches) high, 43.18 cm (17 inches) wide, and 30.48 cm (12 inches) deep.

#### 1.2.1 Disk Canisters

Each modular sheet metal canister contains a 5-1/4 inch ESDI disk drive with 1450 Mbyte formatted capacity. Figure 1-10 shows a disk canister. Front panel labels identify the disk canister bank number and disk number. The disk canister front panel also has a power switch, controlling the +12 and +5 voltage to the canister. The disk canister rear panel has two panel-mounted connectors:

- 64-pin signal connector
- 4-pin power connector

Figure 1-10 Disk Drive Canister



MKV-04000314-77-MPS

The indicator on the front of the disk canister is green when the disk is spinning and off when the disk is not spinning. (The DA controller controls the disk status.)



### 1.2.2 MUX Printed Circuit Board

Using the MUX PCBs, the DA controller can access up to two banks of disks in any one PDA enclosure. The MUX PCBs also provide disk status to the PDA lightpipe and to the disk canister indicators.

The MUX circuitry can be divided into five functional blocks:

- Read data, read clock, drive selected, and command complete multiplexing
- Write data and clocks from array controller to disks
- Provide status to lightpipe indicators
- Provide status to disk canister indicator
- Provide disk sector-pulse monitoring

#### 1.2.2.1 Data Multiplexing

The MUX circuitry multiplexes four groups of signals, two to one:

- Read data (rd)
- Read clocks (rc)
- Command complete (cc\_)
- Drive selected (ds\_)

The MUX receives read data and read clocks from two different banks, determines which bank is selected, and passes the appropriate read data and clocks to the controller.

#### 1.2.2.2 Write Data

Write data and clock signals from the array controller are routed to the bank selected by the array.

#### 1.2.2.3 Sector Pulse Monitor

The sector monitor is a retriggerable monostable multivibrator. Sector pulses from the disk keep the output active. If the sector pulse is lost, the output goes low, signaling that the disk is spun down.

## 1.3 DPU-PDA Interface Cable

Two 50-pin DPU-PDA interface cables interface the DA controller to the VME 6U interface PCB (mounted on a Parallel VME 6U Adapter PCB) in the DPU. One cable carries control information (command/status); the other cable carries data signals.

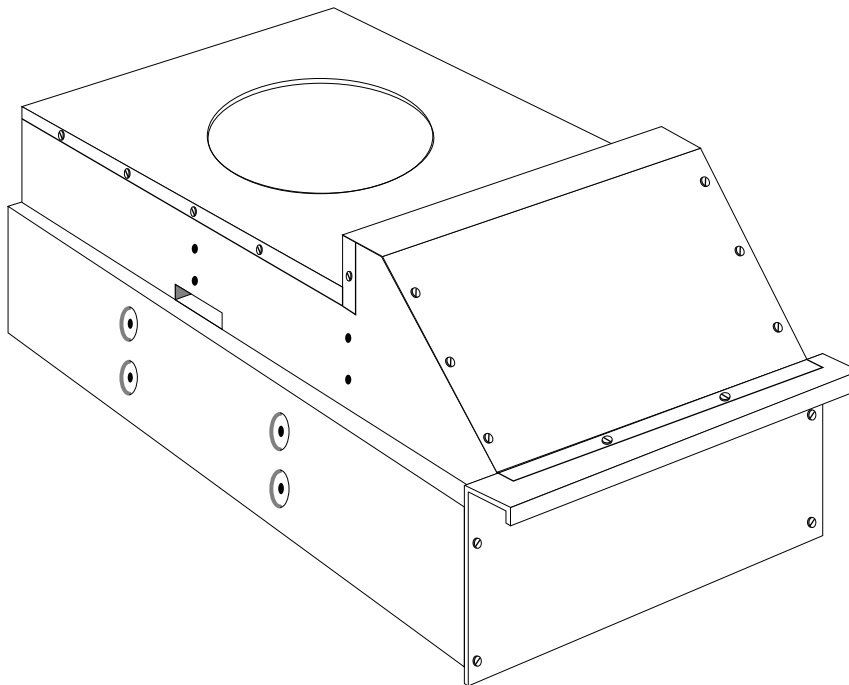
## 1.4 Power/Fan Tray

The power/fan tray mounted in the bottom of the enclosure contains the complete power subsystem and the exhaust fan, including AC power connection, 20 A and 1 A circuit breakers. Figure 1–11 provides front and side views of the power/fan tray.

A set of power distribution connectors allows the complete tray to slide in and out of the enclosure without having to bolt or unbolt power cables. Only the AC power cable, the temperature sensor cable, and the lightpipe power cable need to be detached before removing the tray.

## 1.4 Power/Fan Tray

Figure 1–11 Power/Fan Tray



MKV-04000314-78-MPS

The power selector switch is on the power/fan tray rear panel, inside the PDA enclosure at the rear. It is a toggle switch that must be pulled out slightly before it can be moved. It has three positions:

- **OVERRIDE** (straight up): Powers on the PDA, regardless of other conditions
- **LOCAL** (middle position): Powers on the PDA independent of the DPU
- **REMOTE** (lower position): Not used

---

**Warning**

---

DO NOT USE THE OVERRIDE SETTING. It overrides critical safety systems. The override position is for factory use only.

---

All PDA configurations use the same power/fan tray. Its rating is:

220 Vac, 20A, 2500 W

Table 1–3 shows the power capabilities of the PDA's DC power supply.

**Table 1–3 Power System Capabilities**

| Voltage  | Current |
|----------|---------|
| +12 Vdc  | 80 A    |
| –12 Vdc  | 20 A    |
| +5 Vdc   | 200 A   |
| –5.2 Vdc | 25 A    |

The power/fan tray is 35.56 cm (14 inches) high, 43.18 cm (17 inches) wide, and 60.96 cm (24 inches) deep. It weighs approximately 41.73 kg (92 pounds).

### 1.4.1 Power System

The power system consists of the power-sequencing PCB, the DC power supplies, and power distribution subsystem.

The bus bar system attaches to a high current plug and carries +5 Vdc, +12 Vdc, and ground to the backplane power plates. The –5.2 Vdc and sense lines attach to a low current plug.

DC power distribution begins with a quick disconnect between the power/fan tray and the backplane power cables. The quick disconnect consists of high and low current plugs. One-half of both connectors resides in the power/fan tray (with associated harnesses). The other half of both connectors is attached to a frame member and has a harness and backplane bus bars attached to it.

The backplane supplies power to the disk drives and the MUX PCBs. Table 1–4 shows the backplane power requirements.

**Table 1–4 Backplane Power Requirements**

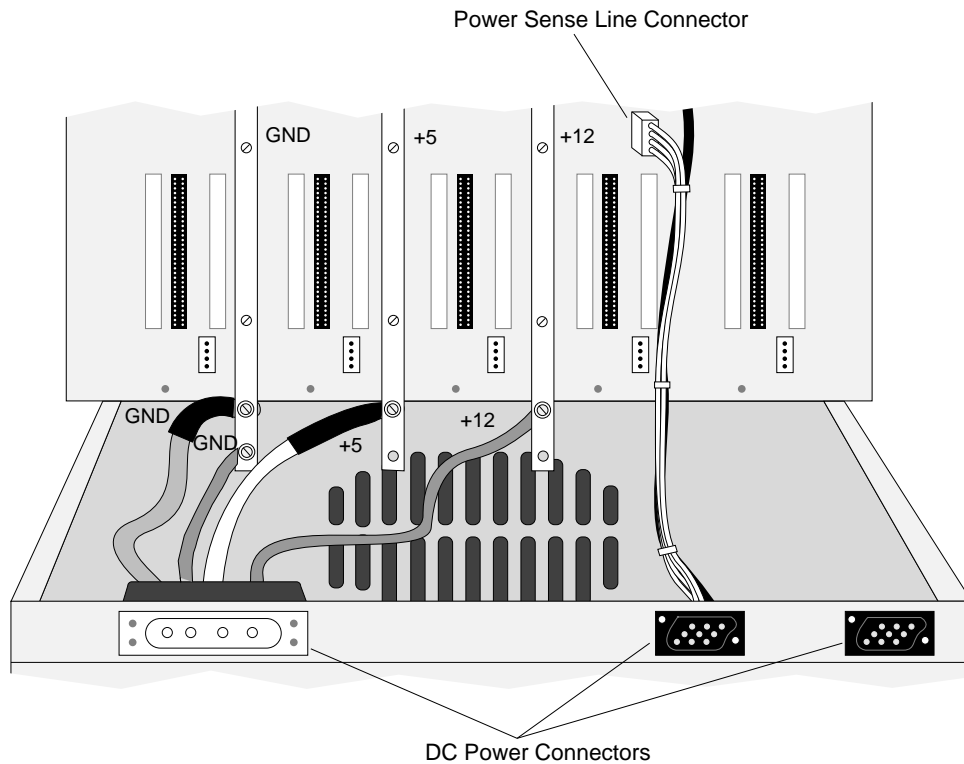
| Purpose          | Volts | Amps | Watts |
|------------------|-------|------|-------|
| Multiplexer PCBs | 5     | 10   | 50    |
| Disk Drives      | 5     | 29   | 145   |
|                  | 12    | 51   | 612   |
| Total            |       |      | 807   |

The chassis ground circuit detects differences in potential between chassis ground and logic ground and shuts down the power/fan tray when the potential exceeds a preset 70 mV threshold.

Figure 1–12 shows the power cables to the DA backplane.

## 1.4 Power/Fan Tray

Figure 1–12 Power Bridge and Cables



MKV-040000314-79-MPS

Figure 1–13 Power/Fan Tray Service Indicators

- Chassis Ground
- Temperature
- Main Power Vcc +5 V
- Main Power Vee -5 V
- Main Power +12 V
- Main Power -12 V
- AUX Power Vcc +5 V
- AUX Power Vee -5 V
- AUX Power +12 V
- AUX Power -12 V

MKV-040000314-55-RAGS

### 1.4.2 Power/Fan Tray Indicators

Figure 1–13 shows the ten indicators located on the PDA power/fan tray rear panel. These indicators provide information about the power/fan tray. They show the status of some part of the power/fan tray and are green during normal operation. If a failure is detected, the corresponding indicator changes to red, and the power tray shuts down. Any red indicators are latched ON, indicating the problem area, and initiate the power shutdown.

Table 1–5 lists the function of each power/fan tray indicator (from the top).

**Table 1–5 Power/Fan Tray Service Indicators**

| Indicator                | Function   |
|--------------------------|--|
| Chassis Ground           | Green — Normal<br>Red — Logic ground to chassis ground short; excess voltage detection |
| Temperature              | Green — Normal<br>Red — Excess temperature   |
| Main power Vcc +5 V      | Green — Normal<br>Red — Failure in +5 V supply   |
| Main power Vee –5 V      | Green — Normal   |
| Main power +12 V         | Green — Normal   |
| Main power –12 V         | Green — Normal   |
| Auxiliary power Vcc +5 V | Green — 15 A breaker is set on<br>Off — indicates problem with power sequencer         |
| Auxiliary power Vee –5 V | Green — 15 A breaker is set on<br>Off — indicates problem with power sequencer         |
| Auxiliary power +12 V    | Green — 15 A breaker is set on<br>Off — indicates problem with power sequencer         |
| Auxiliary power –12 V    | Green — 15 A breaker is set on<br>Off — indicates problem with power sequencer         |

### 1.4.3 Exhaust Fan

The reverse-curve, radial fan in the enclosure bottom exhausts warm air out of the enclosure bottom, creating a low-pressure zone in the plenum in the enclosure center. This low-pressure zone draws air through the air intakes in the top of the enclosure back door, through the PCBs, and into the plenum.

The cooling fan performance is 1200 ft<sup>3</sup>/minute.



---

## Parallel Disk Array Installation

This chapter provides site preparation data and installation procedures for the Parallel Disk Array. Make sure the site meets the requirements listed in Section 2.1. When planning a location, ensure that there is sufficient room for service access.

### 2.1 Parallel Disk Array Site Preparation

This section provides information to help you prepare a site for the parallel disk array (PDA). It includes specifications and a physical description of the PDA enclosure.

#### 2.1.1 Physical Characteristics

##### 2.1.1.1 Size and Service Clearance

The PDA is housed in its own H9A00 series enclosure which is connected to the data parallel unit (DPU) via a high speed VME I/O bus. Refer to Figure 2-1 for overall dimensions of this enclosure.

Space requirements include sufficient area to contain all of the equipment, and to provide service clearance. Leave at least 92 mm (36 inches) of clearance in the front and rear to access the PDA enclosure and to provide adequate ventilation. No space is required on the sides of the enclosure, though there must be some way to access the rear of the enclosure on a regular basis. There must also be ample room for the power cable, which should be out of the way, where it will not be stepped on, tripped over, or otherwise disturbed.

##### 2.1.1.2 Weight

The maximum weight for DECmpp PDA enclosures is 500 kg (1100 pounds).

---

#### Warning

---

Due to the weight of the equipment, Digital recommends that at least two people move the enclosure from the shipping pallet to the final location.

---

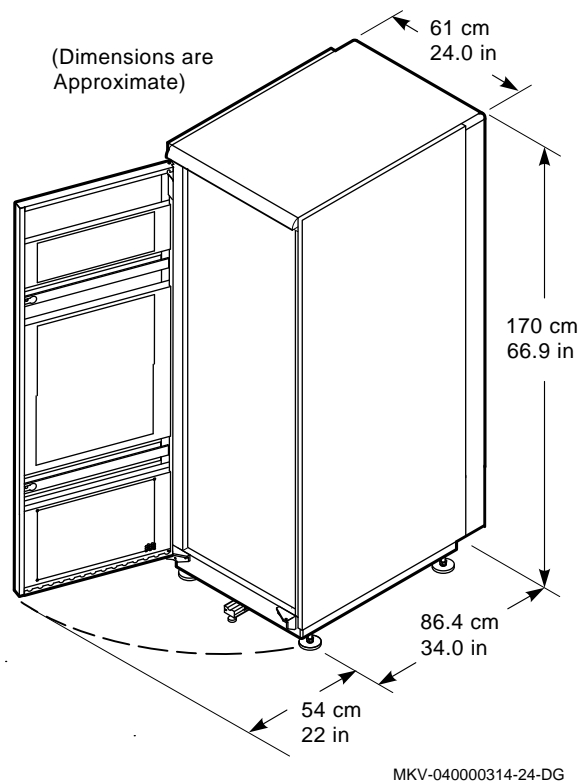
#### 2.1.2 Power and Cooling Requirements

The PDA enclosure for all DECmpp 12000/Sx Series configurations should be installed next to the DPU enclosure; both of which should be in a controlled computer room environment. The area must be free of dust and other environmental hazards.

Cooling air flow for PDA H9A00 series enclosure is such that cool air enters through top louvers in the front door, runs vertically through the equipment and is exhausted out bottom louvers in the front and rear doors.

## 2.1 Parallel Disk Array Site Preparation

Figure 2-1 Parallel Disk Array Enclosure Dimensions



All DECmpp configurations require that the PDA be tied to a dedicated power input. The power requirements are determined by the number of disks and the size of the power tray (either high or low power).

The power requirement for all SMP11-Ax PDA models is as follows:

20A, 220 V/240 V outlet (supports up to 19 disks)

All SMP11-Ax PDA enclosures are shipped with an L6-20P three-prong AC plug. If you are not certain about your power source, contact a qualified electrician.



## 2.1 Parallel Disk Array Site Preparation

### 2.1.3 Power and Environmental Specifications

Table 2-1 provides the PDA electrical and environmental specifications.

**Table 2-1 PDA Specifications**

| <b>Item</b>                  | <b>Specification</b>                     |
|------------------------------|--|
| Line Voltage                 | 220/240 Vac, 20A, 50/60 Hz, single phase |
| Line Voltage Tolerance       | 180 V minimum, 264 V maximum             |
| Maximum Power                | 2500 W                                   |
| Operating Temperature        | 10–40°C (50–104°F)                       |
| Humidity                     | 20–80% noncondensing                     |
| Maximum Wet Bulb Temperature | 28°C (82°F)                              |
| Minimum Dew Point            | 2°C (36°F)                               |
| Maximum Altitude             | 2400 m (8000 ft) at 36°C (96°F)          |
| Cooling Requirement          | 8540 Btu/hour (up to 19 disks)           |

## 2.2 PDA Installation

### 2.2 PDA Installation

The following sections provide instructions for DECmpp 12000 parallel disk array (PDA) installation, connection to the data parallel unit (DPU), and turning on. When these procedures are completed, you can begin operations.

#### 2.2.1 Unpacking and Positioning the PDA

Follow these steps to unpack the PDA:

1. Inspect the container for obvious signs of damage, including dents, scrapes, or water damage. If you find any sign of damage, note this on the bill of lading.
2. Remove the plastic wrap, the corner pads, and any cardboard packaging.

---

**Note**

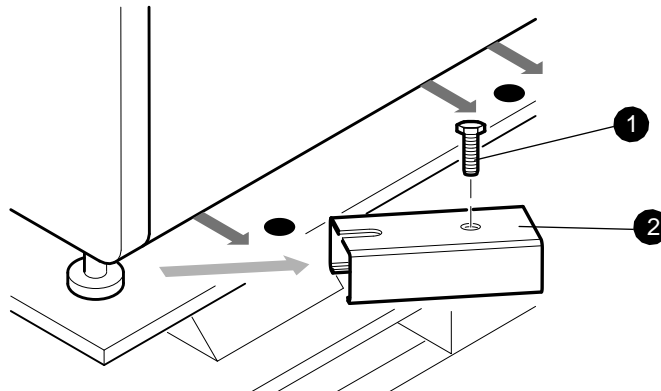
---

Leave a minimum 7-foot clearance in front of the shipping skid.

---

3. Insert the skid ramps into the holes in the front of the skid with arrows on skid lined up with arrows on ramp boards.
4. With a 9/16-inch wrench, loosen and remove the shipping bracket bolts ❶ on the enclosure feet (Figure 2-2).
5. Remove the shipping brackets ❷ and save them for use if the enclosure needs to be reshipped.

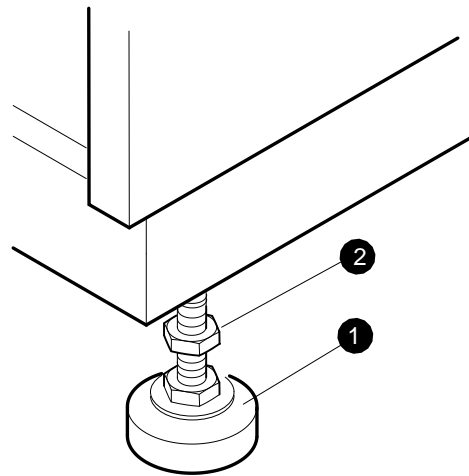
**Figure 2-2 Removing Enclosure Shipping Bracket**



MLO-007680

- ❶ Shipping bracket bolts
- ❷ Shipping bracket

Figure 2-3 PDA Enclosure Levelers



MLO-007745

6. While the enclosure is still on the skid, loosen the 3/4-inch upper (lock) nut ② on each foot by turning clockwise (Figure 2-3). Turn the 9/16-inch bottom nut ① on each foot counterclockwise to raise the feet for skid clearance.

---

**Warning**

---

The PDA enclosure weighs 500 kg (1100 lbs). For safety, two people are needed to roll the enclosure down the ramps.

---

7. Roll the enclosure down the ramp boards. The front two wheels are on swivel casters, the rear two are fixed.
8. Ensure that both enclosure side panels are installed and secure.
9. Position the PDA enclosure to the right (facing front) of the DPU enclosure.
10. Leave a 36-inch clearance at the front and rear of the enclosure for service.

---

**Caution**

---

**All enclosure side panels must be installed.** This is important to maintain proper cooling air flow and to reduce radiated emissions.

---

11. Raise the enclosure up on its feet as follows:
  - a. Lower each foot into position by turning clockwise on the 9/16-inch nut on the foot.
  - b. Tighten each 3/4-inch locknut counterclockwise to lock the feet in place.
12. Ensure that the PDA enclosure is leveled such that its side panel butts against the DPU side panel firmly with no gap. Also, ensure that the height of each enclosure is identical so that horizontal lines are in alignment.

## 2.2 PDA Installation

### 2.2.1.1 Assembling the Enclosure Skirts

Open the PDA enclosure skirt kit, PN H9C00-UA.

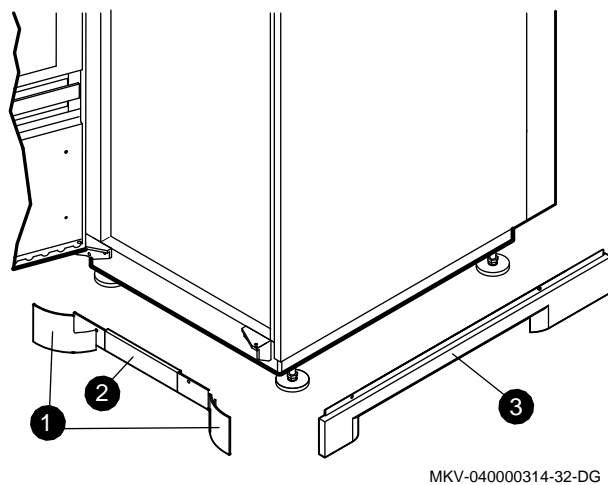
The skirts are installed after the enclosure is installed and raised up off its casters by the leveling feet.

A separate skirt is installed on each side of the enclosure except on the side that butts up against the DPU enclosure. Each skirt is attached to the enclosure by two quarter-turn Phillips head captive screws.

#### Front Skirt

The front skirt, shown in Figure 2-4, has two pieces that must be fitted together, then attached to the enclosure.

Figure 2-4 Front and Side Skirts



- ❶ Front curved pieces, PN 70-29318-01, and PN 7-29318-02
- ❷ Front center piece, PN 74-41733-01
- ❸ Right side piece, PN 70-29320-02 (left side piece, PN 70-29320-01)

#### Side Skirt

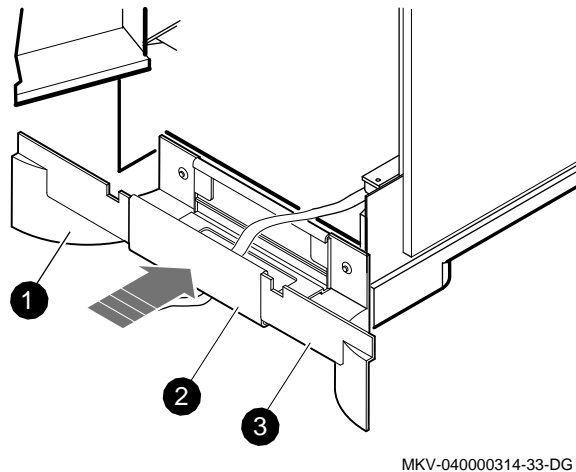
One side skirt is shown in Figure 2-4, item ❸. Attach the side skirt(s) to the side(s) of the enclosure frame that do not butt up against another enclosure.

#### Rear Skirt

The center piece of the rear skirt has a rectangular opening through which run all external power and signal cables. The rear skirt panels are shown in Figure 2-5.

The curved pieces, ❶ and ❸, are attached to the center piece ❷ with plastic push-in fasteners, and attached to the frame with the push-in captive screws. The cable cover, PN 74-41660-00, (not shown in Figure 2-5), fits into the notches on the two side pieces.

Figure 2-5 Rear Skirt



- ❶ PN 70-29318-01
- ❷ PN 74-41733-01
- ❸ PN 70-29318-02

### 2.2.2 Electrical Installation

---

#### Note

---

Notify users that the DECmpp 12000/Sx system will be off line during PDA installation. Before beginning any installation procedures, make sure that the DPU is turned off.

1. Turn off the front-end processor and DPU, as described in the *DECmpp 12000/Sx Hardware Installation Guide* and the *DECmpp 12000/Sx Hardware Service Manual*.
  2. Set the circuit breaker on the DPU rear OFF. Leave the DPU power cable connected to maintain the ground connection.
- 

#### 2.2.2.1 Printed Circuit Boards

The printed circuit boards (PCBs) for the PDA may be packaged separately for additional protection. If so, you should install them now.

Always be sure to:

- Handle all PCBs by the edges
- Wear antistatic equipment (wrist or ankle strap)
- Do any work on a static-free surface

## 2.2 PDA Installation

### 2.2.2.2 Installing Disk Array Controller PCBs

Figure 1–5 shows the disk array (DA) controller front cover. The disk array (DA) controller slots are shown in Figure 1–6 and described in Section 1.1.1.

Follow these steps to install the DA controller PCBs in the DA controller card cage:

1. Open the PDA front door.
2. Remove the four screws attaching the DA controller front cover, exposing the card cage.
3. Align the top and bottom edges of each PCB in the nylon card guides, and slide the PCB into the appropriate slot.

If necessary, push the extractor levers down to seat the card firmly in the connector.

4. Use four screws to reattach the DA controller front cover.

### 2.2.3 Installing MUX PCBs

The MUX PCBs are installed on the DA backplane, behind the rear disk chassis. Figure 1–9 shows the MUX connector locations on the DA backplane. The connectors are labeled as  $JnMn$  and are always to the left of the associated disk interconnect. The MUX PCB itself projects into an open area on the disk canister back panel and butts into a foam strip on the inside of the disk canister. Refer also to Section 2.3 for the disk canister layout pattern.

Follow these steps to install the MUX PCBs:

1. Open the PDA rear door.
2. Holding the MUX PCB by the edges, seat it firmly in the DA backplane connector.

## 2.3 Installing Disk Canisters

Each data disk has been factory-configured for a specific disk bank and disk number. Labels on the disk canister front identify the bank number and disk number. Disks are arranged within the 10-slot disk chassis (5-wide, 2-high). Refer to Figure 1–2 through Figure 1–4 for bank and drive locations. To maintain proper air flow, a blank panel closes any unused slots within the disk chassis.

### 2.3.1 Jumpers and Terminators

On the PCB component side of the disk drive, jumpers and terminators are accessible through the open sides of the disk canister.

In bank 1 only (the rear bank), all the disk drives must have the terminator pack installed at TM223 as shown in Figure 2–6. Also, jumpers must be installed on pin sets JP224 1–2, 3–4, and 5–6, and a jumper on JP213 1–2. Be sure to install these disk canisters only in bank 1.

In bank 2 only (the front bank), all the disk drives must have the terminator pack removed at TM223, no jumpers installed on pin sets JP224, and a jumper installed on JP213 3–4. Both configurations are shown in Figure 2–6.

## 2.3 Installing Disk Canisters

---

### Caution

---

Be sure to install disk canisters only in bank 1 if they are configured for bank 1. Install disk canisters configured for bank 2 only in bank 2.

---

---

### Caution

---

Do not change any other jumpers on the disk drive PCB. Do not compare jumpers from one disk drive PCB to another. Many jumpers (especially the black-colored ones) are factory-set during testing to optimize performance.

---

Disk slot layouts for both front and rear disk banks are as follows. (The chassis canister slots are also numbered in this order.) If used, the optional standby must be in bank #1 (rear bank).

Rear Bank (Bank 1):

|              |   |   |   |   |   |                           |
|--------------|---|---|---|---|---|---------------------------|
| 4-disk bank: | S | - | - | - | - | (S = standby disk option) |
|              | P | 8 | 6 | 4 | 2 | (P = parity)              |
| 8-disk bank: | S | 7 | 5 | 3 | 1 | (S = standby disk option) |
|              | P | 8 | 6 | 4 | 2 | (P = parity)              |

Front Bank (Bank 2):

|              |   |   |   |   |   |                       |
|--------------|---|---|---|---|---|-----------------------|
| 8-disk bank: | 1 | 3 | 5 | 7 | - | (Tenth slot not used) |
|              | 2 | 4 | 6 | 8 | P | (P = parity)          |

Perform the following steps to install the disk canisters, as shown in Figure 2-7.

---

### Caution

---

Be sure to slide the canister squarely into the guide rails. Guide rails can break if the canister is twisted to one side.

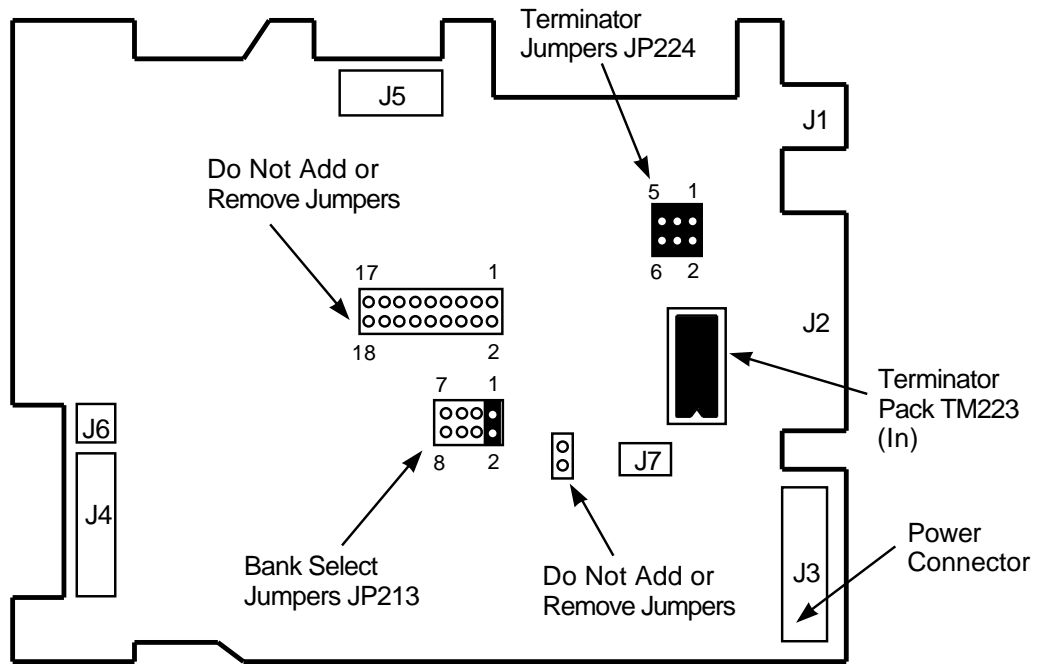
---

1. Open the PDA front or rear door to access the proper disk bank.
2. Align the disk canister top and bottom edges in the nylon guide rails, and push the canister into the disk chassis, seating it against the DA backplane.
3. Tighten the two captive screws, securing the disk canister to the chassis.
4. Set the disk canister switch ON (up). Pull the switch handle out and up.

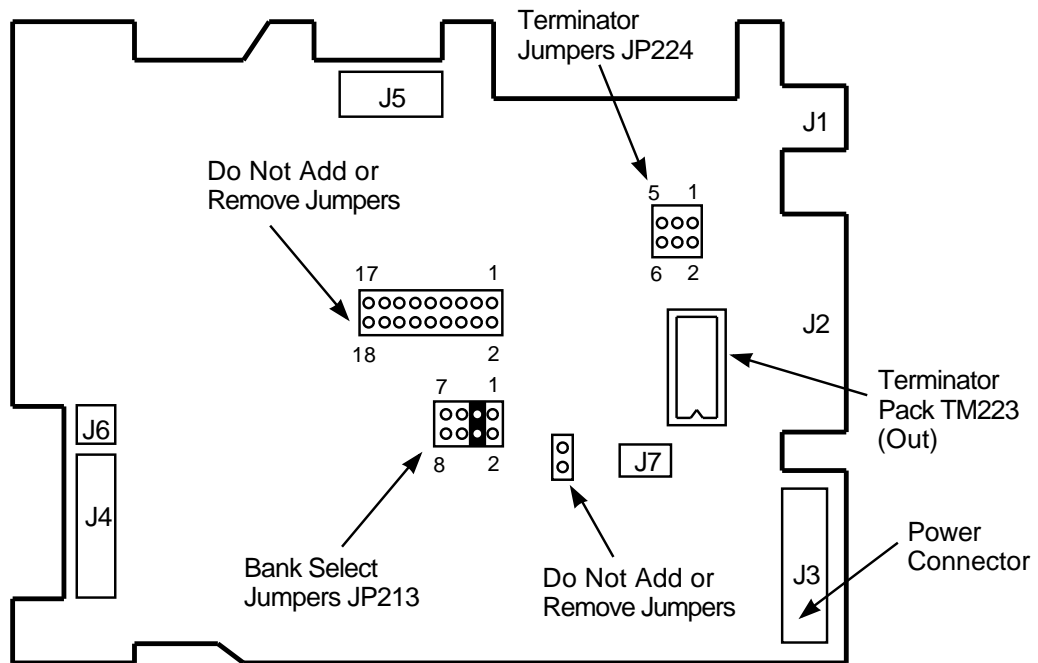
After system powerup, refer to Chapter 6 for details on the lightpipe and disk status.

## 2.3 Installing Disk Canisters

Figure 2-6 Disk Drive PCB (Component Side)



Bank 1 (Rear) Configuration

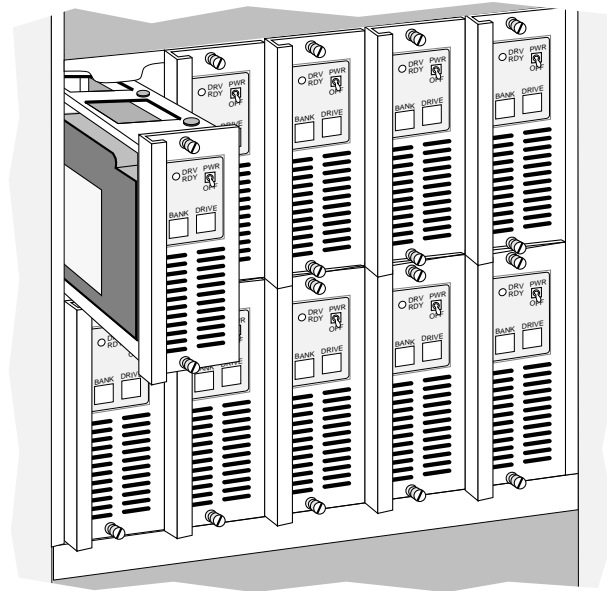


Bank 2 (Front) Configuration

MKV-040000314-53-RAGS



Figure 2-7 Inserting Disk Canister



## 2.4 Connecting DPU and PDA

Two 50-pin cables carry control (command/status) information and data between the DA controller and the VME 6U interface in the DPU.

On the DA controller backplane, the connecting plug marked control attaches to connector P12 on the upper (A) row, and the connecting plug marked data attaches to connector P11 on the lower (B) row. The connectors are held with a strain release clamp on the DA controller backplane and with another clamp at the top of the enclosure. The cables pass through a slot in the top front air baffle and down the side of the enclosure, between the enclosure side rails and the side panel. For shipping, the loose cabling is rolled and placed next to the power supply extension on the bottom of the enclosure.

### 2.4.1 Installing the DPU PCBs

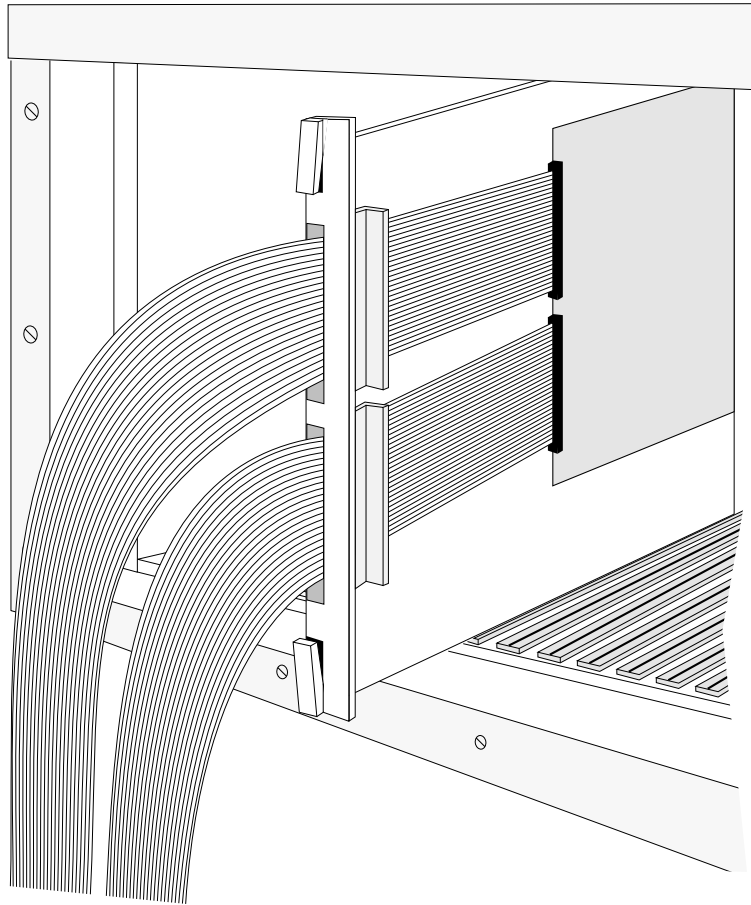
One Parallel VME 6U Adapter PCB connects the VME 6U interface PCB to the DPU. All the Parallel VME 6U Adapter switches and jumpers are factory-set and do not need adjusting. For more information on the Parallel VME 6U Adapter, refer to the *DECmpp 12000/Sx Parallel VME Reference Manual*.

Referring to Figure 2-8, follow these steps to install the VME 6U interface PCB in the DPU. The DPU-PDA cabling is easier if you connect the cables to the VME 6U interface PCB (on the Parallel VME 6U Adapter PCB) before inserting it in the card cage.

The front panel of the fascia holding the VME 6U interface PCB has cutouts for cable access. Behind the front panel, each cutout has a corresponding U-bracket, which functions as a cable clamp, as shown in Figure 2-9. Two nuts attached to studs fasten each U-bracket to the Parallel VME 6U Adapter.

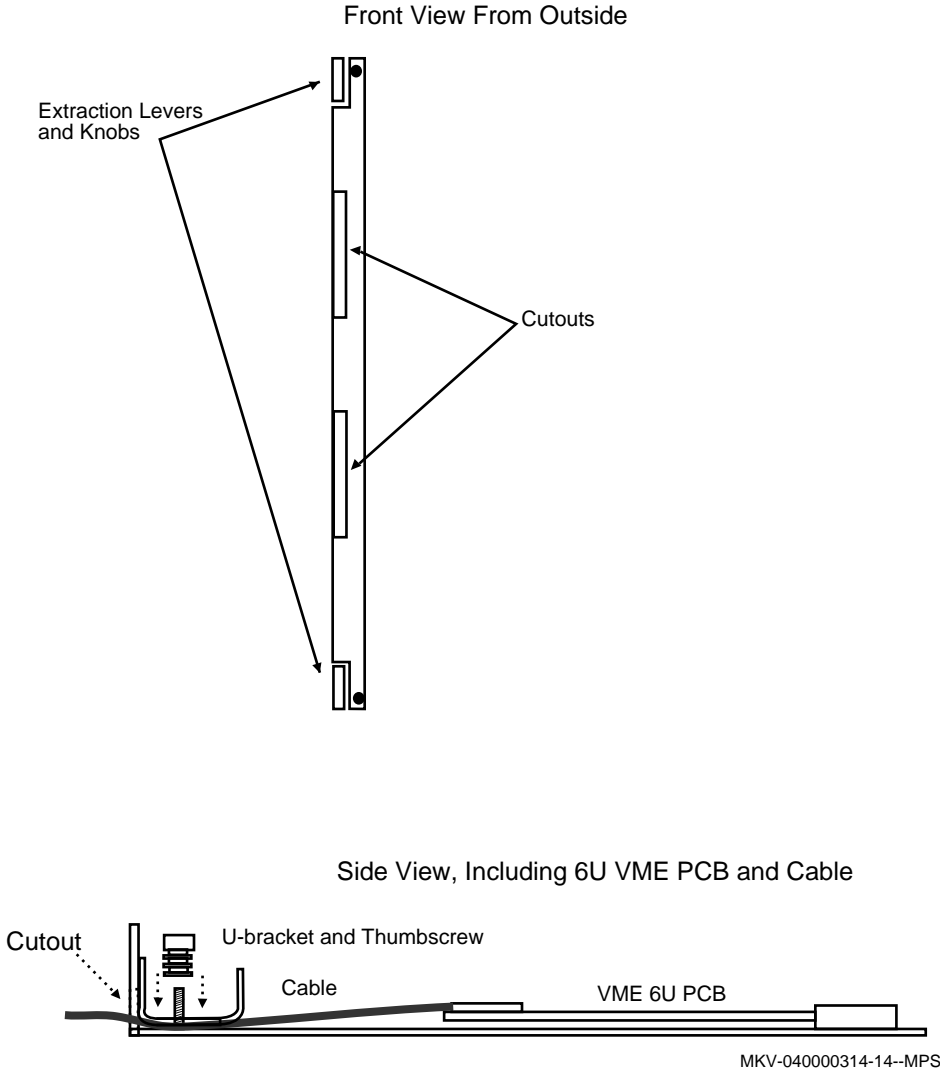
## 2.4 Connecting DPU and PDA

Figure 2–8 DPU Card Cage



1. Loosen the two nuts on both U-brackets until you can move them enough to feed the interface cables through.
2. Feed the cables under the U-brackets and through the cutouts.
3. Connect the cable connector marked JC or control to the upper connector on the VME 6U interface PCB.
4. Connect the cable connector marked JD or data to the lower connector on the VME 6U interface PCB.
5. Tighten the nuts until the U-brackets hold the cables firmly.
6. Insert the Parallel VME 6U Adapter PCB into DPU backplane slot IO04 (Figure 2–10). It will work in any of the DPU I/O slots, but slot IO04 is reserved specifically for this PCB. Slot IO00, the one next to the array control unit (ACU), is reserved for the front-end VME interface (MVIB), and slot IO08 is reserved for the PVME PCB.

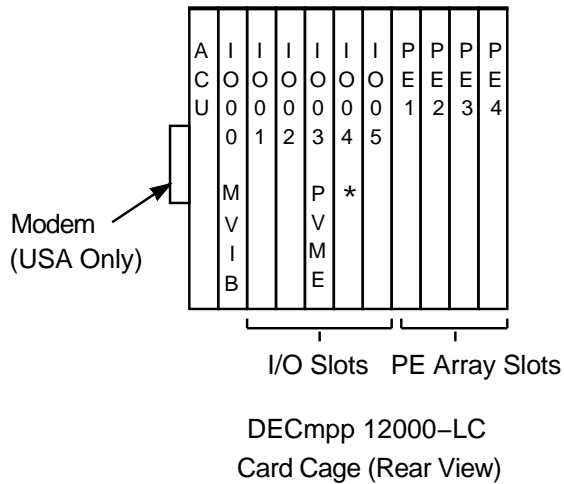
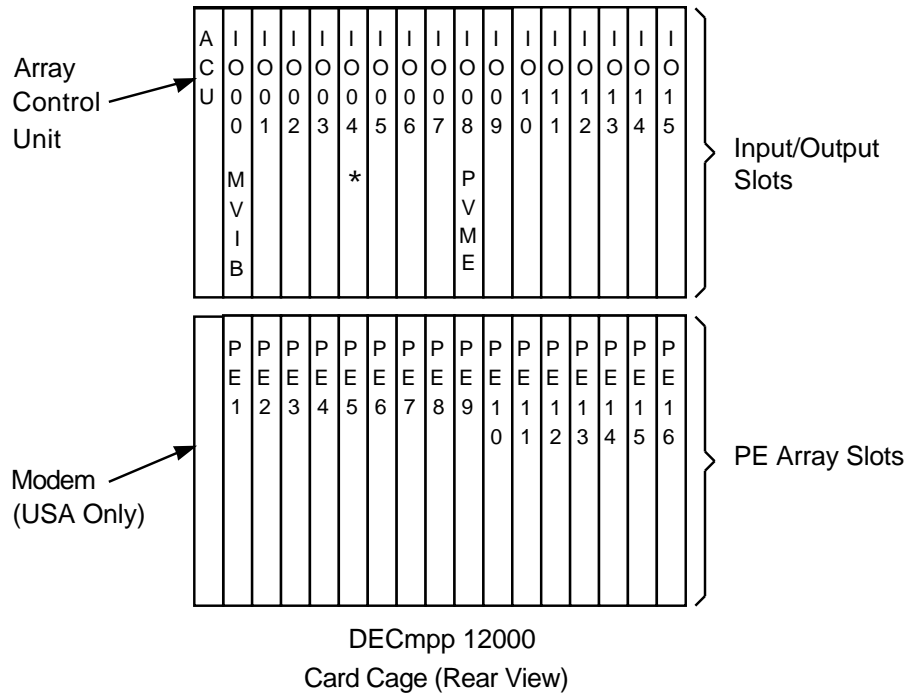
Figure 2-9 DA Interface PCB Cable Connections



The Parallel VME PCB (KF100-AA option) ships standard with the DECmpp 12000/Sx. If the Parallel VME PCB is to be installed at this time in an early model DECmpp system, refer to the *DECmpp 12000/Sx Parallel VME Reference Manual* for installation procedures.

## 2.4 Connecting DPU and PDA

Figure 2-10 Data Parallel Unit Card Cage Slots



\* Slot IO04 Reserved for Optional PDA Interface

MKV-040000314-03-RAGS

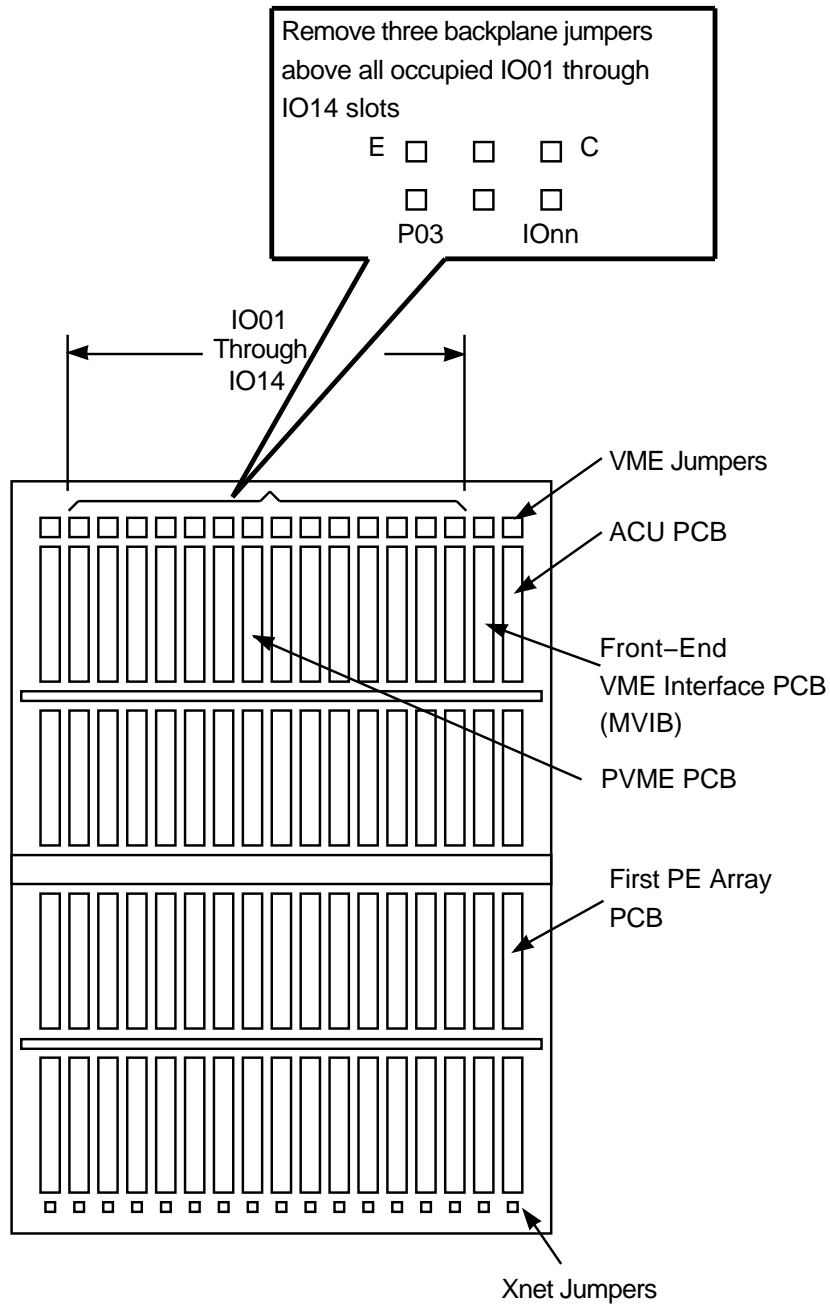
### 2.5 Setting DPU Jumpers

Follow these steps to reconfigure the jumpers on the front of the DPU backplane.

1. Open the DPU front door. Figure 2–11 shows the location of the jumpers for the DECmpp 12000.
2. Identify the IO slot containing the Parallel VME 6U Adapter/VME 6U interface PCB pair. Normally this is slot IO04.
3. Remove the three backplane jumpers above the slot occupied by the Parallel VME 6U Adapter.
4. Close the DPU front door.

## 2.5 Setting DPU Jumpers

Figure 2-11 DPU Backplane Jumper Locations



MKV-040000314-04-RAGS

### 2.6 Turning On the DPU and PDA

Follow these steps to turn on both the DPU and the PDA.

1. Open the PDA rear door.
2. Turn the PDA circuit breaker ON.
3. Open the DPU rear door.
4. Turn the DPU circuit breaker ON.
5. Open the DPU front door.
6. Turn the DPU power keyswitch ON.
7. Turn on the front-end server.
8. Make sure that the DPU and PDA have powered up successfully:
  - DPU POWER indicator is green
  - PDA POWER indicator is green
  - DA controller CPU indicator is green
  - Indicator on each disk canister is green
9. Close all enclosure doors.
10. Boot the system.

You can now begin the system administration procedures described in Chapter 3.





---

## Parallel Disk Array System Administration

Chapter 3 provides procedures for bringing up the DECmpp 12000/Sx Parallel Disk Array (PDA), initializing the disk array (DA) controller and the disk banks, and other administrative procedures. For standard DECmpp 12000/Sx system administration procedures, refer to the *DECmpp 12000/Sx System Administration Guide*.

The PDA has been configured, inspected and tested in the factory. The information presented in this chapter is not required to install a PDA. It is intended for Digital Services personnel.

---

### Note

---

The PDA configuration must be correct before executing `doconfig` on the DECmpp 12000/Sx system.

If you are also installing a new DECmpp 12000/Sx system, you must initialize and configure the PDA before configuring the DECmpp 12000/Sx. If you are adding a PDA to your DECmpp 12000/Sx system or upgrading your existing PDA, you must verify the DA controller configuration.

---

## 3.1 Establishing PDA Communications

### 3.1 Establishing PDA Communications

Make sure the PDA has been installed correctly, and verify that the kernel can communicate with the PDA by entering the following command:

```
#dastat /dev/da0a
```

A message similar to the following should display:

```
(dpu0) Tue Jul 21 15:40:10 1992 Starting up; Version 2.2.0
(dpu0) Tue Jul 21 15:40:11 1992 loading microcode file: "/usr/mpp/etc/mp12ucode.wo"
(dpu0) Tue Jul 21 15:40:18 1992 loading ACU kernel file: "/usr/mpp/etc/acuk"
(dpu0) Wed Jul 22 12:10:55 1992 ACU kernel timeout (command 6)
ECSR=0x4002, QCSR=0x0, PTACCESS=0, CPC=0xffff00a8
(dpu0) Wed Jul 22 12:10:55 1992 Save of context failed, killing job; pid = 10333
(dpu0) Wed Jul 22 12:10:55 1992 ACU kernel command error (!ECSR<Run>)
(dpu0) Wed Jul 22 12:10:55 1992 unable to abort user -- reloading ACU kernel
(dpu0) Wed Jul 22 12:10:55 1992 Job context lost in system reset; pid = 10333
(dpu0) Wed Jul 22 12:10:55 1992 Job context lost in system reset; pid = 10355
(dpu0) Wed Jul 22 12:10:55 1992 loading ACU kernel file: "/usr/mpp/etc/acuk"
(dpu0) Wed Jul 29 10:54:21 1992 Starting up; Version 2.2.0
(dpu0) Wed Jul 29 10:54:22 1992 loading microcode file: "/usr/mpp/etc/mp12ucode.wo"
(dpu0) Wed Jul 29 10:54:41 1992 loading ACU kernel file: "/usr/mpp/etc/acuk"
(dpu0) Tue Aug 11 08:19:50 1992 Starting up; Version 2.2.0
(dpu0) Tue Aug 11 08:19:51 1992 loading microcode file: "/usr/mpp/etc/mp12ucode.wo"
(dpu0) Tue Aug 11 08:20:12 1992 loading ACU kernel file: "/usr/mpp/etc/acuk"
```

If the dastat command fails, check the following areas:

- Device Entry
- PDA Device
- PVME PCB
- Parallel VME 6U Adapter PCB
- VME 6U Interface PCB

If there is no Device Entry, verify the device file is present by entering the following command:

```
# ls -l /dev/da0a
```

If the Device Entry is not present, make the device, using the following command:

```
# cd /dev
# ./MAKEDEV da0
```

Then reboot by entering:

```
# /etc/reboot
```

If there is no PDA Device, verify the kernel exists by looking at the most recent system startup message by entering the following command:

```
# uerf -R | more
```

## 3.1 Establishing PDA Communications

The following lines should display:

```
----- EVENT INFORMATION -----
EVENT CLASS                                OPERATIONAL EVENT
OS EVENT TYPE                               300. SYSTEM STARTUP
SEQUENCE NUMBER                             0.
OPERATING SYSTEM                            ULTRIX 32
OCCURRED/LOGGED ON                          Tue Aug 11 08:07:13 1992 EDT
OCCURRED ON SYSTEM                          mpdemo.mps.m
SYSTEM ID                                    x82020230
                                           HW REV: x30
                                           FW REV: x2
                                           CPU TYPE: R2000A/R3000
PROCESSOR TYPE                               KN02/R3000
MESSAGE                                     ULTRIX V4.2A (Rev. 47 MPSG) System #5:
                                           _Tue May 12 13:33:00 EDT 1992
                                           real mem = 92274688
                                           avail mem = 79495168
                                           using 1228 buffers containing 9224192
                                           _bytes of memory
                                           KN02 processor - system rev 48
                                           cpu0 ( version 3.0, implementation 2 )
                                           fpu0 ( version 3.0, implementation 3 )
                                           asc0 at ibus5
                                           rz0 at asc0 slave 0 (RZ56)
                                           rz1 at asc0 slave 1 (RZ57)
                                           rz4 at asc0 slave 4 (RRD42)
                                           tz5 at asc0 slave 5 (TK50)
                                           ln0 at ibus6
                                           ln0: DEC LANCE Ethernet Interface,
                                           _hardware address: 08:00:2b:1d:31:25
                                           dc0 at ibus7
                                           fb0 at ibus0
                                           fb: Module type PMAG-BA 1024X864
                                           vba0 at slot 2 (3VIA/MVIB)
                                           ms0 at vba0 csr 0xfc800000 vmea32d32
                                           _vec 0xbe priority 1
                                           da0 at ms0 slave 0
                                           (DK515-78 - 4 data drives)
                                           acu0 at vba0 csr 0xfcc00000 vmea32d32
                                           _vec 0xd0 priority 1
                                           iqprobe at be889000
                                           iq0 at vba0 csr 0xdc000000 vmea32d32
                                           _vec 0xc0 priority 1
```

If the Parallel VME 6U Adapter (`mpa0`) line is not present, the wrong kernel has been booted. Reboot with the correct kernel by entering the following commands:

```
# mv vmunix vmunix.old
# mv genvmunix vmunix
# /etc/reboot
```

After the line MasPar Bus Adapter (`mpa0`), the following line should appear:

```
da0: da_config <-5c8f
```

If this line does not appear, either:

- The kernel is not configured for a PDA
- The hardware is not installed correctly

The following message shows the successful attachment of both the disk array (`da0`) and the data parallel unit (DPU) (`acu0`). This message should appear during the bootstrap sequence (and in the SYSTEM STARTUP of `uerf(8)` file).

### 3.1 Establishing PDA Communications

```
----- EVENT INFORMATION -----
EVENT CLASS                      OPERATIONAL EVENT
OS EVENT TYPE                    300. SYSTEM STARTUP
SEQUENCE NUMBER                  0.
OPERATING SYSTEM                 ULTRIX 32
OCCURRED/LOGGED ON              Tue Aug 11 08:07:13 1992 EDT
OCCURRED ON SYSTEM              mpdemo.mps.m
SYSTEM ID                        x82020230 HW REV: x30
                                  FW REV: x2
                                  CPU TYPE: R2000A/R3000
PROCESSOR TYPE                  KN02/R3000
MESSAGE                          ULTRIX V4.2A (Rev. 47 MPSG) System #5:
                                  _Tue May 12 13:33:00 EDT 1992
                                  real mem = 92274688
                                  avail mem = 79495168
                                  using 1228 buffers containing 9224192
                                  _bytes of memory
                                  KN02 processor - system rev 48
                                  cpu0 ( version 3.0, implementation 2 )
                                  fpu0 ( version 3.0, implementation 3 )
                                  asc0 at ibus5
                                  rz0 at asc0 slave 0 (RZ56)
                                  rz1 at asc0 slave 1 (RZ57)
                                  rz4 at asc0 slave 4 (RRD42)
                                  tz5 at asc0 slave 5 (TK50)
                                  ln0 at ibus6
                                  ln0: DEC LANCE Ethernet Interface,
                                  _hardware address: 08:00:2b:1d:31:25
                                  dc0 at ibus7
                                  fb0 at ibus0
                                  fb: Module type PMAG-BA 1024X864
                                  vba0 at slot 2 (3VIA/MVIB)
                                  ms0 at vba0 csr 0xfc800000 vmea32d32
                                  _vec 0xbe priority 1
                                  da0 at ms0 slave 0
                                  (DK515-78 - 4 data drives)
                                  acu0 at vba0 csr 0xfcc00000 vmea32d32
                                  _vec 0xd0 priority 1
                                  iqprobe at be889000
                                  iq0 at vba0 csr 0xdc000000 vmea32d32
                                  _vec 0xc0 priority 1
```

Once the devices are recognized correctly, you need to create a new kernel by running `doconfig`.

If the previous message does not contain a reference to `da0` after the line MasPar Bus Adapter (`mpa0`), the VME 6U interface PCB is not being properly accessed. The problem could be an improperly seated Parallel VME 6U Adapter or VME 6U interface PCB.

A message such as `da0: da_mpvme adapter has unexpected value = 0x66` indicates that the Parallel VME 6U Adapter CSR (OPCSR) is not being properly reset.

The message `da0: da_dmaisr has unexpected value = 0xff` indicates that the VME 6U interrupt status register is not coming up properly after a reset.

The message `dawaitcmd: CMDDBUSY in PSR won't clear` usually indicates a cabling problem between the DPU and the PDA. It could also indicate that power to the PDA has been shut off.

The message `daslave: CONTROLLER WON'T SYNC` indicates some type of initialization problem with the DA controller. Try using the DA controller RESET command.

## 3.2 Controller Configuration

The DA controller must be programmed with the configuration of the PDA disk banks and the disks within each bank. Each bank must contain the number of data disks specified, as well as the same type of disk. All disk drives within the PDA must be the same type and capacity.

1. Make sure that the controller is configured correctly by entering the following command:

```
dastat /dev/da0a
```

The following information displays:

```
(dpu0) Tue Jul 21 15:40:10 1992 Starting up; Version 2.2.0
(dpu0) Tue Jul 21 15:40:11 1992 loading microcode file: "/usr/mpp/etc/mp12uocode.wo"
(dpu0) Tue Jul 21 15:40:18 1992 loading ACU kernel file: "/usr/mpp/etc/acuk"
(dpu0) Wed Jul 22 12:10:55 1992 ACU kernel timeout (command 6)
ECSR=0x4002, QCSR=0x0, PTACCESS=0, CPC=0xffff00a8
(dpu0) Wed Jul 22 12:10:55 1992 Save of context failed, killing job; pid = 10333
(dpu0) Wed Jul 22 12:10:55 1992 ACU kernel command error (!ECSR<Run>)
(dpu0) Wed Jul 22 12:10:55 1992 unable to abort user -- reloading ACU kernel
(dpu0) Wed Jul 22 12:10:55 1992 Job context lost in system reset; pid = 10333
(dpu0) Wed Jul 22 12:10:55 1992 Job context lost in system reset; pid = 10355
(dpu0) Wed Jul 22 12:10:55 1992 loading ACU kernel file: "/usr/mpp/etc/acuk"
(dpu0) Wed Jul 29 10:54:21 1992 Starting up; Version 2.2.0
(dpu0) Wed Jul 29 10:54:22 1992 loading microcode file: "/usr/mpp/etc/mp12uocode.wo"
(dpu0) Wed Jul 29 10:54:41 1992 loading ACU kernel file: "/usr/mpp/etc/acuk"
(dpu0) Tue Aug 11 08:19:50 1992 Starting up; Version 2.2.0
(dpu0) Tue Aug 11 08:19:51 1992 loading microcode file: "/usr/mpp/etc/mp12uocode.wo"
(dpu0) Tue Aug 11 08:20:12 1992 loading ACU kernel file: "/usr/mpp/etc/acuk"
```

2. Verify that the standby disk mode is correct. If the line Bank 1 standby serves all banks appears instead of Each bank has standby, enter:

```
# dainit -m0 /dev/da0a
```

to set the configuration so that each bank uses its own standby disk.

3. Use the dabstat command to check the bank configurations.

This example checks Bank 1:

```
# dabstat -b1 /dev/da0a
Drive ID          1      Reserved
Sub ID           0
Mode             8 data drives
                Use primary system area
                Check drive array addresses
                Log soft and hard errors
                Recover policy is R/E/P
SUBS/DSBL        0 / 0
Bank Status      On line, Mounted 000000000-
Total blocks     703740
Reconstruct Address -1
Drive status     30 30 30 30 30 30 30 30 30 11
                Drive 1      Drive online/mounted
                Drive 2      Drive online/mounted
                Drive 3      Drive online/mounted
                Drive 4      Drive online/mounted
                Drive 5      Drive online/mounted
                Drive 6      Drive online/mounted
                Drive 7      Drive online/mounted
                Drive 8      Drive online/mounted
                Drive 9      Drive online/mounted
                Drive 10     Drive won't select
```

Drive 10 is absent unless it is the standby disk.

## 3.2 Controller Configuration

4. Use the `dabinit` command to set the correct bank configurations. For example, to initialize Bank 1 with eight data disks, enter:

```
# dabinit -b1 -m8 /dev/da0a
```

To disable Bank 2, enter:

```
# dabinit -b2 -c /dev/da0a
```

Whenever you change the DA controller or bank configuration, you must reboot the DECmpp 12000/Sx front-end server.

### 3.3 Setting Up File Systems

The following procedure describes how to set up your file systems on the disk array.

---

**Note**

---

The *h* partition is reserved for diagnostic use. The PDA diagnostics use this partition as a scratch area. The program `chpt(8)` will not modify partition *h* or modify any other partition so that it overlaps partition *h*.

---

1. To check the default file partitions, enter the following command:  

```
# chpt -q /dev/da0a
```
2. To create a new file system, run `/etc/newfs` by entering the following command:  

```
# /etc/newfs [-v] [-n] (special device and device type)
```

For the PDA, *device type* reflects the number of data disks.

The default partitions for 4- and 8-data disk banks are provided in Section 3.3.1.

#### 3.3.1 Default Partitions

The following is an example of display from `etc/fstab`:

**4 Disks:**

| Partition | Bottom  | Top     | Size    | Overlap   |
|-----------|---------|---------|---------|-----------|
| a         | 0       | 1403039 | 1403040 | e,g       |
| b         | 1403040 | 2810519 | 1407480 | e,g       |
| c         | 2810520 | 4191359 | 1380840 | f,g       |
| d         | 4191360 | 5621039 | 1429680 | f         |
| e         | 0       | 2810519 | 2810520 | a,b,g     |
| f         | 2810520 | 5621039 | 2810520 | c,d,g     |
| g         | 0       | 4191359 | 4191360 | a,b,c,e,f |
| h         | 5621040 | 5629919 | 8880    |           |

**8 Disks:**

| Partition | Bottom   | Top      | Size    | Overlap |
|-----------|----------|----------|---------|---------|
| a         | 0        | 2095679  | 2095680 | e,h     |
| b         | 2095680  | 4191359  | 2095680 | e       |
| c         | 4191360  | 6287039  | 2095680 | f       |
| d         | 6287040  | 8382719  | 2095680 | f       |
| e         | 0        | 4191359  | 4191360 | a,b,h   |
| f         | 4191360  | 8382719  | 4191360 | c,d     |
| g         | 8391600  | 11250959 | 2859360 |         |
| h         | 11250960 | 11268719 | 17760   |         |

## 3.4 Upgrade Procedures

### 3.4 Upgrade Procedures

Follow the steps in the following sections when upgrading your PDA from 4 to 8 data disks or when adding a new disk bank (upgrading to 16 data disks).

#### 3.4.1 Upgrading to 8 Disks

1. Save contents of all of the partitions on a tape using the server's drive or over the network.
2. Run `dabinit(8)`, as described in Section 3.2.
3. Format new disks, as described in Chapter 4.
4. Restore all of the partitions from the tape.

#### 3.4.2 Upgrading to 16 Disks

1. Run `dainit(8)` if the standby mode is different.
2. Run `dabinit(8)`, as described in Section 3.2.
3. Format new disks, as described in Chapter 4.



---

## Disk Formatting

Chapter 4 provides instructions for formatting the DECmpp 12000/Sx parallel disk array (PDA) disks. The PDA has been configured, inspected, and tested in the factory. The information presented in this chapter is not required to install a PDA. It is intended for Digital Services personnel.

Because both the disk system area and the user area must be formatted, disk formatting is a two-step process. Read the associated man page(s) first.

The following example shows a typical format sequence for a new bank:

```
# daformat -b1 -f2 -A /dev/da0a
# daformat -b1 -f3 -A /dev/da0a
```

Always use function 2 (-f2) when formatting the system area. Function 2 reads the manufacturer's media defect information and saves it in the system area. During the user area formatting, bad blocks are mapped using this defect information.

In the following descriptions, all numbers are in decimal form except when followed by the letter *h* (for example, 10h), which indicates a hexadecimal value.

The first time a controller is brought up, run the `dainit(8)` command to initialize it.

### 4.1 The FORMAT Command

Formatting is the process of writing sector headers on each track of a disk. The FORMAT (`daformat(8)`) command formats selected disks within a bank.

When formatting disks, the disk array (DA) controller reserves at least one sector per track as a spare. If a single media flaw is logged on a track, the spare sector is placed on the single media flaw. (Every track can tolerate one bad sector.) If two or more sectors within one track have logged media flaws, the track is considered bad and is assigned an alternate track during format.

Formatting a disk consists of two distinct operations:

- Initializing the system area
- Formatting the user area (system area already exists)

Formatting operations occur independently of the drive mount status or the bank online status; the target drive needs only to be on line (power on and ready or spun up).

A bank's Flaw Descriptor Table (FDT) is generated from the Raw Flaw Table (RFT) and is used to assign alternate tracks. The FDT is generated only during a Bank Mount Sequence. Certain formatting operations (`func 0`, `func 1`, and `func 2`) do not alter a bank's FDT in any way.

After formatting, run the MOUNTBANK command (`damount(8)`), as appropriate.

## 4.1 The FORMAT Command

The `function` field defines the function performance as follows:

- **0 — Normalize System Area**  
Reads the existing system area, reformats only the tracks used to store the system area, and then writes the system area to disk with cleared Growth Error Table (GET) and the reset disk array position address.

---

**Caution**

---

The function 1 option can destroy the drive's OEM flaw list.

---

- **1 — Initialize System Area**  
Formats only the tracks used to store the system area and writes the system area to disk with cleared RFT and cleared GET.
- **2 — Read Media Defect List, Initialize System Area**  
Reads the manufacturer-written media list to initialize the RFT, formats only the tracks used to store the system area, and writes the system area to disk with initialized RFT and cleared GET.

---

**Note**

---

Always use function 2 when formatting the system area. Function 2 reads the manufacturer's media defect information and saves it in the system area. During the user area formatting, bad blocks are mapped using this defect information.

---

- **3 — Discard GET, Format User Area**  
Reads the existing system area, clears the GET, writes the system area back to disk, then formats all tracks in the user area. (The DA controller automatically runs a Bank Mount Sequence on the selected bank.)
- **4 — Merge GET into RFT, Format User Area**  
Reads the existing system area, merges the GET into the RFT, clears the GET, writes the system area back to disk, then formats all tracks in the user area. (The DA controller automatically runs a Bank Mount sequence on the selected bank.)
- **5 — Merge GET into RFT, Reformat Bad Tracks Only in User Area**  
Reads the existing system area, attempts to recover data from flawed tracks (retries, ECC correction, RECONSTRUCT), reformats flawed tracks or assigns an alternate, writes original data back, merges the GET into the RFT, clears the GET, and writes the system area back to disk. (The DA controller automatically runs a Mount Bank sequence on the selected bank.)  
The DA controller executes this command entirely (including a Mount Bank sequence) before executing any other commands (noninterleaving of commands). (This is a special feature of `func 5` only.)

Whenever a system area is written, a read check always follows to ensure validity.

### 4.1.1 Operation Examples

This section contains examples showing how to bring up a new bank.

#### 4.1.1.1 Bring Up a New Bank

The following steps provide an example of how to initialize, format, and bring up a new bank.

1. Enter the `dabinit(8)` command to define bank configuration parameters.
2. Enter either the `daformat(8) -A func=1` command to initialize target disks, or the `daformat(8) func=2` command to read the media defect list, then initialize target disks.

```
# daformat -b1 -f1 -A /dev/da0a
# daformat -b1 -f2 -A /dev/da0a
# daformat -b1 -f3 -A /dev/da0a
```

3. Enter the `daformat(8)` command, using the option `-A` and setting `func = 3` to format the user area on the target disks.
4. Enter the `dabstat(8)` command to verify that all the valid disks in the bank are mounted.

#### 4.1.1.2 Bringing Up a New Bank with Flaw Data on Host

The following steps show how to initialize, format, download flaw data, and bring up a new bank.

1. Enter the `dabinit(8)` command to define bank configuration parameters.
2. Enter the `daformat(8)` command, using the option `-A` and setting `func = 1` to initialize target disks.
3. Enter the `daspecflaw(8)` command to download flaw data to target disks. This data is recorded in the GET.
4. Enter the `daformat(8)` command, using the option `-A` and setting `func = 4` (merge GET) to format the user area on the target disks.
5. Enter the `dabstat(8)` command to verify that all valid disks in the bank are mounted.

#### 4.1.1.3 Formatting Newly Installed Disks Within an Online/Mounted Bank

The following steps show how to disable an old disk drive and install a new drive in an online, mounted bank.

1. Disable the old drive before removing it by entering the `dadsbl(8)` command to define the Disabled (DSBL) field of the appropriate Bank Configuration Table. The `dadsbl(8)` command initiates a Bank Mount sequence to activate parity replacement for subsequent host data transfers.
2. Physically replace the old drive with a new drive.
3. Enter the `daformat(8) func = 1` command to initialize target drives, or `daformat(8) func = 2` to read the media defect list, then initialize target drives.
4. Enter the `daformat(8) func = 3` command to format the user area on the target drives.
5. Use the `darecon(8)` command to reconstruct data back to new drive.

## 4.1 The FORMAT Command

### 4.1.1.4 Reformatting Online/Mounted Banks with Growth Errors Logged

The following steps show how to initialize, format, download flaw data, and bring up a new bank.

1. Enter the `daformat(8) func = 5` command (merge the GET) to reformat flawed tracks only on the target drives.

`daformat func = 5` is the only format mode that attempts to read and recover the user data during the format operation.

- If the NON-TRANSFER COMMAND SUCCESS message displays, all data is preserved, the reformat is complete, and the GET is empty. The procedure is complete.
- If unrecoverable data is encountered, the blocks with corrupted data are logged in the Bad Block Table (BBT), and `daformat(8)` immediately terminates, displaying the NON-TRANSFER COMMAND ERROR (RECONSTRUCT NOT POSSIBLE) message. If `daformat(8) func = 5` is reentered, the BBT entries are reformatted/mapped out (data still corrupted), cleared from the GET, cleared from the BBT, and merged into the RFT.

Corrupted data possibly exists until restored as described below.

If unrecoverable data is encountered (NON-TRANSFER COMMAND ERROR (RECONSTRUCT NOT POSSIBLE)), continue with the procedure:

2. Temporarily halt all host transfer commands.
3. Log the BBT entries.
4. Reenter the `daformat(8) func = 5` command. If the NON-TRANSFER COMMAND ERROR (RECONSTRUCT NOT POSSIBLE) message displays, repeat step 2. Otherwise, go to step 5.
5. Restore data blocks for all logged BBT entries.
6. Enable host transfer commands.
7. Use the READCONFIG command to read the BBT.

## 4.2 The MOUNTBANK Command

The disk array formatting commands automatically initiate a Bank Mount sequence when necessary. When the PDA is turned on or after a system reset, the DA controller executes the following Bank Mount sequence on a target bank for which the Bank Configuration Table (DRIVE ID) is nonzero. If valid drives on a bank are selected but not READY, the controller waits for a drive timeout period (refer to Table 4-1), retrying the MOUNTBANK command until the drive becomes READY or the timeout period expires.

During a Bank Mount sequence, if valid drives on a bank are selected but not READY, the controller does not wait and flags the drives not mounted.

After a Bank Mount sequence, run the `dabstat(8)` command to verify correct bank status. The MOUNTBANK command is normally initiated after bank configuration parameters are changed. The DA controller should not be reset.

The DA Controller Bank Mount sequence is as follows:

1. Select the bank and run DRIVE INITIALIZATION, which identifies drives with the status of SELECTED, NOTBUSY, READY, ONCYLINDER, NOFAULT, and NOSEEKERROR. These drives are considered ONLINE.

## 4.2 The MOUNTBANK Command

2. Attempt to read the system area from ONLINE drives. Drives for which this operation is successful are considered MOUNTED. The RFTs of mounted drives are combined to form the bank's FDT.
3. Establish MOUNTBANK status. The DA controller checks to see if all drives required for the current bank configuration are MOUNTED, and, if so, sets the Bank Mount bit. (You can perform read/write operations only with MOUNTED banks.)
4. Determine which drive, if any, must be replaced using parity replacement. The Bank Mount sequence also detects a special state of Substituted/Disabled (SUBS/DSBL), where SUBS equals X, DSBL equals Y, and Y equals 10. This state would be set if a standby drive were inoperable while it was actively substituting for another drive. When this state is detected, the DA controller immediately sets DSBL equal to X and SUBS equal to 0, deassigning the standby drive and restoring the disabled status of the previously substituted drive.
5. Initialize the target bank's Reconstruct Disk Address.

## 4.3 The RECONSTRUCT Command

The RECONSTRUCT (`darecon(8)`) command initiates data/parity drive reconstruction.

The target is valid only when DSBL is nonzero, SUBS is nonzero, and DSBL is not equal to SUBS. For this condition, target selects the target drive for the reconstruct operation:

0 = drive identified by DSBL  
1 = drive identified by SUBS

Using the `dainit(8)` command, there are two selectable standby configurations:

- One standby drive per disk array bank
- One standby drive serving all disk array banks

In configuration 1, the standby drive must be assigned from the same bank as the drive being replaced. This configuration must be selected if there are two or more banks with different drive types.

In configuration 2, the single available standby drive must be connected to bank 1. This configuration can be selected only if all drives of all banks are the same type.

The following variables determine inoperable drive configuration. Use the `dabstat(8)` command to examine them and use the `dadsbl(8)` and `dasubs(8)` commands to set them:

|      |  |
|------|--|
| DSBL | 0 = no drive inoperable                                      |
|      | X = drive X [1 . . . 10] inoperable (disabled)               |
| SUBS | 0 = no drive substituted                                     |
|      | X = drive X [1 . . . 9] substituted (assigned standby drive) |

## 4.3 The RECONSTRUCT Command

### 4.3.1 Disabling and Replacing a Drive

The following steps disable a disk drive:

1. Check the DSBL field. If the DSBL field is already nonzero, do not disable an additional drive in the bank.
2. Use the `dadsbl(8)` command to disable a drive. You can physically replace the disabled drive at this time.
3. Use the `darecon(8)` command to set the DSBL field to 0 after a complete reconstruction cycle.

### 4.3.2 Assigning a Standby Drive

The following instructions assign the standby drive:

1. Run the `dasubs(8)` command to assign a standby drive.
  - Standby Mode 1 (each bank has its own standby drive): if the target bank SUBS field is already nonzero, an additional standby cannot be assigned.
  - Standby Mode 2 (Standby on bank 1 serves all banks): if any bank SUBS field is already nonzero, an additional standby cannot be assigned.

After a standby is assigned to substitute for a newly disabled drive, the SUBS field equals the DSBL field.

### 4.3.3 Reconstructing a Drive

The following steps reconstruct a drive:

1. Use the `damount(8)` command to initialize the target bank's Reconstruct Disk Address to all 1's. The read-only Reconstruct Disk Address value plus 1 specifies a bank starting-disk address for a `darecon(8)` command.
2. Use read-only Reconstruct Disk Address value plus 1 to specify a bank starting-disk address for RECONSTRUCT command.

If an unrecoverable media error is encountered during a reconstruct operation:

- The reconstruct operation ends immediately.
- The Reconstruction Disk Address is updated with the host reference position of the error. The physical disk position is logged in the GET.
- A NON-TRANSFER COMMAND ERROR message is returned to the host.
- The host logs the Reconstruction Disk Address value and continues the reconstruct operation by reentering the `darecon(8)` command without a `damount(8)` command.

On successful completion of the reconstruct (NON-TRANSFER COMMAND SUCCESS), a Bank Mount sequence is automatically initiated, and the host then restores any logged data blocks with errors.

The RECONSTRUCT action varies, depending on the Bank Mount Sequence of the target bank configuration defined by DSBL and SUBS. There are five cases to consider, shown in Table 4-1.

## 4.3 The RECONSTRUCT Command

**Table 4–1 Bank Configurations**

|        | <b>Configuration</b>                            | <b>Explanation</b>   |
|--------|---|--|
| Case 1 | DSBL = 0 and SUBS = 0                           | Reconstruction is invalid in this configuration.   |
| Case 2 | DSBL = nonzero, and SUBS = 0                    | The drive identified by DSBL is the RECONSTRUCT target. Upon successful reconstruction, DSBL is automatically set to 0.  |
| Case 3 | DSBL = nonzero, SUBS = nonzero, and DSBL = SUBS | The drive identified by DSBL/SUBS is reconstructed onto the standby drive. Upon successful reconstruction, DSBL is automatically set to 0, and SUBS is left unchanged.   |
| Case 4 | DSBL = nonzero, SUBS = nonzero, DSBL ≠ SUBS     | This is the same as Case 2, except one other drive is already assigned as standby. <ul style="list-style-type: none"><li>• target = 0: The drive identified by DSBL is the RECONSTRUCT target. Upon successful reconstruction, DSBL is automatically set to 0.</li><li>• target = 1: The drive identified by SUBS is the RECONSTRUCT target. Upon successful reconstruction, SUBS is automatically set to 0 (deassigning the standby drive).</li></ul> |
| Case 5 | DSBL = 0, SUBS = nonzero                        | The drive identified by SUBS is the RECONSTRUCT target. Upon successful reconstruction, SUBS is automatically set to 0 (deassigning the standby drive).  |

When a drive is deemed inoperable, use the `dadsbl(8)` command to load the DSBL field of the appropriate Bank Configuration Table with the inoperable drive number. All subsequent read operations on the bank use the parity replacement drive to provide correct data to the host, and all write operations avoid the disabled drive.

You now have two options:

- If no standby drive is available, physically replace the inoperable drive with a new drive:

1. Format (using the `daformat(8)` command) the new drive.
2. Reconstruct (using the `darecon(8)` command) the original drive's data.

When RECONSTRUCT completes without error, the DA controller automatically clears the DSBL field. This action falls under Case 2 in Table 4–1.

- If a standby drive is available, you can assign it to replace the inoperable drive:

1. Use the `dasubs(8)` command to set the SUBS field equal to the DSBL field of the Bank Configuration Table.
2. Enter the `darecon(8)` command to reconstruct the data.

When RECONSTRUCT completes without errors, the controller automatically clears the DSBL field. This action falls under Case 3 in Table 4–1.

## 4.3 The RECONSTRUCT Command

Once a standby drive is in operation (replacing an inoperable data/parity drive), the bank can tolerate another drive failure. If the next drive to fail is another data/parity drive, mark it disabled, using the `dadsbl(8)` command to set the DSBL field to the failed drive's number. Entering the `darecon(8)` command at this point results in either the DSBL drive or the SUBS drive being reconstructed, depending on the value of `target`. This action falls under Case 4 in Table 4–1.

When you physically replace an inoperable data/parity drive with a new drive and a standby drive is in operation replacing the original drive, bring the new drive on line by entering the `damount(8)` command, followed by the `darecon(8)` command. On successful reconstruction, the SUBS field is automatically set to 0, deassigning the standby drive. This action falls under Case 5 in Table 4–1.

Finally, if a standby drive becomes inoperable while it is assigned to replace a data/parity drive:

- If no other drive is inoperable (DSBL = 0 and SUBS = X):
  1. Set DSBL = Ah.
  2. Enter a `damount(8)` command.

The DA controller immediately sets DSBL equals X and SUBS equals 0, deassigning the standby and restoring the disabled status of the previously substituted drive. This deassigns the standby drive and informs the DA controller that the drive it was formerly replacing is still inoperable.

- If another drive is inoperable (DSBL nonzero), the bank is unusable.

## 4.4 The SCRUB Command

The SCRUB command (`dascrub(8)`) initiates a digital surface analysis of selected disk drives (previously formatted) within a bank.

The `func` field defines the function's performance as follows:

- 0h — Read check only. Preserves disk data.
- 1h — Write 00000000h, then read check. Overwrites disk data.
- 2h — Write FFFFFFFFh, then read check. Overwrites disk data.
- 3h — Write AAAAAAAAh, then read check. Overwrites disk data.
- 4h — Write 55555555h, then read check. Overwrites disk data.
- 5h — Write CCCCCCCh, then read check. Overwrites disk data.
- 6h — Write 33333333h, then read check. Overwrites disk data.
- 7h — Write 6DB6DB6Dh, then read check. Overwrites disk data.
- 8h — Write 92492492h, then read check. Overwrites disk data.
- 9h — Write C6DEC6DEh, then read check. Overwrites disk data.
- Ah — Write 39213921h, then read check. Overwrites disk data.
- Bh, Ch, Dh, Eh — Reserved (0s). Overwrites disk data.
- Fh — Write random, then read check. Overwrites disk data.

Depending on the `func` field, the operation will read-only the entire disks or write/read the entire disks, and then log any media errors encountered into the GET.



## 4.4 The SCRUB Command

If any errors are logged into the GET, reformat (merge the GET into the RFT) to map out new media errors.

## 4.5 The SPECIFYFLAW Command

Use the `daspecflaw(8)` command to specify the location of a media defect. (Used primarily to download flaw information before reformatting.)

Data disks are numbered 1 through 8, parity is 9, and standby is 10.

Flaw information specified using this command is entered into the disk GET. You can merge this data into the RFT when performing a FORMAT operation.



---

## Removal and Replacement Procedures

Chapter 5 describes how to remove and replace the following DECmpp 12000/Sx Parallel Disk Array (PDA) components:

- DA controller PCBs
- Disk canisters
- MUX PCBs
- Lightpipe PCB
- Power/Fan tray
- DA backplane power cables
- Clock PCB
- PDA–DPU interface cable
- Disk drive cables
- Lightpipe signal cable
- Lightpipe power cable
- Temperature sensor cable

---

### Warning

---

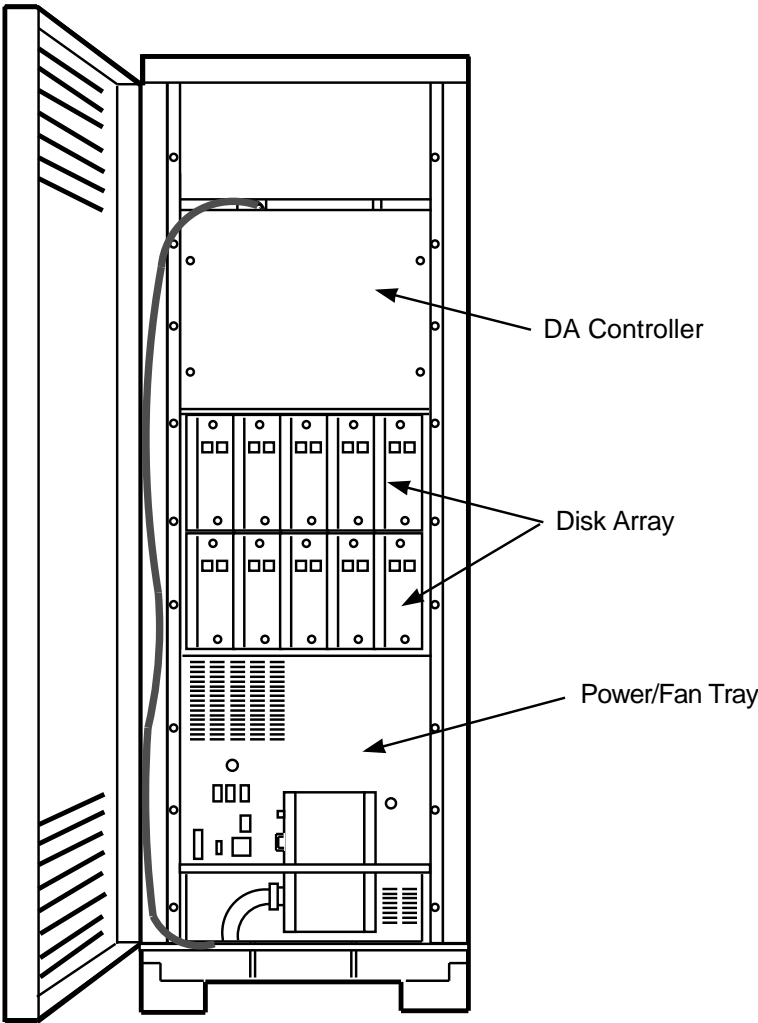
To avoid personal injury and damage to the equipment, turn off the front-end server and the data parallel unit (DPU), and turn the PDA circuit breaker OFF before beginning any service procedures.

---

Several maintenance procedures require you to extend the disk array (DA) controller chassis on its slides. This procedure is described in Section 5.1 and is not repeated in the individual maintenance procedures.

Figure 5–1 shows a rear view of the PDA enclosure and its internal components.

Figure 5-1 PDA Enclosure (rear)



MKV-040000314-85-RAGS

### 5.1 Extending the DA Controller Chassis

Follow these steps to extend and tilt the DA controller chassis. The process is shown in Figure 5–2.

---

**Caution**

---

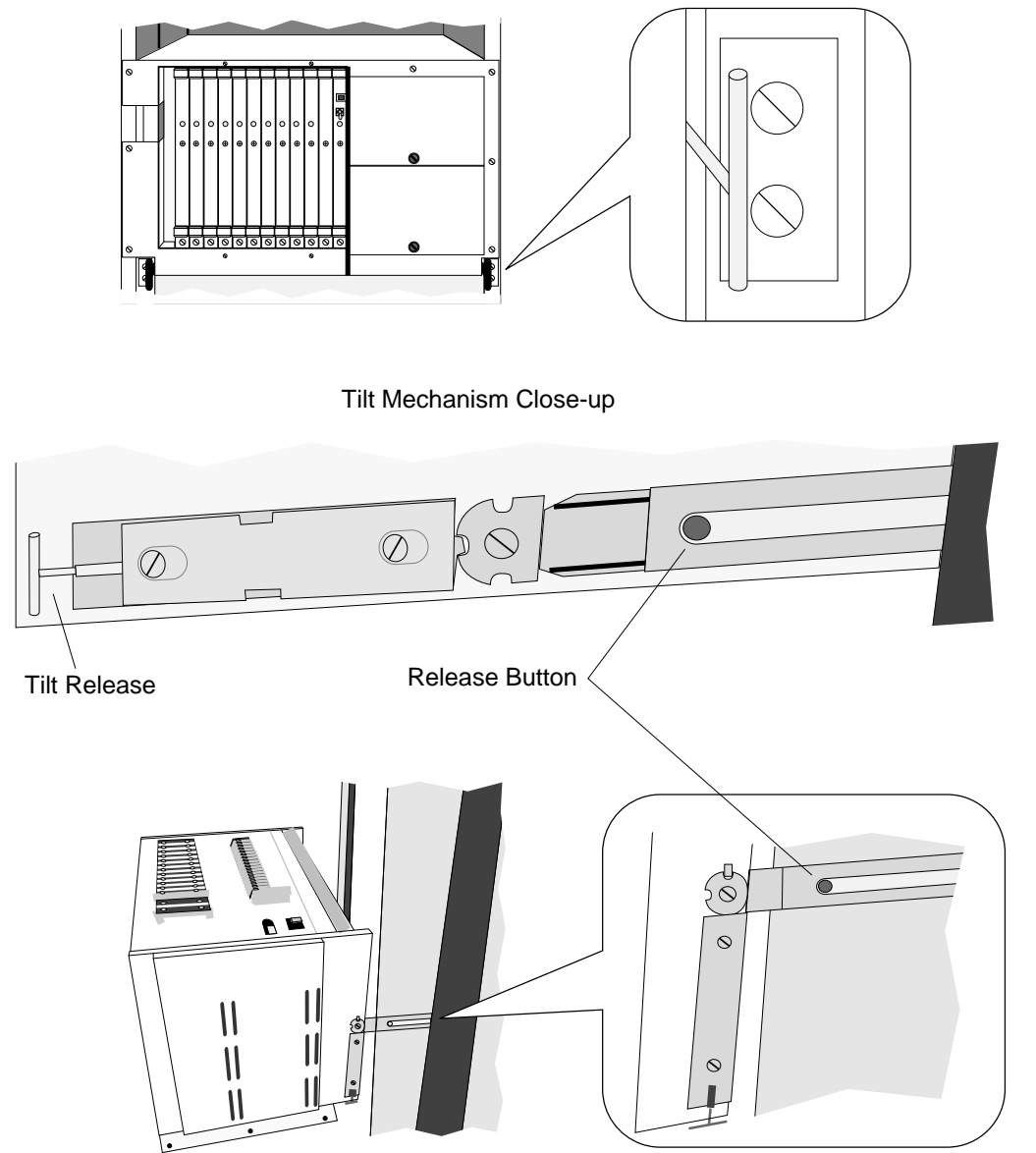
To avoid overheating while the DA controller chassis is extended for long periods of time, always use auxiliary fans while the PDA is turned on.

---

1. Turn the PDA circuit breaker OFF.
2. Open the PDA front door.
3. Remove and save the six screws attaching the DA controller to the PDA enclosure frame.
4. Pull the DA controller chassis forward until it stops (the slides lock).
5. Pull out the T-shaped tilt-release handles at the bottom corners of the DA controller chassis, allowing it to tilt forward. The chassis tends to fall forward as soon as you pull the handles. Make sure you are supporting it and controlling its movement. The chassis should lock in the 90-degree tilt position. If it doesn't, manually slide the tilt-release handles back into a locking position.
6. Close the DA controller chassis by pulling forward on the tilt-release handles, and tilting the DA controller chassis up until it locks in place. The chassis may tend to fall backward past its normal position. Make sure you are supporting it and controlling its movement.
7. Press the slide-release buttons on each rail and push the chassis back into the PDA enclosure.
8. Use the six screws to secure the DA controller chassis to the PDA enclosure frame.

## 5.1 Extending the DA Controller Chassis

Figure 5-2 Extending the Disk Array Controller



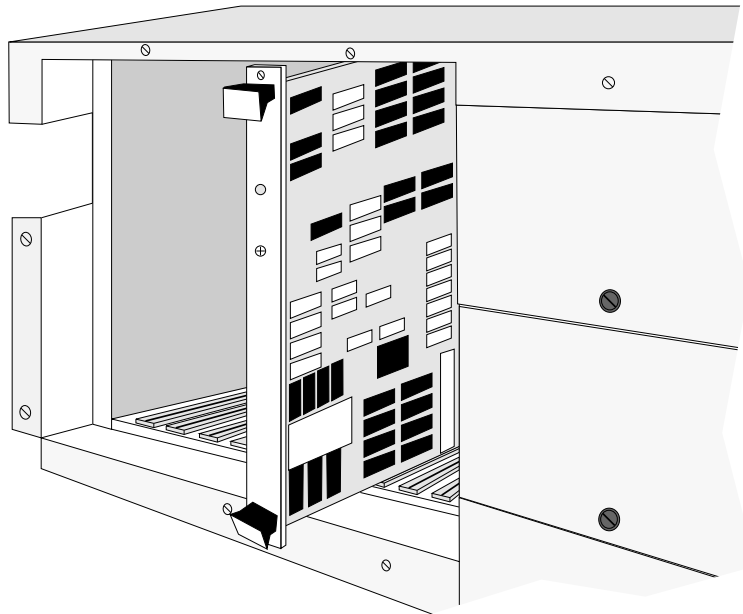
MKV-040000314-94-MPS

### 5.2 Replacing DA Controller PCBs

Follow these steps to remove and replace PCBs in the DA controller chassis, shown in Figure 5–3.

1. Turn the PDA circuit breaker OFF.
2. Open the PDA front door.
3. Remove the two top screws and loosen the two bottom screws attaching the DA controller front panel, shown in Figure 1–5. Remove the cover by lifting it off the bottom screws.
4. Put on an antistatic wrist strap as a precautionary measure.
5. Press open the top and bottom extraction levers on the PCB front edge, loosening it from the backplane.
6. Remove the PCB, pulling it straight out.
7. Holding the replacement PCB by the front edge, install it in the DA controller card cage, pushing the top and bottom extraction levers closed to seat it firmly on the backplane.
8. Replace the DA controller front panel by sliding it over the two bottom screws. Use the two original screws at the top.
9. Close the PDA front door.
10. Turn the PDA circuit breaker ON.

**Figure 5–3 Replacing the DA Controller PCB**



## 5.3 Replacing Disk Canisters

### 5.3 Replacing Disk Canisters

Follow these steps to remove and replace a disk canister.

1. Spin down the disk, following the disable procedures outlined in Chapter 4.
2. Open the PDA front or rear door to access the proper disk bank.
3. Put on an antistatic wrist strap as a precautionary measure.
4. Write the appropriate disk number and disk bank number on the replacement canister labels.
5. Set the disk power switch OFF (down).
6. Make sure the LED on the front of the disabled disk canister is OFF.

---

**Caution**

---

Be sure to slide the canister squarely out of and into the guide rails. Guide rails can break if the canister is twisted to one side.

---

7. Loosen the two captive screws on the disk canister front, and remove the disk canister by sliding it straight out of the chassis.
8. Set the jumpers and terminators on the new disk to the correct bank. Use the jumpers and terminator on the disabled disk as a guide. (Refer also to Section 2.3.1.)

---

**Caution**

---

Change only the bank select jumpers/terminator. Do not change any other jumpers on the disk drive PCB. Do not compare jumpers from one disk drive PCB to another. Many jumpers (especially the black-colored ones) are factory-set during testing to optimize performance.

---

9. Align the replacement disk canister top and bottom edges in the nylon guide rails, and slide the canister squarely and firmly into the DA backplane.
10. Tighten the two captive screws.
11. Set the disk power switch ON (up).
12. Make sure the green LED on the disk canister front is ON.
13. Close the PDA door (front or rear).
14. Format the new disk (refer to Chapter 4).

Refer to Chapter 7 for PDA upgrade procedures (installing additional disk canisters).



### 5.4 Replacing MUX PCBs

Follow the steps below to remove and replace MUX PCBs. Refer to Figure 1–9 for MUX PCB connectors.

1. Turn the PDA circuit breaker OFF.
2. Open the PDA rear door.
3. Put on an ESD wrist strap as a precautionary measure.
4. Remove the associated disk canister that covers the MUX PCB. For additional working space, remove the adjacent disk canisters also.
5. Remove the MUX PCB.
6. Holding the replacement MUX PCB by the front edge, install it in the PDA, seating it firmly on the backplane.
7. Replace the disk canisters.
8. Close the PDA rear door.
9. Turn the PDA circuit breaker ON.

## 5.5 Replacing the Lightpipe PCB

### 5.5 Replacing the Lightpipe PCB

Follow these steps to remove and replace the lightpipe PCB in the PDA.

1. Turn the PDA circuit breaker OFF.
2. Open the PDA front door. The lightpipe PCB is mounted to the inside surface of the front door.
3. Disconnect the lightpipe signal cable and power cable.
4. Loosen but do not remove the four screws that secure the lightpipe PCB to the door.
5. Lift the lightpipe PCB bracket up and off the mounting studs and screws.
6. Put the replacement lightpipe PCB into position over the mounting studs and screws, checking the alignment of the lightpipe and the lightpipe cutouts in the door.
7. Secure the bracket of the replacement PCB to the door by turning the four screws. Do not tighten; the PCB should be able to be moved slightly.
8. Connect the lightpipe power supply cable and the 50-pin ribbon cable to the lightpipe PCB.
9. Turn the PDA circuit breaker ON.
10. Place the lightpipe LED test switch in the TEST position.
11. Fine tune the alignment of the lightpipe and the cutouts in the door to ensure maximum light transmission through the cutouts.
12. Tighten the four screws securing the PCB bracket to the door.
13. Place the lightpipe LED test switch in the NORMAL position.
14. Close the PDA front door.

### 5.6 Replacing Power/Fan Tray

Follow these steps to remove and replace the PDA power/fan tray, as shown in Figure 5-4 and Figure 5-5.

---

#### Warning

---

This procedure requires two people. The power/fan tray weighs 41.73 kg (92 pounds). To avoid injury, use proper lifting techniques when removing or installing the power/fan tray.

---

---

#### Caution

---

Check the specification label on the replacement tray. Make sure that the replacement tray provides the correct power level. Arrays with more than 10 disk drives require a power supply rated at 2500 W.

---

1. Open the PDA front and rear doors.
2. Turn the PDA circuit breaker OFF.
3. Unplug the PDA power cable from the source.
4. Disconnect the PDA power cable where it enters the power fan tray by pulling sideways to the left.
5. Disconnect the lightpipe power cable and the temperature sensor cable.
6. Remove the two screws attaching the power/fan tray rear to the air exhaust plenum.
7. Remove the four screws attaching the power/fan tray front to the enclosure side rails.

---

#### Caution

---

The next step removes the power/fan tray. When the tray reaches the end of its travel, ensure that the rear is supported so that it does not fall and damage the AC input box.

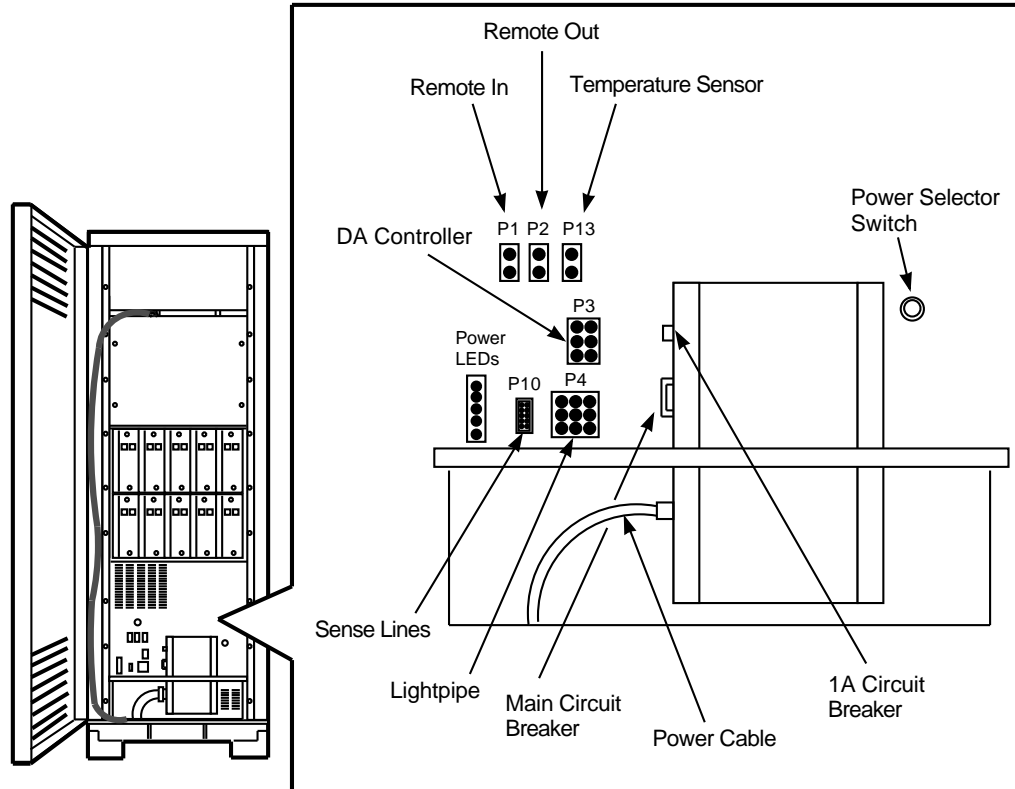
---

8. Pull the tray straight out of the enclosure from the front. (You may want someone to brace the PDA enclosure as you pull the tray out.)
9. Slide the replacement tray into the enclosure.
10. Using the screws removed in steps 6 and 7, secure the power/fan tray to the enclosure with four screws in the front and two screws in the rear.
11. Connect the lightpipe power cable and the temperature sensor cable.
12. Connect the PDA power cable to the AC input box.
13. Plug in the PDA power cable.
14. Turn the PDA circuit breaker ON.
15. Observe the front power indicator and the power/fan tray status indicators in the rear. Ensure that all indicators display a normal condition.

## 5.6 Replacing Power/Fan Tray

16. Close the PDA front and rear doors.

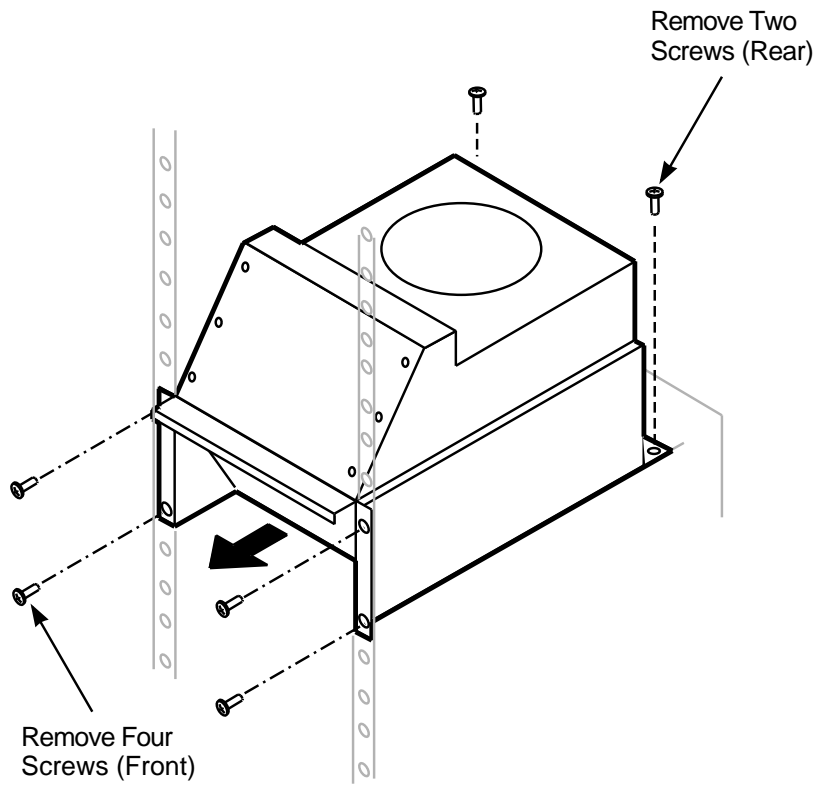
Figure 5-4 Power/Fan Tray (rear)



MKV-04000314-86-RAGS

## 5.6 Replacing Power/Fan Tray

Figure 5-5 Replacing the Power/Fan Tray



MKV-040000314-87-RAGS

## 5.7 Replacing Power Cables to the DA Backplane

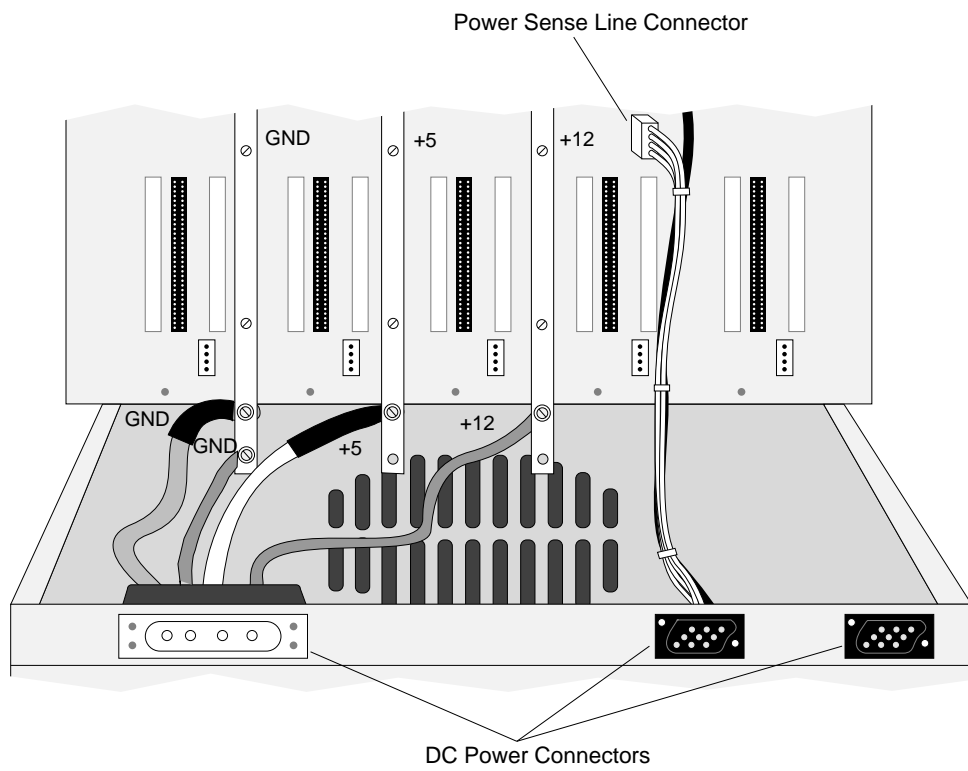
### 5.7 Replacing Power Cables to the DA Backplane

Follow these steps to replace the +5 V, +12 V, or ground cable between the power supply and the DA backplane bus bar system. Figure 5–6 shows the cable connections. Put on an ESD wrist strap as a precautionary measure.

1. Turn the PDA circuit breaker OFF.
2. Unplug the PDA from the power source.
3. Unplug all DA controller PCBs and slide them out 1 inch from the backplane.
4. Unplug all disk canisters and slide them out 1 inch from the DA backplane.
5. Label all power cables.
6. Remove the power/fan tray (Section 5.6).
7. Remove those disk canisters blocking access to the DA backplane bus bar.
8. Remove the nut fastening the cable to the appropriate power bridge high-current connector.
9. Remove the screw and nut connecting the cable to the backplane bus bar.
10. Fasten the replacement cable to the backplane bus bar, using a screw and nut.
11. With the nut removed above, fasten the replacement cable to the power bridge high-current connector.
12. Using an ohmmeter, verify that there are no shorts between cables or to ground.
13. Replace the power/fan tray.
14. Return all disk canisters to their slots and plug in all disk canisters and DA controller PCBs.
15. Turn the PDA circuit breaker ON.

## 5.7 Replacing Power Cables to the DA Backplane

Figure 5-6 Power Bridge and Cables



MKV-04000314-79-MPS

## 5.8 Replacing Clock PCB

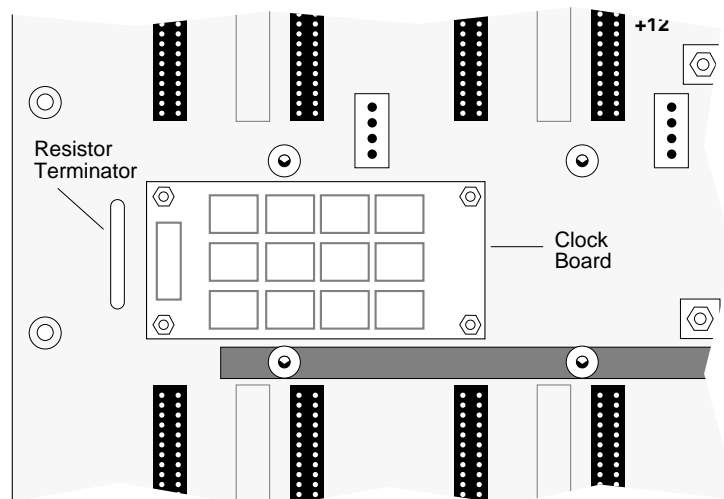
### 5.8 Replacing Clock PCB

The clock PCB is on the far left of the DA backplane rear side, shown in Figure 1–9. Follow these steps to remove and replace the clock PCB, referring to Figure 5–7.

Workspace is very constricted because of the disk chassis frame.

1. Open the PDA back door.
2. Turn the PDA circuit breaker OFF.
3. Put on an ESD wrist strap as a precautionary measure.
4. Remove those left-most disk canisters blocking access to the clock PCB. In an 8-disk bank, removing the first two upper and lower canisters on the left should be sufficient.
5. Remove and save the four plastic nuts attaching the clock PCB corners to the plastic standoffs on the DA backplane.
6. The 20-pin connector is on the bottom-left side of the clock PCB. Grasping the board by the edges, remove the PCB.
7. Install the replacement clock PCB by sliding it carefully behind the disk chassis rear cross-piece and seating it firmly in the 20-pin connector.
8. Secure the clock PCB, screwing the four plastic nuts onto the standoffs. Do not overtighten!
9. Replace the disk canisters removed in step 4 in their original positions.
10. Turn the PDA circuit breaker ON.

Figure 5–7 Replacing the Clock PCB





### 5.9 Replacing DA Controller Backplane Cabling

The following cables connect to the DA controller backplane:

- Two 50-pin PDA-DPU interface cables: control and data
- Ten 40-pin disk control/status cables (34-pin at the DA backplane)
- Ten 20-pin disk data cables
- One flat-braid ground cable
- One flat-braid +5 V cable
- One jacketed multiconductor +12 V cable

Individual cable replacement is described in sections that follow.

#### 5.9.1 Replacing PDA-DPU Interface Cable

Follow these steps to replace the interface cable between the DPU and the PDA.

This procedure is easier if you feed the new cable along at the same time that you are removing the old one.

1. Open the PDA and DPU enclosure front and rear doors.
2. Turn off the DPU with the front keyswitch.
3. Turn both the PDA and DPU circuit breakers OFF.
4. Extend the DA controller chassis.
5. Remove screws fastening the top air baffle, and remove the baffle. There is one top baffle in both the 1- and 2-bank configurations.
6. From the rear, remove the air baffle panel behind the DA controller chassis.
7. Disconnect the original interface cable from the DA controller backplane (50-pin connectors at P13 or P11).
8. Remove one or two screws on the cable strain relief clamp on the DA controller rear, easing it enough to slide the old cable out.
9. Four spring-loaded plungers secure the cable clamp to the PDA frame. Pull and twist each plunger, locking it in the open position and unfastening the clamp from the frame.
10. The cable guide clamp is held together with six alternate-position screws. Remove two or three screws at the end holding the interface cables, and slip the old cable out of the clamp sideways.  
Keep the cable clamp partially fastened to remove an individual cable without disturbing the others.
11. Remove all cable ties securing the cable along its travel down the PDA enclosure.
12. Label both ends of the replacement cable.
13. Connect the replacement cable connectors to the DA controller backplane.
14. Slide the cable into the cable strain relief clamp, and replace the screws in the clamp.

## 5.9 Replacing DA Controller Backplane Cabling

15. Slide the replacement cable into the cable guide clamp, positioning it beneath the other ribbon cable. Adjust the replacement cable slack, matching it to the other ribbon cable as a guide, and tighten the clamp screws until the cables are held firmly.
16. Put the cable guide clamp back into position, and release each plunger by twisting it slightly until the end goes into the hole in the PDA frame.
17. Draw both the old and replacement cables through both the PDA and DPU enclosures, following the path of the other interface cable through both enclosures, and feed the new cable into the DPU.
18. At the DPU card cage, remove the Parallel VME 6U Adapter PCB with the VME interface PCB attached.
19. Two nuts attached to studs fasten a U-bracket to the front of the Parallel VME 6U Adapter PCB. This U-bracket functions as a cable clamp for the interface cable. Loosen the nuts until you can move the U-bracket enough to pull the old cable through.
20. Disconnect the old cable from the 50-pin PCB connector, and pull it through the U-bracket.
21. Feed the replacement cable through the U-bracket and connect it to the PCB.
22. Tighten the nuts on the Parallel VME 6U Adapter PCB U-bracket, securing the cable.
23. Replace the Parallel VME 6U Adapter PCB in the DPU card cage.
24. Replace the PDA rear and top air baffles.
25. Close the DA controller chassis, making sure that no cables are pinched.
26. Turn the DPU and PDA circuit breakers ON.
27. Turn on the DPU with the front keyswitch.

## 5.9 Replacing DA Controller Backplane Cabling

### 5.9.2 Replacing Backplane Disk Cables

Follow these steps to replace the disk control/status or data cables from the DA controller backplane to the DA backplane.

The control/status cable has a 40-pin connector to the DA controller backplane and a 34-pin connector to the DA backplane.

1. Open the PDA front and rear doors.
2. Turn the PDA circuit breaker OFF.
3. Extend the DA controller chassis.
4. Remove the screws fastening the top air baffle and remove it.
5. Disconnect the original disk signal cable from the DA controller backplane and from the DA backplane.
6. Remove one or two screws on the cable strain relief clamp on the DA controller rear, easing it enough to slide the old cable out.
7. Label both ends of the replacement cable.
8. Using the old cable as a guide, fold the replacement cable to match.
9. Connect the replacement cable to the DA controller backplane.
10. Slide the cable into the cable strain relief clamp, and replace the screws in the clamp.
11. Connect the opposite end to the DA backplane.
12. Replace the PDA top air baffle.
13. Close the DA controller chassis.
14. Turn the PDA circuit breaker ON.
15. Close the PDA front and rear doors.

## 5.10 Replacing the Lightpipe Signal Cable

### 5.10 Replacing the Lightpipe Signal Cable

Follow these steps to replace the lightpipe signal cable from the DA backplane to the lightpipe.

The lightpipe split-ribbon cable has a 50-pin connector to the lightpipe and three 10-pin connectors (one for each disk bank) to the DA backplanes.

1. Open the PDA front and rear doors.
2. Turn the PDA circuit breaker OFF.
3. Disconnect the lightpipe signal cable from the 50-pin connector at the bottom edge of the lightpipe PCB.
4. Remove any cable ties securing the ribbon cable to the enclosure or chassis and push the cable to the back of the PDA, between the air baffles and the top.
5. At the rear, remove the far right disk canisters to access the lightpipe signal connector.
6. Disconnect the ends of the cable from the DA backplane 10-pin connectors.
7. Pull the upper end of the cable down from the enclosure top, through the side bracket, and remove the cable.
8. Connect the replacement cable 10-pin connectors to the DA backplane.
9. Feed the 50-pin end through the backplane side bracket, and push the end over the air baffle to the front.
10. From the front, draw the cable forward, and connect it to the lightpipe PCB 50-pin connector.
11. Replace the disk canisters in the disk chassis.
12. Turn the PDA circuit breaker ON.
13. Close the PDA front and rear doors.

### 5.11 Replacing Lightpipe Power Cable

Follow these steps to replace the lightpipe power cable from the power supply to the lightpipe PCB.

1. Open the PDA front and rear doors.
2. Turn the PDA circuit breaker OFF.
3. Disconnect the lightpipe power cable from the lower-left side of the power/fan tray.
4. Clip off cable ties securing the power cable to the cable mounts on the PDA enclosure frame, following the cable path to the front of the PDA.
5. Disconnect the power cable from the lightpipe PCB.
6. Connect the replacement power cable to the lightpipe PCB.
7. Feed the cable toward the side and rear of the enclosure, using cable ties to fasten it to cable mounts on the enclosure frame.
8. Connect the cable to the power/fan tray.
9. Turn the PDA circuit breaker ON.
10. Close the front and rear doors.

## 5.12 Replacing Temperature Sensor Cable

### 5.12 Replacing Temperature Sensor Cable

Follow these steps to replace the temperature sensor cable from the power/fan tray to the DA controller chassis enclosure.

1. Open the PDA front and rear doors.
2. Turn the PDA circuit breaker OFF.
3. Disconnect the temperature cable from the lower left side of the power/fan tray.
4. Clip off cable ties securing the sensor cable to the cable mounts on the PDA enclosure frame, following the cable path to the front of the PDA.
5. Clip the cable tie fastening the sensor cable to the vented top of the DA controller chassis under the right fan.
6. Ensure that the sensor is level and in the same location as the original.
7. Use a cable tie to fasten the replacement sensor cable to the top of the DA controller chassis.
8. Feed the cable toward the side and rear of the enclosure, using cable ties to fasten it to cable mounts on the enclosure frame.
9. Connect the cable to the power/fan tray.
10. Turn the PDA circuit breaker ON.
11. Close the front and rear doors.

---

## Parallel Disk Array Diagnostics

Chapter 6 provides descriptions of the lightpipe indicator displays showing the disk status, and descriptions of the diagnostic procedures and status error codes.

### 6.1 Lightpipe Indicator Status

Fourteen indicators in the DECmpp 12000/Sx parallel disk array (PDA) front panel lightpipe (shown in Figure 6-1) provide status for the PDA and for disks in the disk array (DA) banks.

The first two indicators show the system power status:

- Off — Power system off
- Yellow — Power system sequencing up, or Fault condition detected
- Green — Power system up, no faults detected

The other indicators, in three groups of four, show the disk status within each disk bank. Bank 3 is not used.

- OD — Disks 1, 3, 5, 7
- EV — Disks 2, 4, 6, 8
- P — Parity disk
- S — Standby disk

A sector pulse from the disk drives the disk status lightpipe display. If there is no sector pulse, the indicator is yellow. If there is no disk installed, there is no sector pulse, and the indicator is off:

- Driving yellow — Disk not spinning
- Driving green — Disk spinning and accessed (read, write, or seek)
- None (indicator off) — Disk spinning and idle (not accessed) or no disk installed

Because the indicators provide status for multiple disks in a given bank, the lightpipe display priority is as follows:

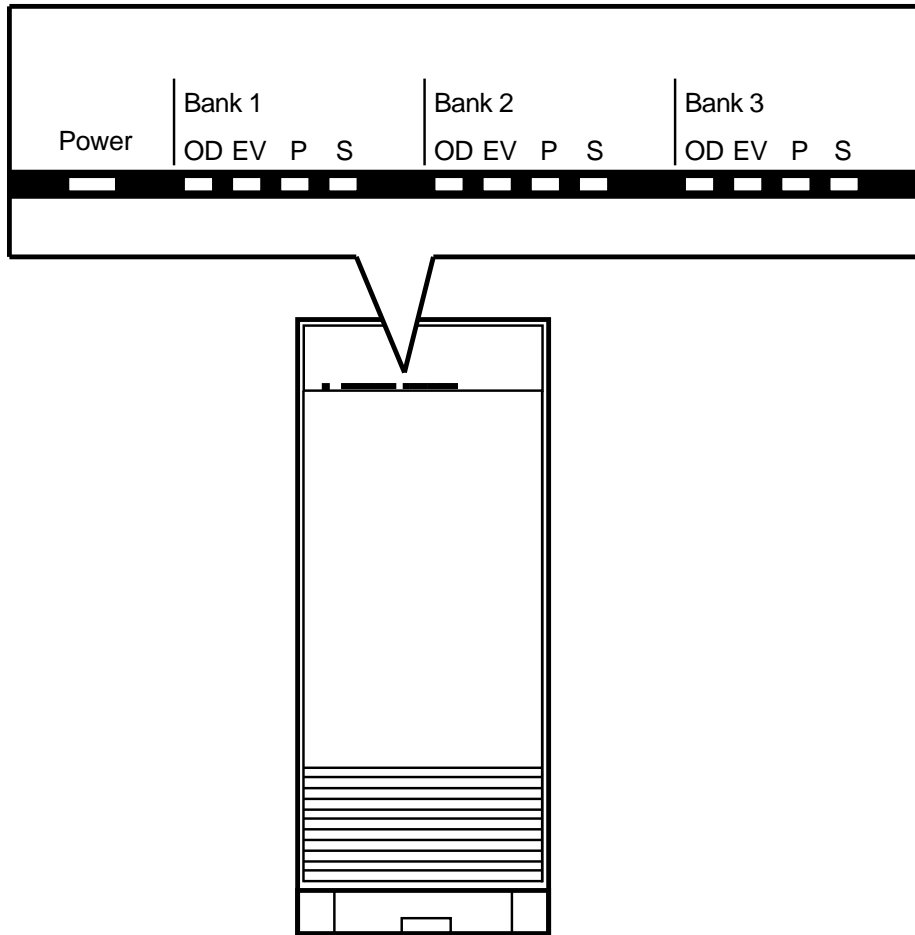
- Yellow — Any disk driving yellow
- Green — Any disk driving green, and no disk driving yellow
- Off — No disk driving yellow, and no disk driving green

## 6.1 Lightpipe Indicator Status

The indicator on the disk canister front provides status for each disk:

- Green when the disk is spinning
- Off when the disk is not spinning

Figure 6–1 PDA Front Indicators



MKV-040000314-84-RAGS



## 6.2 Diagnostic Procedures

The standard DECmpp 12000/Sx system diagnostics test the front-end interface, the array control unit (ACU), the processor element (PE) array, and the PE array/router connections. Additional diagnostics test the Parallel VME 6U Adapter PCB, the VME 6U interface PCB, and the PDA disk array controller.

---

### Caution

---

Running diagnostic programs can cause the system to crash, especially if the programs encounter problems. You should warn the users and prepare the system for a possible crash before running diagnostics. It is a good policy not to run diagnostics while users are on the system.

Avoid aborting any diagnostic program. In particular, do not abort the process while any diagnostic is running. Diagnostic programs change the system microcode, and then replace it when they are done. If they are aborted, they leave the system in an unknown state. If this should happen, you can recover by entering either:

```
# dpuboot -v
```

or:

```
# mpshutdown
# dpumanager
```

---

## 6.3 Diagnostic Environment

Table 6-1 shows the PDA top-level programs and the level-2 programs and describes what each one tests. The individual diagnostics are described following this table.

**Table 6-1 Diagnostic Programs**

| Top Level | Description                         | Level Two | Description                      |
|-----------|-------------------------------------|-----------|----------------------------------|
| vme_diags | DECmpp 12000<br>/Sx extended<br>VME | mpda      | Parallel VME 6U Adapter CSR test |
|           |                                     | vme       | Memory 6U Adapter test           |
| da_diag   | Tests PDA                           | da_intern | DA controller PCB test           |
|           |                                     | da_config | R/W configuration test           |
|           |                                     | da_buff   | R/W buffer test                  |
|           |                                     | da_mount  | Mount bank test                  |
|           |                                     | da_seek   | Read/seek test                   |
|           |                                     | da_disk   | R/W disk test                    |

---

## 6.3 Diagnostic Environment

### 6.3.1 Automatic Mode

To run a top-level test program, enter the test program name at the system prompt. It executes all the level-2 tests, scrolls messages to the screen, and copies the same messages to the log file in the current directory. All actual testing is done by the individual tests comprising level-2 programs.

## 6.4 VME Adapter Tests

The diagnostic `vme_diags` writes and reads the Parallel VME 6U Adapter CSR registers of both the PDA disk array interface PCB and the PVME PCB.

The `vme_diags` diagnostic expects the PDA 6U Adapter CSR address to be at 0xFC800014 and the VME memory 6U Adapter CSR to be at 0xE8000014. (Refer to the *DECmpp 12000/Sx Parallel VME Reference Manual*.)

## 6.5 PDA Test Programs

To begin PDA diagnostic tests, enter:

```
# da_diag
```

All of the level-2 tests execute in the order shown in Table 6-1.

Each level-2 test is described in the following sections. You can run these tests individually by entering the test name at the system prompt.

### 6.5.1 Internal DA Controller Test (`da_intern`)

This test runs the disk array controller internal diagnostic. It checks the boards in the array controller.

### 6.5.2 Read/Write Configuration Test (`da_config`)

---

**Caution**

---

User data may be overwritten with this test.

---

This test allows you to read/write portions of bank 4's system configuration table and also compare them, if you wish. The patterns to be read/written are a variety of test patterns (hex 00, 0F, F0, FF, 33, 55, AA, CC). (Bank 4 is used as a scratch pad for diagnostics.)

### 6.5.3 Read/Write Buffer Test (`da_buff`)

This test causes DMA-type data transactions between the front end and the disk buffer space in the array controller. It is similar to R/W tests to the disks, but no data flows between the buffer space and the disk drives.

### 6.5.4 Mount Bank Test (`da_mount`)

This test reads the system configuration table, determines the status of the various drives in the user-specified bank, and reports any unexpected drive status code (including not on line/not mounted) for the drives that are supposed to be mounted.

### 6.5.5 Read/Seek Test (da\_seek)

This test performs reads of blocks on the disk in an ordered sequence that is designed to exercise the seek operations of the drives.

### 6.5.6 Read/Write Disk Test (da\_disk)

---

**Caution**

---

User data may be overwritten with this test.

---

Read/Write disk test is a read/write/compare test. It runs within a given partition in a factory-specified area, and data transfers occur in packet sizes that are factory-defined.

## 6.6 DA Controller Tests

You can also test the DA controller boards by running diagnostics via the RS-232 com port on the CPU board:

1. Connect a standard telephone cable from the printer port of the front-end workstation to the communication port on the DA controller CPU printed circuit board (PCB).

2. At the prompt, enter:

```
# tip da.9600
```

The screen displays `connect /dev/tty03` in response.

3. To access the diagnostic menu, enter:

```
# Ctrl/C
```

4. To run offline disk diagnostics, enter:

```
# b
# 1aa (4 drives)
# 1ff (8 drives)
```

5. To run offline board diagnostics, enter:

```
# a
```

## 6.7 Status Operations

The following sections describe the status error codes and command-related error codes.

### 6.7.1 Disk Error Logging

The Disk Error Logging Status Messages provide current information about detected media hard errors and soft (correctable) errors.

When a disk data transfer error occurs, the relevant information is stored in the systems event log. Use `uerf(8)` to see the following disk error information.

## 6.7 Status Operations

FERP Error recovery:  
Bit  
F E R P  
- - - -  
0 0 1 0 corrected by retries (2)  
0 1 X 0 corrected by ECC correction (4,6)  
0 X X 1 corrected by parity replacement (1,3,5,7)  
1 X X X not correctable (8..15)

BANK Indicates the bank number with error:  
1Xh bank # 1  
2Xh bank # 2  
3Xh bank # 3

DISK Indicates the disk number with error:  
1h data disk # 1  
2h data disk # 2  
3h data disk # 3  
4h data disk # 4  
5h data disk # 5  
6h data disk # 6  
7h data disk # 7  
8h data disk # 8  
9h Parity disk # 9  
Ah Stand-by disk # 10

H 0 = data field error  
1 = header field error

CYL Actual physical cylinder number with error

HEAD Actual physical head number with error

SECTOR Actual logical sector number with error

POS Actual physical bit position of sector with error from index

BITLEN Actual physical bit length of sector with error

# OF  
RETRIES Number of retry attempts

MATCH Number of ECC matches for data field  
retry attempts

- 1) If the data field ECC matches for all  
retry attempts and retries fail, ECC  
correction is attempted.
- 2) If the data field ECC does not match for any  
retry attempt and retries fail, ECC correction  
is not attempted.

REQCHS Host block address of error

In the preceding file example, the following codes are used:

- F — Failed, not recovered
- E — ECC correction attempted
- R — Retries attempted
- P — Parity replacement attempted

### 6.7.2 Error Information

When a hard or soft media error occurs, information is stored in the error log. The REQCHS is the disk address in the user-accessible area of the disk. The REQCHS is used as input to the dampachs(8) command to determine the disk array partition and where within that partition the error occurred. The error position is also logged to the GET.

```
MESSAGE\DA_MEDIA_LOG: ferp 0x2, ms0, bnk 1,
\_dsk 4, hdr 1, (c,h,s) 16,0,255,
\_pos,len 29754,8816, rtry 3, mtch 2,
\REQCHS (10,0,255)
```

If a disk drive has a drive fault, the fault status is printed in the error log:

```
MESSAGE\DA_FAULT_STATUS: 800
```

The status bits are defined in Table 6–2.

**Table 6–2 Status Bit Definitions**

| Bit Position | Bit Value | Function                               |
|--------------|-----------|--|
| 15           | 0         | Reserved                               |
| 14           | 0         | Reserved                               |
| 13           | 0         | Reserved                               |
| 12           | C         | Write-protected, fixed medium          |
| 11           | 0         | Spindle synchronized                   |
| 10           | 0         | Reserved                               |
| 9            | C         | Spindle motor stopped                  |
| 8            | C         | Power-on reset conditions exist        |
| 7            | C         | Command data parity fault              |
| 6            | C         | Interface fault                        |
| 5            | C         | Invalid or unimplemented command fault |
| 4            | C         | Seek fault                             |
| 3            | C         | Write gate with track offset fault     |
| 2            | C         | Vendor-unique status available         |
| 1            | C         | Write fault                            |
| 0            | 0         | Reserved                               |

If the disk array cannot successfully complete a data transfer request, a message is printed to the error log. Most of these errors are retried, as indicated by the initial word of the message. If retries fail or the error is such that it cannot be retried, the first word of the message is DA\_XFER\_ERROR:

```
MESSAGE\DA_XFER_RETRY: sdr 8405, code bd
\_bank 1, disk 2, blockcount 0
\_flags 19, dev 5000, bno 0, count 4096
```

- Certain DA maintenance commands can result in the following nontransfer error:

```
MESSAGE\DA_NONXFER_ERROR: sdr 8d01, code f4
\_bank 0, disk 0
```

## 6.7 Status Operations

- If a disk drive fails, the DA driver performs some functions automatically, and the following message appears in the error log.

```
MESSAGE\DA_AUTO_DISABLE: bank 1, drive 2  
\_failed
```

If this message appears or the status indicators show a drive failure, use the `dabstat` command to determine the exact nature of the failure and the appropriate recovery procedures to be done (such as replacing a failed drive).

After a drive fails, several actions can result, depending on the specific system configuration and status. Some or all of the following actions and related messages may occur.

- The DA driver attempts to disable a failed drive. The following message is printed when the drive has been disabled:

```
MESSAGE\DA_AUTO_DISABLE: bank 1, drive 2  
\_disabled
```

- If the drive cannot be disabled, the appropriate bank is disabled, and the following message is logged:

```
MESSAGE\DA_AUTO_DISABLE: multiple drive failure  
\bank 1 disabled
```

- After the drive has been disabled, the DA driver attempts to reconstruct the failed drive data on an available standby drive. If there is no standby, the following message is logged:

```
MESSAGE\DA_AUTO_RECONSTRUCT:  
\standby not available  
\bank 1 operating without protection
```

- A successful reconstruction is logged with the following messages:

```
MESSAGE\DA_AUTO_RECONSTRUCT: bank 1, started  
MESSAGE\DA_AUTO_RECONSTRUCT: bank 1, complete
```

- During reconstruct, it is possible (but unlikely if maintenance has been performed) that a sector cannot be reconstructed, and the following message is logged.

```
MESSAGE\DA_AUTO_RECONSTRUCT:  
\unrecoverable media error  
\bank 1, CHS(23,4,13) not recovered
```

The specified cylinder, head, and sector CHS are provided so that the `damapchs(8)` command, used with the `icheck(8)` command, can be used to locate the file, if any, that contains the unrecoverable data. The bad file can then be dealt with as the user sees fit.

- A catastrophic failure during reconstruct is logged with the following message:

```
MESSAGE\DA_AUTO_RECONSTRUCT:  
\reconstruct aborted  
\bank 1 operating without protection
```

### 6.7.3 Extended Transfer Error Status

The Extended Transfer Error Status provides the enumerated cause of an uncorrectable transfer error (in CODE) and the amount of data successfully transferred (in complete blocks) before the transfer was terminated.

When this type of error occurs, the information is put into the system event log:

CODE Refer to command-related error codes for the definitive command error codes.

BANK Indicates the bank number with error:  
1Xh bank # 1  
2Xh bank # 2  
3Xh bank # 3  
00 can be any bank

DISK Indicates the disk number with error:  
0h no disk  
1h data disk # 1  
2h data disk # 2  
3h data disk # 3  
4h data disk # 4  
5h data disk # 5  
6h data disk # 6  
7h data disk # 7  
8h data disk # 8  
9h Parity disk # 9  
Ah Stand-by disk # 10

BLK\_CNT Indicates the number of blocks written to the disk before the write command was terminated; for read commands it is always zero. (This is a 16-bit unsigned value.)

## 6.7 Status Operations

### 6.7.4 Command/Status Summary

Table 6–3 lists the command-related error codes.

**Table 6–3 Command-Related Error Codes**

| Code       | Description   |
|------------|---|
| -5 (FBh)   | Invalid start block number  |
| -6 (FAh)   | Invalid block count   |
| -8 (F8h)   | Invalid command parameter   |
| -9 (F7h)   | Invalid count field (RdConfig, WrConfig)                          |
| -10 (F6h)  | Invalid bank number   |
| -11 (F5h)  | Invalid drive ID number   |
| -12 (F4h)  | Invalid disk number   |
| -13 (F3h)  | Invalid channel number  |
| -17 (EFh)  | Unknown transaction ID (Not in use)                               |
| -18 (EEh)  | Bank not mounted  |
| -49 (CFh)  | Bank write-protect  |
| -51 (CDh)  | Bank not on line  |
| -52 (CCh)  | Reconstruct not possible  |
| -53 (CBh)  | Bank busy   |
| -65 (BFh)  | CPU parity error  |
| -66 (BEh)  | FIFO or cable parity error  |
| -67 (BDh)  | Disk fault  |
| -68 (BCh)  | Unit not ready  |
| -69 (BBh)  | Repeated seek failure   |
| -70 (bah)  | Unit not selected   |
| -71 (B9h)  | Invalid or missing raw flaw tables                                |
| -72 (B8h)  | Format abort—verify failed  |
| -74 (B6h)  | Too many bad tracks   |
| -75 (B5h)  | GET overflow  |
| -76 (B4h)  | No RCLK (clock from drive)  |
| -81 (AFh)  | Disk Data Controller (DDC) error (FIFO data lost, sector overrun) |
| -82 (AEh)  | ECC data correction failed  |
| -84 (ACh)  | No data sync  |
| -87 (A9h)  | Header CRC error or sector not found                              |
| -90 (A6h)  | Temporal parity fault   |
| -94 (A2h)  | Buffer not present error  |
| -95 (A1h)  | Terminated from Abort Command                                     |
| -96 (A0h)  | Unrecoverable hardware error                                      |
| -114 (8Eh) | Data Channel timeout  |
| -115 (8Dh) | Seek timeout  |
| -116 (8Ch) | DDC timeout   |



---

## Upgrading the Parallel Disk Array

The DECmpp 12000/Sx parallel disk array (PDA) supports configurations of 4, 8, or 16 data disks within a single enclosure. A small configuration can be expanded by adding more storage capacity within the same enclosure or by adding additional PDA enclosures.

---

**Note**

---

Always back up all data on the disk array before beginning any upgrade procedures.

---

## 7.1 Upgrading to Eight Disk Drives

### 7.1 Upgrading to Eight Disk Drives

Upgrading the disk array from four to eight disk drives involves plugging in the new drive canisters and formatting the drives. Follow these steps to upgrade your PDA to eight disk drives.

1. Open the PDA rear door to access the disk chassis.
2. Make sure the PDA circuit breaker is OFF.
3. Put on an antistatic wrist strap as a precautionary measure.
4. Remove the screws securing the blank panels covering the unused disk slots (slot numbers 1, 3, 5, and 7). Remove the blank panels and discard them. Remove the blank panel for slot "S" at this time if you are also installing a standby drive.
5. Referring to Figure 1–9, install four new MUX PCBs on the DA backplane MUX connectors: J1M1, J3M1, J5M1, and J7M1. (Install a fifth MUX printed circuit board (PCB) if the optional standby disk is also being added.)
6. The new disk canisters are factory-configured and labeled. If there is a need to alter the configuration, refer to Section 2.3.1 to set terminators and jumpers on each new disk to correspond to the bank number.

---

**Caution**

---

Be sure to slide the canister squarely into the guide rails. Guide rails can break if the canister is twisted to one side.

---

7. Align each new disk canister top and bottom edges in the nylon guide rails, and push the canister squarely into the disk chassis.
8. Label each new disk with the correct bank and disk numbers if they are not marked so.
9. Tighten the two captive screws securing each disk canister to the chassis.
10. Turn each disk canister switch ON (up).
11. Open the PDA front door.
12. Remove and save four screws attaching the disk array (DA) controller front panel, and remove the panel.
13. Insert four new disk array (DA) controller cards in slots 1, 3, 5, and 7 (and in slot 9 if standby disk is being added).
14. Use four screws to reattach the DA controller front panel.
15. Close both the front and rear doors.
16. Turn the PDA circuit breaker ON.

Factory-supplied disk canisters are shipped preformatted. If your drives are from another source, you can now format the disks, following the procedures in Chapter 4.

### 7.2 Upgrading to Sixteen Disk Drives

Upgrading the disk array from eight to sixteen disk drives involves plugging in the new disk canisters and formatting the disk drives.

The new disk canisters are factory-labeled and are installed in the following pattern:

Front Bank

8-disk bank:            1 3 5 7 ( )    (tenth slot not used)  
                             2 4 6 8 P

Follow these steps to upgrade your PDA to sixteen disk drives.

1. Open the PDA front and rear doors to access the disk chassis and the power/fan chassis.
2. Make sure the PDA circuit breaker is OFF.
3. Put on an antistatic wrist strap as a precautionary measure.
4. Remove the screws securing the blank panels covering the unused disk slots in the front chassis (front slot numbers 1 through 8 and P). Remove the blank panels and discard them.
5. The new disk canisters are factory-configured and labeled. If there is a need to alter the configuration, refer to Section 2.3.1 to set terminators and jumpers on each new disk to correspond to the bank number.

---

#### Caution

---

Be sure to slide the canister squarely into the guide rails. Guide rails can break if the canister is twisted to one side.

---

6. Align each new disk canister top and bottom edges in the nylon guide rails, and push the canister squarely into the disk chassis.
7. Tighten the two captive screws securing each disk canister to the chassis.
8. Label each new disk with the correct bank and disk numbers if they are not marked so.
9. Turn each disk canister switch ON (up).
10. Turn the PDA circuit breaker ON.
11. Close both the front and rear doors.

Factory-supplied disk canisters are shipped preformatted. If your drives are from another source, you can now format the disks, following the procedures in Chapter 4.



---

## Configuration Data

Appendix A contains descriptions of the READCONFIG and WRITECONFIG commands. It also contains configuration and data tables.

### A.1 The READCONFIG Command

The READCONFIG command returns disk array (DA) controller configuration data to the host. Use it to read system configuration data, disk type parameter data, a bank's Bad Block Table (BBT), and the system areas of individual disk drives.

The `sel` and `tab` parameters control which table is selected for readback:

- If `sel` is zero, the DA controller returns the internal DA controller tables.
- If `sel` is nonzero, the system displays the bank BBT or the system area of the disk drive selected with `sel`. In this case, the upper nibble of `sel` selects the bank number from 1 through 4; the lower nibble selects the disk number from 1 through 10.

Data disks are numbered 1 through 8, parity is 9, and standby is 10.

The parameters `offset` and `n` define the starting byte offset and number of bytes to read. Executing the READCONFIG command transmits `n` number of Extended Data Status Message words to the host. A value of 0 for `n` is interpreted as 256.

The System Configuration table is a set of global parameters reflecting the DA Controller current configuration. Some parameters are read/write, some are read-only.

The Disk Parameter tables are all the parameters necessary to characterize a particular type of disk drive. A parameter in the System Configuration table defines the number of valid Disk Parameter tables.

A bank's Bad Block Table consists of address positions of unrecoverable bad blocks during a FORMAT Bad Tracks Only (`func = 5`).

A disk system area consists of a home page, an RFT, and a GET. The home page is always system area page 0. The RFT spans system area pages 1 . . .  $n$ , where  $n$  is from 1 . . . 254. Its current value is stored in the home page. The GET spans system area pages  $(n + 1)$  . . . 255.

The READCONFIG command is followed by the NON-TRANSFER COMMAND SUCCESS status message and  $n$  number of EXTENDED DATA status messages.

Table A-1 illustrates how this is done. Values for `sel` and `tab` are shown in hexadecimal.

## A.1 The READCONFIG Command

**Table A-1 READCONFIG Command Status**

| sel | tab | Table Selected                                    |
|-----|-----|---|
| 00h | 00h | System Configuration table                        |
| 00h | 01h | Disk Type #1 Parameter table (read-only)          |
| 00h | 02h | Disk Type #2 Parameter table (read-only)          |
| ⋮   | ⋮   | ⋮   |
| 00h | FFh | Disk Type #255 Parameter table (read-only)        |
| 10h | 00h | Bank 1, Bad Block Table page 0 (read-only)        |
| ⋮   | ⋮   | ⋮   |
| 10h | FFh | Bank 1, Bad Block Table page 255 (read-only)      |
| 11h | 00h | Bank 1, Disk 1, System Area page 0 (read-only)    |
| 11h | 01h | Bank 1, Disk 1, System Area page 1 (read-only)    |
| ⋮   | ⋮   | ⋮   |
| 11h | FFh | Bank 1, Disk 1, System Area page 255 (read only)  |
| 12h | 00h | Bank 1, Disk 2, System Area page 0 (read-only)    |
| ⋮   | ⋮   | ⋮   |
| 1Ah | FFh | Bank 1, Disk 10, System Area page 255 (read-only) |
| 20h | 00h | Bank 2, Bad Block Table page 0 (read-only)        |
| ⋮   | ⋮   | ⋮   |
| 20h | FFh | Bank 2, Bad Block Table page 255 (read-only)      |
| 21h | 00h | Bank 2, Disk 1, System Area page 0 (read-only)    |
| 21h | 01h | Bank 2, Disk 1, System Area page 1 (read-only)    |
| ⋮   | ⋮   | ⋮   |
| 21h | FFh | Bank 2, Disk 1, System Area page 255 (read-only)  |
| 22h | 00h | Bank 2, Disk 2, System Area page 0 (read-only)    |
| ⋮   | ⋮   | ⋮   |
| 2Ah | FFh | Bank 2, Disk 10, System Area page 255 (read-only) |

## A.2 The WRITECONFIG Command

Use the WRITECONFIG command to select controller configuration parameters. The WRITECONFIG command allows user-modification of write controller parameters. Refer also to the READCONFIG command.

Attempting to write a read-only table element does NOT result in an error status; its value is simply not changed.

The WRITECONFIG command waits for any currently executing transfer command to complete before it starts to execute. Normally, changes in configuration data then take effect for subsequent commands. Exceptions are Data Channel Timeout, which requires a controller system reset, and Bank Configuration parameters, which require a MOUNTBANK command.

## A.3 System Configuration Table

---

### Note

---

The information in this section is provided only for completeness. The data in these tables is initialized at installation with the `dainit(8)` and `dabinit(8)` commands. It is never modified after installation.

---

The System Configuration Table stores global user-configurable parameters and operational variables. Use the `READCONFIG` and `WRITECONFIG` commands to access data in this table. The 256 bytes of this table are organized as shown in Table A-2.

**Table A-2 System Configuration Byte Organization**

| Offset        |                          |
|---------------|--------------------------|
| 00h . . . 1Fh | Controller Configuration |
| 20h . . . 3Fh | Bank 1 Control/Status    |
| 40h . . . 5Fh | Bank 2 Control/Status    |
| 60h . . . 7Fh | Reserved                 |
| 80h . . . 9Fh | Reserved                 |
| A0h . . . BFh | Reserved                 |
| C0h . . . DFh | Reserved                 |
| E0h . . . FFh | Reserved                 |

Read/write parameters are stored in the DA controller EEPROM.

## A.3 System Configuration Table

### A.3.1 Controller Configuration

The 32-byte Controller Configuration block is formatted as shown in Table A-3 and Table A-4.

**Table A-3 Controller Configuration Block Byte Organization**

| Byte Offset      | RW Type | Name                  | Value                     |
|------------------|---------|-----------------------|---------------------------|
| 00h              | RW      | Control               | See below                 |
| 01h              | RW      | Retries               | 00h . . . FFh             |
| 02h              | RW      | Data Chan Timeout     | 00h . . . FFh (byte 1)    |
| 03h              | RW      | Data Chan Timeout     | 00h . . . FFh (byte 0)    |
| 04h              | RW      | Status Filter         | 00h . . . FFh (byte 1)    |
| 05h              | RW      | Status Filter         | 00h . . . FFh (byte 0)    |
| 06h . . .<br>07h | RW      | Reserved              | 00h                       |
| 08h              | RW      | Monitor Enable        | See bitmap below (byte 1) |
| 09h              | RW      | Monitor Enable        | See bitmap below (byte 0) |
| 0Ah              | RW      | Drive Timeout/Spin-up | 00h . . . FFh (byte 1)    |
| 0Bh              | RW      | Drive Timeout/Spin-up | 00h . . . FFh (byte 0)    |
| 0Ch . . .<br>0Fh | RW      | Reserved              | 00h                       |
| 10h              | RO      | System                | 02h                       |
| 11h              | RO      | Rev                   | 01h . . . FFh             |
| 12h              | RO      | # Types               | 01h . . . FFh             |
| 13h              | RO      | Reserved              | 00h                       |
| 14h              | RO      | Buffer Presence       | 00h . . . 03h (byte 1)    |
| 15h              | RO      | Buffer Presence       | 00h . . . FFh (byte 0)    |
| 16h . . .<br>1Fh | RO      | Reserved              | 00h                       |



## A.3 System Configuration Table

**Table A-4 Controller Configuration Block Bit Description**

| Name    | Description  |
|---------|--|
| Control | <p>Byte includes the following global control bits:</p> <ul style="list-style-type: none"><li>• Bit 0 — Disk Addressing Mode<ul style="list-style-type: none"><li>– 0 = Physical</li><li>– 1 = Logical</li></ul></li><li>• Bit 1 — Reserved</li><li>• Bit 3:2 — Disk Error Log Select Bit<ul style="list-style-type: none"><li>– 3=0, 2=0 — Log to GET only</li><li>– 3=0, 2=1 — Log none</li><li>– 3=1, 2=0 — Log to GET and host</li><li>– 3=1, 2=1 — Log to host only</li></ul></li><li>• Bit 4 — Standby Mode<ul style="list-style-type: none"><li>– 0 = Mode 1: each bank has its own standby drive</li><li>– 1 = Mode 2: standby drive on bank 1 serves all banks (all drives must be same DRV ID)</li></ul></li><li>• Bit 5 — Format &amp; Scrub Policy<ul style="list-style-type: none"><li>– 0 = track interleave</li><li>– 1 = cylinder interleave</li></ul></li><li>• Bit 6 — Reconstruction Policy<ul style="list-style-type: none"><li>– 0 = track interleave</li><li>– 1 = cylinder interleave</li></ul></li><li>• Bit 7 — Command Reordering (future)<ul style="list-style-type: none"><li>– 0 = Command reordering disabled</li><li>– 1 = Command reordering enabled</li></ul></li></ul> |

## A.3 System Configuration Table

There are two types of selectable Disk Addressing modes:

- **Physical** — The host sends a cylinder, head, and sector number for read/write type commands. The actual physical cylinder addressed is offset by the number of reserved cylinders (Disk Parameter Table). Reads the Drive ID (Bank Control/Status) to get the following parameters from the Disk Parameter Tables.

Starting Cylinder No. = 0  
Last Cylinder No. = (No. of cylinders) - (No. of reserved cylinders) - 1  
  
Starting Head No. = 0  
Last Head No. = (No. of data heads) - 1  
Starting Sector No. = 0  
Last Sector No. = (No. of sectors/track) - (No. of skip sectors) - 1

- **Logical** — The host sends a logical block address for read/write type commands. Reads the Count parameter in the Bank Control/Status to get the following parameters:

Starting Logical Block = 0  
Last Logical Block = (Count) - 1

The Disk Error Log Select, along with the Log Error Type (hard/soft) control bit, allows you to log selected media errors to the host, the GET, or both. The actual errors logged can be hard errors only or hard and soft errors, depending on how bit 6 is set in the bank-mode byte. Only the SCRUB command does not log errors to the host; it always logs errors to the GET.

You can do FORMAT and RECONSTRUCT operations while the controller is on line. Host commands, if pending, are interleaved with a FORMAT or RECONSTRUCT operation on a track or cylinder basis, as shown in Table A-5.

**Table A–5 Disk Error Log Select Bit Descriptions**

| Byte # | Name                 | Description  |
|--------|----------------------|--|
| 1      | Retries              | Byte defining the number of attempts to reread an error detected by CRC or ECC. A value of zero disables retries.<br>Bit 7:0 Number of retries   |
| 2      | Monitor Enable       | <p>Monitor Status Enable word defines which text messages to enable out the Com Port for monitor display:</p> <ul style="list-style-type: none"> <li>• Bit 0 — Command/Status Monitor               <ul style="list-style-type: none"> <li>– 0 = off</li> <li>– 1 = display command/status</li> </ul> </li> <li>• Bit 1 — Error Recovery Monitor               <ul style="list-style-type: none"> <li>– 0 = off</li> <li>– 1 = display error recovery attempts (parity replacement, retries, ECC correction)</li> </ul> </li> <li>• Bit 2 — Error Logging               <ul style="list-style-type: none"> <li>– 0 = off</li> <li>– 1 = display error logging</li> </ul> </li> <li>• Bit 3 — Reserved</li> <li>• Bit 4 — Formatting/Reconstruction Progress               <ul style="list-style-type: none"> <li>– 0 = off</li> <li>– 1 = display formatting/reconstruct cylinder progress</li> </ul> </li> <li>• Bit 15:5 — Reserved</li> </ul> |
| 3      | Data Channel Timeout | <p>Word used to set the Timeout period for the data channel. Each increment of the Timeout value equals 1 second. To invoke a new Timeout value, initiate a controller system reset. When the DA controller executes a Transfer Command, the following sequence is initiated:</p> <ol style="list-style-type: none"> <li>1. Timer is set to the Timeout period.</li> <li>2. If the timer expires before the transfer completes, go to 3, else done.</li> <li>3. If any data activity has occurred, go to 1, else terminate the Transfer Command (Transfer Command Error).</li> </ol> <p>A Timeout value of 0 disables the Timeout function:</p> <ul style="list-style-type: none"> <li>• Bit 15:0 — Data Channel Timeout               <ul style="list-style-type: none"> <li>– 0 = timeout disabled</li> <li>– 1 . . . 65,535 = timeout period</li> </ul> </li> </ul>   |

(continued on next page)

## A.3 System Configuration Table

**Table A-5 (Cont.) Disk Error Log Select Bit Descriptions**

| Byte # | Name          | Description   |
|--------|---------------|---|
| 4      | Status Filter | <p>Word defining which status message class types generate an interrupt to the host by having bit 15 set in each word of the status message. In the hardware, the host interface receives all status messages, but only those with bit 15 set are decoded by the host interface to generate an interrupt:</p> <ul style="list-style-type: none"> <li>• Bit 0 — Transfer Command Intermediate, end of message</li> <li>• Bit 1 — Reserved</li> <li>• Bit 2 — Transfer Command Success, end of message</li> <li>• Bit 3 — Transfer Command Success, more status follows</li> <li>• Bit 4 — Reserved</li> <li>• Bit 5 — Transfer Command Error, Extended Data follows</li> <li>• Bit 6 — Reserved</li> <li>• Bit 7 — Reserved</li> <li>• Bit 8 — Extended Data, end of message</li> <li>• Bit 9 — Extended Data, Extended Data follow</li> <li>• Bit 10 — Nontransfer Command Success, end of message</li> <li>• Bit 11 — Nontransfer Command Success, more status follows</li> <li>• Bit 12 — Reserved</li> <li>• Bit 13 — Nontransfer Command Error, Extended Data follows</li> <li>• Bit 14 — Controller Status, end of message</li> <li>• Bit 15 — Reserved               <ul style="list-style-type: none"> <li>– 0 = set (1) Bit 15 in return status message, interrupt host</li> <li>– 1 = reset (0) Bit 15 in return status message, inhibit host interrupt</li> </ul> </li> </ul> <p>Examples of host interrupt status:</p> <ul style="list-style-type: none"> <li>• Value 0000h: all status message class types interrupt the host.</li> <li>• Value 0004h: all status message class types except Transfer Command Success types interrupt the host. For this case, the host "DMA Done" (interrupt without Transfer Command Error) signals successful completion.</li> </ul> |

(continued on next page)

## A.3 System Configuration Table

Table A-5 (Cont.) Disk Error Log Select Bit Descriptions

| Byte # | Name                  | Description   |
|--------|-----------------------|---|
| 5      | Drive Timeout Spin-up | Word used for two functions: <ul style="list-style-type: none"><li>• Use the lower byte (byte 0) to set the Timeout period for the Power Up Drive Ready period (drive spin-up time). Each increment equals one second. If the drives are not READY within the Timeout period after power is applied, the controller recognizes the drives as off line.</li><li>• Use the upper byte (byte 1), bits 8-11, to set the Motor Spin-Up Delay interval for ESDI drives (ignored for SMD drives). Each increment equals one second. If Bank Mount sequence is run and the drives are NOT spun-up, the controller issues the spin-up command to each drive delayed by the Motor Spin-Up Delay interval. Bit 15 is used to enable the automatic spin-down for only ESDI drives. If a DSBL field is 0 and is set to nonzero followed by a MOUNTBANK command, the drive identified by the DSBL field is issued a spin-down command. Any MOUNTBANK thereafter will issue a spin-up command.</li></ul> |
| 6      | Rev                   | Byte defining the current revision of the controller firmware.  |
| 7      | # Types               | Byte indicating the number of drive types supported by the current revision of firmware.  |
| 8 & 9  | Buffer Presence       | Bytes indicating which buffer cards are present: <ul style="list-style-type: none"><li>• Bit 0 — Buffer 1 present</li><li>• Bit 1 — Buffer 2 present</li><li>• Bit 2 — Buffer 3 present</li><li>• Bit 3 — Buffer 4 present</li><li>• Bit 4 — Buffer 5 present</li><li>• Bit 5 — Buffer 6 present</li><li>• Bit 6 — Buffer 7 present</li><li>• Bit 7 — Buffer 8 present</li><li>• Bit 8 — Buffer 9 present</li><li>• Bit 9 — Buffer 10 present</li><li>• Bit 15:10 — Reserved</li></ul>  |

## A.3 System Configuration Table

### A.3.2 Bank Control/Status Configuration

Each 32-byte Bank Control/Status block is formatted as shown in Table A-6 and Table A-7.

**Table A-6 32-Byte Bank Control/Status Block**

| Bank 1 | Bank 2 | Bank 3 | Bank 4 | Description (Access, Name, Address)               |
|--------|--------|--------|--------|---|
| 20h    | 40h    | 60h    | 80h    | RW — Drive ID; 00h . . . 0Fh                      |
| 21h    | 41h    | 61h    | 81h    | RW — Sub Drive ID; 00h . . . FFh                  |
| 22h    | 42h    | 62h    | 82h    | RW — Mode; See below                              |
| 23h    | 43h    | 63h    | 83h    | RW — Reserved; 00h                                |
| 24h    | 44h    | 64h    | 84h    | RW — SUBS (Substitute);00h . . . 09h              |
| 25h    | 45h    | 65h    | 85h    | RW — DSBL (Disable);00h . . . 0Ah                 |
| 26h    | 46h    | 66h    | 86h    | RW — Reserved                                     |
| 27h    | 47h    | 67h    | 87h    | RW — Reserved                                     |
| 28h    | 48h    | 68h    | 88h    | RO — Bank Status; See below (byte 1)              |
| 29h    | 49h    | 69h    | 89h    | RO — Bank Status; See below (byte 0)              |
| 2Ah    | 4Ah    | 6Ah    | 8Ah    | RO — Reserved                                     |
| 2Bh    | 4Bh    | 6Bh    | 8Bh    | RO — Reserved                                     |
| 2Ch    | 4Ch    | 6Ch    | 8Ch    | RO — Count; 00h . . . FFh (byte 3)                |
| 2Dh    | 4Dh    | 6Dh    | 8Dh    | RO — Count; 00h . . . FFh (byte 2)                |
| 2Eh    | 4Eh    | 6Eh    | 8Eh    | RO — Count; 00h . . . FFh (byte 1)                |
| 2Fh    | 4Fh    | 6Fh    | 8Fh    | RO — Count; 00h . . . FFh (byte 0)                |
| 30h    | 50h    | 70h    | 90h    | RO — Drive 1 Status; See below                    |
| 31h    | 51h    | 71h    | 91h    | RO — Drive 2 Status; See below                    |
| 32h    | 52h    | 72h    | 92h    | RO — Drive 3 Status; See below                    |
| 33h    | 53h    | 73h    | 93h    | RO — Drive 4 Status; See below                    |
| 34h    | 54h    | 74h    | 94h    | RO — Drive 5 Status; See below                    |
| 35h    | 55h    | 75h    | 95h    | RO — Drive 6 Status; See below                    |
| 36h    | 56h    | 76h    | 96h    | RO — Drive 7 Status; See below                    |
| 37h    | 57h    | 77h    | 97h    | RO — Drive 8 Status; See below                    |
| 38h    | 58h    | 78h    | 98h    | RO — Drive 9 Status; See below                    |
| 39h    | 59h    | 79h    | 99h    | RO — Drive 10 Status; See below                   |
| 3Ah    | 5Ah    | 7Ah    | 9Ah    | RO — Reconstruct Disk Address; See below (byte 3) |
| 3Bh    | 5Bh    | 7Bh    | 9Bh    | RO — Reconstruct Disk Address; See below (byte 2) |
| 3Ch    | 5Ch    | 7Ch    | 9Ch    | RO — Reconstruct Disk Address; See below (byte 1) |
| 3Dh    | 5Dh    | 7Dh    | 9Dh    | RO — Reconstruct Disk Address; See below (byte 0) |
| 3Eh    | 5Eh    | 7Eh    | 9Eh    | RO — Reserved                                     |
| 3Fh    | 5Fh    | 7Fh    | 9Fh    | RO — Reserved                                     |

## A.3 System Configuration Table

**Table A-7 #2 Byte Bank Control/Status Block Description**

| Name        | Description  |
|-------------|--|
| Drive ID    | <p>Byte defining the disk drive type of drives on this bank. The Drive ID is a unique number assigned to a specific drive model that has been qualified for use with the controller.</p> <ul style="list-style-type: none"> <li>• Bit 7:0 — Drive ID               <ul style="list-style-type: none"> <li>– 0 = bank disabled</li> <li>– 1 . . . 255 = drive type</li> </ul> </li> </ul>   |
| Subdrive ID | <p>Byte defining the subtype of Drive ID</p> <ul style="list-style-type: none"> <li>• Bit 3:0 — Subtype               <ul style="list-style-type: none"> <li>– 0 = 1024 data byte sector size: 4096 Block size (4-drive bank) or 8192 Block size (8-drive bank)</li> <li>– 1 . . . 15 = See Strategy 2 Drive ID Codes, SID002, for supported drive values.</li> </ul> </li> <li>• Bit 7:4 reserved, readback 0</li> </ul>  |
| Mode        | <p>Byte defining the basic bank configuration:</p> <ul style="list-style-type: none"> <li>• Bit 1:0 — Reserved, readback 0</li> <li>• Bit 2 — 4/8 data drives               <ul style="list-style-type: none"> <li>– 0 = 4-data drive bank (data drives 2, 4, 6, 8 valid only)</li> <li>– 1 = 8-data drive bank (all data drives 1 . . . 8 valid)</li> </ul> </li> <li>• Bit 3 — Reserved</li> <li>• Bit 4 — If set, use alternate system area</li> <li>• Bit 5 — If set, inhibit drive array position check</li> <li>• Bit 6 — Error Logging Policy               <ul style="list-style-type: none"> <li>– 0 = Log hard errors only in the GET or host</li> <li>– 1 = Log hard and soft errors in the GET or host</li> </ul> </li> <li>• Bit 7 — Error Recovery Policy               <ul style="list-style-type: none"> <li>– 0 = PRE Error Recovery Policy Sequence:                   <ul style="list-style-type: none"> <li>– On-the-fly parity replacement</li> <li>– Retries</li> <li>– ECC correction</li> </ul> </li> <li>– 1 = REP Error Recovery Policy Sequence:                   <ul style="list-style-type: none"> <li>– Retries</li> <li>– ECC correction</li> <li>– On-the-fly parity replacement</li> </ul> </li> </ul> </li> </ul> |

## A.3 System Configuration Table

Once a drive bank is written in either a 4- or 8-data drive configuration, do not change the configuration.

Error Checking and Correcting (ECC) code is appended to the data field of every sector. Errors can be detected, and, depending on the type of error, can also be corrected. Table A-8 shows the bit descriptions and Table A-9 shows the status conditions.

In addition to the ECC code error detection, the DA controller performs two types of data parity checking:

- All data paths carry an additional parity bit used to detect any word errors between the host and the drives.
- Temporal parity fault detection is used as an additional cross-drive check of all data drives against the parity drive on read operations. If a temporal parity fault is detected, the transfer is terminated. The temporal parity fault detection is used only if all data drives and the parity drive are operational (no detected drive errors). Before reading new drives, write the entire drive bank to allow successful temporal parity fault detection.

Cyclic Redundancy Checking (CRC) code is appended to the header field of every sector. The code detects errors in the header field.

A hard error is a media error that is uncorrectable.

A soft error is a media error that is corrected by either retries or ECC correction.

For disk read operations:

- PRE Error Recovery Policy Sequence
  1. If a single drive error is detected, on-the-fly parity replacement is invoked, and the error is not logged.
  2. If multiple drive errors are detected, parity replacement is invoked on the first-encountered drive error, and retries are attempted on any additional drives in error.
  3. If the retry attempts fail, ECC correction (data field only) is attempted.
  4. If ECC correction fails, the transfer is terminated, and the errors are automatically logged in the GET.
- REP Error Recovery Policy Sequence
  1. If drive errors are detected, retries are attempted.
  2. If retry attempts fail, ECC correction is attempted.
  3. If ECC correction fails, the errors are automatically logged in the GET and on-the-fly parity replacement is invoked if a single drive is in error. If multiple drive errors still exist, the transfer is ended.



## A.3 System Configuration Table

**Table A–8 Error Checking and Correcting Bit Descriptions**

| Name        | Description  |
|-------------|--|
| SUBS        | <p>The following is a breakdown of the SUBS field:</p> <ul style="list-style-type: none"> <li>• Bit 3:0 — drive # being replaced               <ul style="list-style-type: none"> <li>– 0 = none</li> <li>– 1 . . . 9 = drive 1 . . . 9</li> </ul> </li> <li>• Bit 7:4 — Reserved</li> </ul>   |
| DSBL        | <p>Field defining which drive is inoperable. If nonzero, do not write this byte. After writing this byte, run a MOUNTBANK command to reestablish the active bank. This byte is automatically reset to 0s after a new replacement is reconstructed.</p> <ul style="list-style-type: none"> <li>• Bit 3:0 — Drive # being replaced               <ul style="list-style-type: none"> <li>– 0 = none</li> <li>– 1h . . . Ah = drive 1h . . . Ah</li> </ul> </li> <li>• Bit 7:4 — Reserved</li> </ul>   |
| Bank Status | <p>Word reflecting the current active status of the bank. The bank-mounted bits (bit 15:12) indicates whether the bank is MOUNTED. The drive-mount bits (bits 9:0) indicate the mount status of each drive since the last Bank Mount Sequence occurred. Normal host read/write is permitted only if bit 15 is set (bank-mounted). If any one or more drives required for a normal host read/write is write-protected, the bank as a whole is protected from write Transfer commands. However, formatting is still possible on any drives not write-protected at the drive level.</p> <ul style="list-style-type: none"> <li>• Bit 9:0 — Drives mounted               <ul style="list-style-type: none"> <li>– 0 = not mounted</li> <li>– 1 = mounted (System Area correctly read)</li> </ul> </li> <li>• Bit 11:10 — Reserved</li> <li>• Bit 15:12 — Bank mounted               <ul style="list-style-type: none"> <li>– 15=0, 14=0, 13=0, 12=0 — Bank drive ID zero or invalid</li> <li>– 15=0, 14=0, 13=0, 12=1 — Not enough mounted drives for valid configuration</li> <li>– 15=0, 14=0, 13=1, 12=0 — Too many drive flaws</li> <li>– 15=1, 14=0, 13=0, 12=0 — Bank mounted (read/write)</li> <li>– 15=1, 14=0, 13=0, 12=1 — Bank mounted (read only)</li> </ul> </li> </ul> |
| Count       | <p>Word specifying the bank size in number blocks. Maximum number of blocks in bank.</p>   |

(continued on next page)

## A.3 System Configuration Table

**Table A–8 (Cont.) Error Checking and Correcting Bit Descriptions**

| Name                    | Description  |
|-------------------------|--|
| Drive 1 . . . 10 Status | <p>Byte reflecting the status of the disk at initialization. Initialization occurs at power on, controller system reset, or MOUNTBANK command.</p> <ul style="list-style-type: none"> <li>• Bit 3:0 — Current sequence (defined below)</li> <li>• Bit 5:4 — Current state: <ul style="list-style-type: none"> <li>– 0 = initialization<br/>In states 0 and 1, read/write/format is not possible with this drive.</li> <li>– 1 = error</li> <li>– 2 = warning<br/>In state 2, only format is possible.</li> <li>– 3 = success<br/>In state 3, read/write/format possible.</li> </ul> </li> <li>• Bit 7:6 — Reserved (readback 0)</li> </ul> |

**Table A–9 Bit Status and Conditions**

|                      | State       | Sequence    |  |
|----------------------|-------------|-------------|--|
| <b>Status</b>        | <b>7654</b> | <b>3210</b> | <b>Condition</b>   |
| Initialization Error | 0000        | 0000        | Buffer fault or not present  |
|                      | 0001        | 0000        | Drive not assigned   |
|                      | 0001        | 0001        | Drive not selected (drive not present or unit select switch wrong)         |
|                      | 0001        | 0010        | Drive is busy (reserved by other channel)                                  |
|                      | 0001        | 0011        | Drive does not become READY (check drive)                                  |
|                      | 0001        | 0100        | Drive fault does not clear (check drive)                                   |
|                      | 0001        | 0101        | Drive seek error does not clear or on-cylinder not set (check drive)       |
|                      | 0001        | 0110        | Drive Read_Clock not valid (check drive)                                   |
|                      | 0010        | 0111        | Error reading System Area (drive not formatted)                            |
|                      | 0010        | 1000        | System Area not valid (drive must be reformatted)                          |
| Warning (ONLINE)     | 0010        | 1001        | System Area bank/drive address position doesn't match (drive cables wrong) |
| Mount Success        | 0011        | 0000        | Drive ONLINE/MOUNTED   |
| Mount Success        | 0011        | 0001        | Drive ONLINE/MOUNTED (write-protected)                                     |

## A.3 System Configuration Table

The "Drive Not Assigned" condition occurs for the standby drive only when it is configured in Standby Mode 2 (see control byte of controller configuration) and it is not assigned to this bank.

The Reconstruct Disk Address is initialized to all 1's during a Bank Mount sequence for the associated bank. Any reconstruction operation to the selected bank starts at the Reconstruct Disk Address value plus 1. The Reconstruct Disk Address value is interpreted as follows:

- All 1's — Initial state
- Logical Addressing — Host logical address block value where the last reconstruct was attempted.
- Physical Addressing — Host physical address block value where the last reconstruct was attempted.

During a reconstruct operation, any unrecoverable media error can cause the operation to abort immediately. Log the block address of the failure and continue by resending only the RECONSTRUCT command.

### A.3.3 Disk Parameter Table Overview

The Disk Parameter Tables store the supported disk parameters for each supported Drive ID. Use the READCONFIG command to read the data in these tables. The 256 bytes of each table are organized in 16-byte groups (Sub-DrvID) as shown in Table A-10.

**Table A-10 Disk Parameter Table Byte Organization**

| Sub-DrvID | Byte Offset   | RW Type | Name                        | Value                  |
|-----------|---------------|---------|-----------------------------|------------------------|
| 0         | 00h           | RO      | Drive type                  | 00h . . . FFh          |
| 0         | 01h           | RO      | Flag bit                    | 00h . . . FFh          |
| 0         | 02h           | RO      | Number of cylinder          | 00h . . . FFh (byte 1) |
| 0         | 03h           | RO      | Number of cylinder          | 00h . . . FFh (byte 0) |
| 0         | 04h           | RO      | Number of data head         | 00h . . . FFh          |
| 0         | 05h           | RO      | Number of sectors/track     | 00h . . . FFh          |
| 0         | 06h           | RO      | Number of skip sector       | 00h . . . FFh          |
| 0         | 07h           | RO      | Number of reserved cylinder | 00h . . . FFh          |
| 0         | 08h           | RO      | Unformatted bytes/track     | 00h . . . FFh (byte 1) |
| 0         | 09h           | RO      | Unformatted bytes/track     | 00h . . . FFh (byte 0) |
| 0         | 0Ah           | RO      | Unformatted bytes/sector    | 00h . . . FFh (byte 1) |
| 0         | 0Bh           | RO      | Unformatted bytes/sector    | 00h . . . FFh (byte 0) |
| 0         | 0Ch           | RO      | Number of bytes/PLO sync    | 00h . . . FFh          |
| 0         | 0Dh . . . 0Fh | RO      | Reserved                    |                        |
| ⋮         | ⋮             | ⋮       | ⋮                           | ⋮                      |
| 5         | 50h           | RO      | Drive type                  | 00h . . . FFh          |
| 5         | 51h           | RO      | Flag bits                   | 00h . . . FFh          |

(continued on next page)

## A.3 System Configuration Table

**Table A–10 (Cont.) Disk Parameter Table Byte Organization**

| Sub-Drvid | Byte Offset   | RW Type | Name                         | Value                  |
|-----------|---------------|---------|------------------------------|------------------------|
| 5         | 52h           | RO      | Number of cylinders          | 00h . . . FFh (byte 1) |
| 5         | 53h           | RO      | Number of cylinders          | 00h . . . FFh (byte 0) |
| 5         | 54h           | RO      | Number of data heads         | 00h . . . FFh          |
| 5         | 55h           | RO      | Number of sectors/track      | 00h . . . FFh          |
| 5         | 56h           | RO      | Number of skip sectors       | 00h . . . FFh          |
| 5         | 57h           | RO      | Number of reserved cylinders | 00h . . . FFh          |
| 5         | 58h           | RO      | Unformatted bytes/track      | 00h . . . FFh (byte 1) |
| 5         | 59h           | RO      | Unformatted bytes/track      | 00h . . . FFh (byte 0) |
| 5         | 5Ah           | RO      | Unformatted bytes/sector     | 00h . . . FFh (byte 1) |
| 5         | 5Bh           | RO      | Unformatted bytes/sector     | 00h . . . FFh (byte 0) |
| 5         | 5Ch           | RO      | Number of bytes/PLO sync     | 00h . . . FFh          |
| 5         | 5Dh . . . 5Fh | RO      | Reserved                     |                        |
| 6         | 60h           | RO      | Reserved                     | 00h                    |
| ⋮         | ⋮             | ⋮       | ⋮                            | ⋮                      |
| F         | FFh           | RO      | Reserved                     | 00h                    |

The upper 4 bits of the Byte Offset reference the Sub-Drvid code.

Table A–11 lists the drive type and flag bit definitions.

**Table A–11 Drive Type and Flag Bit Definitions**

| Name       | Description  |
|------------|--|
| Drive Type | <p>The following is a breakdown of the Drive Type field:</p> <ul style="list-style-type: none"> <li>• Bit 7:0 — Drive type <ul style="list-style-type: none"> <li>– 0 = ESDI, 256-byte Media Defect List sector size</li> <li>– 1 = ESDI, 512-byte Media Defect List sector size</li> </ul> </li> <li>• 2 . . . 3 — Reserved</li> <li>• 4 — SMD</li> <li>• 5 . . . 255 — Reserved</li> </ul> |

(continued on next page)

## A.3 System Configuration Table

**Table A–11 (Cont.) Drive Type and Flag Bit Definitions**

| Name      | Description   |
|-----------|---|
| Flag Bits | The following is a breakdown of the Flag Bit field: <ul style="list-style-type: none"> <li>• Bit 0 — Data strobe offset capability present if set</li> <li>• Bit 1 — Track offset capability present if set</li> <li>• Bit 2 — Programmable sector size supported if set (ESDI only)</li> <li>• Bit 3 — Reserved</li> <li>• Bit 4 — Reserved</li> <li>• Bit 5 — Reserved</li> <li>• Bit 6 — Reserved</li> <li>• Bit 7 — Reserved</li> </ul> |

By selecting Drive ID 1, you can program the Disk Parameter Table (using WRITECONFIG command) with the parameters listed in Table A–12.

**Table A–12 Values for Disk Parameter Table**

| Sub-Drvid | Byte Offset | RW Type | Name                         | Value                |
|-----------|-------------|---------|------------------------------|----------------------|
| 0         | 00h         | RW      | Drive Type                   | 00h ... FFh          |
| 0         | 01h         | RW      | Flag bits                    | 00h ... FFh          |
| 0         | 02h         | RW      | Number of cylinders          | 00h ... FFh (byte 1) |
| 0         | 03h         | RW      | Number of cylinders          | 00h ... FFh (byte 0) |
| 0         | 04h         | RW      | Number of data heads         | 00h ... FFh          |
| 0         | 05h         | RW      | Number of sectors/track      | 00h ... FFh          |
| 0         | 06h         | RW      | Number of skip sectors       | 00h ... FFh          |
| 0         | 07h         | RW      | Number of reserved cylinders | 00h ... FFh          |
| 0         | 08h         | RW      | Unformatted bytes/track      | 00h ... FFh (byte 1) |
| 0         | 09h         | RW      | Unformatted bytes/track      | 00h ... FFh (byte 0) |
| 0         | 0Ah         | RW      | Unformatted bytes/sector     | 00h ... FFh (byte 1) |
| 0         | 0Bh         | RW      | Unformatted bytes/sector     | 00h ... FFh (byte 0) |
| 0         | 0Ch         | RW      | Number of bytes/PLO sync     | 00h ... FFh          |
| 0         | 0Dh         | RW      | Reserved                     |                      |
| ⋮         | ⋮           | ⋮       | ⋮                            | ⋮                    |
| 3         | 30h         | RW      | Drive Type                   | 00h ... FFh          |
| 3         | 31h         | RW      | Flag bits                    | 00h ... FFh          |
| 3         | 32h         | RW      | Number of cylinders          | 00h ... FFh (byte 1) |
| 3         | 33h         | RW      | Number of cylinders          | 00h ... FFh (byte 0) |

(continued on next page)

## A.3 System Configuration Table

**Table A–12 (Cont.) Values for Disk Parameter Table**

| Sub-DrvID | Byte Offset | RW Type | Name                         | Value                  |
|-----------|-------------|---------|------------------------------|------------------------|
| 3         | 34h         | RW      | Number of data heads         | 00h . . . FFh          |
| 3         | 35h         | RW      | Number of sectors/track      | 00h . . . FFh          |
| 3         | 36h         | RW      | Number of skip sectors       | 00h . . . FFh          |
| 3         | 37h         | RW      | Number of reserved cylinders | 00h . . . FFh          |
| 3         | 38h         | RW      | Unformatted bytes/track      | 00h . . . FFh (byte 1) |
| 3         | 39h         | RW      | Unformatted bytes/track      | 00h . . . FFh (byte 0) |
| 3         | 3Ah         | RW      | Unformatted bytes/sector     | 00h . . . FFh (byte 1) |
| 3         | 3Bh         | RW      | Unformatted bytes/sector     | 00h . . . FFh (byte 0) |
| 3         | 3Ch         | RW      | Number of bytes/PLO sync     | 00h . . . FFh          |
| 3         | 3Dh         | RW      | Reserved                     |                        |
| 4         | 40h         | RO      | Reserved                     |                        |
| ⋮         | ⋮           | ⋮       | ⋮                            | ⋮                      |
| F         | FFh         | RO      | Reserved                     |                        |

The upper 4 bits of the Byte Offset reference the Sub Drive ID code.

### A.3.4 Bad Block Table

Entries in the Bad Block Table (BBT) are in ascending sorted order. The 256 bytes of a BBT page are organized as shown in Table A–13.

**Table A–13 Bad Block Table Byte Organization**

| Byte Offset | RW Type | Name     | Value                  |
|-------------|---------|----------|------------------------|
| 00h         | RO      | Cylinder | 00h . . . 0Fh (byte 1) |
| 01h         | RO      | Cylinder | 00h . . . FFh (byte 0) |
| 02h         | RO      | Head     | 00h . . . FFh          |
| 03h         | RO      | Sector   | 00h . . . FFh          |
| 04h         | RO      | Block    | 00h . . . FFh (byte 3) |
| 05h         | RO      | Block    | 00h . . . FFh (byte 2) |
| 06h         | RO      | Block    | 00h . . . FFh (byte 1) |
| 07h         | RO      | Block    | 00h . . . FFh (byte 0) |
| 08h         | RO      | Cylinder | 00h . . . 0Fh (byte 1) |
| 09h         | RO      | Cylinder | 00h . . . FFh (byte 0) |
| 0Ah         | RO      | Head     | 00h . . . FFh          |
| 0Bh         | RO      | Sector   | 00h . . . FFh          |
| 0Ch         | RO      | Block    | 00h . . . FFh (byte 3) |
| 0Dh         | RO      | Block    | 00h . . . FFh (byte 2) |
| 0Eh         | RO      | Block    | 00h . . . FFh (byte 1) |

(continued on next page)

## A.3 System Configuration Table

**Table A–13 (Cont.) Bad Block Table Byte Organization**

| Byte Offset | RW Type | Name     | Value                  |
|-------------|---------|----------|------------------------|
| 0Fh         | RO      | Block    | 00h . . . FFh (byte 0) |
| ⋮           | ⋮       | ⋮        | ⋮                      |
| F8h         | RO      | Cylinder | 00h . . . 0Fh (byte 1) |
| F9h         | RO      | Cylinder | 00h . . . FFh (byte 0) |
| FAh         | RO      | Head     | 00h . . . FFh          |
| FBh         | RO      | Sector   | 00h . . . FFh          |
| FCh         | RO      | Block    | 00h . . . FFh (byte 3) |
| FDh         | RO      | Block    | 00h . . . FFh (byte 2) |
| FEh         | RO      | Block    | 00h . . . FFh (byte 1) |
| FFh         | RO      | Block    | 00h . . . FFh (byte 0) |

Eight bytes of FFh follow the last valid flaw address. The maximum number of bad block entries will not exceed the number of sectors per track.

Each bad block is stored twice in BBT:

- First 4 bytes = host physical block address
- Next 4 bytes = host logical block address

The BBT is cleared of a flaw each time a Reformat Flawed Tracks Only operation (FORMAT func = 5 command) is executed, and the flaw is mapped out. At the end of a Non-Transfer Command Success (from FORMAT func = 5 or a system reset), the BBT is cleared.

### A.3.5 Bank X, Disk X, System Area Configuration

The Bank X, Disk X, system area consists of a home page, an RFT, and a GET. The home page is always system area page 0. The RFT spans system area pages 1 . . .  $n$ , where  $n$  is from 1 . . . 254. Its current value is stored in the home page. The GET spans system area pages ( $n+1$ ) . . . 255. Use the ReadConfig command to read the data in these tables. The tables are organized as follows:

```

Tab
-----
00h           Home Page
01h . . . N   RFT
N+1 . . . FFh GET

```

## A.3 System Configuration Table

### A.3.6 System Area Home Page

The 256 bytes of the home page are organized as shown in Table A-14; byte group descriptions are given in Table A-15.

**Table A-14 Home Page Byte Descriptions**

| Byte Offset   | RW Type | Name             | Value         |
|---------------|---------|------------------|---------------|
| 00h           | RO      | System           | 02h           |
| 01h           | RO      | Rev              | 00h . . . FFh |
| 02h           | RO      | Bank Number      | 01h . . . 04h |
| 03h           | RO      | Drive number     | 01h . . . 0Ah |
| 04h           | RO      | Drive ID         | 00h . . . FFh |
| 05h           | RO      | Sub Drive ID     | 00h . . . 0Fh |
| 06h . . . 0Fh | RO      | Reserved         | 00h           |
| 10h           | RO      | Reserved         | 00h           |
| 11h           | RO      | RFT Span         | 01h . . . FFh |
| 12h           | RO      | Reserved         | 00h . . . FFh |
| 13h           | RO      | Alternate Tracks | 00h . . . FFh |
| 14h . . . FFh | RO      | Reserved         | 00h           |

**Table A-15 Home Page Byte Group Descriptions**

| Name            | Description   |
|-----------------|---|
| System          | System model used when the drive was formatted        |
| Rev             | EPROM revision number when the disk was formatted     |
| Bank Number     | Bank position when the drive was formatted            |
| Drive Number    | Drive position (1 to 10) when the drive was formatted |
| Drive ID        | Drive type specified when the drive was formatted     |
| Sub Drive ID    | Subdrive type specified when the drive was formatted  |
| RFT Span        | Number of pages of the RFT                            |
| Alternate Track | Number of alternate tracks in use                     |



## A.3 System Configuration Table

### A.3.7 System Area RFT and GET Pages

Entries in the RFT and GET are in ascending sorted order.

The 256 bytes of a RFT or GET page are organized as follows:

| Byte Offset | RW Type | Name       | Value                  |
|-------------|---------|------------|------------------------|
| 00h         | RO      | F/Cylinder | 00h . . . 8Fh (byte 1) |
| 01h         | RO      | Cylinder   | 00h . . . FFh (byte 0) |
| 02h         | RO      | Head       | 00h . . . FFh          |
| 03h         | RO      | Reserved   | 00h . . . FFh          |
| 04h         | RO      | Position   | 00h . . . FFh (byte 1) |
| 05h         | RO      | Position   | 00h . . . FFh (byte 0) |
| 06h         | RO      | Bit Length | 00h . . . FFh (byte 1) |
| 07h         | RO      | Bit Length | 00h . . . FFh (byte 0) |
| 08h         | RO      | F/Cylinder | 00h . . . 8Fh (byte 1) |
| 09h         | RO      | Cylinder   | 00h . . . FFh (byte 0) |
| 0Ah         | RO      | Head       | 00h . . . FFh          |
| 0Bh         | RO      | Reserved   | 00h . . . FFh          |
| 0Ch         | RO      | Position   | 00h . . . FFh (byte 1) |
| 0Dh         | RO      | Position   | 00h . . . FFh (byte 0) |
| 0Eh         | RO      | Bit Length | 00h . . . FFh (byte 1) |
| 0Fh         | RO      | Bit Length | 00h . . . FFh (byte 0) |
| ⋮           | ⋮       | ⋮          | ⋮                      |
| F8h         | RO      | F/Cylinder | 00h . . . 8Fh (byte 1) |
| F9h         | RO      | Cylinder   | 00h . . . FFh (byte 0) |
| FAh         | RO      | Head       | 00h . . . FFh          |
| FBh         | RO      | Reserved   | 00h . . . FFh          |
| FCh         | RO      | Position   | 00h . . . FFh (byte 1) |
| FDh         | RO      | Position   | 00h . . . FFh (byte 0) |
| FEh         | RO      | Bit Length | 00h . . . FFh (byte 1) |
| FFh         | RO      | Bit Length | 00h . . . FFh (byte 0) |

Eight bytes of FFh follow the last valid flaw address, followed by all 0's until the table ends.

If set, F bit (bit 15) of the cylinder indicates a bad track.



# B

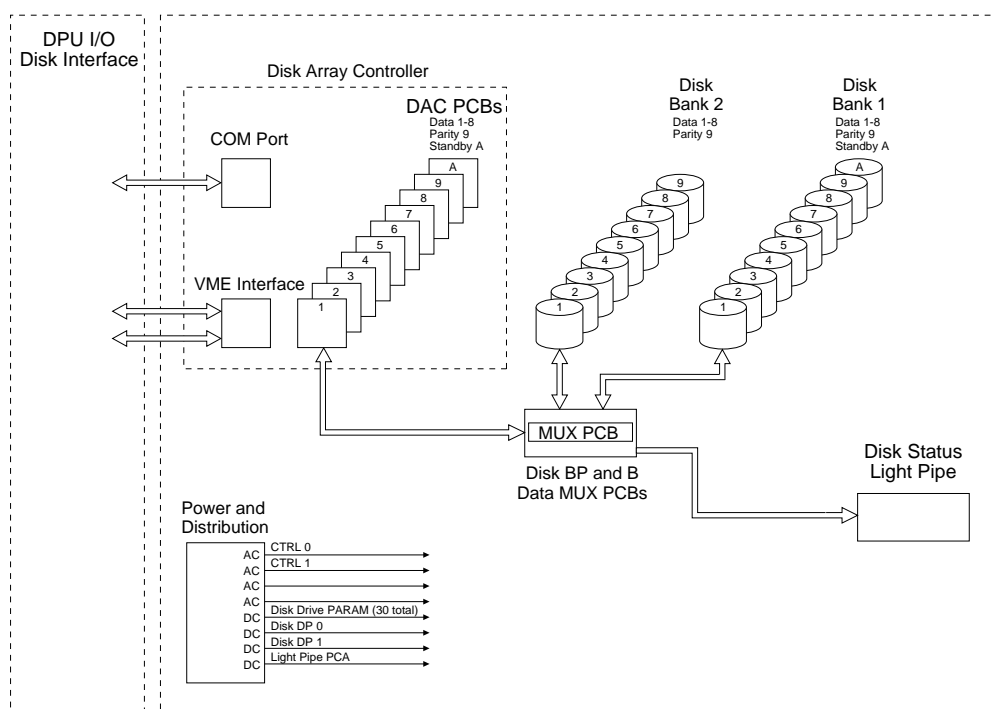
## Parallel Disk Array System Block Diagrams

Appendix B contains the DECmpp 12000/Sx parallel disk array (PDA) system block diagrams.

### B.1 Parallel Disk Array System Block Diagrams

Figure B-1 shows the full system block diagram. Figure B-2 shows the DA controller block diagram, and Figure B-3 shows the DA backplane block diagram.

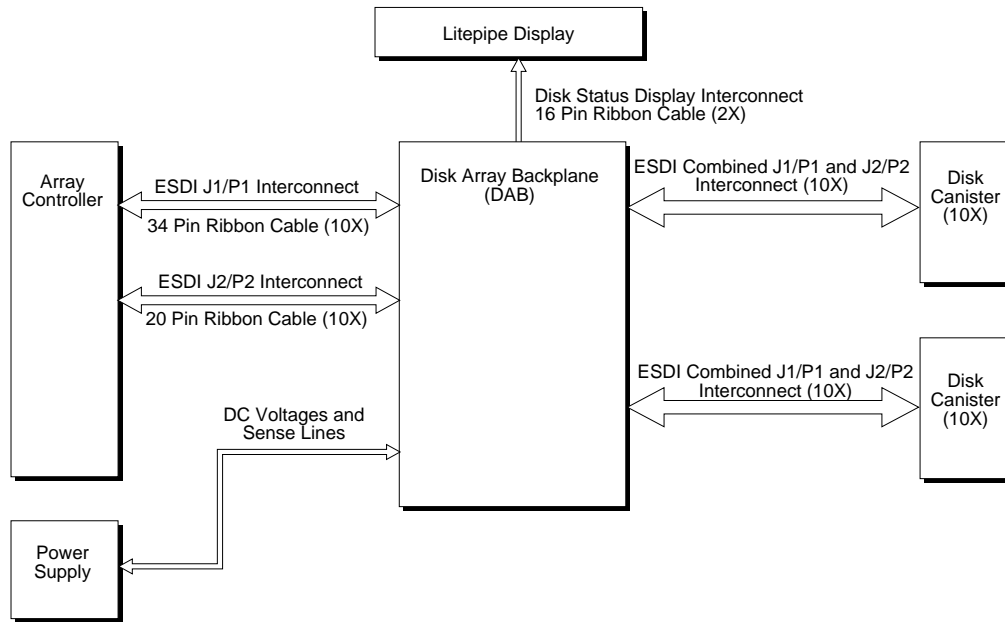
Figure B-1 Full System Block Diagram



MKV-040000314-90-MPS

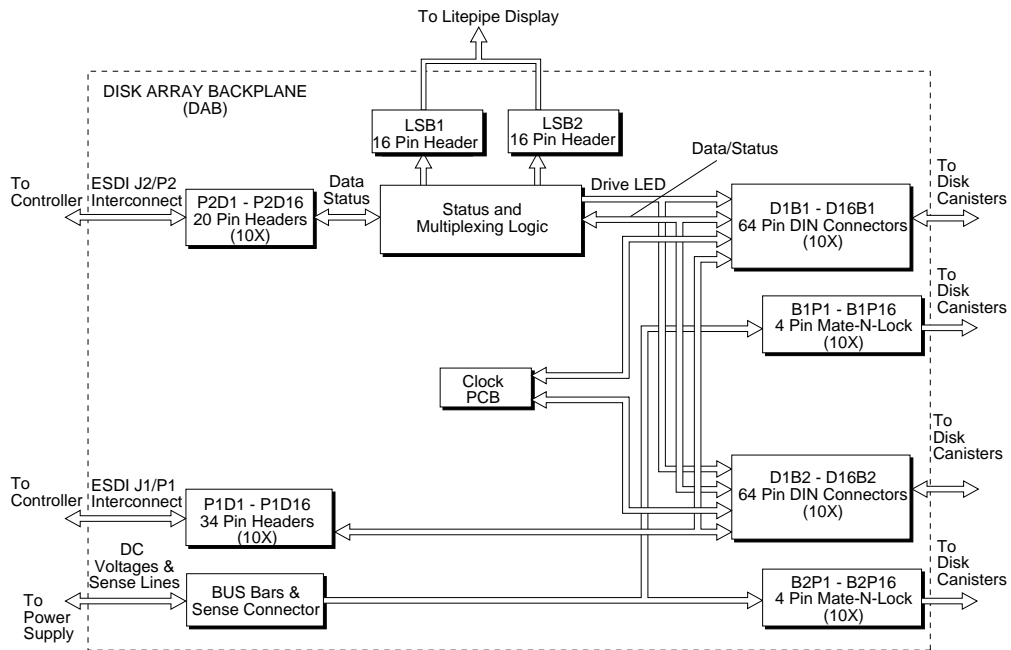
## B.1 Parallel Disk Array System Block Diagrams

Figure B-2 Disk Array Controller Block Diagram



MKV-04000314-91-MPS

Figure B-3 Disk Array Backplane Block Diagram



MKV-04000314-92-MPS

# C

## Recommended Spares List

Table C-1 provides a list of recommended spares for the DECmpp 12000/Sx Series Parallel Disk Array.

**Table C-1 Parallel Disk Array RSL**

| DEC PN      | Vendor PN    | Description  |
|-------------|--------------|--|
| 29-29380-01 | 3400-0034-00 | Disk array multiplexer PCB <sup>1,2</sup>            |
| 29-29391-01 | 3400-0035-00 | Disk array clock PCB <sup>1,2</sup>                  |
| 29-29393-01 | 4200-0035-00 | Temperature sensor cable <sup>2</sup>                |
| 29-30510-01 | 4300-0081-00 | 1.5 GB disk drive <sup>1,2</sup>                     |
| 29-29409-01 | 8010-0012    | MS-2 CPU PCB <sup>1,2</sup>                          |
| 29-29410-01 | 8010-0013    | MS-2 parity PCB <sup>1,2</sup>                       |
| 29-29411-01 | 8010-0014    | MS-2 ESDI controller PCB <sup>1,2</sup>              |
| 29-29412-01 | 8010-0015    | MS-2 VME interface PCB (SVME) <sup>1,2</sup>         |
| 29-29388-01 | 4200-0071-00 | SVME DPU to DA backplane ribbon cable <sup>1,2</sup> |
| 70-30313-02 | N/A          | 2500 W power supply (modified) <sup>2</sup>          |
| 70-30318-01 | N/A          | Lightpipe PCB <sup>2</sup>                           |

<sup>1</sup>Branch-level spare

<sup>2</sup>Geography-level spare



---

## Disk Array Variations and Options

Table D-1 lists all of the part numbers for the DECmpp Disk Array configurations along with upgrade kits, spare drives, and the DECmpp VMEbus interface options.

**Table D-1 DECmpp Disk Array Variations**

| Part Number | Description   |
|-------------|---|
| SMP11-AA    | DECmpp Parallel Disk Array System w/4 RMP11-AA Data Disks, 5.8GB. Requires KF100-AA VME Interface.            |
| SMP11-AB    | DECmpp Parallel Disk Array System w/8 RMP11-AA Data Disks, 11.5GB. Requires KF100-AA VME Interface.           |
| SMP11-AC    | DECmpp Parallel Disk Array System w/16 RMP11-AA Data Disks, 23GB. Requires KF100-AA VME Interface.            |
| SMP12-AA    | DECmpp Parallel Disk Array Upgrade from 4 RMP11-AA Data Disks (SMP11-AA) to 8 RMP11-AA Data Disks (SMP11-AB)  |
| SMP12-AB    | DECmpp Parallel Disk Array Upgrade from 4 RMP11-AA Data Disks (SMP11-AA) to 16 RMP11-AA Data Disks (SMP11-AC) |
| SMP12-AD    | DECmpp Parallel Disk Array Upgrade from 8 RMP11-AA Data Disks (SMP11-AB) to 16 RMP11-AA Data Disks (SMP11-AC) |
| RMP11-AA    | DECmpp Hot Standby 1.5GB Removable Disk Drive w/Controller  |
| KF100-AA    | DECmpp VME I/O Controller (PVME) w/8MB ECC RAM for DECmpp 12000/Sx Series Parallel Disk Array Systems         |

---





---

## Parallel Disk Array Reference Pages

This appendix contains the following parallel disk array reference pages:

- `dabinit(8)`
- `dabstat(8)`
- `dadisk(8)`
- `dadsbl(8)`
- `daflaw(8)`
- `daformat(8)`
- `dainit(8)`
- `damapchs(8)`
- `damount(8)`
- `darecon(8)`
- `dascrub(8)`
- `daspecflaw(8)`
- `dastat(8)`
- `dasubs(8)`
- `daxformat(8)`
- `daxrdc(8)`
- `daxscrub(8)`
- `daxwrc(8)`
- `da_error_log(8)`
- `da_error_report(8)`

## dabinit(8)

---

## dabinit(8)

`dabinit` — Initializes controller, bank configuration information on a DECmpp Sx disk array, Version 1.1

### Syntax

```
dabinit -b bank [-t disktype] -m 4or8drives diskarray
```

### Description

Installing a disk array, upgrading a disk array bank, or adding a new disk array bank requires initialization of configuration parameters on the disk array controller. The `dabinit` command is used when a bank is upgraded or a new bank is added. The disk array controller's Bank Control/Status Configuration is initialized.

The argument *diskarray* is a character special file associated with the disk array controller to which the command should be sent.

### Options

**-b *bank***

Use this option to specify the bank number; valid *banks* are 1–4.

**-t *disktype***

Use this option to specify the type of disks that make up the bank. The variable *disktype* is a digit. The type of disk that corresponds to *disktype* can be determined with the `dadisk(8)` command.

**-m *4or8drive***

Use this option to specify the number of data disks in the bank; valid numbers are 4 and 8.

### Restrictions

Root privileges are required to execute this command.

### Files

`/dev/da?` — Disk array controller device

### See Also

`da(4)`, `dastat(8)`, `dabstat(8)`, `dadisk(8)`, `dainit(8)`

---

## dabstat(8)

dabstat — Displays configuration and status information from the DECmpp Sx disk array controller, Version 1.1

### Syntax

```
dabstat -b bank [diskarray]
```

### Description

The configuration and status of a *bank* of disk drives is examined with the dabstat command. This is an example of the dabstat command output:

```
Drive ID          34      DK515-78
Sub ID           0
Mode             4 data drives
                 Use primary system area
                 Check drive array addresses
                 Log soft and hard errors
                 Recover policy is R/E/P
SUBS/DSBL        0 / 0
Bank Status      On line, Mounted  -0-0-0-00-
Total blocks     659340
Reconstruct Address -1
Drive status     11 30 11 30 11 30 11 30 30 11
  Drive 1       Drive not selected
  Drive 2       Drive online/mounted
  Drive 3       Drive not selected
  Drive 4       Drive online/mounted
  Drive 5       Drive not selected
  Drive 6       Drive online/mounted
  Drive 7       Drive not selected
  Drive 8       Drive online/mounted
  Drive 9       Drive online/mounted
  Drive 10      Drive not selected
```

(Drive 10 will always appear as not selected if there is no standby drive in this bank.)

The optional argument *diskarray* is a character special file associated with the disk array controller to which the command should be sent. If not specified, *diskarray* defaults to the first controller (/dev/da0).

### Restrictions

Root privileges are required to execute these commands.

### Files

/dev/da? — Disk array controller device

### See Also

da(4), dabstat(8), dadisk(8)

## dadisk(8)

---

## dadisk(8)

**dadisk** — Displays configuration and status information from the DECmpp Sx disk array controller, Version 1.1

### Syntax

```
dadisk [-t disktype] [diskarray]
```

### Description

The parameters of a disk drive are examined with the `dadisk` command by using the *disktype* parameter. This is an example of the `dadisk` command output:

```
Disk ty processor element (PE)      DK516-15
Type bits                          0
Flag bits                          7
Cylinders                          2235
Heads                               15
Sectors/track                      41
Skip sectors                        1
Rsvd cylinders                      9
Available cylinders                2226
Available sectors                  1335600
```

The command `dadisk` with no arguments outputs a list of known *disktype* and *disknames*.

The optional argument *diskarray* is a character special file associated with the disk array controller to which the command should be sent. If not specified, *diskarray* defaults to the first controller (`/dev/da0`).

### Restrictions

Root privileges are required to execute these commands.

### Files

`/dev/da?` — Disk array controller device

### See Also

`da(4)`, `dabstat(8)`, `dastat(8)`

---

## dadsbl(8)

dadsbl — Disables an inoperable drive, Version 1.1

### Syntax

```
dadsbl -b bank -d disk diskarray
```

### Description

When a drive is considered inoperable, the array controller is directed to stop using the bad drive by running the `dadsbl` command, with the appropriate *bank* and *disk* indicated.

The current disabled/substituted (DSBL/SUBS) state of a bank can be determined with the `dabstat(8)` command.

The argument *diskarray* is a character special file associated with the disk array controller to which the command should be sent. The following options are processed by `getopt(3)` and should precede *diskarray*.

### Options

**-b *bank***

Use this option to specify the bank to be disabled; legal values for *bank* are 1–4.

**-d *disk***

Use this option to specify the disk to be disabled; legal values for *disk* are 1–10.

### Restrictions

Root privileges are required to execute this command.

### Files

`/dev/da?` — Disk array controller device

### See Also

`da(4)`, `dastat(8)`, `dabstat(8)`, `dasubs(8)`, `daformat(8)`, `dascrub(8)`

## daflaw(8)

---

## daflaw(8)

daflaw — Reads and writes a DECmpp Sx array controller disk's flaw tables, Version 1.1

### Syntax

```
daflaw -b bank -ddisk [-r] [diskarray]
```

### Description

Two media flaw tables are maintained for each disk on an array controller: the raw flaw table (RFT) and the growth error table (GET). The RFT tells the controller what sectors and tracks are bad and should be spared out. Media errors, including retries, are logged to the GET as well as to the host. The `daformat(8)` command can be used to merge the GET into the RFT, effectively sparing the sectors indicated by the GET.

The `daflaw` command reads the specified flaw table from the disk and writes the information in ASCII decimal to the standard output.

Each line of the file describes a media defect. The line is formatted as follows:

```
cylinder head position length [ trackflag ]
```

The *cylinder* and *head* variables are in decimal; *position* is the bit position of the media defect start on the specified track. The *length* variable is the number of bits that the media defect covers. The optional *trackflag* variable is any ASCII string; when present, it indicates the the entire track is to be considered bad.

This is an excerpt from the output of the `daflaw` command:

```
214  5 28076    3
281  1 21299  8968  BAD_TRACK
```

The optional argument *diskarray* is a character special file associated with the disk array controller to which the command should be sent. If not specified, *diskarray* defaults to the first controller (`/dev/da0`). The following options are processed by `getopt(3)` and should precede *diskarray*.

### Options

**-r**

Use this option to select the RFT to be read and output. The default table to read is the GET.

**-b *bank***

Use this option to select the bank; legal values for *bank* are 1–4.

**-d *disk***

Use this option to select the disk; legal values for *disk* are 1–10.

### Restrictions

Root privileges are required to execute this command.

**Files**

`/dev/da?` — Disk array controller device

**See Also**

`da(4)`, `daformat(8)`, `daspecflaw(8)`

## daformat(8)

---

## daformat(8)

daformat — Formats disk(s) on the DECmpp Sx disk array, Version 1.1

### Syntax

```
daformat -b bank -f function [-t target] [-A] diskarray
```

### Description

There are usually two reasons to use the `daformat` command: when a new bank of disks is installed and when a disk is being installed to replace an inoperable disk. Without the `-A` option, `daformat` automatically determines which drive to format. It examines the disabled (DSBL) and substituted (SUBS) fields of the specified *bank* and selects the disk marked inoperable. After the format, `darecon(8)` is run to reconstruct data onto the newly installed drive.

For this command and options, a list of drives to format is output, and positive confirmation is required from the user.

The argument *diskarray* is a character special file associated with the disk array controller to which the command should be sent. The following options are processed by `getopt(3)` and should precede *diskarray*.

### Options

#### **-b *bank***

Use this option to specify the bank; legal values for *bank* are 1–4.

#### **-f *function***

Use this option to specify the function; legal values for *function* are 0–5.

#### **-t *target***

In the case that DSBL is not equal to 0, SUBS is not equal to 0, and DSBL is not equal to SUBS, both the disk drives indicated by DSBL and SUBS are candidates for format. In this case, use the `-t` option to select the drive. When *target* is equal to 0, the disk drive indicated by DSBL is selected. When *target* is equal to 1, the disk drive indicated by SUBS is selected. The variable *target* can have no other value.

#### **-A**

When the `-A` option is used, the status of the selected *bank* is examined, and all drives present in the bank are selected for format.

### Restrictions

Root privileges are required to execute this command.

### Files

`/dev/da?` — Disk Array controller device



**See Also**

`da(4)`, `dabstat(8)`, `dadsbl(8)`, `darecon(8)`, `daxformat(8)`

## dainit(8)

---

## dainit(8)

`dainit` — Initializes controller, bank configuration information on a DECmpp Sx disk array, Version 1.1

### Syntax

```
dainit [-m standbymode] diskarray
```

### Description

Installing a disk array, upgrading a disk array bank, or adding a new disk array bank requires initialization of configuration parameters on the disk array controller. The `dainit` command is used when a disk array is first installed or if some disks are added, changing the standby mode. The disk array controller's System Configuration Table is initialized.

The argument *diskarray* is a character special file associated with the disk array controller to which the command should be sent.

### Options

#### **-m *standbymode***

There are two standby modes possible for the disk array, one standby for all banks or one standby for each bank. If *standbymode* is 1, all banks share the standby on bank 1. If *standbymode* is 0, each bank has its own specified standby. If this option is not specified, the standby mode is set to one standby for all banks.

### Restrictions

Root privileges are required to execute this command.

### Files

`/dev/da?` — Disk Array controller device

### See Also

`da(4)`, `dastat(8)`, `dabstat(8)`, `dadisk(8)`, `dabinit(8)`

---

## damapchs(8)

damapchs — Converts error location on the DECmpp Sx disk array to file system partition and block number, Version 1.1

### Syntax

```
damapchs -b bank cylinder head sector [diskarray]
```

### Description

Most of the error-reporting associated with the disk array provide cylinder, head, and sector numbers specifying the location of the error. The `damapchs` command uses these and reports which file system partition the sector is in and the block number.

Note that the disk array has a reserved data area that is inaccessible. The accessible area starts at 0. The *cylinder*, *head*, and *sector* used for this command are assumed to be addresses in the accessible portion of the disk array.

The optional argument *diskarray* is a character special file associated with the disk array controller to which the command should be sent. If not specified, *diskarray* defaults to the first controller (`/dev/da0`).

Here is an example of output from the `damapchs` command:

```
translate cyl 1317, head 5, sector 1 to partition and block
      partition      blocks      (array sector)
d          75608 - 75615      (9451)
f          1627568 - 1627575      (203446)
g          3182768 - 3182775      (397846)
```

The following options are processed by `getopt(3)` and should immediately follow the command name.

### Options

**-b *bank***

Use this option to specify the bank; legal values for *bank* are 1–4.

### Restrictions

Root privileges are required to execute these commands.

### Files

`/dev/da?` — Disk Array controller device

### See Also

`da(4)`, `darecon(8)`

## damount(8)

---

## damount(8)

damount — Initializes a bank on the DECmpp Sx disk array, Version 1.1

### Syntax

```
damount -b bank [-ddisk] [diskarray]
```

### Description

The `damount` command initiates a disk array bank mount sequence and establishes the bank's status. When maintenance is being performed, and physical changes are being made to the disk array, the `damount` command will usually be given before a `dabstat(8)` command is given.

The optional argument *diskarray* is a character special file associated with the disk array controller to which the command should be sent. If not specified, *diskarray* defaults to the first controller (`/dev/da0`). The following options are processed by `getopt(3)` and should precede *diskarray*.

With the `-d` option, specified drives can be spun down, or all drives can be commanded to spin up.

### Options

**-b *bank***

Use this option to specify the target bank; legal values for bank are 1–4.

**-d *disk***

Use this option to specify the target drive (*disk*) to be spun down. If the variable *disk* is 0, all drives are commanded to spin up.

### Restrictions

Root privileges are required to execute these commands.

### Files

`/dev/da?` — Disk Array controller device

### See Also

`dabstat(8)`, `da(4)`

---

## darecon(8)

darecon — Performs data/parity drive reconstruction, Version 1.1

### Syntax

```
darecon -b bank [-t target] diskarray
```

### Description

After an inoperable drive has been replaced and the new drive formatted, use the `darecon` command to reconstruct the data/parity drive. The drive selected for reconstruction is indicated by the disabled/substituted (DSBL/SUBS) fields of the bank's status.

During reconstruction, unrecoverable media errors might be encountered. The `darecon` command writes to standard-out the cylinder, head, and sector numbers where data could not be recovered. The command `damapchs(8)` is used to convert the cylinder, head, and sector numbers to a UNIX block number in a file system. If unrecoverable errors occurred, files associated with the bad blocks should be repaired, if possible, or otherwise removed.

At the completion of the `darecon` command, the DSBL/SUBS fields of the bank's status are modified to indicate the new drive in use.

The argument *diskarray* is a character special file associated with the disk array controller to which the command should be sent. The following options are processed by `getopt(3)` and should precede *diskarray*.

### Options

#### **-t *target***

This option is only valid when DSBL is nonzero, SUBS is nonzero, and DSBL is not equal to SUBS. When the variable *target* is set to 0, the drive identified by DSBL is selected for reconstruction. When *target* is set to 1, the drive identified by SUBS is selected for reconstruction.

### Restrictions

Root privileges are required to execute these commands.

### Files

`/dev/da?` — Disk array controller device

### See Also

`da(4)`, `daxdsbl(8)`, `dabstat(8)`

## dascrub(8)

---

### dascrub(8)

dascrub — Performs a digital surface analysis of disks on the DECmpp Sx disk array, Version 1.1

#### Syntax

```
dascrub -b bank -f function [-t target] [-A] diskarray
```

#### Description

If a disk is suspected of having media defects that are not known to the controller, the `dascrub` command scans disks, normally performing a write/read, looking for media defects. The drive must already be formatted. Media errors are logged to the growth error table (GET). Without the `-A` option, the `dascrub` command automatically determines which drive to scan for media defects. It examines the DSBL and SUBS fields of the specified *bank* and selects the disk marked inoperable. After the scrub, the `daformat(8)` is normally run to merge the GET into the RFT.

When you use this command and its options, a list of drives to scrub is output, and positive confirmation is required from the user.

The argument *diskarray* is a character special file associated with the disk array controller to which the command should be sent. The following options are processed by `getopt(3)` and should precede *diskarray*.

#### Options

**-b *bank***

Use this option to specify the target bank; legal values for *bank* are 1–4.

**-f *function***

Use this option to specify the pattern to be used, or the read-check only function; legal values for *function* are 0–15.

**-t *target***

In the case that DSBL is not equal to 0, SUBS is not equal to 0, and DSBL is not equal to SUBS, both the disk drives indicated by DSBL and SUBS are candidates for scrub. Use the `-t` option to select the drive. When *target* is equal to 0, the disk drive indicated by DSBL is selected. When *target* is equal to 1, the disk drive indicated by SUBS is selected. The variable *target* may have no other value.

**-A**

When the `-A` option is used, the status of the selected *bank* is examined, and all drives present in the bank are selected for scrub.

#### Restrictions

Root privileges are required to execute these commands.

**Files**

`/dev/da?` — Disk array controller device

**See Also**

`da(4)`, `dabstat(8)`, `dadsbl(8)`, `daformat(8)`, `darecon(8)`, `dascrub(8)`

## daspecflaw(8)

---

## daspecflaw(8)

`daspecflaw` — Reads and writes a DECmpp Sx array controller disk's flaw tables, Version 1.1

### Syntax

```
daspecflaw -b bank -d disk -f flawfile diskarray
```

### Description

Two media flaw tables are maintained for each disk on an array controller: the raw flaw table (RFT) and the growth error table (GET). The RFT tells the controller what sectors and tracks are bad and should be spared out. Media errors, including retries, are logged to the GET as well as to the host. The `daformat(8)` command can be used to merge the GET into the RFT, effectively sparing the sectors indicated by the GET.

The `daflaw` command reads the specified flaw table from the disk and writes the information in ASCII decimal to the standard output. The `daspecflaw` command uses a file in `daflaw`-generated format to add entries to the GET.

The optional argument *diskarray* is a character special file associated with the disk array controller to which the command should be sent. If not specified, *diskarray* defaults to the first controller (`/dev/da0`). The following options are processed by `getopt(3)` and should precede *diskarray*.

### Options

**-b *bank***

Use this option to specify the target bank; legal values for *bank* are 1–4.

**-d *disk***

Use this option to specify the target disk; legal values for *disk* are 1–10.

**-f *flawfile***

Use this option to specify the name of the file that contains the media defect information.

### Restrictions

Root privileges are required to execute this command.

### Files

`/dev/da?` — Disk array controller device

### See Also

`da(4)`, `daflaw(8)`, `daformat(8)`



---

## dastat(8)

dastat — Displays configuration and status information from the DECmpp Sx disk array controller, Version 1.1

### Syntax

```
dastat [diskarray]
```

### Description

Global controller configuration parameters and operational variables are examined with the dastat command. This is an example of the dastat command output:

```
Control          Physical addressing
                 Disk logging is enabled
                 OS logging is enabled
                 Each bank has standby
                 Format interleave is by track
                 Reconstruct interleave is by track
Retries          10
Data Chan TMO    5
Ready TMO        255
Spin up delay    2
Status filter    0001
Monitor          0000
Buffer presence  0000000000
```

The optional argument *diskarray* is a character special file associated with the disk array controller to which the command should be sent. If not specified, *diskarray* defaults to the first controller (/dev/da0).

### Restrictions

Root privileges are required to execute these commands.

### Files

/dev/da? — Disk array controller device

### See Also

da(4), dabstat(8), dadisk(8)

## dasubs(8)

---

### dasubs(8)

dasubs — Substitutes a standby drive for an inoperable drive, Version 1.1

#### Syntax

```
dasubs -b bank -d disk diskarray
```

#### Description

When a drive is considered inoperable and a standby drive is available, the array controller is directed to stop using the bad drive by running the `dasubs` command. After `dasubs` is run, `darecon(8)` should be run to reconstruct the inoperable drive onto the standby.

The current disabled/substituted (DSBL/SUBS) state of a bank can be determined with the `dabstat(8)` command.

The argument *diskarray* is a character special file associated with the disk array controller to which the command should be sent. The following options are processed by `getopt(3)` and should precede *diskarray*.

#### Options

**-b *bank***

Use this option to specify the bank that the bad drive is in; legal values for *bank* are 1–4.

**-d *disk***

Use this option to specify the disk that the standby drive is to be substituted for; legal values for *disk* are 1–9.

#### Restrictions

Root privileges are required to execute these commands.

#### Files

`/dev/da?` — Disk array controller device

#### See Also

`da(4)`, `dastat(8)`, `dadsbl(8)`, `dabstat(8)`, `daformat(8)`, `dascrub(8)`

---

## daxformat(8)

daxformat — Formats disk(s) on the DECmpp Sx disk array, Version 1.1

### Syntax

```
daxformat -b bank -f function [-d drive] diskarray
```

### Description

The `daxformat` command allows specific drives within a bank to be selected for format. This command is not normally used (refer to the `daformat` command). There are generally two reasons to use this command: when a new bank of disks is installed and when a disk is being installed to replace an inoperable disk.

For this command and options, a list of drives to format is output, and positive confirmation is required from the user.

The argument *diskarray* is a character special file associated with the disk array controller to which the command should be sent. The following options are processed by `getopt(3)` and should precede *diskarray*.

### Options

**-b *bank***

Use this option to specify the target bank; legal values for *bank* are 1–4.

**-f *function***

Use this option to specify the function to be performed; legal values for *function* are 0–5.

**-d *drive***

Use this option to specify the target drive. There can be multiple `-d` options; thus an arbitrary set of drives can be specified for format.

### Restrictions

Root privileges are required to execute this command.

### Files

`/dev/da?` — Disk array controller device

### See Also

`da(4)`, `dabstat(8)`, `dadsbl(8)`, `daformat(8)`, `darecon(8)`

## daxrdc(8)

---

## daxrdc(8)

daxrdc — Reads the configuration data of a DECmpp Sx disk array controller, Version 1.1

### Syntax

```
daxrdc [-s sel] [-t tab] [-o offset] [-n n] diskarray
```

### Description

The `daxrdc` utility is not intended for general use. It gives direct access to the disk array controller's configuration data. The format of this data area is described elsewhere. Other commands are available to examine and manipulate this data in a meaningful way.

The argument *diskarray* is a character special file associated with the disk array controller to which the command should be sent.

The options are processed by `getopt(3)` and should precede *diskarray* for the `daxrdc` command and *hexvalue* for the `daxwrc` command.

### Restrictions

Root privileges are required to execute these commands.

### Files

`/dev/da0` — Disk Array controller device

### See Also

`da(4)`, `dastat(8)`, `dabstat(8)`, `daflaw(8)` `daxwrc(8)`

---

## daxscrub(8)

daxscrub — Performs a digital surface analysis of disks on the DECmpp Sx disk array, Version 1.1

### Syntax

```
daxscrub -b bank -f function [-d drive] diskarray
```

### Description

If a disk is suspected of having media defects that are not known to the controller, the `daxscrub` command scans a specific disk, normally performing a write/read, looking for media defects. The drive must already be formatted. Media errors are logged to the growth error table (GET). After the scrub, the `daformat(8)` is normally run to merge the GET into the RFT. This command is not normally used.

When you use this command and its options, a list of drives to scrub is output, and positive confirmation is required from the user.

The argument *diskarray* is a character special file associated with the disk array controller to which the command should be sent. The following options are processed by `getopt(3)` and should precede *diskarray*.

### Options

**-b *bank***

Use this option to specify the target bank; legal values for *bank* are 1–4.

**-f *function***

Use this option to specify the pattern to be used, or the read-check only function; legal values for *function* are 0–15.

**-d *drive***

Use this option to set an arbitrary set of drives specified for scrub; there can be multiple -d options.

### Restrictions

Root privileges are required to execute these commands.

### Files

`/dev/da?` — Disk Array controller device

### See Also

`da(4)`, `dabstat(8)`, `dadsbl(8)`, `daformat(8)`, `darecon(8)`, `dascrub(8)`

## daxwrc(8)

---

## daxwrc(8)

`daxwrc` — Writes the configuration data of a DECmpp Sx disk array controller, Version 1.1

### Syntax

```
daxwrc [-s sel] [-t tab] [-o offset] hexvalue diskarray
```

### Description

The `daxwrc` utility is not intended for general use. It gives direct access to the disk array controller's configuration data. The format of this data area is described elsewhere. Other commands are available to examine and manipulate this data in a meaningful way.

The argument *diskarray* is a character special file associated with the disk array controller to which the command should be sent. The argument *hexvalue* is a byte in hex that should be written at the selected location. Note that the `daxwrc` command only writes a single location.

The options are processed by `getopt(3)` and should precede *hexvalue* for the `daxwrc` command.

### Restrictions

Root privileges are required to execute these commands.

### Files

`/dev/da0` — Disk Array controller device

### See Also

`da(4)`, `dastat(8)`, `dabstat(8)`, `daflaw(8)` `daxwrc(8)`

---

## da\_error\_log(8)

da\_error\_log — Extracts DECmpp Sx parallel disk array (MPDA) related entries from uerf

### Syntax

```
da_error_log [days_to_check]
```

### Description

The `da_error_log` command examines the `uerf(8)` output for the specified period and prints only entries relating to the DECmpp Sx parallel disk array (MPDA).

This program is usually run by `cron(1)` to give periodic status reports on the MPDA.

### Options

#### **days\_to\_check**

This optional parameter specifies how many full days to summarize. A full day goes from midnight to midnight. For example, if the command is given at 8:00A.M., and `days_to_check` is 1, the last 32 ( $1 * 24 + 8$ ) hours will be summarized. If this option is zero, activity since midnight will be checked.

### Files

`$MP_PATH/etc/MPDA.mail` — File containing mail addresses

### See Also

`cron(1)`, `dabstat(8)`, `da_error_report(8)`, `uerf(8)`

## da\_error\_report(8)

---

### da\_error\_report(8)

`da_error_report` — Summarizes DECmpp Sx parallel disk array (MPDA) errors, Version 1.1

#### Syntax

```
da_error_report [-m] [days_to_check]
```

#### Description

The `da_error_report` command examines the DECmpp Sx parallel disk array (MPDA) status and the `uerf(8)` output. If certain MPDA related errors or anomalies are found, `da_error_report` summarizes these errors in a table. The `da_error_report` command with no options prints a summary of the previous day to standard out.

This program is usually run by `cron(1)` to give periodic status reports on the MPDA.

#### Options

**-m**

This option specifies that the output should be mailed to *mpservice* (generally an alias). This option is generally used when the command is invoked by `cron(1)`. Note that nothing is mailed if no problems are encountered.

**days\_to\_check**

This optional parameter specifies how many full days to summarize. A full day goes from midnight to midnight. For example, if the command is given at 8:00A.M., and *days\_to\_check* is 1, the last 32 (1 \* 24 + 8) hours will be summarized. If this option is zero, activity since midnight will be checked.

#### Files

`$MP_PATH/etc/MPDA.mail` — File containing mail addresses

#### See Also

`cron(1)`, `dabstat(8)`, `da_error_log(8)`, `uerf(8)`



## B

---

- Backplane, 1–9
  - connector blocks, 1–8
  - disk canister, 1–11
  - installing MUX PCBs, 2–8
  - interconnect, 1–11
  - lightpipe, 1–11
  - multiplexer interconnect, 1–11
  - PCB slot assignments, 1–7
  - power, 1–15
  - power interconnect, 1–11
  - replacing disk cables, 5–17
- Bad block table, A–1, A–18
- Bank
  - configuration, 3–5
  - control/status block, A–10
- Banks
  - assigning a standby drive, 4–6
  - bringing up, 4–3
  - diagnostic, 6–4
  - disabling a drive, 4–6
  - formatting a disk, 4–3
  - mount diagnostic, 6–4
  - reconstructing a drive, 4–6
  - status, 6–1

## C

---

- Cable clamps, 1–8
- Cables
  - interface, 1–13
  - replacing +12 V, 5–12
  - replacing +5 V, 5–12
  - replacing backplane disk, 5–17
  - replacing internal ground, 5–12
  - replacing lightpipe power, 5–19
  - replacing lightpipe signal, 5–18
  - replacing PDA-DPU interface, 5–15
  - replacing temperature sensor, 5–20
- Chassis ground circuit, 1–15
- Chassis, DA controller
  - extending, 5–3
- Circuit breaker
  - DPU, 2–7
  - PDA, 2–17, 5–10

- Clock PCB, replacing, 5–14
- Clock signal routing, 1–13

Com port, 1–6

### Commands

- dabinit, E–2
- dabstat, E–3
- dadisk, E–4
- dadsbl, E–5
- daflaw, E–6
- daformat, E–8
- dainit, E–10
- damapchs, E–11
- damount, E–12
- darecon, E–13
- dascrub, E–14
- daspecflaw, E–16
- dastat, E–17
- dasubs, E–18
- daxformat, E–19
- daxrdc, E–20
- daxscrub, E–21
- daxwrc, E–22
- da\_error\_log, E–23
- da\_error\_report, E–24

### Communication

- for the PDA, 3–2

Components, 1–5

Configuration data, A–1

Configurations, 1–1

Connector blocks, 1–8

Controller configuration, 3–5, A–4

Controller, disk array, 1–6

Cooling requirement, PDA, 2–1

Cyclic redundancy checking, A–12

## D

---

- dabinit, 3–5
- dabinit command, E–2
- dabstat command, E–3
- dadisk command, E–4
- dadsbl command, E–5
- daflaw command, E–6
- daformat command, E–8
- daformat(8), 4–1
- dainit command, E–10

- Damage
  - to shipping container, 2-4
- damapchs command, E-11
- damount command, E-12
- damount(8), 4-1
- darecon command, E-13
- darecon(8), 4-5
- dascrub command, E-14
- dascrub(8), 4-8
- daspecflaw command, E-16
- daspecflaw(8), 4-9
- dastat command, E-17
- dasubs command, E-18
- Data multiplexing, 1-13
- Data parallel unit
  - connecting PDA, 2-11
  - connecting to PDA, 1-13
  - I/O jumper settings, 2-15
  - installing PCBs, 2-11
  - turning on, 2-17
- daxformat command, E-19
- daxrdc command, E-20
- daxscrub command, E-21
- daxwrc command, E-22
- DA\_AUTO\_DISABLE, 6-8
- DA\_AUTO\_RECONSTRUCT, 6-8
- da\_error\_log command, E-23
- da\_error\_report command, E-24
- DA\_FAULT\_STATUS, 6-7
- DA\_MEDIA\_LOG, 6-7
- DA\_NONXFER\_ERROR, 6-7
- DA\_XFER\_ERROR, 6-7
- DA\_XFER\_RETRY, 6-7
- DC power distribution, 1-15
- Device entry, 3-2
- Diagnostics
  - aborting, 6-3
  - automatic mode, 6-4
  - command/status summary, 6-10
  - da\_buff test, 6-4
  - da\_config test, 6-4
  - da\_disk test, 6-5
  - da\_intern test, 6-4
  - da\_mount test, 6-4
  - da\_seek test, 6-5
  - DCA controller, 6-5
  - disk error logging, 6-5
  - error codes, 6-5
  - extended transfer error status, 6-9
  - internal disk array controller test, 6-4
  - mount bank test, 6-4
  - PDA programs, 6-4
  - programs list, 6-3
  - read/seek test, 6-5
  - read/write buffer test, 6-4
  - read/write configuration test, 6-4
  - read/write disk test, 6-5
  - running, 6-4

- Diagnostics (cont'd)
  - VME adapter, 6-4
- Disk addressing modes, A-6
- Disk array, 1-1
  - controller, 1-6
    - functions, 1-6
- Disk array backplane, 1-5
- Disk array controller, 1-1, 1-5
  - backplane, 1-8
  - backplane PCB slot assignments, 1-7
  - com port, 1-6
  - configuration, 3-5
  - diagnostic, 6-4
  - diagnostics, 6-5
  - extending the chassis, 5-3
  - installing PCBs, 2-8
  - interconnect, 1-11
  - replacing PCBs, 5-5
  - reset switch, 1-6
  - RS-232 port, 1-6
- Disk canisters, 1-11, 1-12
  - installing, 2-8
  - layout, 2-9
  - replacing, 5-6
  - status, 6-2
- Disk chassis, 1-5, 1-12
- Disk drives
  - configurations, 1-1
  - replacement, 1-1
- Disk formatting, 4-1
  - assigning a standby drive, 4-6
  - Bank Mount sequence, 4-4
  - disabling a drive, 4-6
  - examples, 4-3
  - FORMAT command, 4-1
  - func field, 4-1
  - MOUNTBANK command, 4-4
  - RECONSTRUCT command, 4-5
  - reconstruction, 4-6
  - SCRUB command, 4-8
  - SPECIFYFLAW command, 4-9
- Disk parameter table, A-1, A-15
- Disks
  - error logging, 6-5
  - upgrading to 16, 7-3
  - upgrading to 8, 7-2
- doconfig, 3-4

## E

---

- Enclosure
  - extending DA controller chassis, 5-3
- Error checking and correction, A-12
- Error codes, 6-5
  - extended transfer error status, 6-9
- Error logs, A-6
- Error recovery operations, A-12

Errors, media, 6-7  
Exhaust fan, 1-17  
Expansion, 1-1

## F

---

Fan, 1-5  
Fault tolerance, 1-1  
File partitions  
  checking, 3-7  
  default, 3-7  
File systems, 3-7  
Flaw descriptor table, 4-1  
FORMAT command, 4-1

## G

---

GET page, A-21  
Gravity guide bar, 1-8

## H

---

Hard errors, A-12  
Home page, A-1, A-20

## I

---

I/O jumper settings, 2-15  
Indicators  
  disk drive, 1-12, 6-2  
  lightpipe, 6-1  
Installation  
  connecting DPU and PDA, 2-11  
  DA controller PCBs, 2-8  
  disk canisters, 2-8  
  DPU PCBs, 2-11  
  MUX PCBs, 2-8  
  PCBs, 2-7  
Interface cable, 2-11

## J

---

Jumpers  
  disk drive, 2-8  
  DPU settings, 2-15

## K

---

Kernel  
  creating, 3-4

## L

---

Lightpipe  
  display, 6-1  
  indicators, 6-1  
  interconnect, 1-11  
  replacing lightpipe power cable, 5-19  
  replacing PCB, 5-8

Lightpipe (cont'd)  
  replacing signal cable, 5-18

## M

---

Media errors, 6-7  
MOUNTBANK command, 4-1, 4-4  
Multiplexer  
  connectors, 1-9  
  PCB, 1-13  
MUX, 1-9, 1-13  
  installing PCBs, 2-8  
  interconnect, 1-11  
  replacing PCB, 5-7

## P

---

Parallel disk array, 1-1, 2-1  
  backplane, 1-5  
  components, 1-5  
  configuration, 3-5  
  configurations, 1-1  
  connecting DPU, 2-11  
  connecting to DPU, 1-13  
  controller, 1-5, 1-6  
  cooling, 1-13  
  cooling requirement, 2-1  
  diagnostics, 6-1  
  disk canister, 1-12  
  disk chassis, 1-5  
  establishing communications, 3-2  
  installing controller PCB, 2-8  
  installing MUX PCBs, 2-8  
  lightpipe, 6-1  
  MUX PCB, 1-13  
  power, 1-13  
  power requirement, 2-1  
  power system, 1-15  
  power/fan tray, 1-5  
  setting up file systems, 3-7  
  space requirements, 2-1  
  specifications, 2-3  
  storage capacity, 1-1  
  turning on, 2-17  
  upgrading, 7-1  
Parallel VME 6U Adapter, 2-11, 3-3  
Parity disk, 1-1  
PCBs  
  replacing clock, 5-14  
  replacing DA controller, 5-5  
  replacing lightpipe PCB, 5-8  
  replacing MUX PCBs, 5-7  
PDA  
  unpacking instructions, 2-4  
PDA-DPU interface  
  replacing cable, 5-15  
Power interconnect, 1-11

Power requirement, PDA, 2-1  
Power selector settings, 1-14  
Power supply, 1-5  
Power system, 1-15  
    status, 6-1  
Power/fan tray, 1-5, 1-13  
    indicators, 1-16  
    replacing, 5-9

## R

---

Raw Flaw table, 4-1  
READCONFIG command, A-1  
RECONSTRUCT command, 4-5  
RFT page, A-21  
RS-232 port, 1-6, 6-5

## S

---

SCRUB command, 4-8  
Sector pulse, 6-1  
Sector pulse monitor, 1-13  
Shipping container damage, 2-4  
Signal routing, 1-11  
Soft errors, A-12  
Space requirement, PDA, 2-1  
Specifications, PDA, 2-3  
SPECIFYFLAW command, 4-9  
Standby drive, 1-1, 4-6  
Storage capacity, 1-1  
System administration  
    default partitions, 3-7

    establishing PDA communications, 3-2  
    setting up file systems, 3-7  
    upgrade procedures, 3-8  
System block diagrams, B-1  
System configuration table, A-1, A-3

## T

---

Tables  
    bad block, A-1, A-18  
    disk parameter, A-1, A-15  
    system configuration, A-1, A-3  
Temperature sensor  
    replacing cable, 5-20  
Terminators  
    setting, 2-8

## U

---

Unpacking instructions, 2-4  
Upgrades, 3-8  
Upgrading, 1-1

## V

---

VME adapter, diagnostics, 6-4

## W

---

Write signal routing, 1-13  
WRITECONFIG command, A-2