xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

xxxxxxxxxxxxxxxxxxxxxxxxxxxxx

xxxxxxxxxxxxxxxxxxxxxxxxxxx

xxxxxxxxxxxxxxxxxxxxxxxxx

xxxxxxxxxxxxxxxxxxxxxxx

xxxxxxxxxxxxxxxxxxxxx

xxxxxxxxxxxxxxxxxxx

xxxxxxxxxxxxxxxxx

xxxxxxxxxxxxxxx

xxxxxxxxxxxxx

xxxxxxxxxxx

xxxxxxxxx

xxxxxxx

xxxxx

xxx

x

RF31

x

xxx

xxxxx

xxxxxxx

xxxxxxxxx

xxxxxxxxxxx

xxxxxxxxxxxxx

xxxxxxxxxxxxxxx

xxxxxxxxxxxxxxxxx

xxxxxxxxxxxxxxxxxxx

xxxxxxxxxxxxxxxxxxxxx

xxxxxxxxxxxxxxxxxxxxxxx

xxxxxxxxxxxxxxxxxxxxxxxxx

xxxxxxxxxxxxxxxxxxxxxxxxxxx

xxxxxxxxxxxxxxxxxxxxxxxxxxxxx

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

RF31 Integrated Storage Element User Guide

Order Number EK-RF31D-UG-001

Digital Equipment Corporation

May 1990

This document was prepared and published by Educational Services Development and Publishing, Digital Equipment Corporation.

Contents

5 Performing Troubleshooting Procedures

6 Digital Repair Services

Index

Figures

CHAPTER 1

1 General Information

Introduction

The RF31 integrated storage element (ISE) is a half-height, 5-1/4 inch, disk storage device based on the Digital storage architecture (DSA) and using the Digital Storage System Interconnect (DSSI) bus and interface.

DSSI Bus

The DSSI bus is a logical equivalent to the CI bus used on larger Digital systems. It allows one or more hosts to communicate directly with storage devices, using the systems communications architecture (SCA) protocols. As many as eight nodes (ISEs and adapters) can be connected to one DSSI bus.

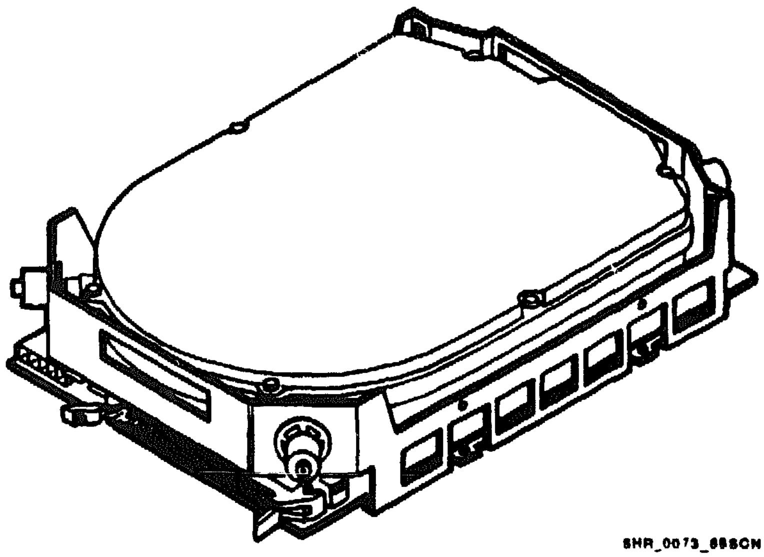
Physically, the DSSI bus is a 50-conductor cable. Inside an enclosure, it may be a flat, ribbon cable or a round bundle of twisted pairs. Between enclosures, it is a shielded, round cable approximately one-half inch in diameter.

Integrated Storage Element

An integrated storage element is a DSSI device that contains an embedded, intelligent controller and an on-board Mass Storage Control Protocol (MSCP) server. Each ISE executes commands and transfers data independently of other ISEs attached to the DSSI bus. Commands and data are transferred over the DSSI bus in small packets, so multiple ISE transfers are efficiently multiplexed.

Physical Description

The RF31 ISE is shown in Figure 1–1. It is a half-height, 5-1/4 inch fixed disk storage device. Its dimensions are 20.9 cm (8.21 in) by 14.7 cm (5.79 in) by 4.1 cm (1.62 in).



SHR_0073_69SCN
SHR_X1075_69_SCN

Figure 1–1 The RF31 ISE

Performance Features

The RF31 ISE offers powerful performance features that are not typically available on disk storage devices in this form factor. The following table lists some of these features, many of which are described in more detail in Chapter 2.

Feature	Function
Multihost support	This allows a single ISE to be used by two hosts at the same time. For example, two MicroVAX 3800 systems can be booted from a single ISE.
Seek ordering	When more than one I/O command is outstanding, the ISE performs the commands in an order that minimizes seek time. The commands considered for seek ordering include the commands from all hosts.
Request fragmentation	This technique breaks single I/O requests into smaller pieces that may be optimized independently. The result is lower rotational latency and, consequently, faster access time for large requests.

Feature	Function
Quadruplicated headers	The headers preceding each data block are replicated four times to make sure data is not lost due to header errors.
264-bit ECC	DSSI ISEs store a large, 264-bit error correction code (ECC) in each block, capable of correcting up to 120 erroneous bits.
Controller-initiated BBR	With controller-initiated bad block replacement (BBR), the ISE presents the host with a set of logically contiguous blocks, and disk capacity never shrinks because bad blocks are detected and automatically moved to spare blocks.
RCT cache	DSSI ISEs cache the replacement control table (RCT), allowing replaced blocks to be located without the time required for seeks to and from the RCT stored on the media.

Performance Specifications

The following table summarizes the performance specifications of the RF31 ISE.

Specification	Value
Data storage capacity	381 Mbytes, formatted
Average seek time	15.3 ms
Average access time	23.6 ms
Peak transfer rate to DSSI bus	4.0 Mbytes/second
Start time	<60 seconds, total
Time to attain full r/min	<15 seconds
Internal diagnostics	<33 seconds
Spin-down	<15 seconds

Current and Power Consumption

The following table summarizes the maximum current and typical power consumption specifications for the RF31 ISE.

Specification	Value
5 V supply current	1.2 A max.
12 V supply current	3.0 A (peak, first 3 seconds (nominal) of spin-up) 0.7 A max. (idle) 1.3 A max. (continuous random seeks)
Total power	12.6 W typical (idle) 18.7 W typical (continuous random seeks)

Media Specifications

The following table summarizes the specifications for the RF31 storage media.

Specification	Value
Tracks per inch	1,875
Bits per inch	30,520
Areal density	57.35 Mbits/in^2
Data tracks per surface	1,861
Sectors per track	50 data sectors 1 replacement sector
Data surfaces per drive	8
Disk type	Thin film
Servo	Fully embedded
Positioner	Rotary
R/W code	RLL 1,7

Environmental Specifications

The following table summarizes the environmental specifications for the RF31 ISE.

Specification	Value
Temperature	
Operating	10 - 50°C (50 - 122°F), ambient, with a gradient of 11°C (20°F) per hour (as introduced to the drive enclosure)
Non-operating	-40 - 66°C (-40 - 151°F), ambient, with a gradient of 20°C (36°F) per hour
Relative humidity	
Operating	10 - 90% with maximum wet bulb temperature of 28°C (82°F) and a minimum dew point of 2°C (36°F), with no condensation
Non-operating (storage/shipping)	8 - 95%, with no condensation
Altitude	
Operating	2,438 meters (8,000 feet)
Non-operating	4,876 meters (16,000 feet)
Noise (closed office environment)	<5.1 Bels, idle <5.6 Bels, seeking
Air flow	10 ft^3/min (minimum)
Agency compliance	UL, CSA, IEC 380, IEC 435, VDE 804

CHAPTER 2

2 Theory of Operation

Physical Description

Overview

The RF31 ISE consists of two basic components: a head-disk assembly (HDA) that contains the disks and drive mechanisms, and a printed circuit board called the "drive module" which contains the disk controller and drive electronics.

Figure 2–1 is a diagram of the RF31 ISE, showing the basic components.

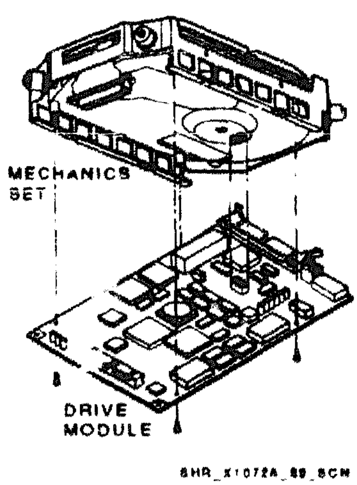


Figure 2–1 RF31 ISE Basic Components

HDA

The HDA contains the disks, heads, spindle motor, and rotary positioner. The RF31 HDA has four disks and eight heads. The ISE uses an embedded digital servo for track following, so there is no dedicated servo surface. All surfaces are used for user data storage.

The RF31 HDA is composed of 1,861 data "cylinders". Each cylinder is composed of 8 tracks, with each track having 50 sectors available for customer data.

Drive Module Interfaces

The drive module combines the functions of disk controller and drive electronics onto a single, eight-layer, surface-mount-technology printed circuit board. Figure 2–2 is a diagram of the drive module, showing the five interfaces.

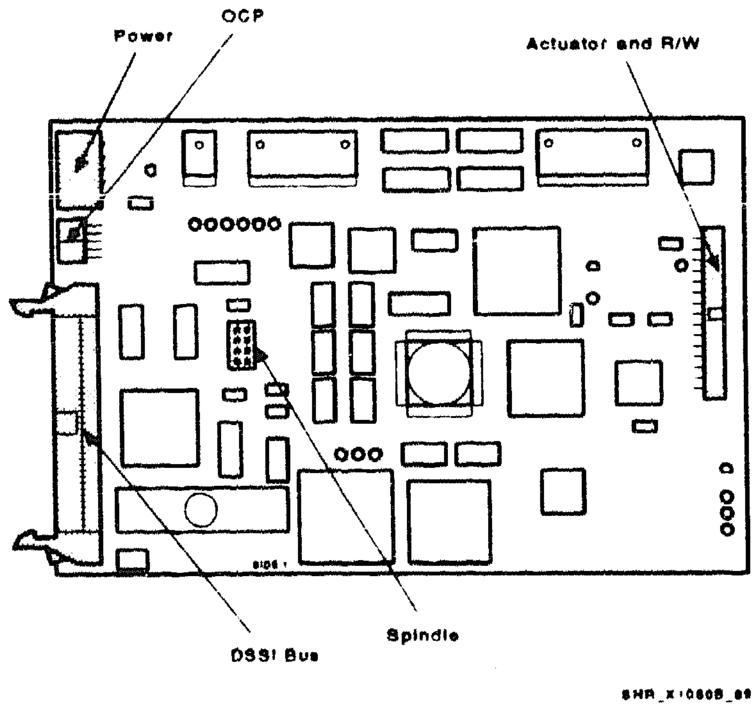


Figure 2–2 Drive Module Interfaces

As shown in Figure 2–2, each interface is a connector. The following table describes the function of each interface.

Interface	Description	Function
DSSI bus	50-pin flat cable connector	Connects ISE to DSSI bus
Power	5-pin Molex connector	Connects +5V, +12V, and ACLO to ISE
OCP	10-pin flat cable connector	Connects ISE to operator control panel
Actuator and R/W	34-pin flat cable connector that mates to the flexible circuit strip from the HDA	Connects ISE to read/write preamps and to rotary positioner
Spindle	10-pin DIP connector that mates to the HDA	Connects spindle motor to drive module

Functional Description

Drive Module Electronics

To help you understand the drive module, think of it as being partitioned into six major functional areas (Figure 2–3). Large-scale integration is used extensively, so the function of an area typically includes the function of one or a few of the chips in the corresponding area in Figure 2–3.

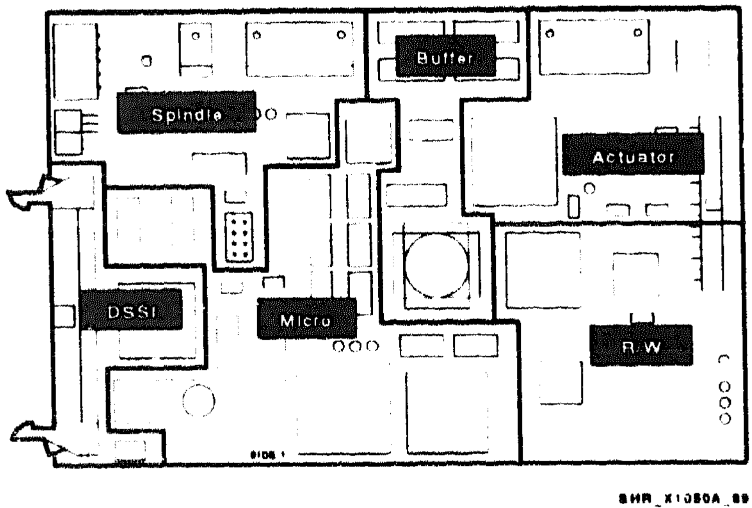


Figure 2–3 Drive Module Electronics

The following table describes the function of each partitioned area in Figure 2–3.

Area	Function
Micro	The on-board intelligence resides here, in a quarter Mbyte of firmware driving a 16 2/3-MHz MC68000[1] microprocessor with 128 Kbytes of static scratchpad RAM and 2 Kbytes of EEROM.
DSSI	A single chip (Swift) interfaces the Micro area to the DSSI bus. It provides a queued-packet interface to the Micro area, segregates data from commands, and generates and checks error detection codes.
Buffer	A single chip (Phoenix) acts as a 3-port RAM controller, serving the 128 Kbytes of static RAM as a data buffer and read cache for the Micro, R/W, and DSSI areas. Most of the digital R/W tasks, such as ECC generation and checking, are also done by this chip.
Spindle	A single chip (and associated active and passive components) manages the spin-up of the disk stack and keeps the disks spinning at 3600 rpm.
Actuator	A single chip (and associated active and passive components) controls the rotary positioner used to track the position of the heads over a track.
R/W	These three chips (and associated active and passive components) handle the remaining digital and analog read/write chain.

[1]MC68000 is a trademark of Motorola, Inc.

Drive Module Block Diagram

Figure 2–3 shows how the drive module is laid out. Figure 2–4 is a block diagram of the same module.

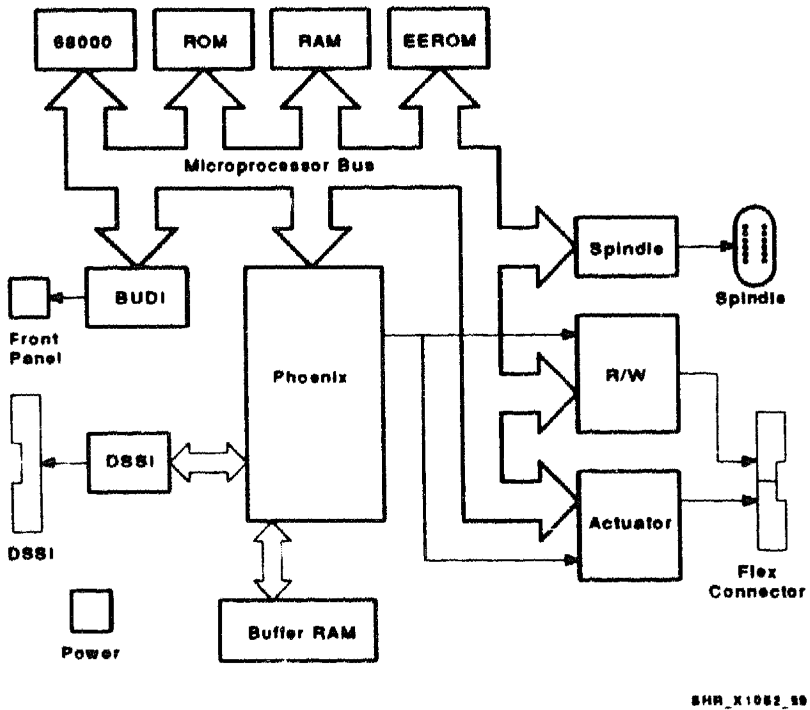


Figure 2–4 Drive Module Block Diagram

Data Flow

As Figure 2–4 shows, almost every functional area is connected to the microprocessor bus. However, the microprocessor is not directly involved with the data flow. Data is segregated or collected in the DSSI area, temporarily stored in the Buffer area, and I/O is performed by the Phoenix chip using the R/W hardware. Figure 2–5 diagrams the data flow.

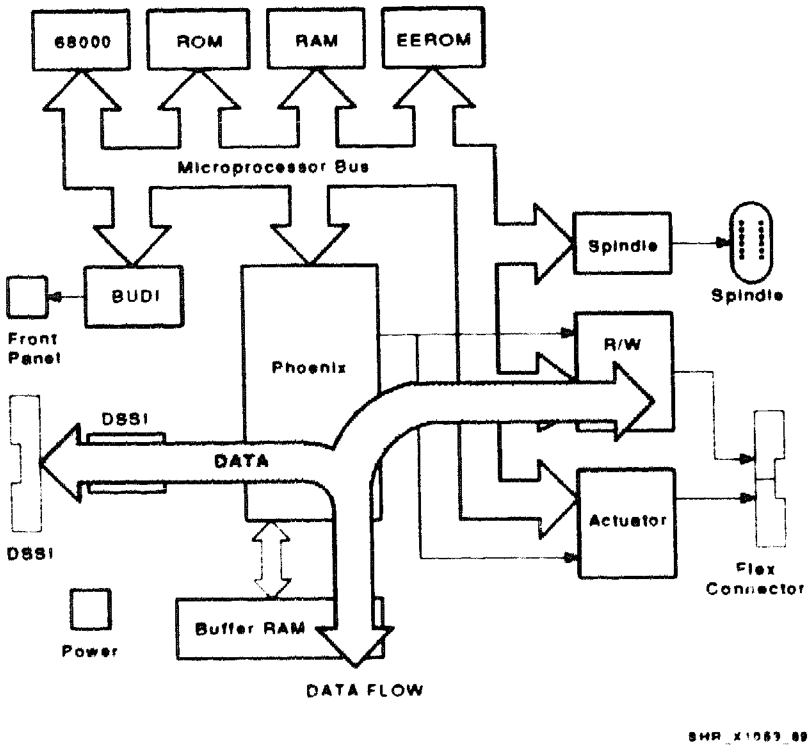


Figure 2–5 Data Flow Through the Drive Module

The data is protected while it travels through the drive module. The Swift chip in the DSSI area generates and checks DSSI parity and error detection code (EDC). To protect the data in buffer RAM, another EDC is generated and appended to each 512-byte block of data. This EDC travels with the data through the Phoenix chip, where the 264-bit ECC is generated before the data is written to the media. To further protect against media-addressing errors, the block number where the data was written is "added" to the Swift-generated EDC.

The ECC is checked as data is being read from the media. At the same time, the block number where the data should be located is "subtracted" from the EDC. The EDC is then checked after the DSSI parity and EDC are generated by the DSSI area, as the data is transmitted onto the DSSI bus.

Command Flow

Commands are segregated into packets by the DSSI area and stored in the buffer RAM. The microprocessor then copies these commands (typically 36 bytes or less) into its local RAM before it acts on them. As commands are completed, the process is reversed. Figure 2–6 shows how commands are routed through the drive module.

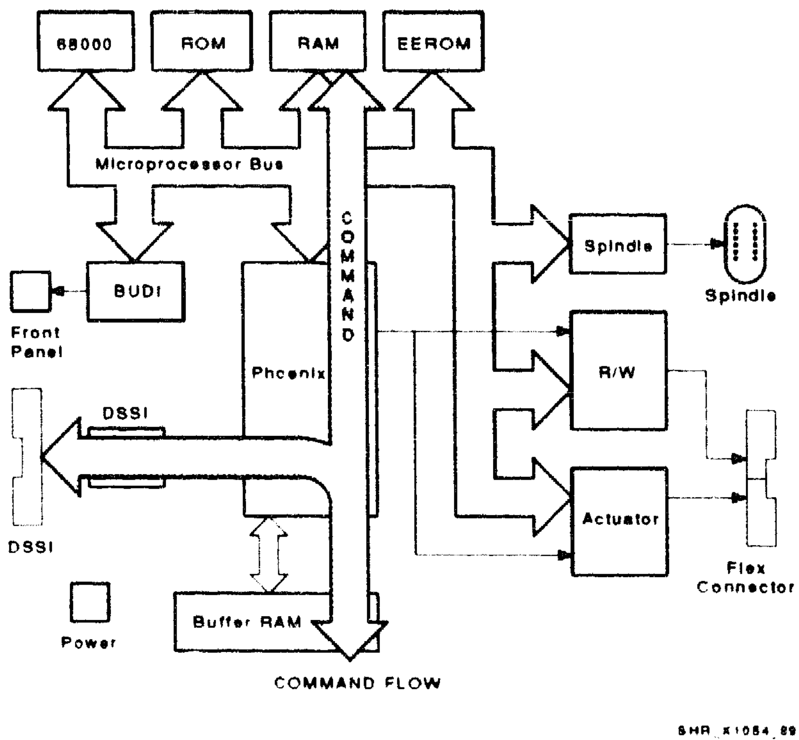


Figure 2–6 Command Flow Through the Drive Module

Commands are also protected as they travel through the drive module. On the DSSI bus, parity and EDC are appended to the commands. The Swift chip in the DSSI area checks and generates DSSI parity and EDC. To protect the commands during their short stay in the buffer RAM, the Swift chip generates an EDC for them. The microprocessor then copies the commands to its parity-protected scratchpad RAM. After the copy is complete, the microprocessor compares the EDC calculated by the BUDI chip in the Micro area with the EDC generated by Swift.

When commands are transmitted out to the DSSI bus (such as the end packet for an I/O operation), the process is reversed. The EDC is generated by the BUDI chip when the microprocessor copies the command to the buffer RAM. This EDC is checked by the Swift chip after it generates the DSSI bus parity and EDC, as the command is transmitted onto the DSSI bus.

Firmware

Firmware Structure

The RF31 ISE is called an integrated storage element because the drive, controller, and even some host functions are all integrated into the same package. Figure 2–7 illustrates the firmware structure that makes this integration possible.

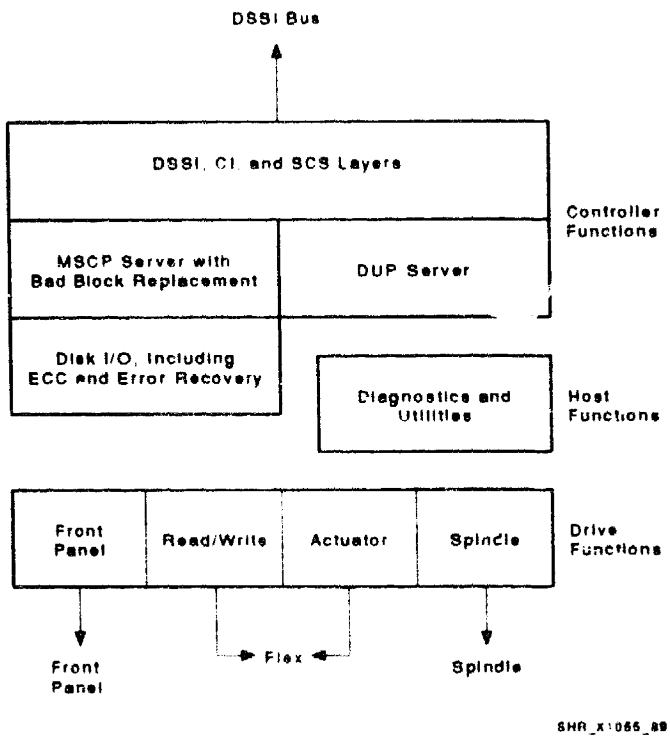


Figure 2–7 Firmware Structure

Several aspects of the RF31 firmware structure are not shown in Figure 2–7. For example, a mini-operating system runs the multiple programs in the ISE. Most of the firmware is coded in C, and a C run-time library is included. At initialization, power-on self-tests check the various chips on the module, start the spin-up sequence, and so on. Periodic diagnostics and calibrations adjust the servo and R/W systems for changing conditions.

STATUS THREADS Command

The programs that run in the ISEs are displayed when you access the PARAMS local program and type the STATUS THREADS command. The following is an example of such a display.

```
PARAMS>STATUS THREADS
 Pid  TCB Addr  Process Name   WQ Addr   State Pri      CPU
  0  FFFFB52C  NULL            00000000  RDY  255 0 00:06:49.46
  1  FF840C06  PERIODICS       FFFF9F74  BLK  254 0 00:00:16.10
 65  FF8414A4  PRFMON          FFFFB57A  IDL  130 0 00:00:00.00
 67  FF841544  DRVEXR          FF8415F4  IDL  124 0 00:00:00.00
 68  FF841594  DRVTST          FF841544  IDL  121 0 00:00:00.00
 69  FF8415E4  HISTRY          FF841594  IDL  118 0 00:00:00.00
 70  FF841634  DIRECT          FF8415E4  IDL  115 0 00:00:00.00
 71  FF841684  ERASE           FF841634  IDL  112 0 00:00:00.00
 72  FF8416D4  VERIFY          FF841684  IDL  109 0 00:00:00.00
 73  FF841724  DKUTIL          FF8416D4  IDL  106 0 00:00:00.00
 74  FF841774  PARAMS          FFFFB582  CUR  103 0 00:00:03.54
  2  FF841814  MSCP$DUP        FFFFA3E6  TIM   28 0 00:00:01.17
  3  FF8420C8  SCS$DIRECTORY   FFFFA516  BLK   25 0 00:00:00.07
  5  FF843.AA  MSCP$BBR        FFFFA3CA  BLK   19 0 00:00:00.00
  6  FF8439FA  PPT$DISK        FFFF8B7A  BLK   16 0 00:00:01.37
  7  FF84424A  SCS$DSSI        FFFF9FDE  BLK   13 0 00:00:02.09
  8  FF8488CE  MSCP$DISK       FFFFA17A  BLK   10 0 00:00:00.00
```

The threads listed in the above example correspond to the firmware layers shown in Figure 2–7. For instance, the DSSI, CI, and SCS layers are implemented by the SCS$DSSI thread. The SCS$DSSI thread interfaces the DSSI hardware to the system applications (SYSAPs) that run in the ISE. For each application, SCS$DSSI provides communication services to the other nodes on the bus. The two important SYSAPs are directly beneath the SCS$DSSI: the disk MSCP server and the diagnostic and utility protocol (DUP) server.

MSCP Server

The MSCP server is implemented by the MSCP$DISK thread, and the bad block replacement layer is implemented by the MSCP$BBR thread. The MSCP server is a multihost server because it can maintain a command dialogue with multiple hosts at the same time. Figure 2–8 illustrates a multihost server.

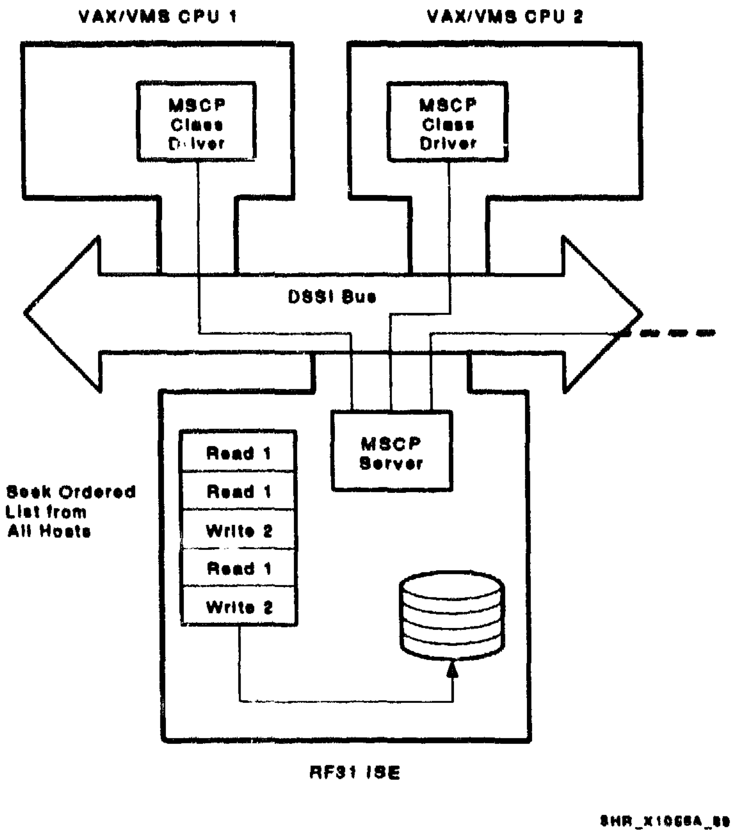


Figure 2–8 Multihost Server

Seek Ordering

I/O commands received from all hosts are combined to form a seek-ordered list of commands. This list determines the order of command execution in a way that minimizes the amount of time the drives spend seeking. Since seek time is typically the largest component of the total access time, seek ordering improves overall performance.

Request Fragmentation

To minimize the effects of rotational latency, the disk I/O firmware (implemented by the PPT$DISK thread) uses request fragmentation. Request fragmentation works by cutting up a long transfer into smaller pieces, and then completing the smaller pieces optimally.

Assume, for example, that a full track read is issued to an RF31 ISE. Typically, as in Figure 2–9, the 50-sector request is broken up into seven fragments, the first six fragments having eight sectors, and the last fragment having only two sectors. Each fragment is delivered to the host as it is read, so that the apparent rotational latency is only .07 revolutions, as opposed to the expected .50 revolutions.

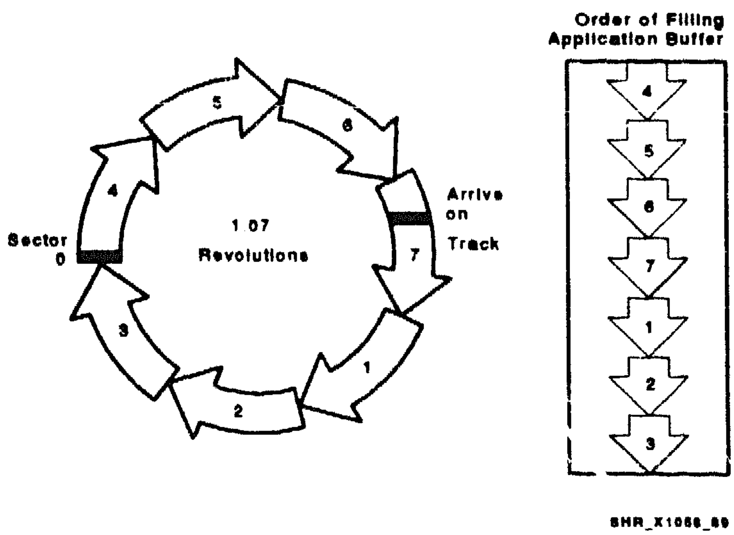


Figure 2–9 Request Fragmentation

Read Cache

To further reduce latency, the disk I/O firmware manages a four-track read cache. After a request is satisfied, and until a new command pre-empts, the sectors on the current track are read into one of the track lines in the cache. Write requests try to disturb the cache as little as possible by using the least recently used (LRU) track line(s).

RCT Caching

The MSCP server caches the entire replacement control table (RCT). The RCT contains the list of bad blocks and the replacement block numbers (RBNs) that hold the data for each bad block in the list. In the past, tertiary replacements (those made to an RBN not on the track of the defective LBN) caused the controller to seek to the RCT to locate the RBN. With RCT caching, the performance impact of tertiary revectors is greatly reduced, as shown in Figure 2–10.

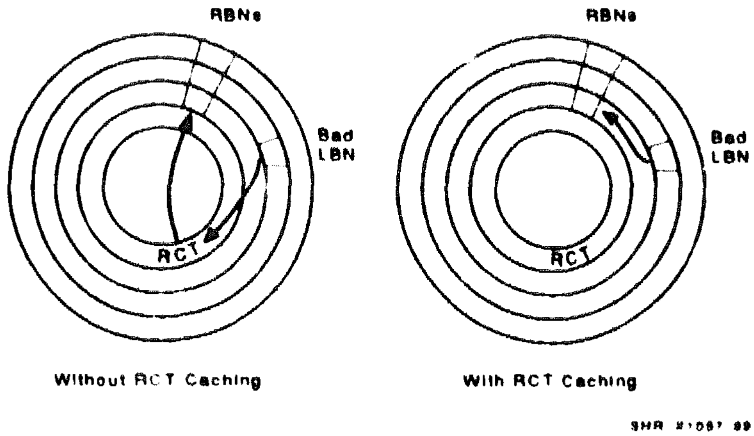


Figure 2–10 RCT Is Cached in Controller RAM

Sector Format and Error Handling

Other functions implemented by PPT$DISK include error recovery and ECC correction. The format of each disk sector (Figure 2–11) help explain error handling.

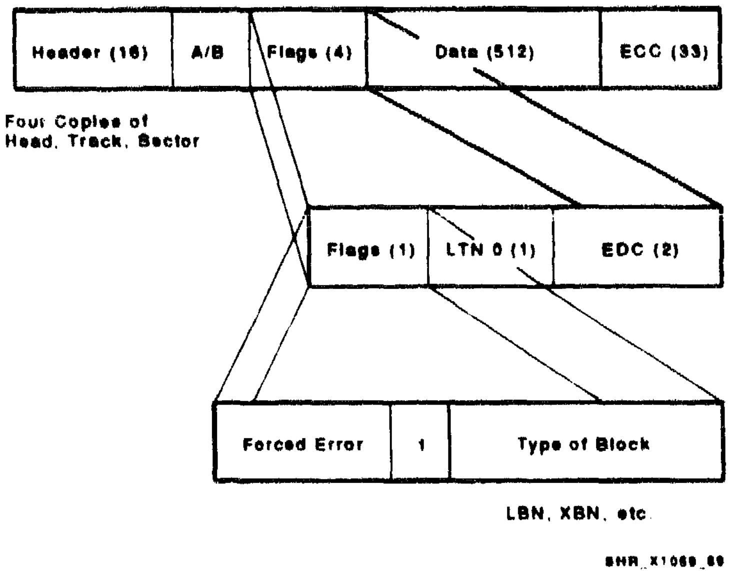


Figure 2–11 Sector Format

Each sector header has four copies of its address (head, track, and sector number). PPT$DISK instructs the Phoenix chip to match all four copies prior to a read or write operation. If the full header compare fails on a write operation, the block is replaced. If the full header compare fails on a read operation, one of the levels of error recovery reduces the number of copies that must match to three, then to two, and finally just one. Once the data is recovered, the block is replaced. Note that the header is read only and represents the physical address of the block.

The read/write portion of the block starts with the four-byte Flags field. It contains the customer data in 512 bytes, and ends with a 33-byte ECC field. ECC is one of the error recovery mechanisms, and can correct up to 15 bytes in a single sector. However, a much lower error threshold (number of bytes in error in the sector) causes the block to be replaced to ensure that plenty of correction capability remains in the ECC.

The Flags field is expanded in Figure 2–11 to include a flags byte, an LTN0 byte, and a word of EDC. The value in the EDC field is what is generated by the Swift chip when the data arrives over the DSSI bus with the block address "added" to it.

The flags byte contains the forced error indicator and the type of block. The forced error indicator is a bad data mark. It means that the data is logically "bad" even though the sector it resides in may be defect-free. The type of block (LBN, RBN, DBN, and so on) is also checked on read operations.

The LTN0 field is used to compensate for head and track scatter, as shown in Figure 2–12. On an RF31 ISE, the heads on the head stack are not aligned above one another. In fact, the tracks themselves are not aligned above one another. This lack of vertical alignment is not a product defect. At 1,875 tracks per inch, the distance between track centers is only 5-1/3 ten thousandths of an inch (0.53×10^{-3} in).

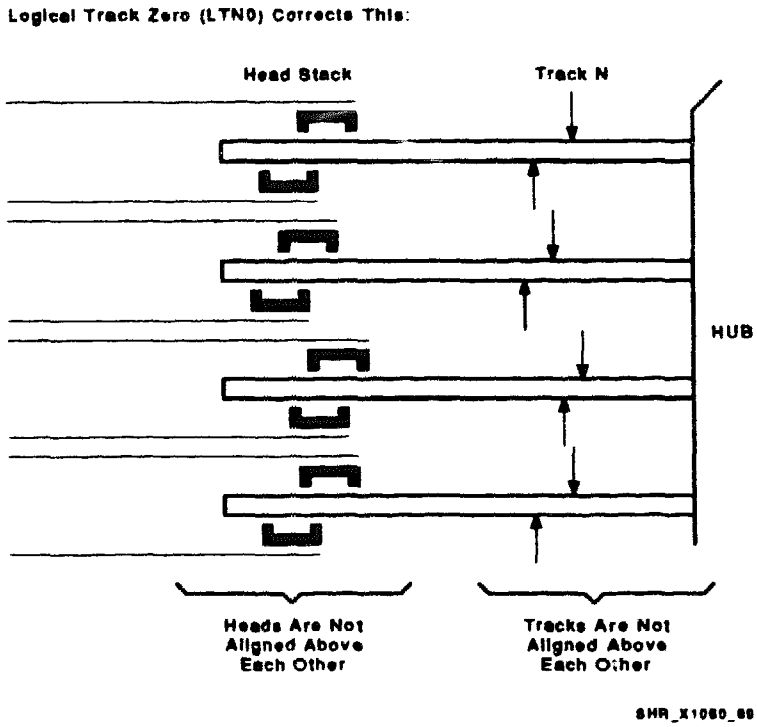


Figure 2–12 Head and Track Scatter Compensation

To improve performance and retain the concept of a cylinder (a vertically aligned set of tracks), the value in the LTN0 byte is the physical track number of the head that is directly above or below the LTN0 track for all other heads. This LTN0 track is the reference point for each surface (Figure 2–13).

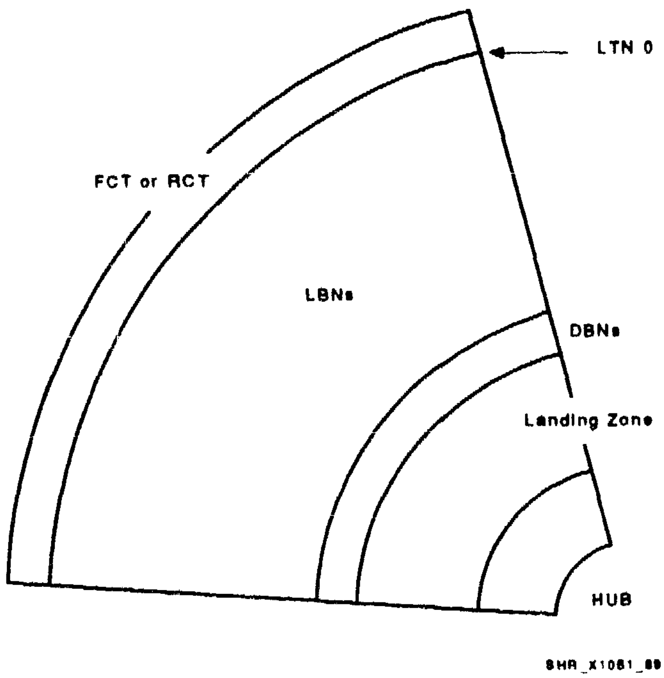


Figure 2–13 On-Disk Format Anchored at LTN0 on Each Surface

Diagnostics and Utilities

Introduction

The rectangle labeled "Diagnostics and Utilities" in Figure 2–7 represents those user-accessible programs that run under the diagnostic and utility protocol (DUP) server. These programs are listed later in this section, and are described in greater detail in Chapter 4.

POST

The other diagnostic program that runs at system initialization is called power-on self-test (POST). The POST firmware tests the hardware on the drive module and in the HDA, and achieves almost complete coverage. The only hardware not verified by the POST is a small part of the DSSI area (shaded part of Figure 2–14).

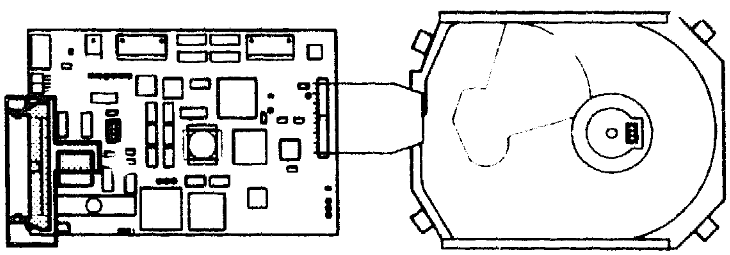


Figure 2–14 POST Coverage

POST, and the diagnostics and calibrations that run after head load, verify:

- That each head in the HDA works
- That the drive can seek across all tracks
- That each head can read and write correctly

Once POST has run successfully, you need only access the drive over the DSSI bus to achieve full diagnostic coverage.

After POST runs, the diagnostic tasks are picked up by the PERIODICS thread and by the DUP server (MSCP$DUP thread). The PERIODICS thread invokes POST diagnostics and runs calibrations on the drive periodically. This process is automatic, running as the lowest priority thread in the system.

DUP Server

You can think of the DUP server as a window into the ISE-resident diagnostic and utility programs. Like the MSCP server, the DUP server is a multihost server, allowing several diagnostic and utility programs to run in the ISE at the same time (Figure 2–15).

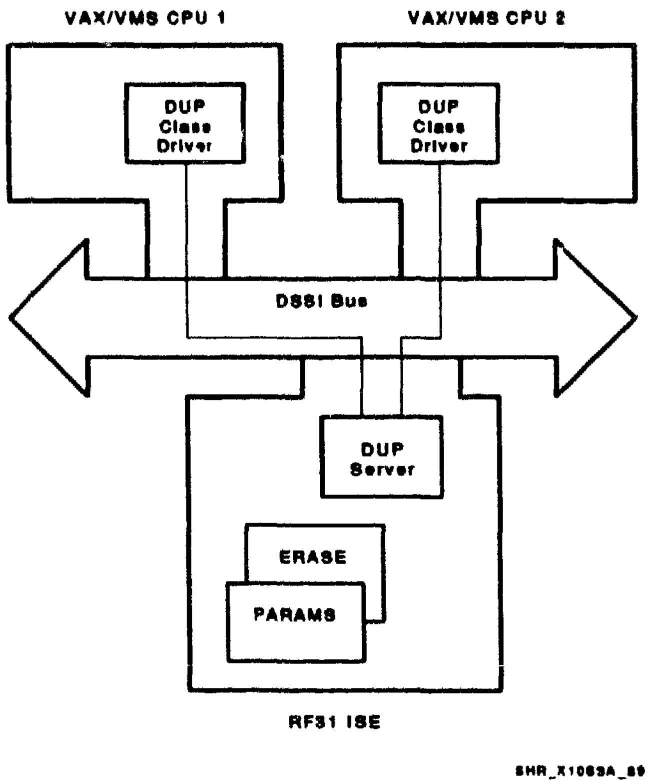


Figure 2–15 Multihost DUP Server

Diagnostic and Utility Programs

Nine diagnostic and utility programs are resident in the RF31 ISE. They can be broken down into three categories: management utilities, diagnostics, and system-level utilities. Figure 2–16 shows how the programs are grouped into these categories.

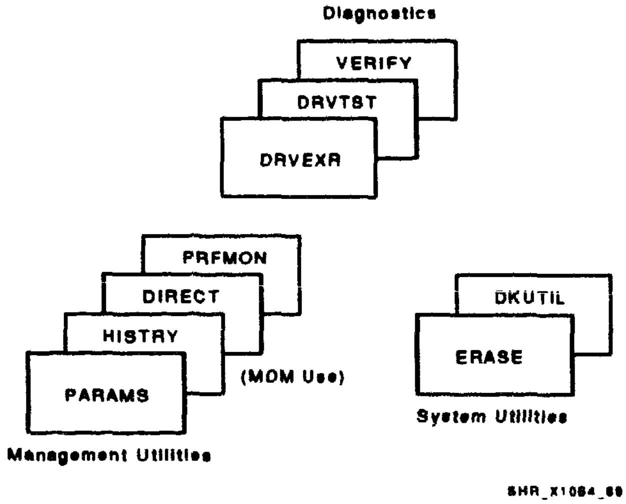


Figure 2–16 Diagnostics and Utilities

The following table provides a brief description of each program. See Chapter 4 for more detailed information.

Category	Program	Description
Management utilities	PARAMS	A SYSGEN-like parameter editor and interactive query utility used to view ISE and DSSI bus status.
	HISTRY	An abbreviated version of a portion of PARAMS. HISTRY is used by host-level software such as MDM.
	DIRECT	A directory program, output only, that lists the available diagnostic and utility programs.
	PRFMON	An abbreviated version of a portion of PARAMS. PRFMON is used by host-level software such as the VAX performance analyzer (VPA).

Category	Program	Description
Diagnostics	DRVEXR	A comprehensive drive exerciser.
	DRVTST	A quick pass/fail test of the drive.
	VERIFY	A complete read check of the disk and verification of the DSDF[1] on-disk structure.
System-level utilities	ERASE	A data "scrubber" utility that writes alternating patterns to each LBN, including the second sector of the RCT and the previous locations of the bad blocks, and then verifies that the scrubbing process succeeded.
	DKUTIL	An interactive block display and replace utility. Typically used to confirm ERASE scrubbing.

[1]DEC Standard Device Format specification

Detecting Errors

The RF31 ISE makes errors visible in a variety of ways. Knowing where to look for errors helps you to communicate with Customer Services personnel.

If the controller portion of the ISE is not functioning, the red fault LED on the drive module and the red fault LED on the operator control panel turn on (Figure 2–17). Also, the ISE is not visible on the DSSI bus. The failing component, in this case, is the drive module itself, because it houses the controller electronics.

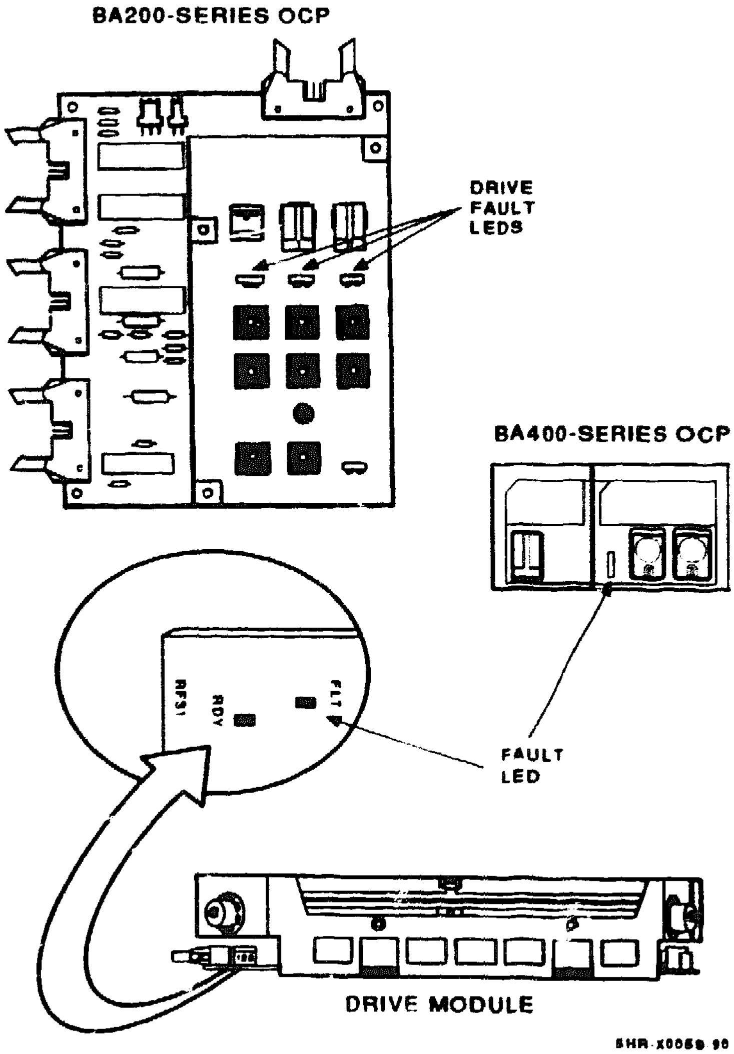


Figure 2–17 Location of the Fault LEDs

Detecting Errors Using PARAMS

If the controller portion of the ISE passes diagnostics but the unit portion does not, the red fault LED still turns on. However, (because the controller is operational), the ISE *is* visible on the DSSI bus and can be queried to further resolve the error condition. At this point, the fault may be within the unit electronics or the HDA.

To obtain a failure code, use the PARAMS utility program as shown in the following example.

```
PARAMS> STAT CONF
 Node R1WSRA is an RF31 controller
 Software RFX X200 built on 8-AUG-1989 03:38:51
 Electronics module name is EN83804656
 Unit is inoperative, error code 9802(X)
 Last known unit failure 9802(X)
 In 1275 power-on hours, power has cycled 767 times
 System time is 19-AUG-1989 18:13:33
```

In the example, the unit error code is 9802(X). This is also shown as the last known error (stored in nonvolatile RAM). These error codes are defined in the service guide. Your Customer Services representative uses them to determine the field replaceable unit (FRU) most likely to have failed.

The errors described above are usually fatal. That is, once the error occurs, the red fault LED remains lit and the ISE is not usable until the error condition is corrected.

Transient errors, however, can also occur. For example, a bit in RAM can flip, or an address collision can occur on the DSSI bus. The most recent 11 transient errors are kept in a log in nonvolatile RAM, which is accessible though the PARAMS utility program. To access this error history, use the STAT LOG command as shown in the following example.

```
PARAMS> STAT LOG
Log History:

Log #000-2B-73-07/0000 21-AUG-1989 13:22:04
FFFF8450 FFFFB354 4E554E55 4E554E55 4E554E55 4E554E55 4E554E55 0000AA01
FF80C1B0 FF80C1B0 000029F6 4E554E55 FF817D5E FFFF83BE 4E554E55 FFFF96BC
5CABF380 009239C1 00002EA0
```

The primary users of the error history are module repair personnel. The error history may also serve as an indication to Customer Services personnel that the system is configured incorrectly or that the drive module should be replaced.

Detecting Soft Errors

Soft errors are errors that are caused by external environmental factors (for instance, temperature or shock) or errors that are expected to occur throughout the lifetime of the product (such as an infrequent bad block replacement). These errors are reported in MSCP error logs.

MSCP error logs are displayed by the error log report generators of many different operating systems. These logs are analyzed by the VAXsimPLUS program to predict failures.

The RF31 ISE produces three kinds of error logs, as summarized in the following table.

MSCP Error Log	Reason Generated
Read/Write Error Log	Any failure to read or write (above certain thresholds).
Bad Block Replacement Summary	Any time a read/write error log is generated, this log summarizes the attempt to replace the "bad" block.
Servo Performance Warning	If, during the past hour, the servo system suffered more than its threshold of performance hits.

Certain soft errors are not reported in logs. For example, a single symbol (byte) error on a read is corrected, but not reported. The threshold for reporting an error (and replacing the block) in the case of an ECC error on a read command is three. This means that until three or more symbols are corrected in a block, the block is not replaced. Three symbols in error is well within the safety margin of the ECC. The ECC can correct up to 15 symbols in error in a single block.

CHAPTER 3

3 Controls and Indicators

Introduction

This chapter describes the controls and indicators associated with RF31 ISE operation. The controls and indicators are located in two places:

- Operator control panel on the system enclosure
- ISE drive module

OCP

The operator control panel (OCP) is a set of controls on the enclosure that overrides the drive module switches and enables the operator to set the DSSI node ID and write-protect mode for the ISE. The OCP also contains a fault LED that indicates if the ISE is malfunctioning.

For BA200-series systems, the OCP contains three identical sets of controls and indicators, one set for each ISE that may be connected to it. In the BA400-series enclosures, each ISE has an individual OCP which is mounted to a panel in front of the ISE.

BA200-Series OCP

The BA200-series enclosures each have a standard DSSI operator control panel. Figure 3–1 shows the location of the controls and indicators on the operator control panel.

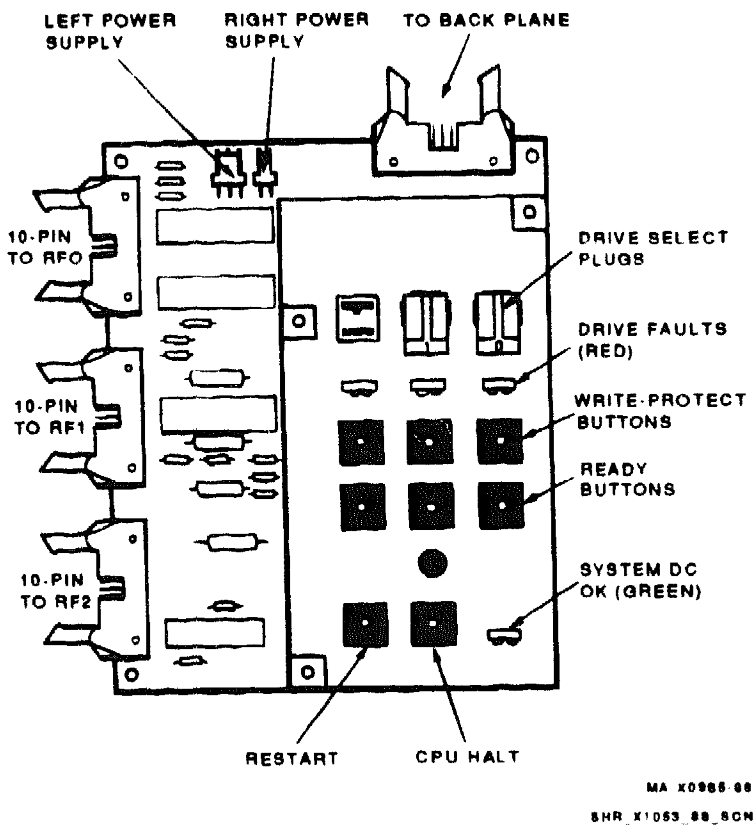


Figure 3–1 BA200-Series DSSI Operator Control Panel

The following table describes the function of the controls on the BA200-Series DSSI operator control panel.

Control	Status	Function
Drive Select Plug	Installed	The DSSI node ID is set to the number specified on the plug. A drive select plug must be installed for every ISE connected to the OCP.
	Removed	The DSSI node ID is undefined. This is a fault condition that causes the drive fault LED to flash at 10 Hz.
Write-Protect Button	Out	The system can read and write to the media (normal operating position).
	In	Write-locked, the system can only read from the media.
Ready Button	Out	ISE is on-line (normal operating position).
	In	ISE is off-line.

The following table describes the function of the indicators on the BA200-series DSSI operator control panel.

Indicator	Status	Function
Fault LED	On	Fault condition present (except during POST).
	Off	No fault present (normal operating condition).
	Slow flash (5 Hz)	Internal ISE calibrations are being performed.
	Quick flash (10 Hz)	OCP failure, or drive select plug is missing.
Write-Protect LED	On	Write-protect enabled.
	Off	Write-protect disabled.
Ready LED	On	ISE is on-line and ready.
	Off	ISE is off-line.

BA400-Series OCP

Each ISE installed in a BA400-series enclosure has a front panel with the following controls and indicators.

- DSSI bus node ID plug
- Fault LED
- Write-protect button
- Run/Ready button

Figure 3–2 is an illustration of the front panel of a BA400-series enclosure.

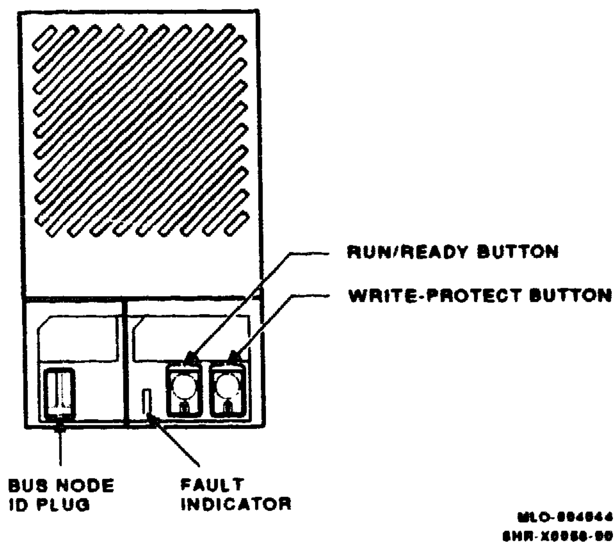


Figure 3–2 BA400-Series OCP for DSSI ISEs

The following table describes the function of the controls and indicators on the BA400-series DSSI operator control panel.

Control	Status	Function
Fault LED	Lit	Indicates an error condition within the ISE.
	Not lit	Indicates an error-free condition within the ISE.
Run/Ready Button	In (lit)	The ISE is on-line. When the ISE is available for use, the green LED is lit. When the ISE is being used, the green LED flickers.
	Out (not lit)	The ISE is off-line and cannot be accessed. The green LED cannot be lit when the Run/Ready button is out.
Write-Protect Button	In (lit)	The ISE is write-protected. System software cannot write to the ISE.
	Out (not lit)	The ISE is not write-protected. This is the normal position for software operation. System software is free to read or write to the ISE.

Changing the DSSI Node ID Plugs

Spare DSSI node ID plugs are supplied with your system. Use these spare plugs to renumber your DSSI system should you need to reconfigure due to adding or removing ISEs, or if you create a multihost configuration.

The DSSI node ID plugs have prongs on the back that indicate the bus node number (and by default, the unit number) of the ISE. To remove a DSSI node ID plug, grasp it firmly and pull it straight out. To insert a new plug, align the two center prongs with the two center slots and press the plug into the slots.

Use the following rules to renumber your ISEs:

- For each DSSI bus, do not use duplicate DSSI node IDs.
- By convention, ISEs are numbered in increasing order from right to left.
- Use a blank DSSI node ID plug where no ISE is present.

NOTE
If you change the bus node ID plugs while the system is operating, you must turn off the system and then turn it back on for the new plug positions to take effect.

Drive Module Controls and Indicators

A DIP switchpack containing three switches and two LEDs is mounted on the edge of the drive module. The switches provide a means of setting the DSSI node ID if an OCP is not connected to the drive, or if the OCP fails. The two LEDs indicate drive status (ready and fault).

Figure 3–3 shows the location of these switches and LEDs on the drive module.

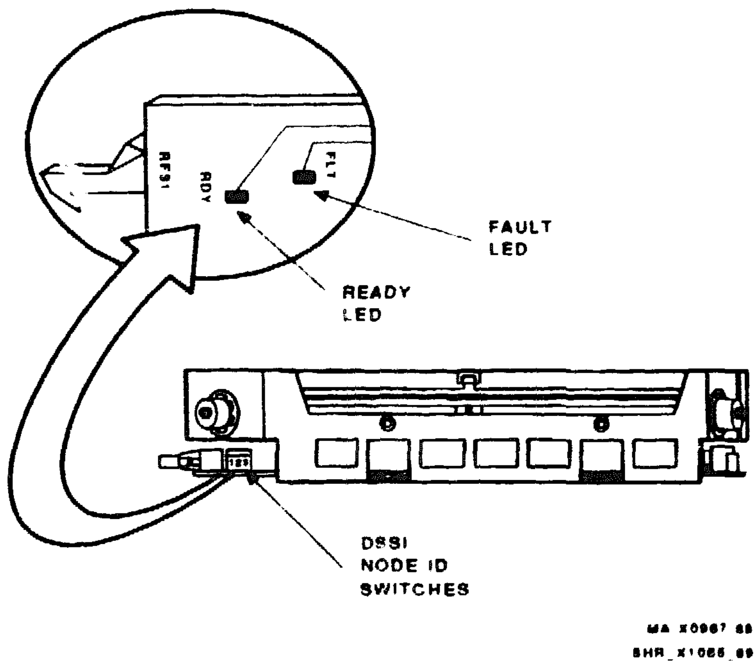


Figure 3–3 Drive Module Switch and LED Locations

Assigning DSSI Node ID

Assignment of the DSSI node ID is done by setting the three switches to the binary equivalent of the selected ID number, as shown in the following table. These switches are ignored when an operator control panel is connected to the ISE.

DSSI Node ID	Switch Positions[1]		
Address	1	2	3
0	Down	Down	Down
1	Down	Down	Up
2	Down	Up	Down
3	Down	Up	Up
4	Up	Down	Down
5	Up	Down	Up
6	Up	Up	Down
7[2]	Up	Up	Up

[1]Up is toward the HDA, down is toward the module.

[2]DSSI address 7 is normally assigned to a host adapter.

Drive Module LEDs

The two LEDs mounted on the drive module monitor ISE status during operation. The following table describes the state of these two LEDs during the various phases of ISE operation.

When . . .	The green LED is . . .	And the red LED is . . .
The ISE is powered up	On	On
POST has run successfully	Off	Off
The read/write heads are on-cylinder and ready	On	Off
Drive activity	Flickering	Off
A read/write or serious physical error is detected	Off	On

CHAPTER 4

4 Local Programs

Accessing Local Programs

Overview

Local programs are diagnostics and utilities that are internal to the ISE. They are accessed in one of three ways, depending on the system you're using:

- Through VMS, using the SET HOST command
- From the console, using the SET HOST command
- Through MDM, using the Device Resident Programs menu

Once a connection is established, operations are performed under the control of the local program. When the program ends, control is returned to the system.

Using VMS

To access a local program from a MicroVAX system running VMS version 5.1 or greater, the command is:

```
$ SET HOST/DUP/SERVER=MSCP$DUP/TASK=taskname nodename
```

Where:

taskname = name of the local program
nodename = node name of the ISE

Descriptions of the available local programs are given later in this chapter. To find the node name of an ISE, type SHOW DEVICES or SHOW CLUSTER at the $ prompt and press [Return].

To produce a file in your directory of what appears on the screen, add the qualifier /log=filename.ext (where filename.ext is what you want to name the file) before you press [Return].

Using Console Commands

MicroVAX 4000 series systems allow you to access the local programs using console commands. The command you use depends on whether your system uses a Q-bus adapter like the KFQSA module, or an embedded adapter such as the KA640 module.

Q-bus Adapters

To access a local program from a system with a Q-bus adapter, the command is:

```
>>> SET HOST/UQSSP/DUP/DISK # taskname
```

Where:

# = port number of the ISE
taskname = name of the local program

Embedded Adapters

To access a local program from a system with an embedded adapter, the command is:

```
>>> SET HOST/DUP/DSSI/BUS:n #
```

Where:

n = the bus number where the ISE is located
# = DSSI node number of the ISE

Then, the system prompts you for the name of the local program you want to run.

To find the DSSI node number and node name, type SHOW DSSI at the >>> prompt. To see a list of the devices on the Q-bus, type SHOW QBUS or SHOW UQSSP at the >>> prompt.

To abort the program and return control to the system, press Ctrl/C or Ctrl/Y.

Using MDM

If neither VMS nor console commands are available on your system, you can run local programs using MDM. Use the following procedure:

1. Boot MDM.
2. Enter the date and time.
3. Select the menus in the following order:
   - Service menu
   - Device menu
   - KFQSAA-KFQSA subsystem menu
   - Device Utilities menu
   - Device Resident Programs menu

   When you select the Device Resident Programs menu, the following is displayed:

```
RUNNING A UTILITY SERVICE TEST

To halt the test at any time and return to the previous menu,
type CTRL-C by holding down the CTRL key and pressing the C key.

KFQSAA started.
KFQSAA pass 1 test number 3 started.
Copyright 1988 Digital Equipment Corporation
Completed.
EXIT                     DRVEXR                  DRVTST
HISTRY                   ERASE                   PARAMS
DIRECT                   DKUTIL                  PRFMON
VERIFY

Please choose a local program or press <RETURN> to continue.
```

4. Type in the name of the local program you want to run and press [Return]. For information about the available local programs, refer to the program descriptions on the following pages.
5. To exit MDM, press the [Break] key.

DIRECT

Description

DIRECT provides a directory of local programs resident in the ISE.

Example

The following is an example of what is displayed when you run the DIRECT program.

```
Copyright © 1989 Digital Equipment Corporation
PRFMON     V1.0   D    21-AUG-1989   13:39:09
DRVEXR     V2.0   D    21-AUG-1989   13:39:09
DRVTST     V2.0   D    21-AUG-1989   13:39:09
HISTRY     V1.1   D    21-AUG-1989   13:39:09
DIRECT     V1.0   D    21-AUG-1989   13:39:09
ERASE      V2.0   D    21-AUG-1989   13:39:09
VERIFY     V1.0   D    21-AUG-1989   13:39:09
DKUTIL     V1.0   D    21-AUG-1989   13:39:09
PARAMS     V2.0   D    21-AUG-1989   13:39:09
Total of 9 programs.
```

DRVEXR

Description

DRVEXR is a diagnostic that applies several types of stress to the ISE.

DRVEXR is accessed in the same way as are the other local programs. Once a connection is established, the user is prompted to answer a series of questions. The responses determine the mode and test duration.

To stop DRVEXR in progress, press Ctrl/C, Ctrl/Y, or Ctrl/Z. When DRVEXR stops, a short report is printed.

Modes

DRVEXR can be run in one of the following modes, depending on your responses to the dialogue questions.

Mode	Function
Read/Write	Writes and reads as many blocks as possible in the given amount of time.
Data Integrity	Similar to Read/Write mode, but with a "manual" check of data buffers done by the ISE processor.
Seek Intensive	Only seeks are performed in this mode.
Max Stress	Reads the inner DBNs and outer DBNs alternately.

Dialogue

The following table describes the DRVEXR dialogue.

Message	Explanation
Copyright © 1989 Digital Equipment Corporation	No response is expected.
Write/read anywhere on the medium? [1=Yes/(0=No)]	This question asks if you want to write to the media.
User data will be corrupted. Proceed? [1=Yes/(0=No)] (This question is omitted if you typed 0 to answer the previous question.)	This question asks if you really want to overwrite existing data on the media.
Test time in minutes? [(10)-100]	Your response determines the length of the test, in minutes.

Message	Explanation
Number of sectors to transfer at a time? [0 - 50]	Your response determines the length of each I/O issued. If you type 0, this is a seek-only test.
Compare after each transfer? [1=Yes/(0=No)] (This question is omitted if you typed 0 to answer the previous question.)	Your response determines whether the processor "manually" compares the results of the read with the expected data (if writing is enabled) or the hardware does the compares after each read.
Test the DBN area? [2=DBN only/(1=DBN and LBN)/0=LBN only]:	Your response determines how to include the DBN area in the test. If you type 2, the test always includes writes, even if you answered the first question with 0.

The following table shows the relationship between the four test modes and the responses to the six questions in the dialogue.

Model	Response to Question					
	1	2	3	4	5	6
Read/Write	1	1	Any	Any	0	Any
Data Integrity	1	1	Any	Any	1	Any
Seek Intensive	0	N/A	Any	0	N/A	0 or 1
Max Stress	0	N/A	Any	0	N/A	2

Example

The following is an example of what is displayed when you run DRVEXR.

```
Copyright © 1989 Digital Equipment Corporation
Write/read anywhere on the medium? [1=Yes/(0=No)] 1
User data will be corrupted. Proceed? [1=Yes/(0=No)] 1
Test time in minutes? [(10)-100] 10
Number of sectors to transfer at a time? [0 - 50] 18
Compare after each transfer? [1=Yes/(0=No)]: 0
Test the DBN area? [2=DBN only/(1=DBN and LBN)/0=LBN only]:

    73990 blocks (512 bytes) read.
    73990 blocks (512 bytes) written.
     18666 DBN blocks (512 bytes) read.
     18666 DBN blocks (512 bytes) written.

Complete.
```

DRVTST

Description

DRVTST provides a comprehensive test of the ISE hardware. Errors detected by this program are isolated to the FRU level.

Dialogue

The following table describes the DRVTST dialogue.

Message	Explanation
Copyright © 1989 Digital Equipment Corporation	No response is expected.
Write/read anywhere on the medium? [1=Yes/(0=No)]	This question asks if you want to write to the media. If you type 0, this is a read-only test. DRVTST does, however, write to a diagnostic area on the disk.
User data will be corrupted. Proceed? [1=Yes/(0=No)]	This question asks if you really want to overwrite existing data on the media. If you type 0, this is a read-only test.
5 minutes to complete.	No response is expected.
Test passed.	This means the test was successful. Choose another local program or return control to the system.

Error Messages

The following table describes DRVTST error messages.

Message	Description
Unit is currently in use.	This can mean that the ISE unit is inoperative, in use by a host, or is currently running another local program.
Operation aborted by user.	This message appears if the user stops the program while it is in progress.
Soft read error on head xx track yyyy. Soft write error on head xx track yyyy. Soft compare error on head xx track yyyy.	These are soft error messages which indicate that an operation succeeded, but that the error recovery firmware was invoked. These messages may indicate a forced-error flag or correctable ECC error, or that the read/write head was temporarily off-track. These are corrected during normal operation.
xxxx - Unit diagnostics failed.	This is a fatal error. Call Digital Customer Services. xxxx is the MSCP error code.
xxxx - Unit read/write test failed.	This is also a fatal error. Call Digital Customer Services. xxxx is the MSCP error code.

Example

The following is an example of what is displayed when DRVTST runs successfully.

```
Copyright © 1989 Digital Equipment Corporation
Write/read anywhere on the medium? [1=Yes/(0=No)] 1
User data will be corrupted. Proceed? [1=Yes/(0=No)] 1
  5 minutes to complete.
Test passed.
```

HISTRY

Description

HISTRY displays ISE information that is used by programs running in the host (such as MDM).

The information is displayed in the following order:

copyright notice
product name
serial number
node name
allocation class
firmware revision level
hardware revision level
power-on hours
power cycles

Example

The following is an example of what is displayed when you run the HISTRY program.

```
Copyright © 1989 Digital Equipment Corporation
RF31
EN92300124
R1EJAA
0
RFX X201
RF31 PCB-3/ECO-00
1
45
Complete.
```

ERASE

Description

ERASE is a utility that overwrites data on RF31 ISEs. Every writable LBN and RBN block on the ISE is written several times with a complementary data pattern, and finally is overwritten with zeros.

ERASE writes every block in the LBN and RBN space twice the specified number of times: once with the pattern 99(X), and once with the complement pattern 66(X). These two values were selected because they are bit complements of each other in both decoded and encoded formats. That is, the bits toggle in controller memory and the fixed frequency pulse trains on the disk are 180 degrees out of phase with one another, assuming the pulse trains start together.

After the specified number of write/write complement operations have been performed, a final pattern of all zeros is written on each block. A failure to write these patterns or to read back all zeros means that the block is "bad" and was previously replaced. The failure and the status of each such block is displayed. If the block is marked bad in the FCT, the user may inhibit this output.

The write/write complement, write of zeros, and read back of zeros is repeated to the second block of each RCT copy. The second block of the RCT is used as an intermediate holding area for data during bad block replacement and is treated as an extension of the user data area by ERASE.

ERASE is accessed in the same way as the other local programs. Once a connection is established, the user is prompted to answer a series of questions.

To stop an ERASE in progress, press Ctrl/C, Ctrl/Y, or Ctrl/Z.

Dialogue

The following table explains the ERASE dialogue.

Message	Explanation
Copyright © 1989 Digital Equipment Corporation	No response is expected.
Write/read anywhere on medium? [1=Yes/(0=No)]	This question asks if you want to write to the media.
User data will be corrupted. Proceed? [1=Yes/(0=No)]	This question asks if you really want to overwrite existing data on the media.
How many times should the disk be pre-written before erasing? [0-99]	This question asks you to specify the number of passes ERASE should make on each sector. A pass is a write with a pattern followed by a write with the pattern complement. A message is displayed indicating your choice.

Message	Explanation
Display errors erasing known bad blocks? [(1=Yes)/0=No]:	This question asks if you want a list of all ERASE errors in the RCT and the FCT. A message is displayed indicating your choice.
Is this information correct? [(1=Yes)/0=No]:	If you type 0, you are asked the following question.
Do you wish to continue? [1=Yes/(0=No)]:	If you type 1, you are once again prompted to specify the number of passes, and so on. If you type 0, the session ends.
n minutes to complete.	This lists the number of minutes (n) until completion of the erase operation. This number is a function of the number of passes you specified and the ISE type. No response is expected.
Erase complete.	This message indicates that the erase operation is complete. The program stops automatically.

Example

The following is an example of the ERASE utility run on an RF31 ISE.

The blocks listed in this example are the blocks that either did not write, or did not contain an all-zeros pattern when ERASE read back each track on the LBN/RBN space. In most cases, the failing blocks are listed as "bad" in the FCT.

```
Copyright © 1989 Digital Equipment Corporation
Write/read anywhere on medium? [1=Yes/(0=No)] 1
User data will be corrupted.  Proceed? [1=Yes/(0=No)] 1
How many times should the disk pre-written before erasing? [1-99]:
Display errors erasing known bad blocks? [(1=Yes)/0=No]:
The disk will be pre-written 3 times.
All errors will be reported, including FCT bad blocks.
Is this information correct? [(1=Yes)/0=No]:

 58 minutes to complete.

        ERASE BAD BLOCK LIST

  LBN   Head  Track  Sector  Block Status
 -----  ----  -----  ------  ------------
 14602   4     132     19    Unable to write sector (marked bad in FCT)
 14603   4     132     20    Unable to read sector (marked bad in FCT)
 14604   4     132     21    Unable to write sector (marked bad in FCT)
 14605   4     132     22    Unable to write sector (not found in FCT)
629079   5    1661     12    Non-zero data in sector (marked bad in FCT)
629080   5    1661     13    Unable to write sector (marked bad in FCT)

Erase complete.
```

VERIFY

Description

VERIFY is a utility that is used to determine the amount of "margin" remaining in on-disk structures (for instance, the number of bad blocks in the RCT and thus the number of RBNs consumed). VERIFY is useful when you verify an installation.

The VERIFY utility is read only. It does not overwrite user data and does not perform bad block replacement.

VERIFY is accessed in the same way as are the other local programs. Once a connection is established, the user is prompted to answer a series of questions.

Dialogue

The following table describes the VERIFY dialogue.

Message	Explanation
Copyright © 1989 Digital Equipment Corporation	No response is expected.
Print informational (non-warning) messages? [1=Yes/(0=No)]	Your response controls the amount of output produced by VERIFY. Type 1 if you want informational messages displayed.
Report transient error by block? [1=Yes/(0=No)]	Again, your response controls the amount of output produced by VERIFY. By default, a transient, correctable error in a block is not displayed as blocks are read. Type 1 if you want transient errors reported.

Example

The following is an example of VERIFY run on an RF31 ISE. The example is interrupted several times by text that describes the next part of the example.

The first part of this example contains the VERIFY dialogue and the FCT header dump. The FCT header dump provides basic information, such as the serial number and the date of last format (typically the date the HDA was manufactured).

```
Copyright © 1989 Digital Equipment Corporation
Print informational (non-warning) messages? [1=Yes/(0=No)] 1
Report transient error by block? [1=Yes/(0=No)] 1

***    FCT Block 0 Information    ***

   Serial Number:         0000123400000000
   Mode:                  ADDE(X)
   First Formatted:       22-MAY-1989 11:58:26
   Date Formatted:        22-MAY-1989 11:58:26
   Format Instance:       1
   FCT:                   VALID
   Bad PBNs in FCT:       34

   Scratch Area Offset:  30032
   Size (Not Last):       2480
   Size (Last):           2480

   Flags:                 0000(X)
   Format Version:        1
```

This next part of the example contains the RCT header dump. Immediately following is a read check of the RCT. Note that one sector of one copy of the RCT (the 5th copy, block 68) is not usable.

```
***     RCT Block 0 Information     ***

   Serial Number:          0000123400000000
   Flags:                  0000(X)

   LBN Being Replaced:       0
   Replacement RBN:          0
   Bad RBN:                  0

*****     Revector Control Table for RlVJAA$DIA3     *****
Copy 5 of RCT block 68 (LBN 745103.) is bad.
```

After the read check of the RCT, each bad block in the RCT is displayed. In this abbreviated syntax, the symbol *-> means that a bad LBN is replaced by an RBN not on its track. The symbol --> means that the bad LBN is replaced by its primary (on-track) RBN.

```
      2 -->       0,   2083 -->     41,   3569 -->     71,
   9566 *->     190,   9564 -->    191,   9563 *->    192,
  14603 -->     292,  15361 -->    307,  15398 *->    308,
  18762 -->     375,  19403 -->    388,  21198 -->    423,
  22695 -->     453,  22788 *->    454,  22791 -->    455,
  22790 *->     457,  22789 *->    458,  23183 *->    462,
  23182 *->     464,  28081 -->    561,  31026 *->    619,
  31027 *->     621,  33589 -->    671,  35495 -->    709,
  45046 -->     900,  45339 -->    906,  48484 -->    969,
  57875 -->    1157,  63818 -->   1276,  70125 -->   1402,
    :             :      :             :      :           :
 (many lines have been deleted from this example)
    :             :      :             :      :           :
Bad RBN:      12041, 602527 -->  12050, Bad RBN:      12073,
609475 -->    12189, Bad RBN:    12217, 613875 -->    12277,
645331 -->    12906, 645330 *->  12907, 645885 -->    12917,
652795 -->    13055, 656745 -->  13134, 657589 -->    13151,
664108 -->    13282, 675509 -->  13510, 679607 -->    13592,
693585 -->    13871, 694130 -->  13882, 694129 *->    13883,
707101 -->    14142, 707100 *->  14143, 711501 -->    14230,
711607 -->    14232, 717484 -->  14349, 721878 -->    14437,
722237 -->    14444, 723439 -->  14468, 729597 -->    14591,
732140 -->    14642, 732372 -->  14647, 735723 -->    14714,
737467 -->    14749, 739349 -->  14786, 742125 -->    14842,
```

The next part of this example is a summary of the contents and state of the RCT.

```
RCT Statistics:
                          47 Bad RBNs
                         185 Bad LBNs
                         140 Primary Revectors
                          45 Non-Primary Revectors
                           1 Bad RCT Blocks
                           0 Bad First Copy RCT Blocks
```

Then a similar scan of the FCT follows. Each bad block in the FCT is displayed. Entries in this display are physical blocks. The syntax DKUTIL uses for PBNs is also used here: h:t:s:, where h is the head number, t is the track number, and s is the sector number. If the bad block is used by the disk format, the corresponding block number is displayed in parentheses.

```
*****    Factory Control Table for R1VJAA$DIA3    *****

1:  96:50(LBN   2083) 1: 161:48(LBN  28081) 1: 181: 0(LBN  36084)
1: 212: 0(LBN  48484) 2: 259: 8(LBN  70125) 1: 374: 0(LBN 113284)
3: 452:23(LBN 142973) 1: 558: 0(LBN 186884) 5: 564:17(LBN 188684)
5: 565:17(LBN 189084) 3: 630:38(LBN 214188) 3: 634:16(LBN 215766)
2: 673:43(LBN 235709) 6: 822: 0(LBN 293100) 3: 938:25(LBN 337375)
1:1009:40(LBN 367273) 5:1160:30(LBN 427097) 7:1208: 1(LBN 447185)
5:1321: 8(LBN 491475) 3:1440: 8(LBN 538158) 3:1442: 8(LBN 538958)
5:1449: 8(LBN 542675) 3:1454: 8(LBN 543758) 3:1459: 8(LBN 545758)
6:1471:40(LBN 552740) 5:1532: 8(LBN 575875) 5:1546: 8(LBN 581475)
5:1559: 8(LBN 586675) 0:1556:20(LBN 587620) 5:1616: 8(LBN 609475)
5:1627: 8(LBN 613875) 6:1731:45(LBN 656745) 6:1778: 9(LBN 675509)
0:1786: 7(LBN 679607)
```

The state of the FCT is displayed next.

```
FCT Statistics:
                          0 Bad FCT Blocks
                          0 Bad First Copy FCT Blocks
```

In the last part of this example, the LBN area is scanned. Each LBN block is read, some LBN statistics are displayed, and VERIFY ends. In this example, transient error reporting was enabled so the transients are displayed.

```
*****    Scan of LBN Area    *****

LBN   2740. has a transient (5 out of 6) error.
LBN   7904. has a transient (3 out of 6) error.
LBN  77100. has a transient (4 out of 6) error.
LBN  95831. has a transient (1 out of 6) error.
LBN 577712. has a transient (1 out of 6) error.
LBN 730484. has a transient (1 out of 6) error.

LBN Statistics:            6 Block(s) with a Transient error.
                           6 Total Block(s) in error.
Complete.
```

Error Messages

The following table describes the VERIFY error messages.

Message	Description
00C5 - Unit diagnostics failed.	The ISE unit diagnostics failed and VERIFY cannot run.
xxxx - Multicopy read operation of xCT block n failed.	Block n of a multicopy structure, the FCT, or the RCT, is not readable. xxxx is the MSCP error code.
Copy n of RCT block m (LBN x) is bad.	Copy n of RCT block m, corresponding to LBN block x, is bad.

Message	Description
xBN n has a forced error.	While scanning the LBN, DBN, or XBN space, block n was marked with a forced error flag.
xBN n has an invalid header error.	While scanning the LBN, DBN, or XBN space, block n had a problem with its header.
xBN n has a data sync timeout error.	While scanning the LBN, DBN, or XBN space, block n had problems syncing up to the data bits.
xBN n has an ECC field correctable error.	While scanning the LBN, DBN, or XBN space, block n suffered an error, but only in the ECC field.
xBN n has an uncorrectable ECC error.	While scanning the LBN, DBN, or XBN space, block n was unreadable due to an uncorrectable ECC error.
xBN n received error status : xxxx(X).	While scanning the LBN, DBN, or XBN space, block n has some other error. xxxx is the MSCP error code.

Warning Messages

The following table describes the VERIFY warning messages.

Message	Description
xCT block n used copy m.	The first copy of block n of a multicopy structure, the FCT, or the RCT, is not readable. Copy m of that block was used instead.
xBN n has a transient (e out of six) error.	While scanning the LBN, DBN, or XBN space, block n suffered a transient error. Out of six reads, the transient error happened e times.
xBN n has a k symbol correctable ECC error.	While scanning the LBN, DBN, or XBN space, block n suffered a correctable ECC error of k bytes.
Table is empty (NO BAD PBNs).	This message is very rare, and indicates that no blocks are marked bad in the FCT. The media was perfect when the ISE unit was formatted.

DKUTIL

Description

DKUTIL is a utility that displays disk structures and disk data. Its primary intent is to verify that an ERASE worked correctly. DKUTIL is also useful in problem diagnosis and manual bad block replacement.

DKUTIL is accessed in the same way as are the other local programs. Once a connection is established, all interaction occurs through the use of commands and responses. DKUTIL has its own command line interpreter. It has a command syntax similar to the hierarchical storage controller (HSC) DKUTIL utility.

DKUTIL prompts the user for a command with the DKUTIL> prompt. In most cases, the user issues a GET command to acquire an ISE unit to which subsequent commands are applied. DKUTIL acquires the unit and then returns to the command mode, prompting for a command, executing it, and prompting for another command.

To stop DKUTIL, press Ctrl/C, Ctrl/Y, Ctrl/Z, or type EXIT at the DKUTIL> prompt.

DKUTIL Commands

The following table lists the available DKUTIL commands. You do not need to type the entire command because the program is set up to recognize the matching command from the abbreviated command. The abbreviated command is shown in uppercase letters.

Command	Definition
DEfault	Changes the default modifiers for subsequent DKUTIL commands.
DIsplay	Displays characteristics, error history, and RCT/FCT information.
DUmp	Dumps the contents of a sector on the disk or the contents of the current buffer.
Exit	Stops the DKUTIL utility.
Get	Acquires the ISE unit from the MSCP server.
Help	Displays information on how to use the DKUTIL commands.
POp	Restores the save buffer to the current buffer.
PUsh	Stores the current buffer in the save buffer.
REplace	Replaces a specified LBN.
Set size	Changes the block size of the ISE unit.

DKUTIL Command Modifiers

A number of modifiers are used with the DKUTIL GET, DISPLAY, and DUMP commands. These modifiers can be typed with the command or they can be stored using the DEFAULT command. The following tables describe the modifiers and their functions.

GET Modifier	Default Setting	Function
IMF	/NOIMF	Ignore media format. If this modifier is specified, the unit is acquired for physical (PBN) access only.
WP	/WP	Write protect. If this modifier is specified, the media is write protected. This inhibits replacements.

DISPLAY Modifier	Default Setting	Function
FULL	/NOFULL	Display full. If this modifier is specified, a full dump of the RCT or FCT header is displayed.
ITEMS	/NOITEMS	Display items. If this modifier is specified, the items in the RCT or FCT are displayed.

DUMP Modifier	Default Setting	Function
HEADER	/HEADER	Display header. If this modifier is specified, the block type and LTN0 fields of the block are displayed.
EDC	/EDC	Display the EDC. If this modifier is specified, the EDC and the expected EDC of the block are displayed.
DATA	/DATA	Display data. If this modifier is specified, the data in the block is displayed.
ECC	/NOECC	Display ECC. If this modifier is specified, the ECC residue of the block is displayed.
ALL	/NOALL	Display all fields. Use of this modifier is equivalent to specifying the modifiers /HEADER/EDC/DATA/ECC.
RAW	/NORAW	Display the raw LBN. If this modifier is specified, the original LBN that was replaced is used instead of the RBN. This modifier affects only the replaced LBNs.
NZ	/NONZ	Display nonzero data. If this modifier is specified, the data field is displayed if it contains all nonzero data. This modifier has no effect if /NODATA is also specified.

DEFAULT

The DEFAULT command changes the default modifiers for subsequent DKUTIL commands. This command "remembers" modifiers so they do not have to be typed each time a command is issued. Modifiers that do not apply to a command are ignored when that command is issued. So, you can specify modifiers that apply to GET, DUMP, and DISPLAY at the same time.

DEFAULT Syntax

The syntax for the DEFAULT command is:

DEFAULT [*modifiers*]

The available *modifiers* are described in the tables on the preceding two pages.

DEFAULT Example

The following example shows how to access the DKUTIL program on an ISE with the unit name R1EJAA.

Once DKUTIL is accessed, the DEFAULT command is issued to set new default values for subsequent commands. GET will apply /NOWP; DUMP will apply /NZ; and DISPLAY will apply /FULL and /ITEMS.

Note that both the DEFAULT command and the /ITEMS modifier are abbreviated.

```
$ SET HOST/DUP/SERVER=MSCP$DUP/TASK=DKUTIL R1EJAA
%HSCPAD-I-LOCPROGEXE, Local program executing
Copyright © 1989 Digital Equipment Corporation

DKUTIL> DE/NOWP/NZ/FULL/ITE
```

DISPLAY

The DISPLAY command displays characteristics, error history, and RCT/FCT information.

DISPLAY Syntax

There are three basic variations to the DISPLAY command:

- DISPLAY CHARACTERISTICS DISK

  This displays the characteristics of the ISE unit.

- DISPLAY [/FULL][/ITEMS] {RCT,FCT}

  This displays the header of the RCT or the FCT. The /FULL modifier controls the amount of RCT or FCT header information displayed. If the /ITEMS modifier is specified, the contents of the RCT or FCT is also displayed.

- DISPLAY ALL

  This is equivalent to typing DISPLAY RCT, DISPLAY FCT, and DISPLAY CHARACTERISTICS DISK.

DISPLAY Modifiers

The following table describes the modifiers used with the DISPLAY command.

Modifier	Default	Function
FULL	/NOFULL	Display full. If this modifier is specified, a full dump of the RCT or FCT header is displayed.
ITEMS	/NOITEMS	Display items. If this modifier is specified, the items in the RCT or FCT are displayed.

DISPLAY Example

The following example shows how the DISPLAY command lists the ISE unit characteristics.

```
DKUTIL> DISP CHA DIS

    Drive characteristics for drive R1EJAA$DIA1

  Type:          RF31 (512 byte mode only)
  Media:         Fixed
  Cylinders:     1861 LBN, 5 XBN, 5 DBN
  Geometry:      1 tracks/group, 8 groups/cylinder,
                 8 tracks/cylinder, 50 LBNs/track,
                 1 RBNs/track, 51 sectors/track,
                 51 XBNs/track, 408 XBNs/cylinder,
                 400 LBNs/cylinder, 8 RBNs/cylinder
  Group Offset:  17 (LBN), 0 (XBN)
  LBNs:          744400 (host), 745416 (total)
  RBNs:          14888
  XBNs:          2040
  DBNs:          1632 (read/write), 408 (read only)
  PBNs:          838032
  RCT:           127 (size), 8 (copies)
  FCT:           127 (size), 8 (copies)
  Retry Limits:  8 (retry), 3 (badblk)
  ECC Threshold:3/2 (normal), 0/0 (format)
  Revision:      4 (hardware), 32 (firmware)
  Drive ID:      124923040310000
```

DUMP

The DUMP command displays the contents of a sector on the disk or the contents of the current buffer. The amount of information displayed can be controlled with modifiers.

DUMP Syntax

There are four basic variations to the DUMP command:

- DUMP BUFFER

  This dumps the contents of the current buffer.

- DUMP {DBN,LBN,RBN,XBN} *block#*

  This dumps the contents of the specified *block#*. The LBN space extends into the RCT, and the XBN space spans all copies of the FCT. These methods of reading the RCT and FCT allow a specific copy to be read.

- DUMP {RCT,FCT} *block#*

  This dumps the contents of the specified *block#*. The block is read using the multiread algorithm. This command also shows you which copy was used to obtain the information.

- DUMP PBN *head#*:*track#*:*sector#*

  This dumps the contents of the specified block by giving the physical location of the sector not adjusted by LTN0.

DUMP Modifiers

The following table describes the modifiers used with the DUMP command.

Modifier	Default	Function
HEADER	/HEADER	Display header. If this modifier is specified, the block type and LTN0 fields of the block are displayed.
EDC	/EDC	Display the EDC. If this modifier is specified, the EDC and the expected EDC of the block are displayed.
DATA	/DATA	Display data. If this modifier is specified, the data in the block is displayed.
ECC	/NOECC	Display ECC. If this modifier is specified, the ECC residue of the block is displayed.
ALL	/NOALL	Display all fields. Use of this modifier is equivalent to specifying the modifiers /HEADER/EDC/DATA/ECC.
RAW	/NORAW	Display the raw LBN. If this modifier is specified, the original LBN that was replaced is used instead of the RBN. This modifier affects only the replaced LBNs.
NZ	/NONZ	Display nonzero data. If this modifier is specified, the data field is displayed if it contains all nonzero data. This modifier has no effect if /NODATA is also specified.

DUMP Example

The following example shows how the DUMP command displays the contents of a block of the RCT.

Note that the first copy of this RCT block was used. Also note that LBN 0 is replaced by RBN 0.

```
DKUTIL> DUMP RCT 2

*** Buffer for RCT block 2 (LBN 744402 used), MSCP Status: 0000

    LTN 0 =   90         Type = 47 (X)
    EDC = 032F(X)        Calculated EDC Difference = 14B3(X)

 Data = 3000 0000 0000 0000 0000 0000 0000 0000
 + 16   0000 0000 0000 0000 0000 0000 0000 0000
 + 32   0000 0000 0000 0000 0000 0000 0000 0000
 + 48   0000 0000 0000 0000 0000 0000 0000 0000
 + 64   0000 0000 0000 0000 0000 0000 0000 0000
 + 80   0000 0000 0000 0000 0000 0000 0000 0000
 + 96   0000 0000 0000 0000 0000 0000 0000 0000
 +112   0000 0000 0000 0000 0000 0000 0000 0000
 +128   0000 0000 0000 0000 0000 0000 0000 0000
 +144   0000 0000 0000 0000 0000 0000 0000 0000
 +160   0000 0000 0000 0000 0000 0000 0000 0000
 +176   0000 0000 0000 0000 0000 0000 0000 0000
 +192   0000 0000 0000 0000 0000 0000 0000 0000
 +208   0000 0000 0000 0000 0000 0000 0000 0000
 +224   0000 0000 0000 0000 0000 0000 0000 0000
 +240   0000 0000 0000 0000 0000 0000 0000 0000
 +256   0000 0000 0000 0000 0000 0000 0000 0000
 +272   0000 0000 0000 0000 0000 0000 0000 0000
 +288   0000 0000 0000 0000 0000 0000 0000 0000
 +304   0000 0000 0000 0000 0000 0000 0000 0000
 +320   0000 0000 0000 0000 0000 0000 0000 0000
 +336   0000 0000 0000 0000 0000 0000 0000 0000
 +352   0000 0000 0000 0000 0000 0000 0000 0000
 +368   0000 0000 0000 0000 0000 0000 0000 0000
 +384   0000 0000 0000 0000 0000 0000 0000 0000
 +400   0000 0000 0000 0000 0000 0000 0000 0000
 +416   0000 0000 0000 0000 0000 0000 0000 0000
 +432   0000 0000 0000 0000 0000 0000 0000 0000
 +448   0000 0000 0000 0000 0000 0000 0000 0000
 +464   0000 0000 0000 0000 0000 0000 0000 0000
 +480   0000 0000 0000 0000 0000 0000 0000 0000
 +496   0000 0000 0000 0000 0000 0000 0000 0000
```

EXIT

The EXIT command stops the DKUTIL utility. If the ISE unit was acquired, the unit is released to the MSCP server.

EXIT Syntax

The syntax for this command is:

EXIT

EXIT Example

The following is an example of what appears on the screen when you EXIT DKUTIL.

```
DKUTIL> EXIT

%HSCPAD-S-REMPGMEND, Remote program terminated - message code 3.
%HSCPAD-S-END, Control returned to node ISMINE
```

GET

The GET command acquires the ISE unit from the MSCP server (if possible). Most DKUTIL commands work only after the GET command succeeds.

The GET command cannot succeed if the ISE unit is on-line to any host.

GET Syntax

The syntax for the GET command is:

GET [*modifier*]

The specified *modifier* controls how the unit is acquired.

GET Modifiers

The following table describes the modifiers used with the GET command.

Modifier	Default	Function
IMF	/NOIMF	Ignore media format. If this modifier is specified, the unit is acquired for physical (PBN) access only.
WP	/WP	Write protect. If this modifier is specified, the media is write protected. This inhibits replacements.

GET Example

The following example shows the outcome of a successful GET command. The unit is acquired from the MSCP server and the unit FCT status is displayed. In this example, /NOWP was previously set by the DEFAULT command and /NOIMF was the default upon entry to DKUTIL.

```
DKUTIL> GET
   Serial Number:          0000000000000000
   Mode:                   ADDE(X)
   First Formatted:         9-AUG-1989 13:25:10
   Date Formatted:          9-AUG-1989 13:25:10
   Format Instance:        1
   FCT:                    VALID
```

HELP

The HELP command displays information on how to use the DKUTIL commands.

HELP Syntax

The syntax for the HELP command is:

HELP

HELP Example

Typing HELP produces the following display.

```
DKUTIL> HELP
   DEFAULT [/qualifiers]
   DISPLAY [/qualifiers] {ALL|FCT|RCT|CHAR DISK}
      /FULL       /ITEMS
   DUMP [/qualifiers] {BUFFER|type n|PBN h:t:s}
      /HEADER     /EDC         /DATA
      /NZ         /ECC         /ALL
      /RAW
   EXIT
   GET [/qualifiers]
      /IMF        /WP
   HELP
   POP
   PUSH
   REPLACE lbn
   SET SIZE 512

DKUTIL>
```

POP

The POP command restores the save buffer to the current buffer.

DKUTIL maintains two I/O buffers while it runs: the current buffer and the save buffer. The DUMP command always uses the current buffer. Use the PUSH command to save data before subsequent DUMPs. Use the POP command, followed by the DUMP BUFFER command, to view the saved data.

POP Syntax

The syntax for this command is:

POP

POP Example

In this example, the contents of the buffer saved by the most recent PUSH command is restored to the current buffer.

```
DKUTIL> POP
```

PUSH

The PUSH command stores the current buffer in the save buffer.

DKUTIL maintains two I/O buffers while it runs: the current buffer and the save buffer. The DUMP command always uses the current buffer. Use the PUSH command to save data before subsequent DUMPs. Use the POP command, followed by the DUMP BUFFER command, to view the saved data.

PUSH Syntax

The syntax for this command is:

PUSH

PUSH Example

In this example, the contents of the buffer loaded by the most recent DUMP command is stored to the save buffer.

```
DKUTIL> PUSH
```

REPLACE

The REPLACE command replaces a specified LBN. The results of the REPLACE command are displayed. The display contains the last few longwords of a Bad Block Replacement Summary error log packet.

REPLACE cannot succeed if the unit is write protected (GET/WP), or if the unit was acquired for physical (PBN) access only (GET/IMF).

REPLACE Syntax

The syntax for the REPLACE command is:

REPLACE *lbn#*

The specified *lbn#* is replaced.

REPLACE Example

In the following example, LBN0 is replaced.

```
DKUTIL> REPLACE 0
Replace summary log packet:
     Flags: 60(X)
     Event: 0014(X)
 Rep flags: 8000(X)
       LBN: 0
   Old RBN: 0
   New RBN: 0
```

SET SIZE

The SET SIZE command changes the block size of the ISE unit. Only 512-byte block sizes are supported. This command is for compatibility with the HSC implementation of DKUTIL.

SET SIZE Syntax

The syntax for the SET SIZE command is:

SET SIZE *blocksize#*

Presently, the only acceptable *blocksize#* is 512.

SET SIZE Example

This example shows the block size of the unit set to 512 bytes.

```
DKUTIL> S S 512
```

PARAMS

Description

PARAMS is a utility that allows you to examine and edit internal ISE parameters such as node name, allocation class, and MSCP unit number. PARAMS is also used to display the state of the ISE and performance statistics maintained by the ISE.

PARAMS is accessed in the same way as the other local programs. Once a connection is established, all interaction occurs through the use of commands and responses. PARAMS has its own command line interpreter. It has a command syntax similar to the VMS SYSGEN utility.

PARAMS prompts the user for a command with the PARAMS> prompt. Once a command is entered, PARAMS executes it and then prompts for another command.

To stop PARAMS, press Ctrl/C, Ctrl/Y, Ctrl/Z, or type EXIT at the PARAMS> prompt.

PARAMS Commands

The following table lists the available PARAMS commands. You do not need to type the entire command because the program is set up to recognize the matching command from the abbreviated command. The abbreviated command is shown in uppercase letters.

Command	Definition
Exit	Stops the PARAMS utility.
Help	Displays information on how to use PARAMS commands.
Locate	Causes a soft fault in the ISE to help you locate it.
SEt	Changes internal ISE parameters.
SHow	Displays the setting of a parameter or a class of parameters.
STatus	Displays module configuration, history, performance counters, and so on.
Write	Records the device parameter changes you make using the SET command in nonvolatile RAM.
Zero	Clears a block of counters or all known blocks of counters.

EXIT

The EXIT command stops the PARAMS utility.

EXIT Syntax

The syntax for the EXIT command is:

EXIT

EXIT Example

The following is an example of what appears on the screen when you **EXIT PARAMS**.

```
PARAMS> EXIT
Exiting ...

%HSCPAD-S-REMPGMEND, Remote program terminated
%HSCPAD-S-END, Control returned to node ISMINE
```

HELP

The HELP command displays information on how to use PARAMS commands. It is not a substitute for this document, but is useful as a quick reference.

HELP Syntax

The syntax for the HELP command is:

HELP

HELP Example

The following is what appears on the screen when you type HELP.

```
PARAMS> help
  ENABLE MSCP
  EXIT
  HELP
  LOCATE
  SET {parameter | .} value
  SHOW {parameter | . | /class}
    /ALL      /CONST    /DRIVE
    /SERVO    /SCS      /MSCP
    /DUP
  STATUS [type]
    CONFIG    LOGS      DATALINK
    PATHS     SYSTEM    SEEKS
    MSCP      THREADS
  WRITE
  ZERO counter
    ALL       MSCP      SEEKS

PARAMS>
```

LOCATE

The LOCATE command causes a soft fault in the ISE to help you locate it. The soft fault does not affect its current operation or state. The red LED on the drive module and the red LED on the operator control panel turn on and stay on until you press Return at the PARAMS> prompt.

LOCATE Syntax

The syntax for the LOCATE command is:

LOCATE

LOCATE Example

The following is what appears on the screen when you type LOCATE.

```
PARAMS> LOCATE
Drive has been soft faulted to help locate it
Press RETURN to continue:
PARAMS>
```

SET

The SET command changes internal ISE parameters. The changes are made to a working copy of the ISE parameters and do not take effect until a WRITE command is successful.

SET Syntax

The syntax for the SET command is:

SET *parameter value*

The *parameter* is the name of the parameter to be set, or the special symbol . (period), which means the last SET or SHOWn parameter. The *value* is the new value you want to assign to the parameter.

The available parameters are listed on the following page. If the parameter name is abbreviated, the first matching parameter is used without regard to uniqueness. We recommend that you use SHOW before you use SET.

SET Example

The following example changes the node name from the default (R1EJAA) to the new string "SUSAN". When entering ASCII strings, you may use single quotes, double quotes, or no quotes at all.

```
PARAMS> SHOW NODE
Parameter      Current          Default       Type     Radix
---------   -------------   ------------   -------   -----
NODENAME          R1EJAA           RF31     String   Ascii    B

PARAMS> SET . "SUSAN"
PARAMS> SHOW .
Parameter      Current          Default       Type     Radix
---------   -------------   ------------   -------   -----
NODENAME          SUSAN            RF31     String   Ascii    B

PARAMS>
```

SET and SHOW Parameters

The following tables describe the SET and SHOW command parameters and their functions.

MSCP Parameter	Function
ALLCLASS	Sets or shows the allocation class of the disk MSCP server.
UNITNUM	Sets or shows the MSCP unit number.
FORCEUNI	Determines whether the parameter UNITNUM or the DSSI node ID is used as the MSCP unit number. The factory setting for this parameter is 1 (true), and the DSSI node ID is used. Until set to 0 (false), the UNITNUM parameter is ignored.
HISPEED	The factory setting for this parameter is 0 (false). If set to 1 (true), only half of the ISE capacity is presented to hosts. This reduces the stroke of the seek, and thus improves performance at the expense of capacity. Average seek time is reduced by approximately 3 ms.
SEEKALG	This parameter selects one of three seek algorithms. The factory setting is the C-scan algorithm (2). You can also select the shortest distance algorithm (0), or the elevator algorithm (1).
FIVEDIME	This parameter determines whether the MSCP server should support five concurrent connections (the factory default of true, or 1) or seven connections with reduced resources allocated to each (false, or 0).

DRIVE Parameter	Function
VOLSERNO	Shows the volume serial number of the HDA.

DUP Parameter	Function
ADD_CR	Appends a [Return] character after each message. The factory setting is 0 (false).
ADD_LF	Appends a [Line Feed] to each message. The factory setting is 0 (false).

SCS Parameter	Function
SYSTEMID	Sets or shows the 48-bit SCS system ID of the ISE.
NODENAME	Sets or shows the SCS node name of the ISE.
FORCENAM	Determines whether the value set by the NODENAME parameter or the string RF31x is returned as the SCS node name. The letter x corresponds to the DSSI node ID (A = 0, B = 1, and so on). The factory setting is 0 (false), and the value set by NODENAME is used.

SHOW

The SHOW command displays the setting of a parameter or a class of parameters. The full name of the parameter, its current value, default value, radix and type, and any flags associated with the parameter are displayed.

SHOW Syntax

The syntax for the SHOW command is:

SHOW *parameter_or_class*

The *parameter_or_class* is a parameter name, the special symbol . (period), which means the last SET or SHOWn parameter, or a SHOW class preceded by a / (slash).

If an abbreviation is used, the first matching parameter or class is displayed.

SHOW Classes

The following table describes the available SHOW classes.

Class	Description
/ALL	All SHOW classes are displayed.
/CONST	ISE constants are grouped in this class.
/DRIVE	Drive parameters, typically those governing drive calibrations, thresholds, and retries, are grouped in this class.
/SERVO	Servo parameters, typically the results of drive calibrations, are grouped in this class.
/SCS	Systems communication services (SCS) parameters are grouped in this class. The node name and system ID are in this class.
/MSCP	Mass Storage Control Protocol (MSCP) parameters are grouped in this class. The parameters that control the unit number are in this class.
/DUP	Dia_ ostic and utility protocol (DUP) parameters are grouped in this class.

SHOW Example

The following example shows various forms of the SHOW command and how they are used to change the MSCP unit number from the default (DSSI bus ID) to a new value of 8402.

```
$ SET HOST/DUP/SERVER=MSCP$DUP/TASK=PARAMS R1EJAA
%HSCPAD-I-LOCPROGEXE, Local program executing
Copyright © 1989 Digital Equipment Corporation

PARAMS> SHOW FORCEUNI
Parameter      Current          Default          Type     Radix
---------  ---------------  ---------------  -------  -----
FORCUNI                  1                1  Boolean    0/1   U

PARAMS> SET . 0
PARAMS> SHOW UNITN
Parameter      Current          Default          Type     Radix
---------  ---------------  ---------------  -------  -----
UNITNUM                  0                0  Boolean    Dec   U

PARAMS> SET . 8402
PARAMS> SHOW /MSCP
Parameter      Current          Default          Type     Radix
---------  ---------------  ---------------  -------  -----
MSCPNVR    2020202020202020 2020202020202020
           2020202020202020 2020202020202020
           2020202020202020 2020202020202020
           2020202020202020 2020202020202020   String   Hex   RO
WRTLOGSZ                  2                2     Word   Dec   RO
UNITID     1B02403192300124    0000000000000 Quadword   Hex   RO
ALLCLASS                  0                0     Byte   Dec   B
MEDIAID            2264601F         2264601F Longword   Hex   RO U
HISPEED                   0                0  Boolean   0/1   U
UNITNUM                8402                0     Word   Dec   U
FORCEUNI                  0                1  Boolean   0/1   U
FORCEID                   1                1  Boolean   0/1   RO
FIVEDIME                  1                1  Boolean   0/1   B
SEEKALG                   2                2     Byte   Dec
CNT_TMO                  60               60     Word   Dec   RO

PARAMS>
```

STATUS

The STATUS command displays a variety of information including module configuration, transient and unit error history, performance counters, and so on. The STATUS type you select determines the type of information displayed.

STATUS Syntax

The syntax for the STATUS command is:

STATUS [*type*]

The *type* is an option that corresponds to a type of information. If omitted, all available types of status information are displayed.

If the *type* is abbreviated, all matching types are displayed.

STATUS Types

The following table describes the available STATUS types.

Type	Function
CONFIG	Displays module configuration information and some system information such as power-on hours and the system time.
LOGS	Displays the last 11 transient errors recorded in nonvolatile RAM. In VMS terms, these are the last 11 machine checks and bug checks known to the controller.
DATALINK	Displays the DSSI data link counters. These include the number of packets received and transmitted, the number of NAKs sent and received, and so on.
PATHS	Displays the topology of the DSSI bus on which this ISE resides by displaying open virtual circuits to other nodes on the bus and related packet counters.
SYSTEM	Displays system statistics and counters.
SEEKS	Displays seek statistics such as average seek times and a seek length histogram.
MSCP	Displays MSCP statistics such as the number and type of I/O commands and various histograms.
THREADS	Displays the known programs in the system, the amount of CPU time consumed, and other information.

STATUS Examples

The following examples show the use of three STATUS commands.

```
PARAMS> STAT CONFIG

Configuration:
 Node R1EJAA is an RF31 controller
 Software RFX V200 built on 22-AUG-1989 13:39:09
 Electronics module name is EN92300124
 In 26 power-on hours, power has cycled 45 times
 System up time is 1 01:25:07.90
 System time is 26-AUG-1989 15:00:12


PARAMS> STATUS DATALINK
Datalink Counters:
    Interval:  91593 seconds
  Pkts Rcv'd:      121535:*******************************
  Pkts Xmt'd:      121536:*******************************
  Naks Rcv'd:       23665:**********
  Nakd Xmt'd:           0:
Resets Rcv'd:       45713:******************
Resets Xmt'd:       21369:*********
 No response:        2290:*
  Duplicates:           0:
    Unrecogs:           0:


PARAMS> STATUS PATHS
 ID  Path Block       ote Node       DG_S  DG_S  MSGS_S  MSGS_S
 --  ---------       ----------- ----- ----- ------- -------
  1 PB FF8451        rnal Path         0     0       0       0
  0 PB FF8452A  R QZT  RFX V103        0     0       0       0
  3 PB FF8453D2 R7Z    RFX V103        0     0       0       0
  4 PB FF8454FE R ZJBA RFX V103        0     0       0       0
  5 PB FF84562A R7ULBA RFX V103        0     0       0       0
  6 PB FF845756 LEDS13 VMS V5.2        0     0   27497   27497
  7 PB FF845882 ISMINE VMS V5.2        0     0   32636   32636

PARAMS>
```

WRITE

The WRITE command records the device parameter changes you make using the SET command in nonvolatile memory. The WRITE command is similar to the VMS SYSGEN WRITE command, although no parameters are available.

When using the WRITE command, you must be aware of system and/or ISE requirements and use the command accordingly or it may not succeed. The WRITE command may fail for one of the following reasons:

- You altered a parameter that required the unit, and the unit could not be acquired (it was not in the available state with respect to all hosts). Changing the unit number is an example of a parameter that requires the unit.

  A parameter requires the unit if the flag U appears at the right of the SHOW display (refer to the SHOW example).

- You changed a parameter that required controller initialization, and you replied negatively to the request for reboot. Changing the node name or the allocation class are examples of parameters that require controller initialization.

  A parameter requires controller initialization if the flag B appears at the right of the SHOW display (refer to the SHOW example).

- Initial drive calibrations were in progress on the unit. The use of the WRITE command is inhibited while these calibrations are running. In this case, the message "Drive calibration in progress, please try later" is displayed.

WRITE Syntax

The syntax for the WRITE command is:

WRITE

WRITE Example

The following is an example of how you use the WRITE command.

```
PARAMS> SHOW ALL
Parameter       Current            Default         Type     Radix
---------  ---------------  ---------------  -------  -----
ALLCLASS                  0                0     Byte     Dec  B

PARAMS> SET . 4
PARAMS> WRITE
Changes require controller initialization, ok? [Y/(N)] N
PARAMS>
```

ZERO

The ZERO command clears a block of counters or all known blocks of counters.

ZERO Syntax

The syntax for the ZERO command is:

ZERO *counter*

The *counter* is a block of performance counters or all known counters.

Counter Types

The following table describes the available counter values.

Counter	Function
ALL	Clears all performance counter blocks.
MSCP	Clears the MSCP performance counters and histograms.
SEEKS	Clears the seek performance counters and histograms.

ZERO Example

The following example of the ZERO command can clear the MSCP performance counters and histograms.

```
PARAMS> STATUS MSCP
MSCP statistics:
        Interval:  92715 seconds
    I/O Commands:  0
 Max Queue Depth:  0
 Queue Depth Histogram:
 Commands Processed Histogram:
  GetUnit         1546:**************************************
   SetCon:           2:*
 Blocks Accessed Histogram (cylinder ranges):
 Read Lengths Histogram (block count ranges):
 Write Lengths Histogram (block count ranges):

PARAMS> ZERO MSCP
PARAMS> STATUS MSCP
MSCP statistics:
        Interval:  2 seconds
    I/O Commands:  0
 Max Queue Depth:  0
 Queue Depth Histogram:
 Commands Processed Histogram:
 Blocks Accessed Histogram (cylinder ranges):
 Read Lengths Histogram (block count ranges):
 Write Lengths Histogram (block count ranges):

PARAMS>
```

CHAPTER 5

5 Performing Troubleshooting Procedures

Introduction

This chapter describes what you should do in the event that your RF31 ISE fails to operate correctly. Also described in this chapter are failure indications and internal self-tests. Remember that repairs to the RF31 ISE should be attempted only by qualified Customer Services engineers.

Self-Tests

POST

The ISE automatically performs a self-test whenever power is applied. This power-on self-test (POST) is designed to detect most faults that occur in the ISE.

Internal Tests and Calibrations

Periodic internal tests and calibrations are performed as a normal part of the operation of the ISE. These tests are run automatically on a periodic basis and cause the device to appear active (although there may be no system activity at the time). This is normal, and does not indicate a problem.

Failure Indications

An ISE may fail during initial power-up or during normal operation. The following table describes the states of the LEDs, what these states mean, and what actions you should take.

When . . .	It means . . .	And you . . .
The green READY LED is on and the red FAULT LED is off	the operating condition is normal	may use the ISE.
The red FAULT and green READY LEDs turn on and stay on	the ISE is unable to execute the POST	power down the system, check DSSI cable, and try again. If failure persists, call Digital Customer Services.
Neither LED turns on	power is not getting to the ISE	check power supply and DSSI cable. If failure persists, call Digital Customer Services.
The red FAULT LED turns on and stays on	a fault condition exists	run local programs, as described in Chapter 4, to isolate failure. If unable to access local programs, check DSSI node ID. If failure persists, call Digital Customer Services.

Before Calling Digital Customer Services

If a failure occurs with the RF31 ISE, perform the steps outlined in the following table before you call Customer Services.

Step	Action
1	Remove the outside panels from the system enclosure (as needed) to access the ISE. Refer to your system maintenance guide for panel removal procedures. CAUTION **Do not operate the ISE for more than 10 minutes with the outside panels removed. Overheating and subsequent damage may result due to changes in air flow.**
2	Check to make sure the green LED on the external terminator is turned on. If it is not on, there is no power on the DSSI bus cable. Check for a bad cable connection or a blown fuse on the DSSI adapter module.
3	Power down the system.
4	Check for correct power cable connections to the ISEs.
5	Make sure the DSSI cable is connected correctly to all DSSI ISEs and to the host adapter.

Step	Action
6	Make sure the ISE has the correct DSSI node ID, and that no other device on the DSSI bus has the same node ID. The following table explains how to do this.

If . . .	Then . . .
An OCP is connected to the ISE	refer to Chapter 3 to make sure the DSSI node ID is set correctly. Now make sure the OCP is functioning. Disconnect it from the ISE, set the node ID using the switches on the drive module, and then try once again to power up the system. NOTE **When the OCP is disconnected, the ISE defaults to no write-protect and on-line.**
No OCP is connected	check the switches on the drive module to make sure they are set correctly.

Step	Action
7	Make sure the terminators are correctly inserted at the ends of the DSSI bus cable. Refer to your system or adapter documentation for details on the DSSI bus cable and the terminators.

Step	Action
8	Reapply power. Watch the status of the READY and FAULT LEDs. The following table explains what you should do.

When . . .	Then . . .
The red FAULT LED and the green READY LED turn on. Then the red FAULT LED turns off while the green READY LED stays on continuously	the drive is ready for operation.
The red FAULT LED stays on, or the green READY LED does not turn on, or both LEDs stay on	call Digital Customer Services.

CHAPTER 6

6 Digital Repair Services

Types of Service Plans

Digital Customer Services offers a range of flexible service plans:

On-Site Service

On-site service offers the convenience of service at your site and insurance against unplanned repair bills. For a monthly fee, you receive personal service from our service specialists. Within a few hours, the specialist is dispatched to your site with equipment and parts to give you fast and dependable maintenance.

Basic Service

Basic service offers full coverage from 8 a.m. to 5 p.m., Monday through Friday. Options are available to extend your coverage to 12-, 16-, or 24-hour periods, and to include Saturdays, Sundays, and holidays. Under the basic service plan all parts, materials, and labor are covered in full.

DECservice

DECservice offers a premium, on-site service providing committed response to remedial service requests made during contracted hours of coverage. Remedial maintenance will be performed continuously until the problem is resolved, which makes this service ideal for customers requiring maximum service performance. Under DECservice all parts, materials, and labor are covered in full.

Carry-In Service

Carry-in service offers fast, personalized response, and the ability to plan your maintenance costs for a smaller monthly fee than On-Site Service. When you bring your unit to one of 160 Digital SERVICenters worldwide, factory-trained personnel repair your unit within 2 days. This service is available on selected terminals and systems. Contact your local Digital Customer Services office to see if this service is available for your unit.

Digital SERVICenters are open during normal business hours, Monday through Friday.

DECmailer

DECmailer offers expert repair at a per use charge. This service is designed for users who have the technical resources to troubleshoot, identify, and isolate the module causing the problem. Mail the faulty module to our Customer Returns Center where the module is repaired and mailed back to you within 5 days.

Per Call Service

Per call service offers a maintenance program on a noncontractual, time-and-materials-cost basis. It is appropriate for customers who have the expertise to perform first-line maintenance, but may occasionally need in-depth support from Digital Customer Services.

Per Call Service is available as a supplement to three of our service plans, as shown in the table below.

Service Plan	Turnaround Time	Coverage Is Available . . .
Basic	ASAP	12-, 16-, or 24-hours a day including weekends and holidays, depending or ption chosen
On-Site	2 to 3 days	24 hours a day, 7 days a week
Carry-In	2 to 3 days	During normal business hours

For More Information

For more information on these Digital Customer Services plans, prices, and special rates for volume customers, call the Digital Customer Services office nearest you.

INDEX

Index