DSA TROUBLESHOOTING COURSE

STUDENT GUIDE

This manual is used as a student guide for the DSA Troubleshooting course. Much of the material contained herein may be used as reference material by the field engineer while troubleshooting problems in the field. The Field Engineer at his/her discretion may remove any or all parts of this manual and incorporate them into other resource documents or notebooks. This manual is the property of DIGITAL EQUIPMENT CORPORATION and is considered for DIGITAL INTERNAL USE ONLY.

Class A Computing Devices:

NOTICE: This equipment generates, uses, and may emit radio radio frequency energy. It has been tested and found to comply with the limits for a Class A computing device pursuant to Subpart J of Part 15 of FCC rules for operation in a commercial environment. This equipment, when operated in a residential area, may cause interference to radio/TV communications. In such event the user (owner), at his/her own expense, may be required to take corrective measures.

Revision/Update Information: Version 1.0, March 1989

This is the first document release from CX/CSSE and supersedes all previous versions used in preliminary DSA Troubleshooting Courses. All revisions and known error corrections up to March/89 have been incorporated into this manual

CX/CSSE 1st Preliminary 1986
CX/CSSE Version 1 March, 1989

The postpaid READER'S COMMENTS form on the last page of this document requests the user's critical evaluation to assist in preparing future documentation.

The following are trademarks of Digital Equipment Corporation:

DEC
DEC/CMS
DEC/MMS
DECnet
DECsystem–10
DECSYSTEM–20
DECUS
DECwriter
DIBOL
EduSystem
IAS
MASSBUS
PDP
PDT
RSTS
RSX
UNIBUS
VAX
VAXcluster
VMS
VT
digital™

The following are also trademarks of Digital Equipment Corporation:

CI
DDCMP
DDIF
DEBET
DSA
DECconnect
DECdirect
DECdisk
DECmail
DECmat
DECmate
DECnet/E
DECnet-RT
DECnet-ULTRIX
DECserver
DECservice
DECtape
DELNI
DELUA
DEMPR
DEQNA
DESTA
DEUNA
DMS
DRB32
DSRVB-AA
HSC
IVIS
KA10
KD11
KDA50-Q
KDB50-A
KDB50-B
KI
KL10
KS10
LA50
LN01
LN03
MicroPDP-11
MicroVAX
MicroVMS
MSCP
PDP-11
Q-bus
RA60
RA70
RA80
RA81
RA90
RC25
RQDX3
RMS-11
RSX-11
RSX-11M
RSX-11S
RX33
SA482
SA550
SA600
SA650
TA78-81
TMS-11
TK50
TOPS-10
TOPS-20
TU78-81
UDA50
UETP
ULTRIX
ULTRIX-11
ULTRIX-32
VAXELN
VAX/VMS
VAXsimPLUS
VMS
RA82

This document was prepared using VAX DOCUMENT, Version 1.0

DSA TROUBLESHOOTING COURSE

Lab Exercise #1

BLOCK.COM

BLOCK.COM
Lab Exercise 1

1. Log into your STUDENTx account
2. $ **SET DEF [STUDENTx.MISC]** (x = your student account #)
3. $ **SET TERMINAL/INQUIRE**
4. $ **@BLOCK**

STEP	BLOCK.COM Program Prompt	Your Response
5	**Would you like help ?**	Y
6	**Create an output file of your conversion Y[N] ?**	Y
7	**What type of DISK would you like ?** Note the response and "allowed selections."	RA87
8	**What type of DISK would you like ?**	RA81
9	**What formatted mode would you like to convert ?** Note that help may be obtained as needed.	H
10	**What formatted mode would you like to convert ?**	16
11	**What TYPE of block would you like to convert ?**	H
12	**What TYPE of block would you like to convert ?**	LBN
13	**Select Group Offset ?** Pressing <return> selects the normal (default) group offset for the selected drive type. Changing the group offset is normally only used during disk engineering development and new product testing.	<return>
14	**Would you like a Status Map displayed Y[N] ?**	Y

STEP	BLOCK.COM Program Prompt	Your Response
15	**Select the desired LBN number(s) ?** Compare the results with the sample calculation shown in the DSDF section of the Student Reference Manual for this LBN on a 16–bit RA81. See LBN conversion example #2.	2498
16	**Select the desired LBN number(s) ?** Compare the results with the sample calculation shown in the DSDF section of the Student Reference Manual for this LBN on a 16–bit RA81. See LBN conversion example #3.	2499
17	**Select the desired LBN number(s) ?** Be sure a comma separates the numbers.	2498,2499
18	**Select the desired LBN number(s) ?** Be sure a colon separates the numbers.	2490:2500
19	**Select the desired LBN number(s) ?** Note that the conversions are performed in ascending order regardless of the order in which the block numbers are entered into the program.	2500:2490
20	**Select the desired LBN number(s) ?** Note the various ways that block numbers may be entered into the program. Entries may contain a mix of numbers using commas and/or colons.	H <ret>
21	**Select the desired LBN number(s) ?** Watch the progression of numbers in the SECTOR # column and the SECTOR # FROM INDEX column (physical sector). Note that when the HEAD number changes from 0 to 1 there is a 14–sector shift in the SECTOR # FROM INDEX column. This is due to the group offset (RA81=14) when switching groups (heads in an RA81).	40:60 <cr>
22	**Select the desired LBN number(s) ?** Press <return> until you are prompted to select the disk type.	
23	Select RA60, 16–bit LBN 6000. Compare the results with the sample calculation shown in the DSDF Section of the Student Reference Manual for this LBN on a 16–bit RA60. See LBN conversion example #5.	

STEP	BLOCK.COM Program Prompt	Your Response
24	Select LBNs in RCT and HOST area (use MAP). The map may be obtained by typing "M" <ret> in response to the prompt that requests the block numbers for the type of block you selected (LBN, RBN, etc.).	
25	Select RA82. (Note that the RA82 is only capable of 16–bit mode.)	
26	Select other disk types and modes.	
27	Select RBN. – Look at the Map while RBN is selected. – Enter some numbers and review the results.	
28	Select DBN. – Look at the map while DBN is selected. – Enter some numbers and review the results. – Note the cylinders that contain these blocks.	
29	Select XBN. – Look at the map while XBN is selected. – Enter some numbers and review the results.	
30	Use the H (help) feature for various prompts.	
31	Deliberately enter erroneous information to some of the responses. – enter an invalid mode number – enter an invalid block type – enter block numbers that are invalid or too large	
32	Exit the program.	EXIT <ret>

STEP	BLOCK.COM Program Prompt	Your Response
33	**$ Type BLOCK.DAT** Note that errors are not included on the output and that the header page is different when you changed parameters while executing the program.	
34	Print the file BLOCK.DAT to obtain a hardcopy for review.	

BLOCK.COM NOTES

- VERSIONS of BLOCK.COM prior to V 3.5 are obsolete and have some calculation errors.
- BLOCK.COM is distributed with VAXSIM–Plus using the file name VAXSIM$LBN.COM. The versions of this conversion utility that were released with VAXSIM–Plus version 1.0 and 1.1 do not support the RA90. Version 1.2 of VAXSIM–Plus contains VAXSIM$LBN.COM version 3.5 which has the latest corrections and will support the RA90.

  In the meantime, use BLOCK.COM version 3.5 to support RA90 troubleshooting.

DSA TROUBLESHOOTING COURSE

Lab Exercise #2

DKUTIL

– For execution on an RA70 Disk Drive –

DKUTIL Execution on an RA70 Disk Drive
Lab Exercise 2,

Save hardcopy of all your activities for reference and questions during subsequent discussion in lecture.

1	**RUN DD1:DKUTIL**	(HSC50)
	RUN DKUTIL	(HSC70)

2 Select the target disk that you have been assigned.

Version 350 of HSC code will prompt you for the disk device number (Dxx). Version 370 and later HSC code will provide you with a message indicating that you will need to enter an additional command (GET Dxx) to select the disk unit number.

3 Verify that the mode is correct and the FCT is VALID from the display.

Use your DKUTIL user guide (in the Student Manual) as reference for the following steps.

	Command(s) To Enter	Notes
4	**DISPLAY CHARA DISK**	Compare the results to the drive characteristics in your student manual for this drive type.
5	**DISPLAY ERRORS**	On RA80/81/82, this displays the contents of the internal error silo which contains the last 16 drive detected error codes. For later drives (RA70, RA90, etc.) this command will dump and format the drive internal error log. For the RA60, an error message will result since the RA60 does not support internal error logging or error silos.
6	**DISPLAY RCT**	Note the LBNs that are replaced and which RBNs they are revectored to. The asterisk (*) next to an entry indicates non-primary replacement. Otherwise, the replacement is primary.
7	**DISPLAY RCT/FULL**	Note the additional information that is displayed from RCT block 0 when using the /FULL modifier.
8	**DISP FCT**	Note the PBNs (Physical block numbers) and which logical blocks they describe that should be considered bad and replaced when formatting the disk. The codes in parentheses represent the reasons why the block is bad and how it is to be treated. This information is in the FCT section of the Student Manual.

	Command(s) To Enter	Notes
9	**DISPLAY ALL**	The information provided here is the total accumulation of data that would be obtained if individually entering each of the previous DISPLAY commands.
10	**DISPLAY CHARA LBN 100**	Note the header information that is supplied. Consult your instructor if you have any questions concerning the format of the header information. This display provides translation of an LBN address (100 in this example) into cylinder, group, track, and position. Position is the physical sector from Index.
11	**DISPLAY CHARA DBN 2**	
12	**DISPLAY CHARA RBN 24**	
13	**DISPLAY CHARA XBN 400**	
14	**DUMP LBN 100/ALL**	Note the contents of the data, the four copies of the header, and the calculated EDC difference. Note the header code.
15	**DUMP/ALL DBN 123**	Note the contents of the data, the four copies of the header and the calculated EDC difference. Note the header code and how it differs from an LBN. Using the RCT display obtained from step 6 above, select an RBN number that is not being used for replacement and use that RBN as part of the following command:
16	**DUMP/ALL RBN xxxx**	Note the contents of the data, the four copies of the header and the calculated EDC difference. Note the header code. Note that EDC is inverted, indicating a forced error Note the data pattern. This is the DEC Standard Format Data Pattern. Using the RCT display obtained from step 6 above, select an RBN number that is being used for replacement and use that RBN as part of the following command:

	Command(s) To Enter	Notes
17	**DUMP/ALL RBN xxxx**	Note the contents of the data, the 4 copies of the header and the calculated EDC difference. Note the header code. In this case, the RBN should contain valid data from some LBN and a correctly written EDC (EDC diff = 0).
18	**DUMP/ALL XBN 0**	Note the contents of the data, the four copies of the header and the calculated EDC difference. Note the header code and how it is different from the other block types. This is the FCT control block. Use your Student Manual to find the mode byte, the FK bit, and their contents.
19	**DUMP/ALL FCT BLOCK 1 COPY 1**	What is different about this from the information obtained in the previous step? Have the instructor clarify this if it is unclear.
20	**DUMP/ALL XBN 1**	This is the first block in the FCT that contains PBN descriptors.
21	**DUMP/ALL FCT BLOCK 2 COPY 1**	What is different about the information displayed here from the previous step?
22	**DUMP/ALL RCT BLOCK 1 COPY 1**	This is the RCT control block, often referred to as block 0 of the RCT (accessed as block 1 when using DKUTIL). Use your student manual to decode the contents of RCT block 0.
23	**DUMP/ALL LBN 547041**	What's different about the contents here as compared to the contents from the previous step?
24	**DUMP/ALL LBN 100**	
25	**MODIFY 32 1111 2222 3333 4444 5555 6666**	

NOTE

An invalid command error message indicates the write patch is not installed. Install it at this time, if necessary. After installing the patch and re-running DKUTIL, continue this exercise starting with step 24.

	Command(s) To Enter	Notes
26	**DUMP/ALL BUFFER**	Note the differences that occurred after the modify. It probably does not appear the way you expected. The next step should provide clarification.
27	**MOD 32 O1111 O2222 O3333 O4444 O5555 O6666 O7777**	Use the letter O (for octal) and not zero (0).
28	**DUMP/ALL BUFFER**	Now notice the changes that occurred.
29	**WRITE LBN 547040**	This will write the contents of the buffer (which you have modified) to LBN 547040.
30	**DUMP LBN 547040**	This is to verify the block was correctly written with the desired modifications.
31	**REVECTOR 547040**	This is actually a command that forces a replacement; i.e., LBN 547040 will be replaced.
32	**DISP/FULL RCT**	Verify that the LBN was replaced and remember which RBN was used.
33	**DUMP/ALL LBN 547040**	
34	**DUMP/ALL RBN xxxx**	xxxx is the RBN number you obtained from step 32 above.
35	**DUMP/ALL/RAW LBN 547040**	Compare the results of step 33 through step 35. Do you understand what is happening? Consult the instructor if you are unclear.
36	**DUMP RCT BLOCK 2**	This shows the contents of the RCT caching block which contains a buffered copy of the host data during the last replacement operation. Notice that this is the same data that was in LBN 547040 which you previously replaced manually.
37	**DUMP RCT BLOCK 3**	This is the first block in the RCT that contains DESCRIPTORS. It mostly zeroes (if not all zeroes).

	Command(s) To Enter	Notes
38	**DUMP RCT BLOCK x**	Substitute 4, 5, 6, etc., for the value x and continue looking at RCT descriptor blocks until you encounter a block that contains something other than all zeros. The non–zero values will be due to an RBN descriptor entry. Use the RCT section of your Student Guide and see if you can decode a descriptor and compare it to the corresponding entry in the RCT table obtained in step 6. Ask your instructor for assistance, as needed.
39	**DUMP RCT BLOCK 9999**	Notice that you received an error message. The message will tell you the maximum LBN that you can enter for the RCT table. This is a convenient technique to locate the LBN number of the last RCT descriptor block in the table for this particular type of disk drive.
40	**DUMP RCT BLOCK xxxx**	Substitute the maximum LBN number you obtained in step 39 for xxxx in this step. This will display the last RCT descriptor block (copy 1 in this case). This descriptor block should contain descriptors with descriptor codes of 10, indicating the end of a valid RCT table copy. You should also see a descriptor entry for the RBN that was allocated for LBN 547040 when you performed the revector command in step 31.

Save all the material you obtained in this exercise for review later in class.

TABLE OF CONTENTS

CHAPTER 6 REPLACEMENT CONTROL TABLE (RCT) 6–1

CHAPTER 7 FORMAT CONTROL TABLE (FCT) 7–1

CHAPTER 8 STANDARD DISK INTERFACE (SDI) 8–1

CHAPTER 9 LEVEL 2 SDI COMMANDS 9–1

CHAPTER 18 FAKDSK (ON HSC) . . . 18–1

EXAMPLES

FIGURES

TABLES

CHAPTER 1

GOALS/OBJECTIVES

1.1 WHAT'S COVERED

- DSDF disk format and structures
- SDI
- BBR
- Drive status decoding
- Basic error log decoding and review
- Special tools and diagnostics
    - DKUTIL
    - Block Conversion Utility
    - RAUTIL HDA Analyzer
    - DKRFCT
    - FORMATTING
    - Error Log Tools
    - Remote Analysis Tools
    - Miscellaneous HSC Utilities
    - Disk Scrubbing
- Logically broken drives (versus) physically broken
- Emphasis will be VMS, HSC, and some tools/concepts in a two–board controller environment
- Troubleshooting information will be supplied throughout the discussions of the various tools, topics, and lab activities
- Lab exercises to familiarize the student with the usage of the tools and problem solving

1.2 WHAT'S NOT COVERED

- MSCP
- Specific controller repair
- Specific drive repair except HDA and communication references
- Emphasis on physically broken equipment
- Tape drive support

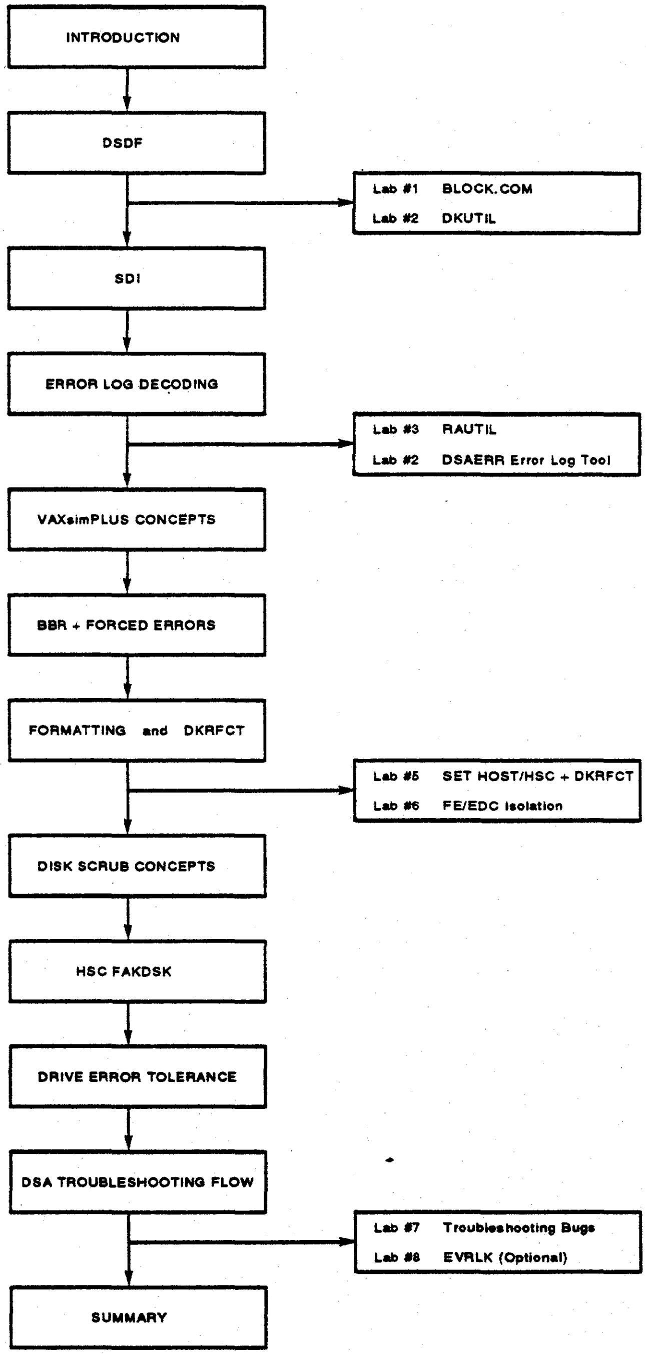
NOTE:

- ALL OF THE DOCUMENTATION MATERIALS AND SPECIAL TOOLS OBTAINED IN THIS COURSE ARE STRICTLY **DIGITAL INTERNAL USE ONLY**. PLEASE TREAT THIS MATERIAL ACCORDINGLY. DO NOT LEAVE ANY OF THE SPECIAL SOFTWARE TOOLS OR DOCUMENTATION ON A SITE THAT IS NOT UNDER DIGITAL SERVICE CONTRACT AGREEMENTS OR A SITE THAT IS ACCESSIBLE BY 3RD PARTY MAINTENANCE.

- DO NOT ATTEMPT TO RE–TEACH THIS COURSE IN THE FIELD !

1.3 COURSE MAP

Figure 1–1: Course Map



CHAPTER 2

DSDF FOR RA60/70/81/81/82/90

DSDF for RA60/70/80/81/82/90

This document describes the location, specification, and function of the various disk internal storage such as platters, heads, and the positioner mechanism, as well as storage components such as cylinders, groups, tracks, and blocks. Bad block replacement, hardware error recovery, and forced errors are also described.

This material stresses Digital Standard Disk Format (DSDF) characteristics unique to a variety of disk drives. It will help you understand the overall function of the drives, interpret error logs, and work with diagnostic information in the field.

2.1 MEDIA COMPONENTS (Physical Geometry, Head Disk Assemblies)

2.1.1 RA80/81/82

The RA80/81/82 HDA contains 4 storage platters attached to a spindle assembly. The 4 platters provide a total of 8 magnetic recording surfaces. A rotary positioner and motor assembly within the HDA contains 8 metal arms. Each metal arm contains 2 head assemblies for a total of 16 heads within the HDA. The heads and arms are attached to the positioner so that 2 heads are located over each of the 8 recording surfaces. The positioner motor is responsible for moving all 16 heads simultaneously across the media surfaces during a SEEK operation.

Fourteen data heads are used for R/W data operations to and from the disk surface in the RA80 and RA81. Fifteen of the data heads are used for R/W data operations to and from the disk surfaces in the RA82. The heads are numbered as shown in Figure 2–1 and Figure 2–2. The last head is a servo head used to read specially recorded servo information from the dedicated servo area of the disk surface.

Figure 2–1: Basic RA80/81 HDA

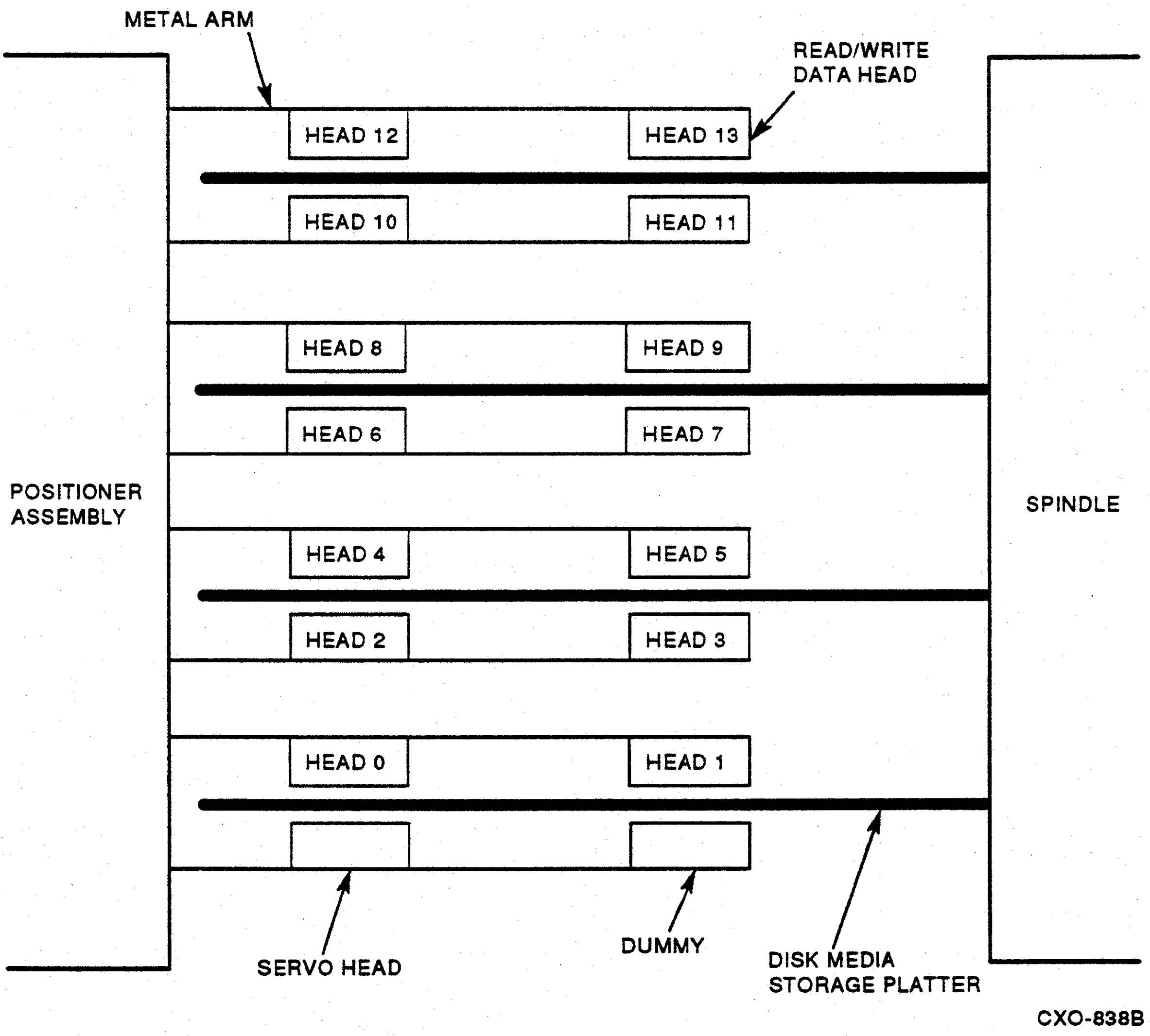
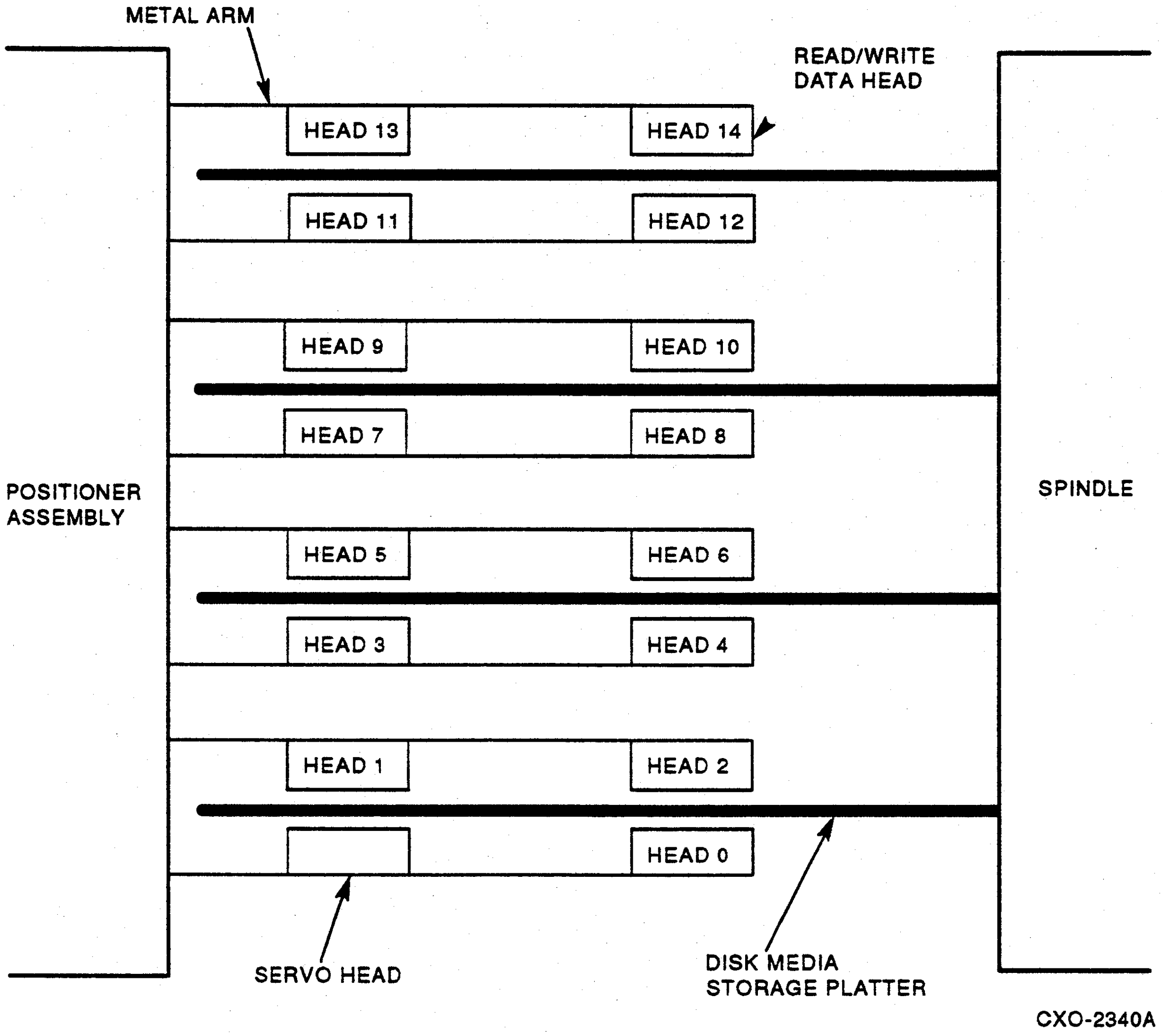


Figure 2–2: Basic RA82 HDA

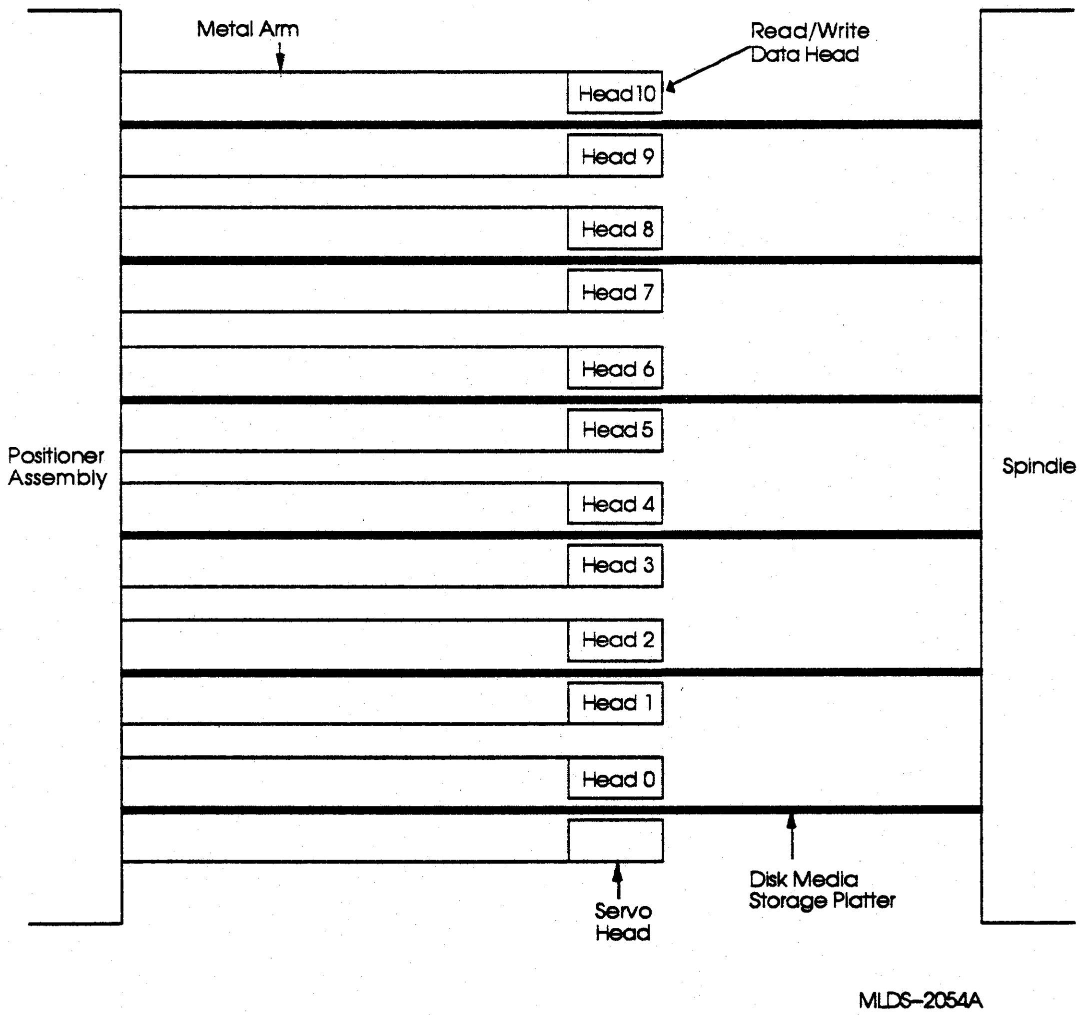


2.1.2 RA70

The RA70 HDA contains 6 storage platters attached to a spindle assembly. The six platters provide a total of 12 magnetic recording surfaces. A linear positioner and motor assembly within the HDA contains 12 metal arms. Each metal arm contains 1 head assembly for a total of 12 heads within the HDA. The heads and arms are attached to the positioner so that 1 head is located over each of the 12 recording surfaces. The positioner motor is responsible for moving all 12 heads simultaneously across the media surfaces during a SEEK operation.

Eleven data heads are used for R/W data operations to and from the disk surface in the RA70. The heads are numbered as shown in Figure 2–3. The last head is a servo head used to read specially recorded servo information from the dedicated servo area of the disk surface.

Figure 2–3: Basic RA70 HDA

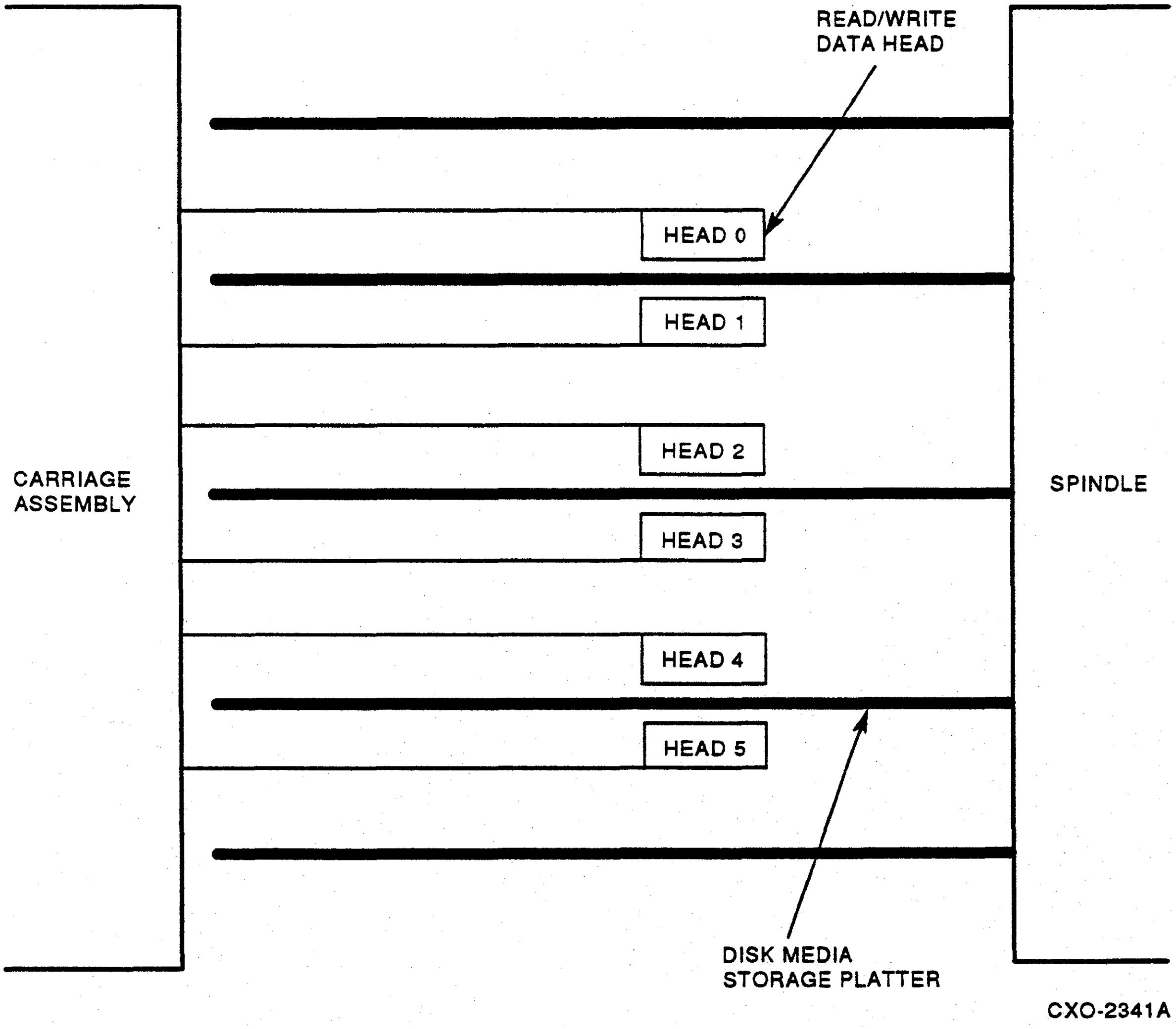


2.1.3 RA60

The RA60 uses a removable pack that contains 3 storage platters attached to a spindle. The 3 storage platters provide a total of 6 magnetic recording surfaces. A carriage assembly within the RA60 drive contains 6 replaceable head/arm assemblies. These assemblies are attached to the carriage so that 1 head will be positioned over each recording surface. The carriage assembly is responsible for moving all 6 heads simultaneously across the media surfaces during a SEEK operation.

All 6 data heads are used for R/W data operations to and from the disk surface in the RA60. The heads are numbered as shown in Figure 2–4. There is no dedicated servo surface in the RA60.

Figure 2–4: Basic RA60

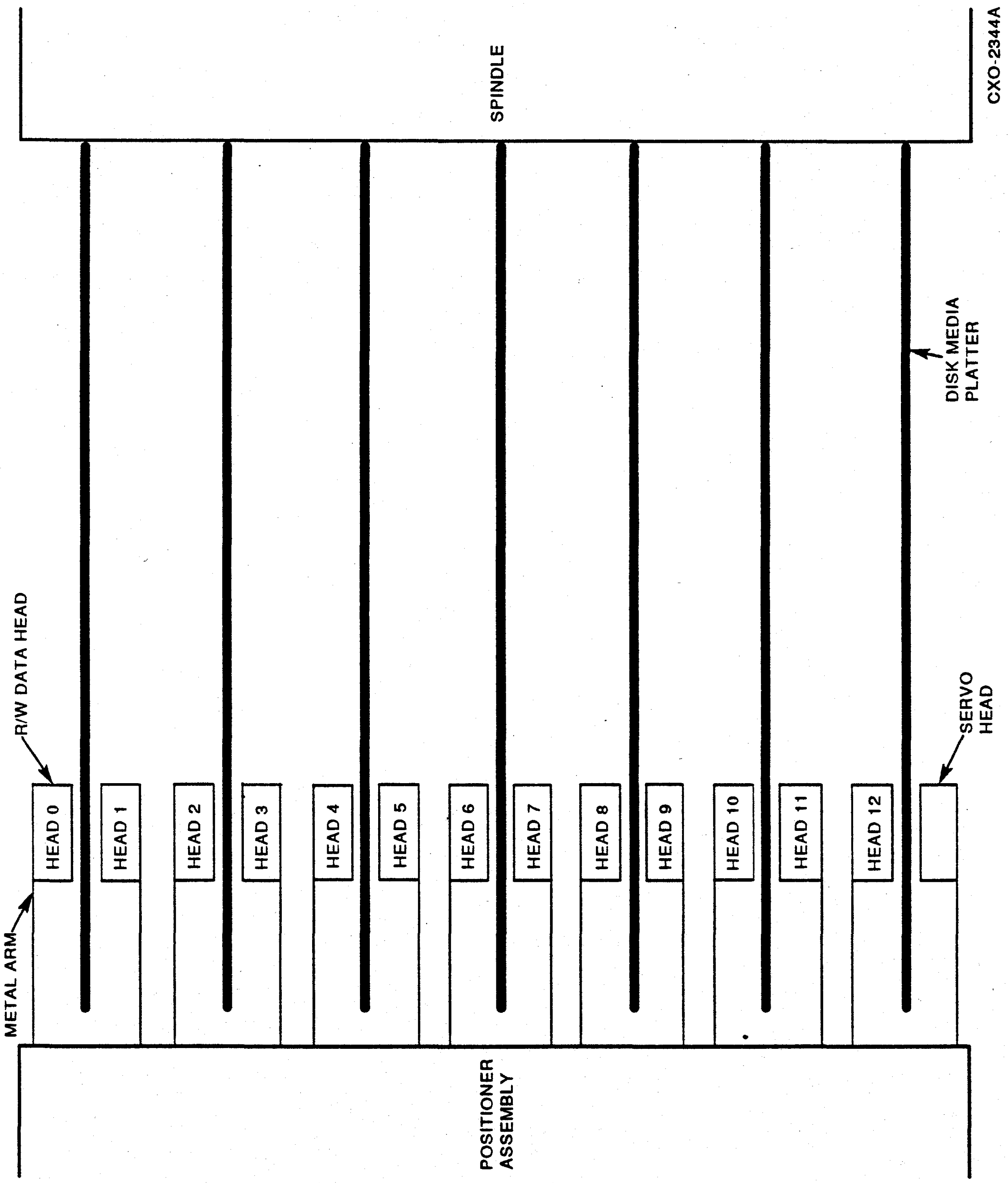


2.1.4 RA90

The RA90 HDA contains 7 storage platters attached to a spindle assembly. The 7 platters provide a total of 14 magnetic recording surfaces. A positioner and motor assembly within the HDA contains 14 metal arms. Each metal arm contains 1 head assembly for a total of 14 heads within the HDA. The heads and arms are attached to the positioner so that 1 head is located over each of the 14 recording surfaces. The positioner motor is responsible for moving all 14 heads simultaneously across the media surfaces during a SEEK operation.

Thirteen data heads are used for R/W data operations to and from the disk surface in the RA90. The heads are numbered as shown in Figure 2–5. The last head is a servo head used to read specially recorded servo information from the dedicated servo area of the disk surface.

Figure 2–5: Basic RA90 HDA



2.2 SERVO INFORMATION

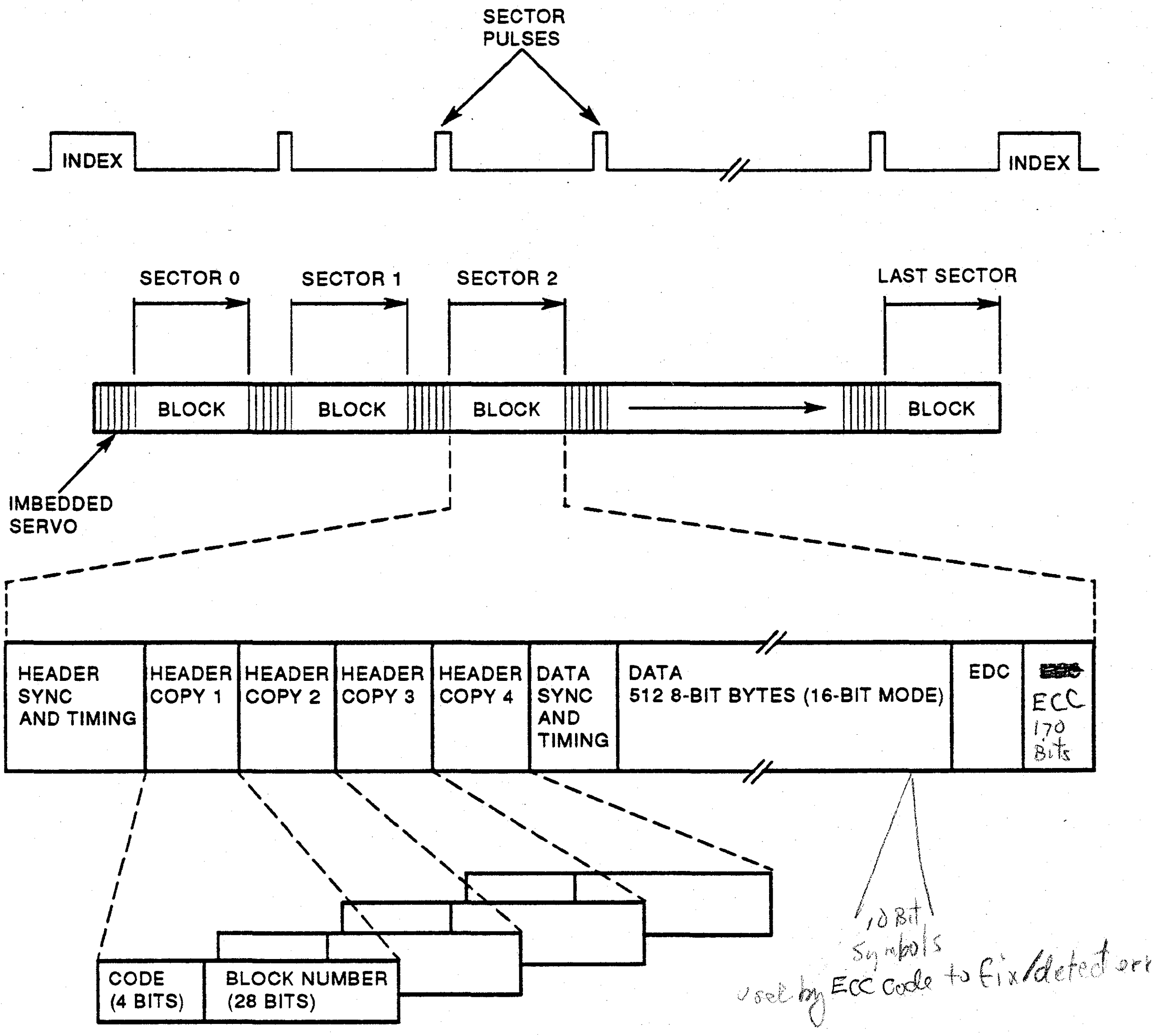
In the RA70, RA80, RA81, RA82, and RA90, information on the dedicated servo surface is written only at the factory during the HDA manufacturing process. For these drives, the dedicated servo surface contains the following:

- Information to identify the inner and outer guard band regions of the media.
- Information to provide index and sector pulses.
- Information to provide proper positioning of the R/W heads over physical cylinders and tracks.

In the RA70, RA81, RA82, and RA90, the positioning pulses from the dedicated servo surface provide only the coarse positioning information. Special embedded servo information (factory written only) is recorded between each sector on the data surfaces to provide fine positioning information for the servo circuits in these drives. Refer Figure 2–6.

The RA60 also provides embedded servo information (factory written only) to provide both coarse and fine positioning information for its servo circuits.

Figure 2–6: Basic Track and Sector



	SECTORS/TRACK	
	16-BIT	18-BIT
RA60	43	39
RA70	34	—
RA80	32	29
RA81	52	47
RA82	58	—
RA90	70	—

CXO-2205A

2.3 DATA INFORMATION

The data surfaces are used for both WRITE and READ data operations. These data recording areas are divided into physical cylinders, tracks, and sectors by the design of the drive. The review of these physical concepts is necessary to better understand the logical organization of these disks in the DSA environment and to troubleshoot and maintain these drives in system environments.

2.4 PHYSICAL SECTOR

The sector is the smallest addressable unit of the disk. It occupies a specific physical position relative to the index pulse, and it is available for reading or writing once per disk revolution. The sector contains the components shown in Figure 2–6.

2.4.1 Header

There are four copies of the 32–bit header in every sector, or a total of 128 bits. Each of the 32–bit copies contains a 28–bit block number for the sector. Each of the four copies contains a 4–bit code to define what type of block it is (LBN, RBN, XBN, or DBN), which logical area it is located in, and the current status of the block. Block types and logical areas will be discussed in more detail later in this course.

The header does not provide any error checking or correcting capability and, for this reason, there are four copies of each header. In the event that one header copy becomes corrupt, the controller uses all four copies and special algorithms to determine the correct block number and code for the sector.

The controller uses the header information to assure that the data in a particular sector is the data it wants during read data or write data operations.

2.4.2 Data

This is the specific information written on the disk by the host applications software, host operating system software, or the controller during read data or write data operations. The data consists of 512 8–bit bytes in the 16–bit formatted mode or 576 8–bit bytes in the 18–bit formatted mode.

Disks formatted for 16–bit mode are used on 16–bit systems (PDP–11) and 32–bit systems (VAX). Disks formatted for 18–bit mode are used on 18–bit systems (PDP–15) and 36–bit systems (DECsystem–10 or DECsystem–20). They can only be set for these modes at the factory. The number of sectors per track for 18–bit disks is less than the number of sectors for 16–bit disks. The number of sectors on each track is derived from the servo information formatted only at the factory.

NOTE

The RA60, RA80, and RA81 are available in both 16–bit and 18–bit configurations.

The RA70, RA82, and RA90 are available in only 16–bit configurations.

2.4.3 Error Detecting Code (EDC)

The Error Detecting Code (EDC) is a 16–bit code used primarily to detect data path problems through the controller. Under some circumstances, the EDC may become inverted (one's complement) to indicate a special condition known as forced error. Forced errors are discussed in more detail later in this course.

2.4.4 Error Correcting Code (ECC)

The Error Correcting Code (ECC) is a 170–bit code appended to the end of a sector by the controller during a write data operation to the disk drive. It is used for both error detection and correction for data passing to and from the disk drive. The calculation of the ECC code is determined by the content of the actual data (including the EDC characters) written to the disk. During a read data operation, the controller again calculates an ECC code determined by the actual data and EDC read from the disk. The calculated ECC is then compared to the ECC read at the end of the sector to determine if errors have developed in the data or ECC.

The ECC code can also be used to correct the errors detected during a read operation. The Reed–Solomon technique is capable of correcting up to eight 10–bit symbol errors in a 604–byte field. This technique is currently used in many of the DSA controllers.

The larger the number of ECC symbol errors, the more severe the error becomes. For example, a 7–symbol ECC error is more severe than a 4–symbol error and requires more effort to correct the data. When the number of symbol errors detected exceeds the capability of the error correcting technique being used, the error is called an uncorrectable ECC error.

DSA disk drives are required to provide a recommended symbol threshold value upon request to the controller. The ECC threshold is the number of ECC errors that the controller should consider abnormal for that drive. If the number of ECC errors detected in a sector equals or exceeds the threshold, the sector (block) should be considered bad and becomes a candidate for replacement. Bad Block Replacement (BBR) is discussed later in this course.

2.5 PHYSICAL TRACK

Sectors on the disk recording surfaces are grouped into tracks. A physical track is a circular collection of contiguous sectors located under one head, during one revolution of any data surface, while the positioner assembly is stationary. Figure 2–6 shows a basic linear representation of a track and how the sectors are grouped within the track. An index pulse defines the start of a track. Sector pulses separate the physical boundaries of each sector.

Figure 2–6 illustrates the location of embedded servo information at the beginning of each sector. As we mentioned earlier, this is used by the drives to provide fine positioning while a head is located over a particular track. The only exception is the RA80 which does not contain embedded servo information between each sector.

The total number of physical sectors per drive depends upon the drive type as shown in Figure 2–6 and Table 2–1.

Table 2–1: Physical Sectors per Track

	16–bit	18–bit
RA60	43	39
RA70	34	–
RA80	32	29
RA81	52	47
RA82	58	–
RA90	70	–

2.6 PHYSICAL CYLINDER

Physical tracks on the disk recording surface are grouped into physical cylinders. A physical cylinder is the collection of each track under every head while the head positioning mechanism is stationary.

If the heads are positioned over a particular physical cylinder and data is requested from a different cylinder, the hardware in the drive must move the positioner to the desired physical cylinder. This would typically occur during a SEEK operation.

Servo information recorded on the disk surface(s) defines how many cylinders exist in the HDA assembly and where they are physically located across the media surfaces.

2.7 LOGICAL DISK ADDRESSING

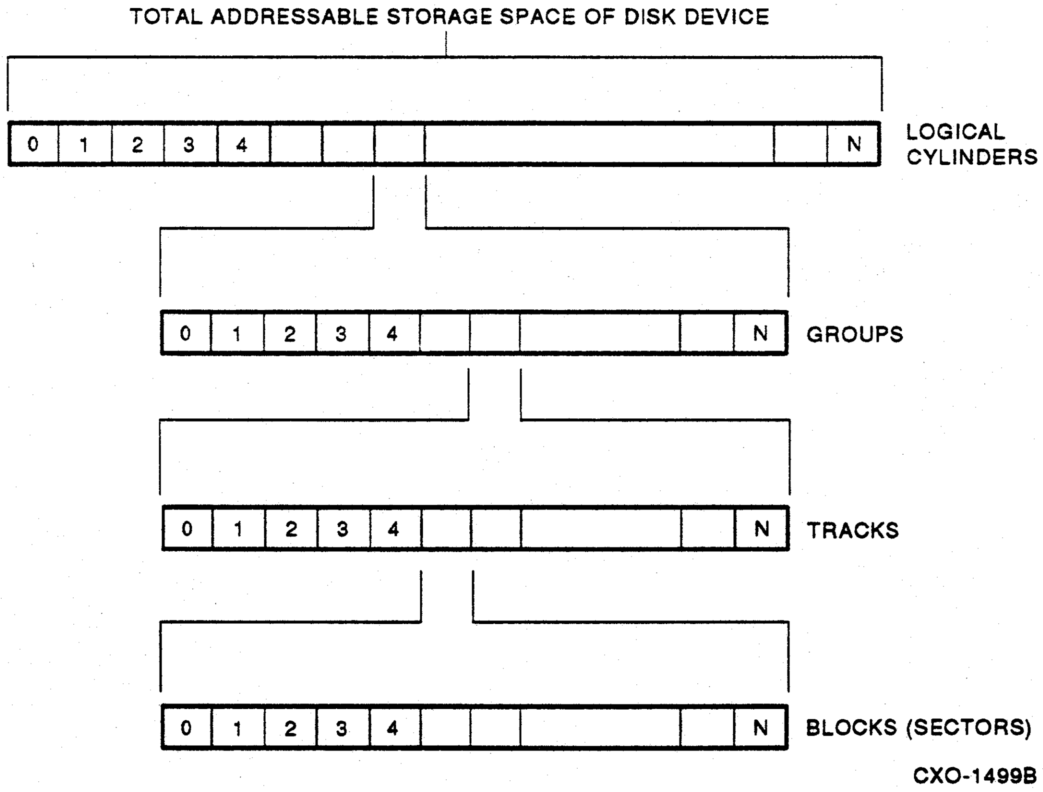
We have previously discussed the physical characteristics of some various disk drives. These included the physical addressing characteristics such as cylinder, track, and sector.

DSDF provides the host processor and a DSA controller with the ability to access data from a DSA disk in a standardized manner without the burden of knowing how or where the data is physically located. The physical location of data within a DSA disk is dependent upon the design of the specific disk drive but is transparent to the host and controller.

To standardize DSA disks, DSDF specifications require the disk be addressable by a common logical addressing language. This removes the burden of the controller having to understand all the unique hardware design characteristics that vary from drive to drive.

Four logical entities make up the addressing of a DSA disk. These are discussed in the following paragraphs. Figure 2–7 is a basic illustration of how these logical addressing entities fit together.

Figure 2–7: Logical Disk Addressing



2.7.1 Logical Block

The logical block, like the physical sector, is the smallest addressable unit of the disk. For the disks described in this course, a physical sector and a logical block are equivalent. The components of a logical block are the same as the physical sector previously discussed in the first section of this document.

2.7.2 Logical Track

A logical track represents a set of logical blocks (sectors) occupying contiguous disk locations in one disk revolution. A logical track has the following properties:

- The time required to access any block on a track from another block on the same track must occur within one revolution of the disk. This time is commonly referred to as the "maximum rotational latency."
- The first sector on a track immediately follows the last sector on the same track.

These properties limit a track to the logical structure of a ring. For the disks described in this document, a physical track and a logical track are equivalent.

2.7.3 Logical Group

A logical group is a collection of logical tracks and has the following properties.

- Selection of a logical group requires a controller-initiated GROUP SELECT command. A level 1 GROUP SELECT command requires one SDI command frame from the controller to the drive to complete the command.
- The same physical sector address is simultaneously available for reading or writing on all tracks within the group.
- The time to switch from one sector in a track to the next sequential sector on any other track within the same group is less than the intersector gap time between two adjacent sectors.

2.7.4 Logical Cylinder

A logical cylinder is the collection of logical groups and has the following properties.

- Selection of a logical cylinder requires a controller-initiated SEEK command. A level 2 SEEK command requires seven SDI command frames from the controller to the drive to complete the command.
- The time required to select new groups within a logical cylinder is LESS than the time required to perform a physical one-cylinder seek.

Switching from one particular group to any other group in the same logical cylinder usually requires an amount of time that is greater than the intersector gap time. To compensate for the seek or head switching time required when switching between groups, a physical offset is applied to the first track in the new group. This allows the target block to be the first block available for reading or writing immediately after the seek or head switch has been accomplished.

The amount of offset is dependent upon drive design. It is referred to as **group offset** because we are effectively switching between groups.

2.7.5 Implementation of Logical Addressing

Two major factors were considered when deciding upon the selection of the logical addressing schemes in conjunction with the specific design characteristic for these drives.

1. The selection scheme must satisfy the DSDF specification for both addressing and selecting or switching time requirements.
2. The selection scheme must provide for the fastest and most efficient throughput when operated in a system environment. This means getting the most amount of data to and from the disk in the least amount of time when using typical operating systems such as VMS, RSTS, etc.

2.7.5.1 RA80 Logical Addressing

The RA80 implements following geometry for logical addressing:

- 1 logical track = 1 physical track
- 1 logical group = 14 logical tracks (1 physical cylinder)
- 1 logical cylinder = 2 logical groups (2 physical cylinders)

Figure 2–8 and Figure 2–9 show the RA80 logical addressing and geometry.

Figure 2–8: RA80 Logical Disk Addressing

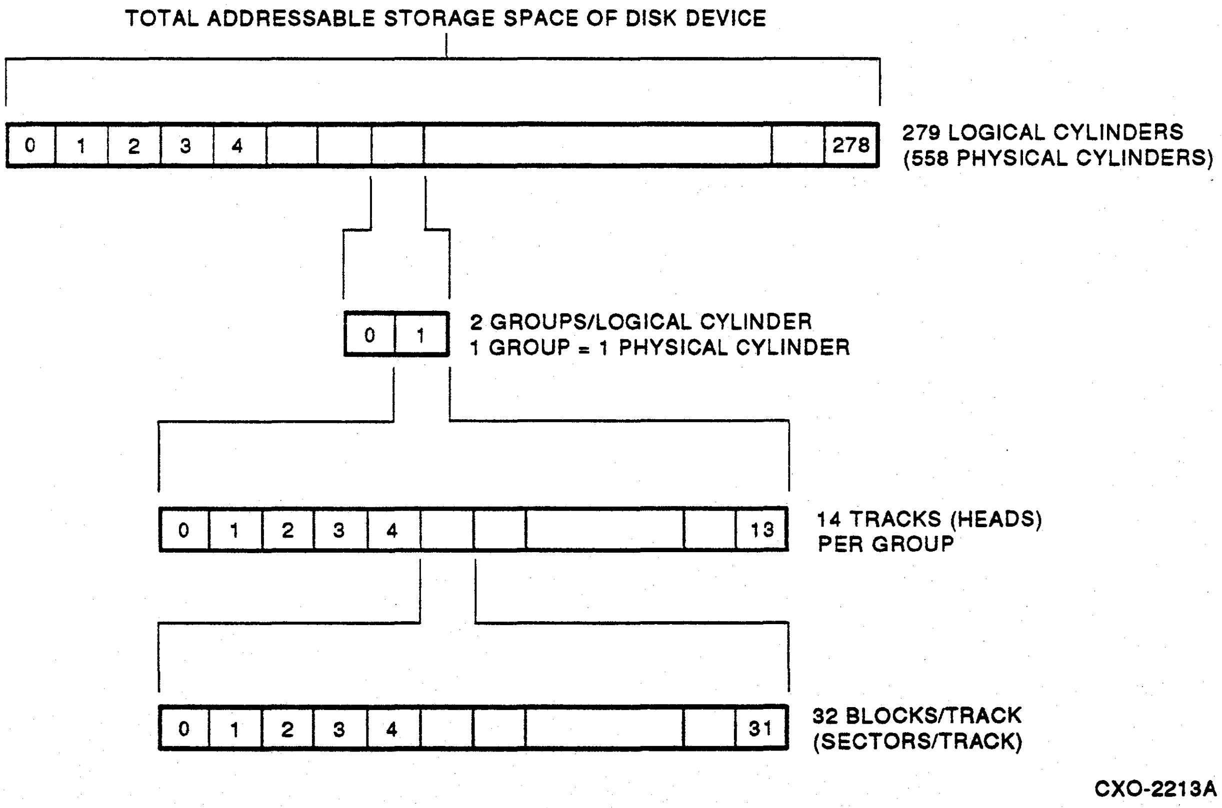
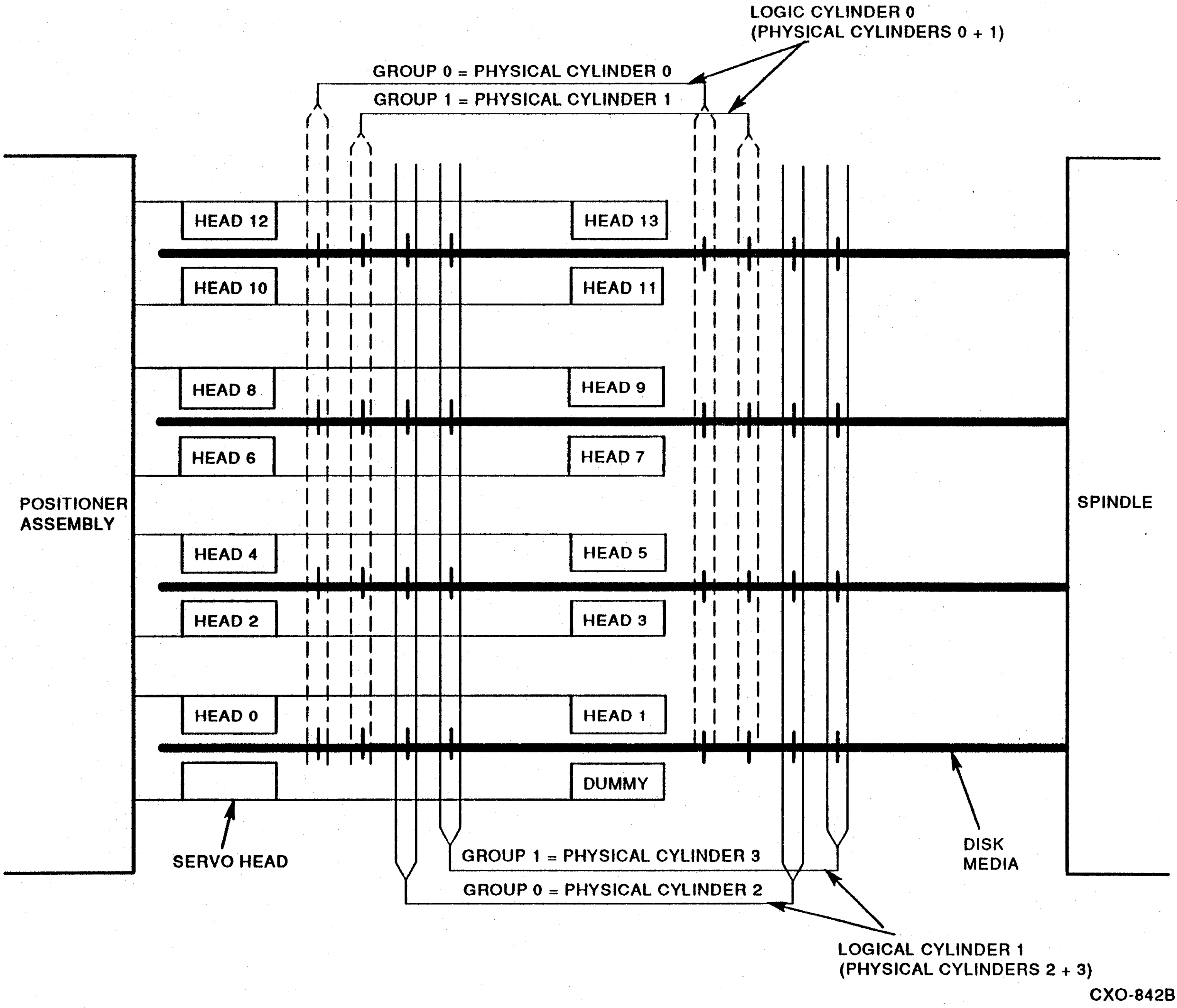


Figure 2–9: RA80 Geometry



The RA80 uses dedicated servo for both coarse and fine positioning control. This dedicated servo data is always in control of the positioning. When performing a head switch in the RA80, we can immediately begin reading data from the next sequential head without any significant latency required to re–establish fine positioning.

Since the latency required for a head switch is less than the intersector gap time, a group in the RA80 is equivalent to 14 tracks.

To switch from a group on one physical cylinder to a group on another adjacent physical cylinder in the RA80 requires the drive to switch heads and internally perform a one–cylinder seek. The RA80 can perform these actions simultaneously as there is little or no additional head settling time required. Therefore, the RA80 is able to equate two groups (or two physical cylinders) into one logical cylinder. This provides the ability to select adjacent physical cylinders using a SELECT GROUP command (requires one SDI command frame from the controller) as opposed to a SEEK command (requires seven SDI command frames from a controller).

You are probably wondering why three physical cylinders (3 groups) were not incorporated into one logical cylinder. One reason is simple: DSDF indicates that the time required to select any new group within a logical cylinder must be less than the time to select a new physical cylinder. Switching from a group 0 address to a group 2 address on a logical cylinder is really two physical cylinders in the RA80. Obviously, a two cylinder seek requires more time than a one cylinder seek.

2.7.5.2 RA81 Logical Addressing

The RA81 implements the following geometry for logical addressing:

- 1 logical track = 1 physical track
- 1 logical group = 1 logical track
- 1 logical cylinder = 14 logical groups (1 physical cylinder)

Figure 2–10 and Figure 2–11 show the RA81 logical addressing and geometry.

Figure 2–10: RA81 Logical Disk Addressing

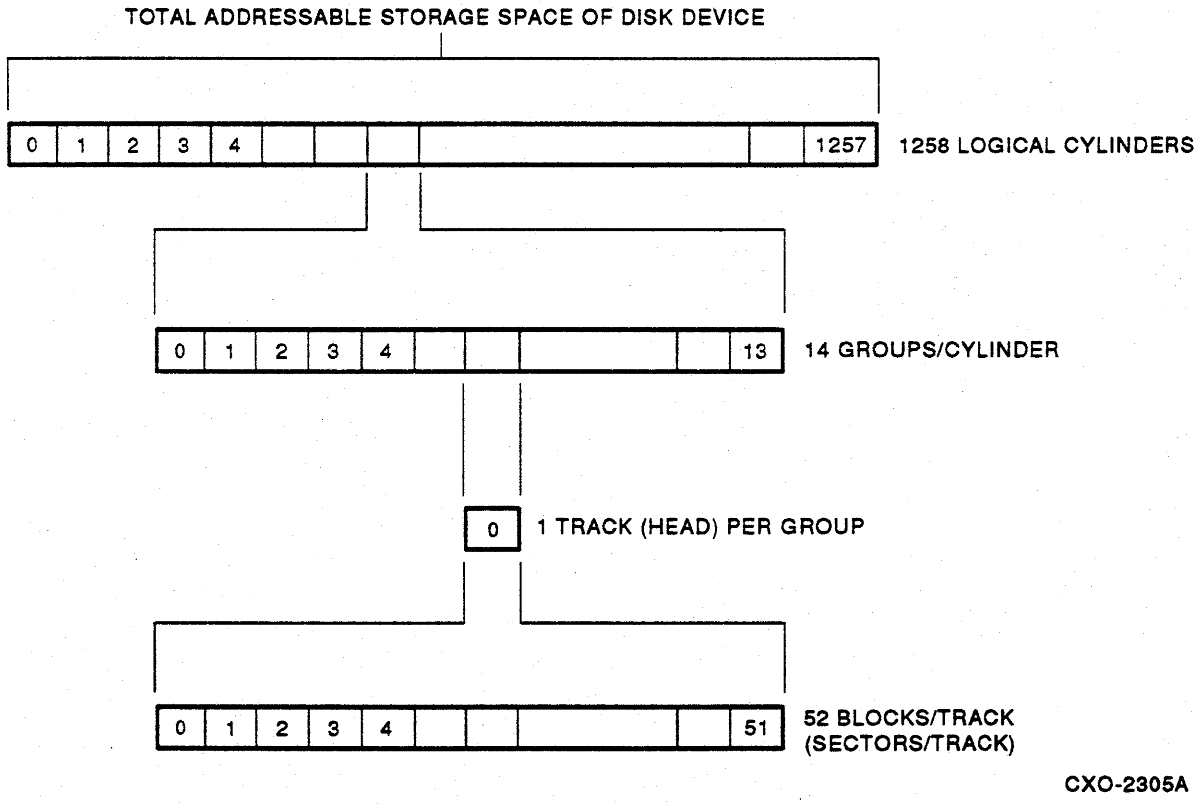
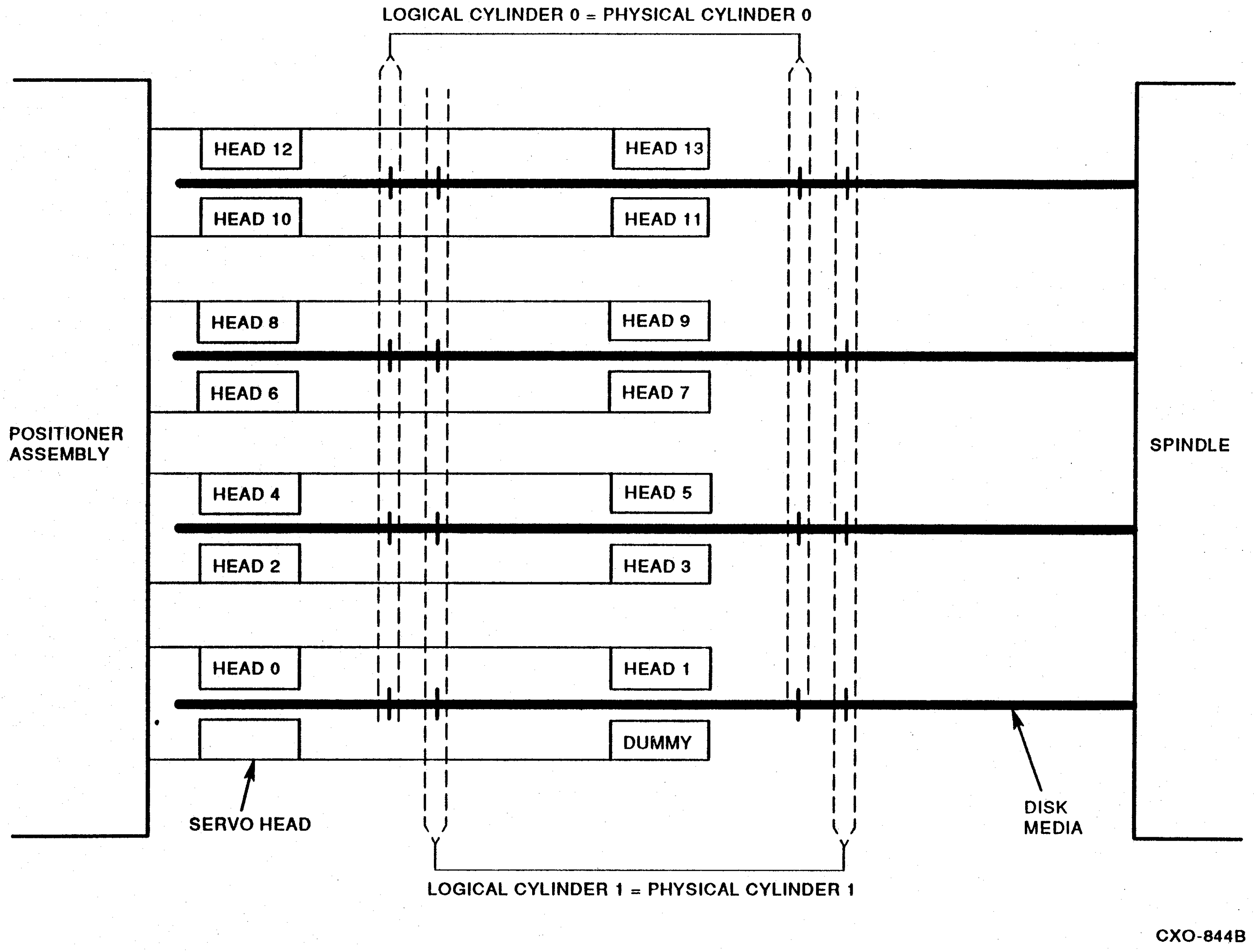


Figure 2–11: RA81 Geometry



The RA81 has dedicated servo for coarse positioning. The increased data and cylinder densities in the RA81 require a more precise mechanism for fine positioning. Therefore, the RA81 incorporates embedded servo to accomplish a finer positioning scheme. The embedded servo exists between every sector on every track. When performing a head switch, the RA81 servo logic must read several embedded servo bursts from a track to establish fine position before it can continue reading or writing.

The time to accomplish a head switch is obviously greater than the intersector gap time. Therefore, the RA81 equates one track per group.

To switch from a group on one physical cylinder to a group on another adjacent physical cylinder in the RA81 requires the drive to switch heads and internally perform a one–cylinder seek. This requires the RA81 to settle (fine position using embedded servo) after performing the seek. The time required to do this is obviously greater than just the seek itself. Therefore, the RA81 equates 14 groups to 1 physical and logical cylinder.

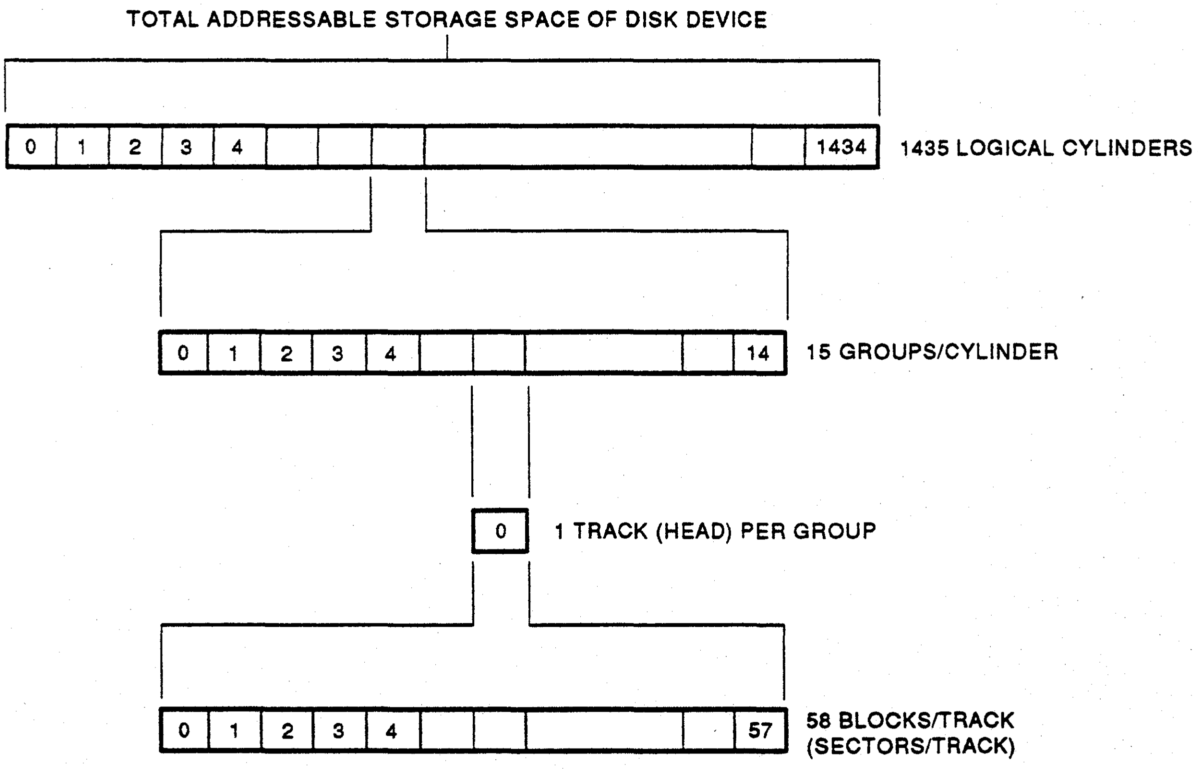
2.7.5.3 RA82 Logical Addressing

The RA82 implements the following geometry for logical addressing:

- 1 logical track = 1 physical track
- 1 logical group = 1 logical track
- 1 logical cylinder = 15 logical groups (1 physical cylinder)

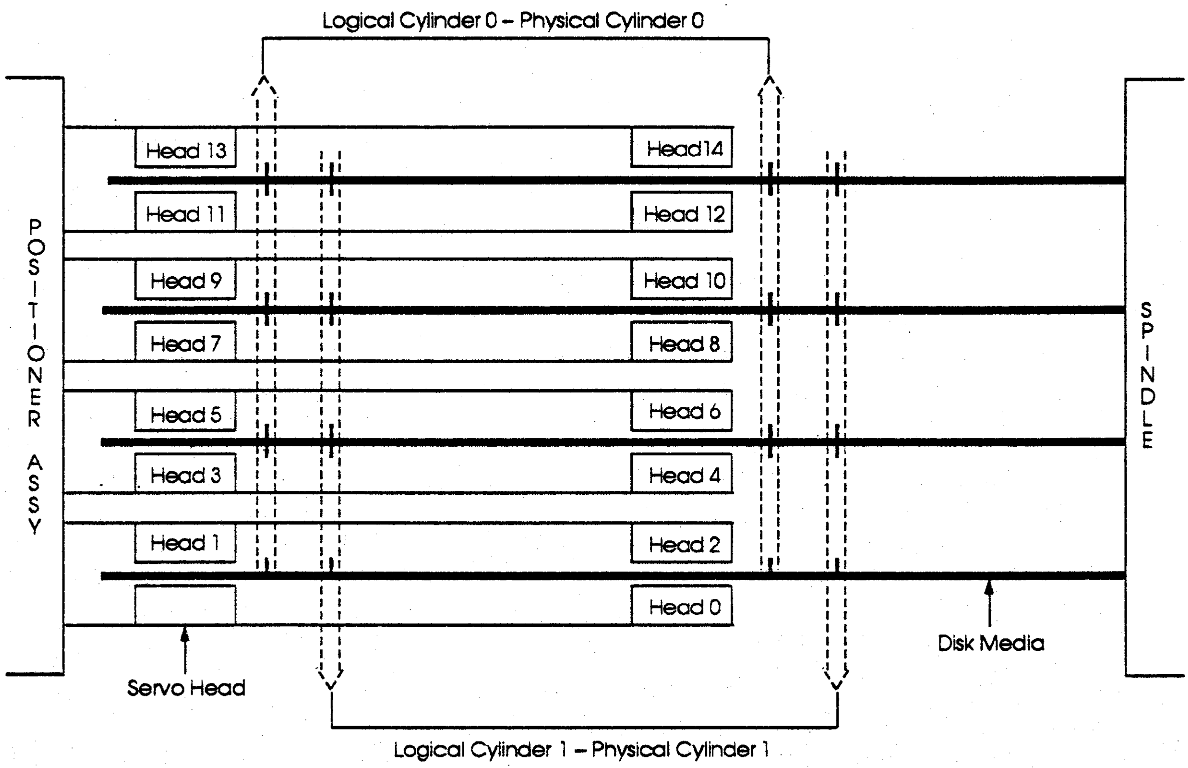
Figure 2–12 and Figure 2–13 illustrate the RA82 logical addressing and geometry.

Figure 2–12: RA82 Logical Disk Addressing



CXO-1500B

Figure 2–13: RA82 Geometry



MLDS-1344A

The RA82 has dedicated servo for coarse positioning. The large amount of data and cylinder densities in the RA82 require a more precise mechanism for fine positioning. Therefore, the RA82 incorporates embedded servo to accomplish a fine positioning scheme. The embedded servo exists between every sector on every track. When performing a head switch, the RA82 servo logic must read several embedded servo bursts from a track to establish fine position before it can continue reading or writing.

The time to accomplish a head switch is obviously greater than the intersector gap time. Therefore, the RA82 equates one track per group.

To switch from a group on one physical cylinder to a group on another adjacent physical cylinder in the RA82 requires the drive to switch heads and internally perform a one–cylinder seek. This requires the RA82 to settle (fine position using embedded servo) after performing the seek. The time required to do this is obviously greater than just the seek itself. Therefore, the RA82 equates 15 groups to 1 physical and logical cylinder.

2.7.5.4 RA70 Logical Addressing

The RA70 implements the following geometry for logical addressing:

- 1 logical track = 1 physical track
- 1 logical group = 1 logical track
- 1 logical cylinder = 11 logical groups (1 physical cylinder)

Figure 2–14 and Figure 2–15 show the RA70 logical addressing and geometry.

Figure 2–14: RA70 Logical Disk Addressing

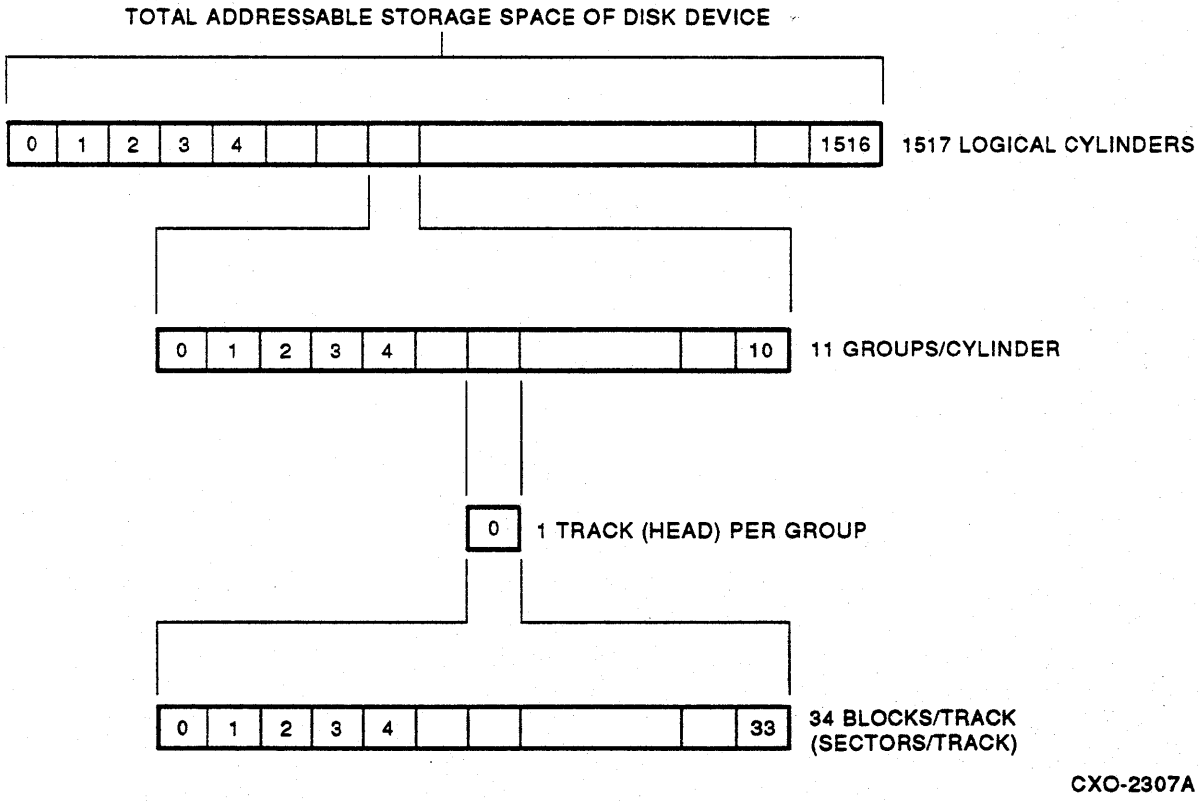
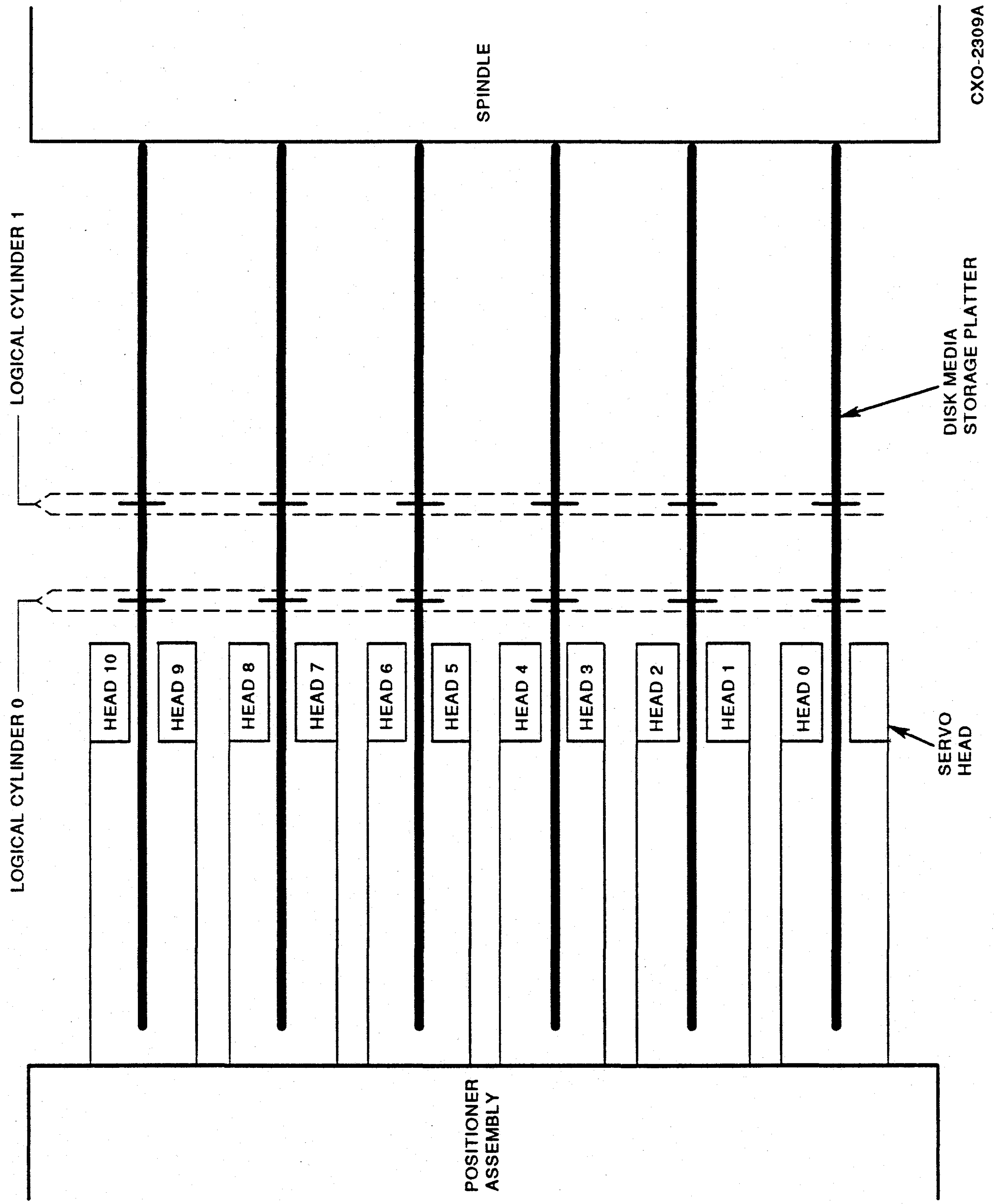


Figure 2–15: RA70 Geometry



The RA70 has dedicated servo for coarse positioning. The large amount of data and cylinder densities in the RA70 require a more precise mechanism for fine positioning. Therefore, the RA70 incorporates embedded servo to accomplish a fine positioning scheme. The embedded servo exists between every sector on every track. When performing a head switch, the RA70 servo logic must read several embedded servo bursts from a track to establish fine position before it can continue reading or writing.

The time to accomplish a head switch is obviously greater than the intersector gap time. Therefore, the RA70 equates one track per group.

To switch from a group on one physical cylinder to a group on another adjacent physical cylinder in the RA70 requires the drive to switch heads and internally perform a one–cylinder seek. This requires the RA70 to settle (fine position using embedded servo) after performing the seek. The time required to do this is obviously greater than just the seek itself. Therefore, the RA70 equates 11 groups to 1 physical and logical cylinder.

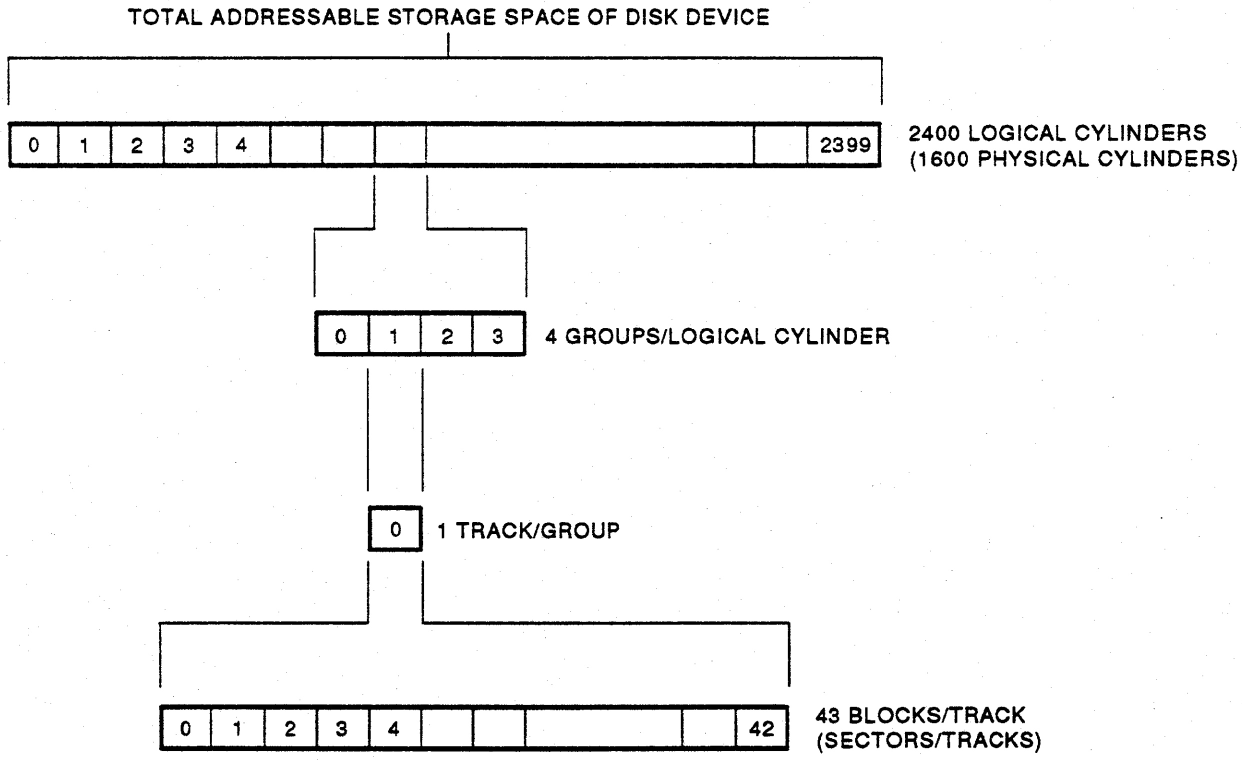
2.7.5.5 RA60 Logical Addressing

The RA60 implements the following geometry for logical addressing:

- 1 logical track = 1 physical track
- 1 logical group = 1 logical track
- 1 logical cylinder = 4 logical groups (4 physical tracks)

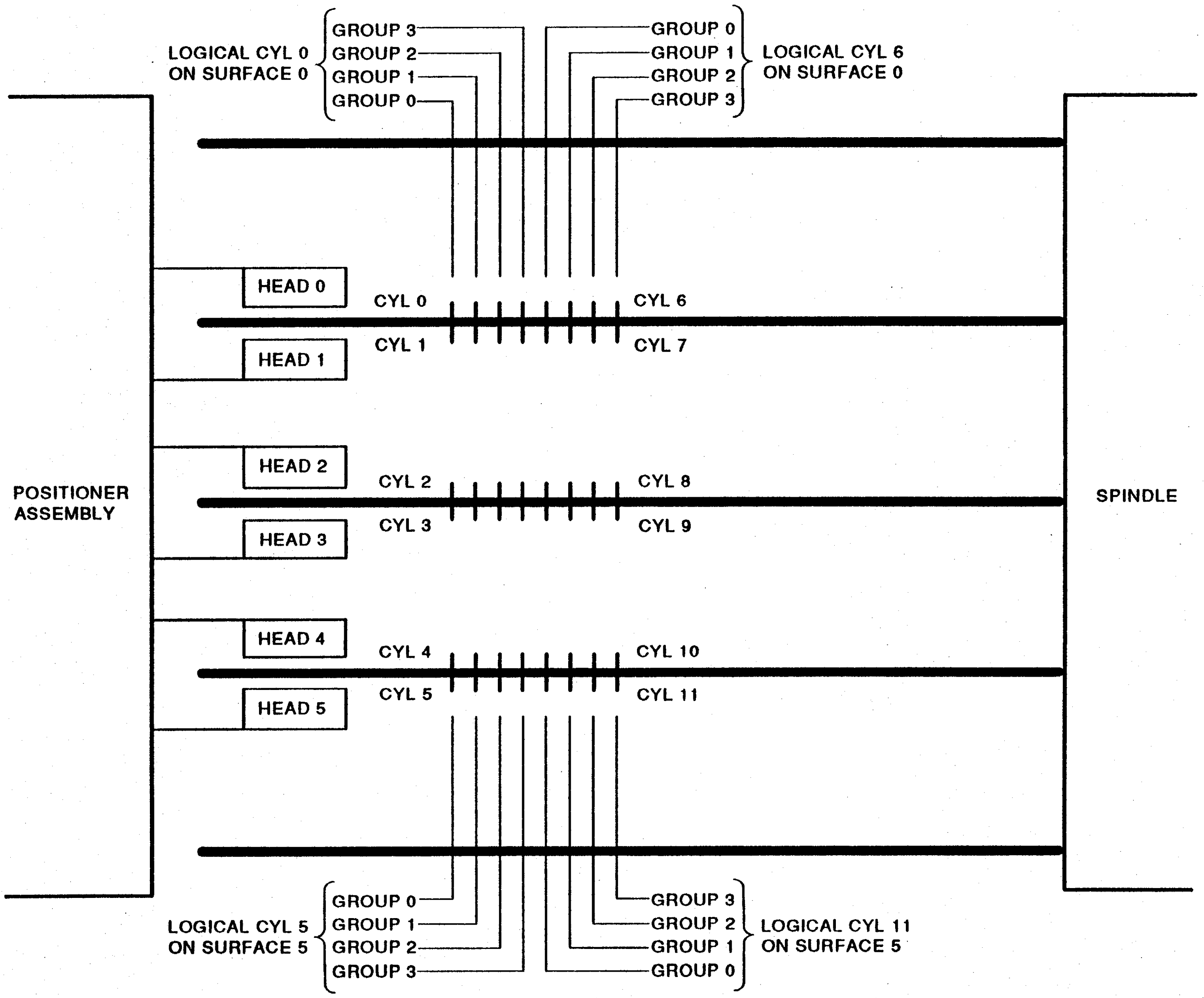
Figure 2–16 and Figure 2–17 show the RA60 logical addressing and geometry.

Figure 2–16: RA60 Logical Disk Addressing



CXO-2326A

Figure 2–17: RA60 Geometry



The RA60 does not incorporate a dedicated servo surface but relies upon embedded servo information to perform servo positioning. The embedded servo exists between every sector on every track. When performing a head switch, the RA60 servo logic must read several embedded servo bursts from a track to establish fine position before it can continue reading or writing.

The time to accomplish a head switch is obviously greater than the intersector gap time. Therefore, the RA60 equates one track per group.

Another unique characteristic of the RA60 is that it can seek faster than it can change head selection. After performing a variety of design evaluations under different operating system environments, a unique physical geometry was established to allow optimum performance of the RA60.

This is better understood by reviewing Figure 2–17. A logical cylinder consists of four adjacent physical tracks (groups) on the same surface read by the same head. For example, logical cylinder 0 consists of groups 0, 1, 2, and 3 all on disk surface 0, read by R/W head number 0. Physically, this equates to the first four physical tracks on surface 0. Logical cylinder 1 consists of the first four tracks (group 0 through group 3) on physical disk surface 1, read by R/W head number 1. As you can see from studying the diagram, a logical cylinder is quite different from a physical cylinder in the RA60.

2.7.5.6 RA90 Logical Addressing

The RA90 implements the following geometry for logical addressing:

- 1 logical track = 1 physical track
- 1 logical group = 1 logical track
- 1 logical cylinder = 13 logical groups (1 physical cylinder)

Figure 2–18 and Figure 2–19 show the RA90 logical addressing and geometry.

Figure 2–18: RA90 Logical Disk Addressing

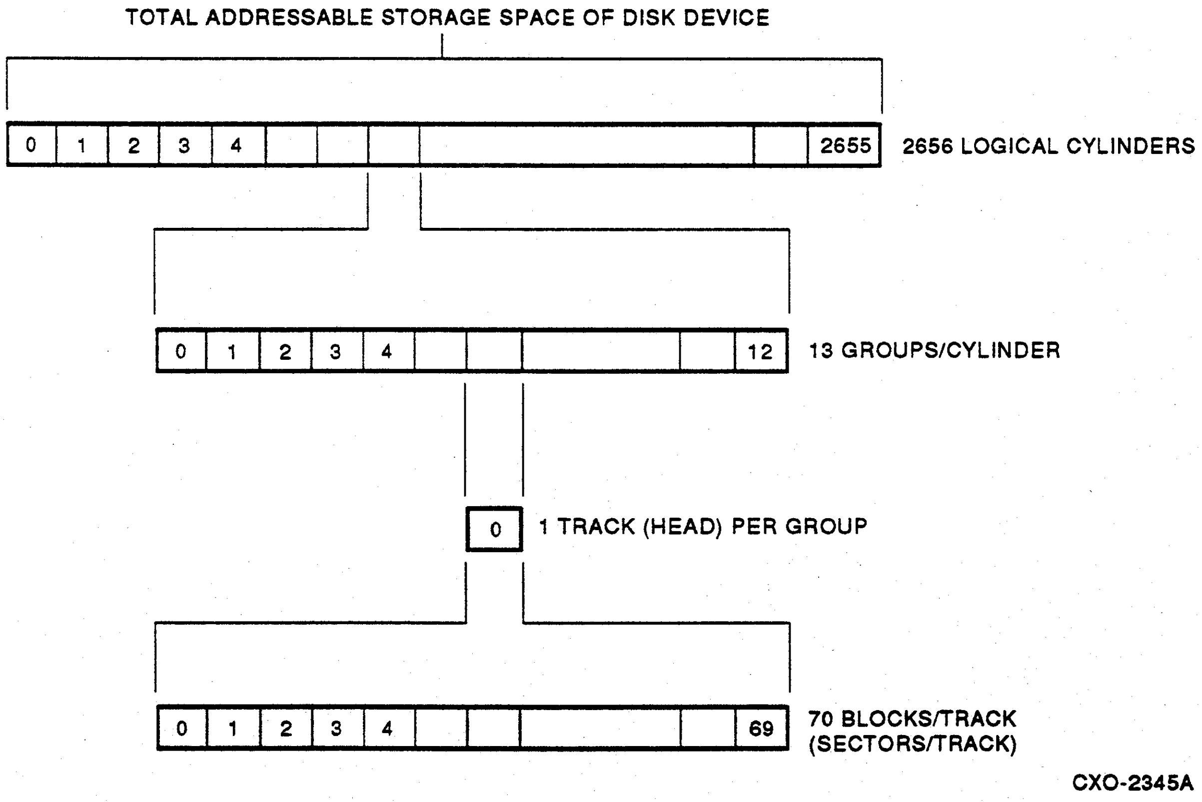
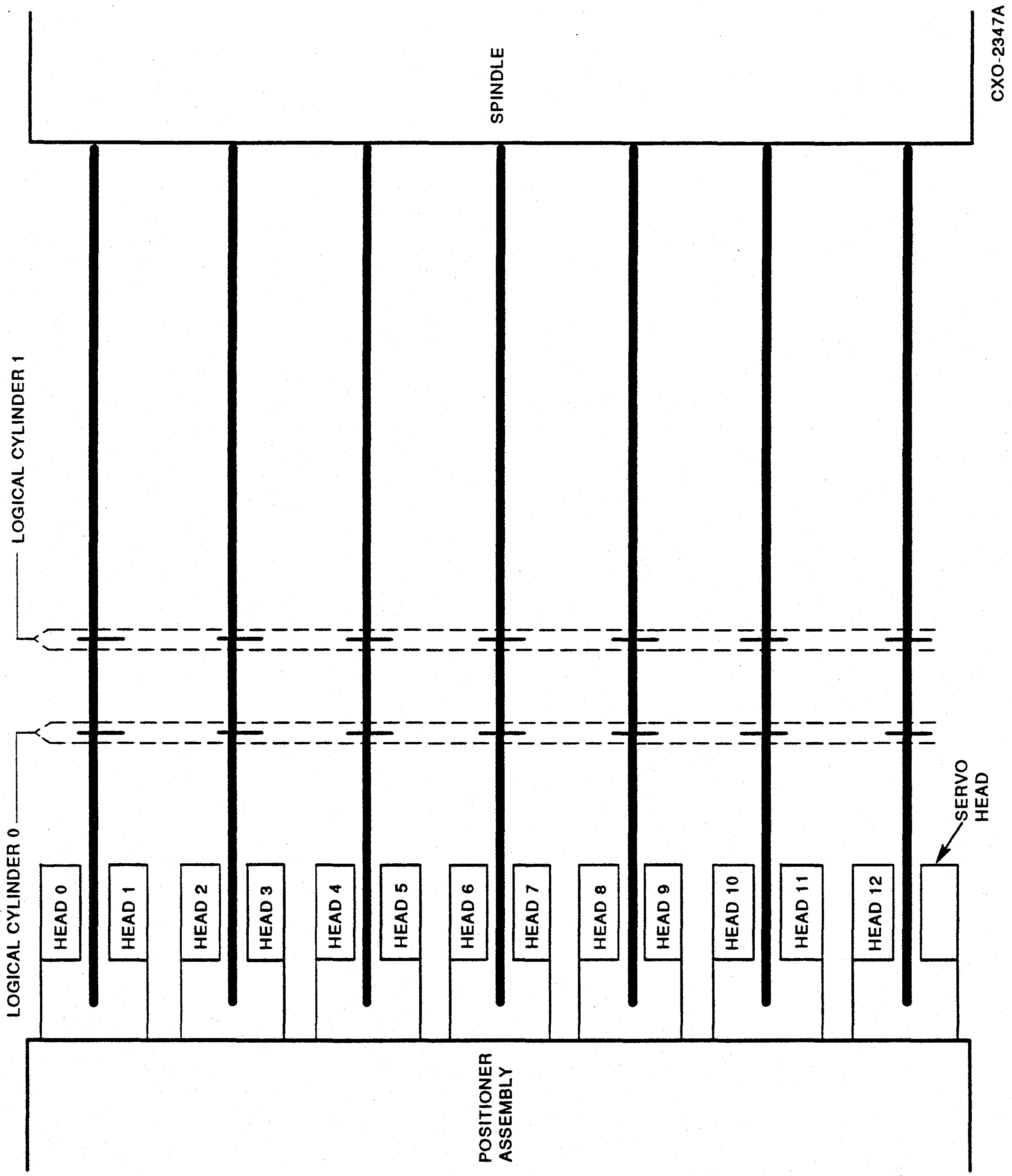


Figure 2–19: RA90 Geometry



The RA90 has dedicated servo for coarse positioning. The large amount of data and cylinder densities in the RA90 require a more precise mechanism for fine positioning. Therefore, the RA90 incorporates embedded servo to accomplish a fine positioning scheme. The embedded servo exists between every sector on every track.

The time to accomplish a head switch is obviously greater than the intersector gap time. Therefore, the RA90 equates one track per group.

To switch from a group on one physical cylinder to a group on another adjacent physical cylinder in the RA90 requires the drive to switch heads and internally perform a one–cylinder seek. This requires the RA90 to settle (fine position using embedded servo) after performing the seek. The time required to do this is obviously greater than just the seek itself. Therefore, the RA90 equates 13 groups to 1 physical and logical cylinder.

2.8 LOGICAL AREAS and LOGICAL BLOCKS

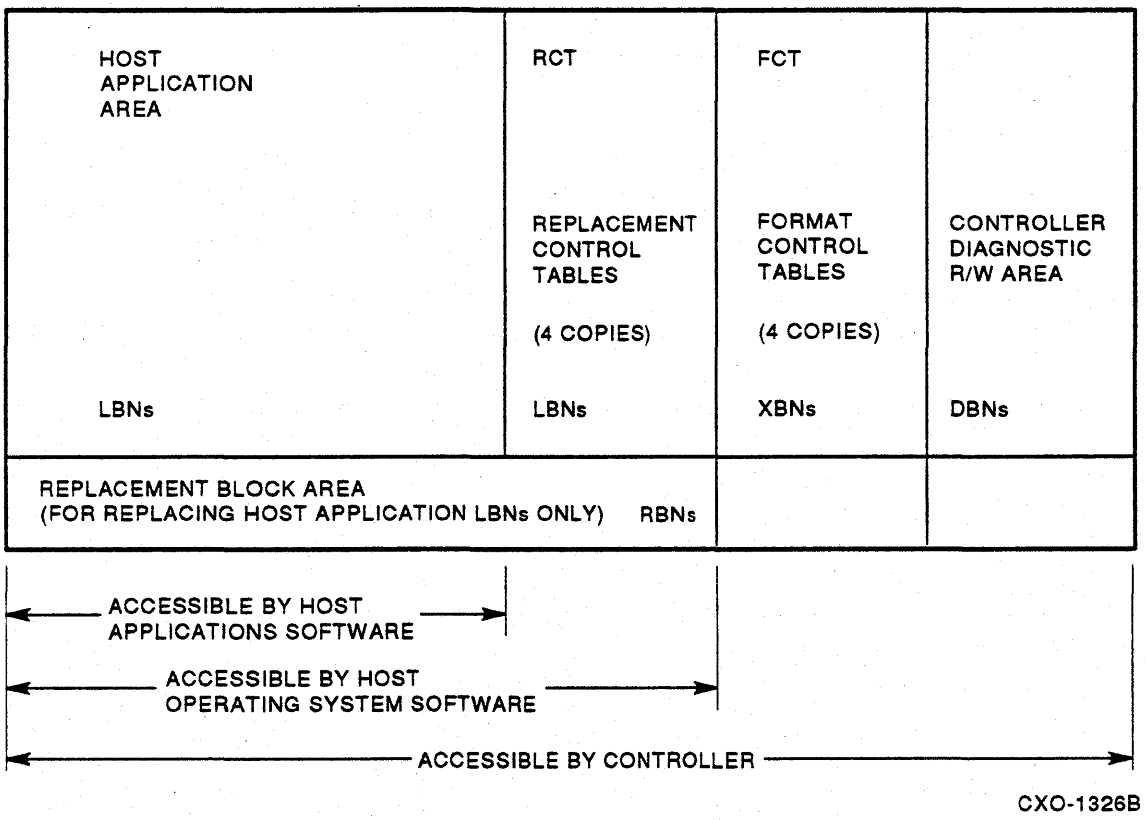
Information on the data recording surfaces of the media is logically organized according to the Digital Standard Disk Format (DSDF) specification. This specification standardizes and defines how a Digital Storage Architecture (DSA) device appears to the host processor and the controller to which it is attached.

The total data storage area of the disk media is divided into physical sectors. In the disks discussed in this document, a sector is equivalent to a logical block. We have already seen how the physical attributes of disk addressing are translated to logical attributes.

The DSA architecture also provides for logical organization of the data blocks on the disk, regardless of the addressing attributes selected. The number of logical blocks is divided into logical areas.

The following paragraphs describe these areas. Figure 2–20 illustrates the areas of the basic topology.

Figure 2–20: Basic Topology



2.8.1 Host Application Area (LBNs)

The host application area is the largest area containing data blocks for use by normal host operating applications as well as system operating software. This is the area where users store data files and/or programs. System files and system operating software are also stored here. This is the normal working area of the disk.

Blocks in the host application area are addressed as logical block numbers (LBNs). This area is also sometimes referred to as the normal LBN area.

2.8.2 Replacement Block Area (RBNs)

Blocks within the replacement block area are used to replace defective blocks in the host applications area. When a block in the normal LBN area becomes unusable, the host operating system or the controller may substitute a replacement block for the defective LBN in the host application area. This is accomplished by a process called Bad Block Replacement (BBR). BBR is discussed later in this course. Blocks in the replacement block area are addressed as RBNs. If an RBN becomes defective, another RBN may be used in its place. RBN's are located in the last logical sector of every track in both the host and RCT area.

2.8.3 Replacement Control Table (RCT) Area (LBNs)

Blocks within the replacement control table area contain information that allows the controller and/or the host operating system software to find blocks from the normal LBN area that have been replaced by blocks in the replacement area. The tables also contain information that identify which RBNs are still available when performing bad block replacement.

Blocks in RCT are also addressed as LBNs. Blocks in this area are not available for access by normal host applications. These blocks are only accessible by the controller and/or host operating system software (VMS, RSTS, etc.).

Blocks in the RCT are not replaced by RBNs when they become defective. For this reason, multiple copies of the RCT tables are maintained in the RCT area. This redundancy permits backup protection in the event that any blocks in this area become unusable.

2.8.4 Format Control Tables (FCT) Area (XBNs, External Blocks)

Blocks within the format control table area contain the following information:

- Media serial number.
- Date of initial factory formatting.
- Date of most recent formatting.
- Mode that the media/HDA was formatted. The HDA/pack is available from the factory in either 16–bit mode (512 8–bit bytes per sector) or 18–bit mode (576 8–bit bytes per sector). PDP–11 and VAX processors, for example, require 16–bit mode media, and DECsystem–10 and DECsystem–20 processors require 18–bit mode media. The format mode of an HDA/pack cannot be changed in the field.
- Information to indicate if the rest of the FCT structure contains any valid data.
- Location of the manufacturing–detected bad blocks (sectors).

NOTE

The RA60, RA80, and RA81 are available in both 16–bit and 18–bit configurations.

The RA70, RA82, and RA90 are available in only the 16–bit configuration.

During the manufacture of an HDA, special factory scanners are used to locate defective blocks found in the media. During factory formatting, this information is recorded into the FCT. Special formatter programs and/or utilities executing within the controller use the manufacturing–detected bad block information in the FCT to create, re–create, or verify the RCT and replace the bad blocks known to exist when the HDA was manufactured. These special programs are either resident or loaded into the controller from the host for execution, but only upon manual request.

Blocks in the FCT are not replaced by RBNs when they become defective. For this reason, multiple copies of the FCT tables are maintained in the FCT area. This redundancy permits backup protection in the event that any blocks in this area become unusable.

Blocks in the FCT are addressed as external block numbers (XBNs). Blocks in the FCT are only accessible by the controller. Most external blocks within the FCT contain data that is used to physically locate manufacturing–detected bad blocks (sectors) on the media. This special data within an XBN is referred to as a physical block number (PBN).

2.8.5 Diagnostic Area (Diagnostic Block Numbers)

Blocks in this area are used by the controller to perform read/write diagnostics to the disk drive. Blocks in this area are addressed as diagnostic block numbers (DBNs).

Blocks in the DBN area are only accessible by the controller. Blocks in this area are not replaced by RBNs when they become defective. Also, no hardware provisions nor DSA specifications govern the protection against defective DBNs. It is the responsibility of the controller and its specially loaded diagnostic software to handle unusable DBNs.

Reformatting the DBN area in the field may restore a previously defective DBN, but use this process with caution.

Read data, write data, or format operations to the DBN area require execution of special diagnostic/formatting microcode within the controller itself. Depending upon the controller type, this code may be resident or loaded into the controller from the host. Manual intervention is usually required to invoke these operations.

2.9 IMPLEMENTATION OF LOGICAL AREAS

2.9.1 Drive Topology Maps

Figure 2–21 through Figure 2–28 illustrate how the topology is implemented into the various disks. For example, refer to Figure 2–25 for a map of the RA82 topology. This diagram illustrates how the logical areas are mapped into the RA82 physical environment for a 16–bit formatted HDA.

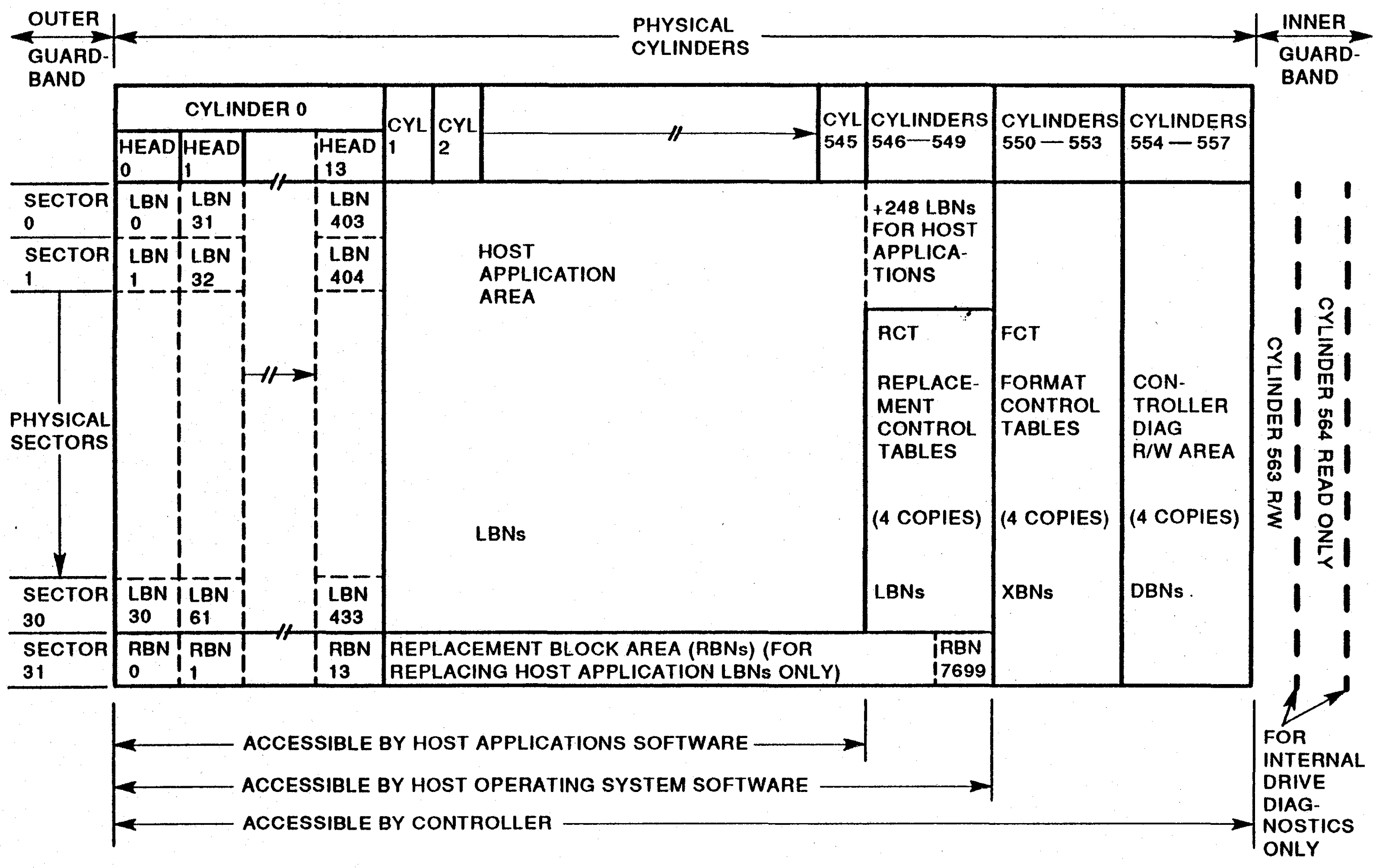
Here we show the physical cylinders across the top of the diagram and physical sectors down the left side. Physical cylinder 0 has been further expanded for clarification. All the sectors within track 0 under head 0 appear in the column marked Head 0. The logical block assignment for LBN 57 in the host applications area, for example, appears at the intersection of sector 0 and head 1 (track 1), etc.

NOTE

These figures are for training purposes only. They do not show the implementation of group offset. Illustrating group offset would cause a more complex representation and result in confusion.

Notice that the last sector in each track is allocated as a replacement block (RBN). Even though the RBN area extends physically into the cylinders allocated for the RCT, RBNs are only used to replace bad LBN blocks for the host application area. The RBNs are assigned numbers independently of LBN numbers. Bad block replacement will be discussed later.

Figure 2–21: RA80 Topology – 16–Bit Format



CXO-847B

Figure 2–22: RA80 Topology – 18–Bit Format

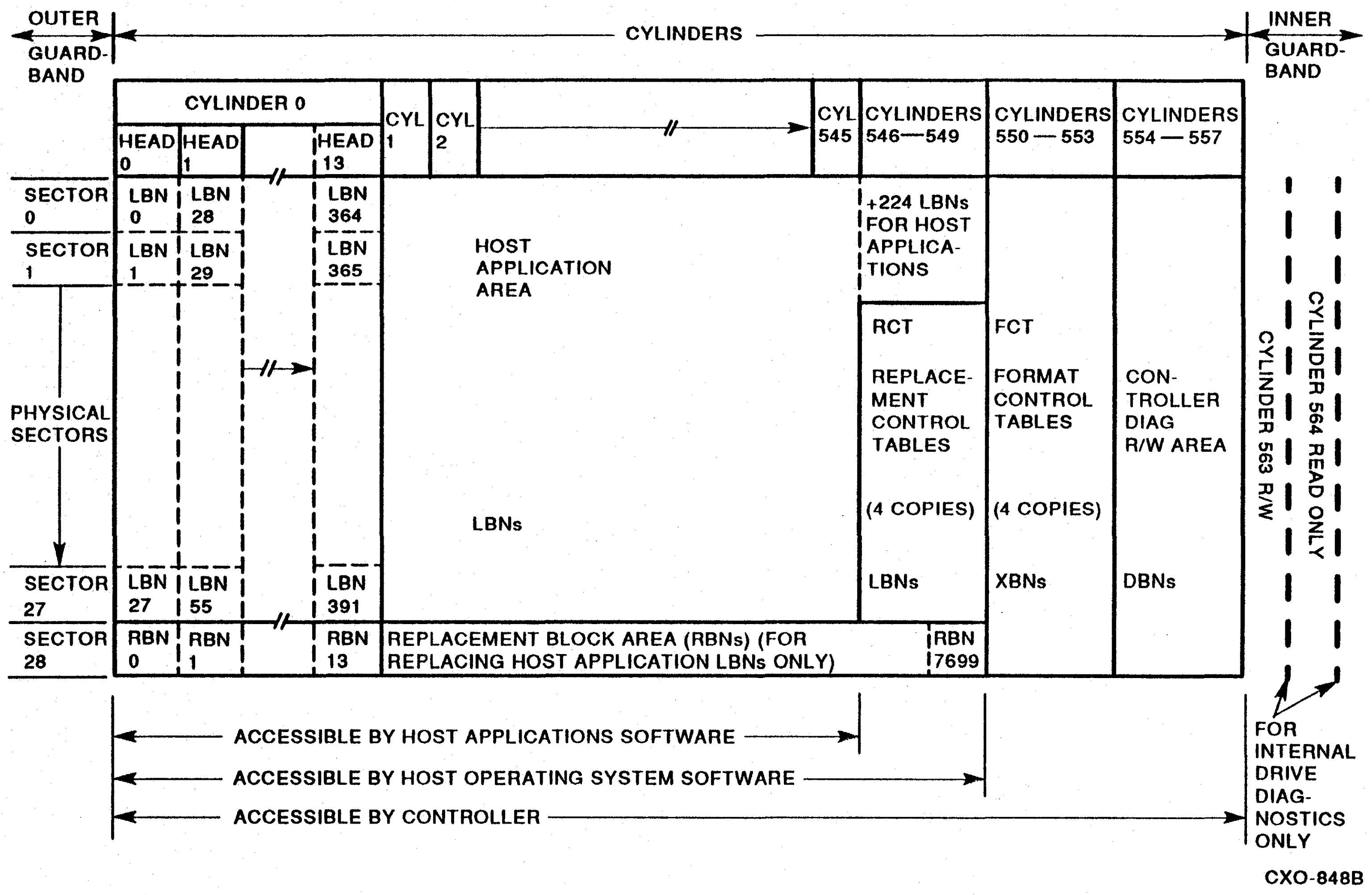
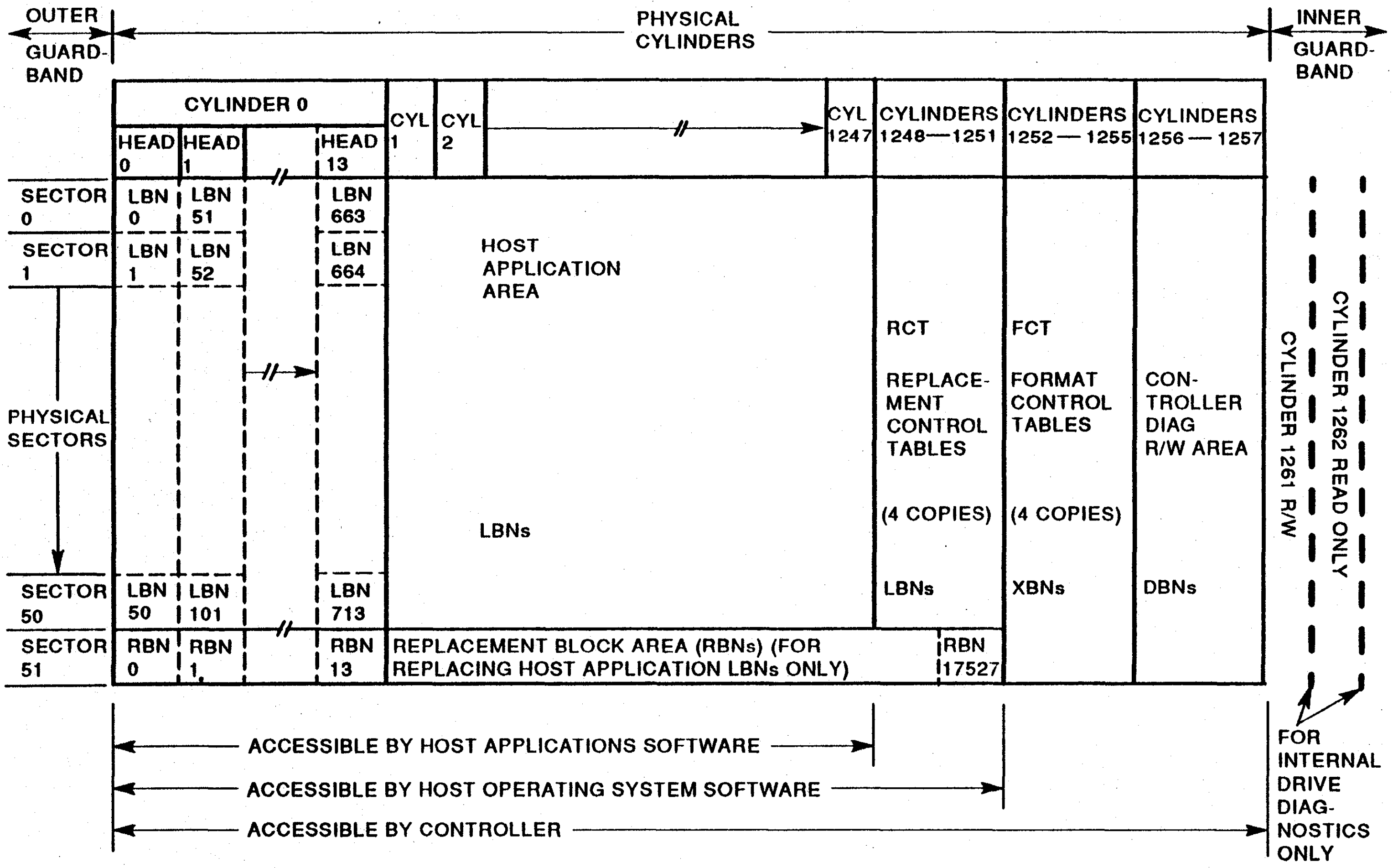


Figure 2–23: RA81 Topology – 16–Bit Format



CXO-849B

Figure 2–24: RA81 Topology – 18–Bit Format

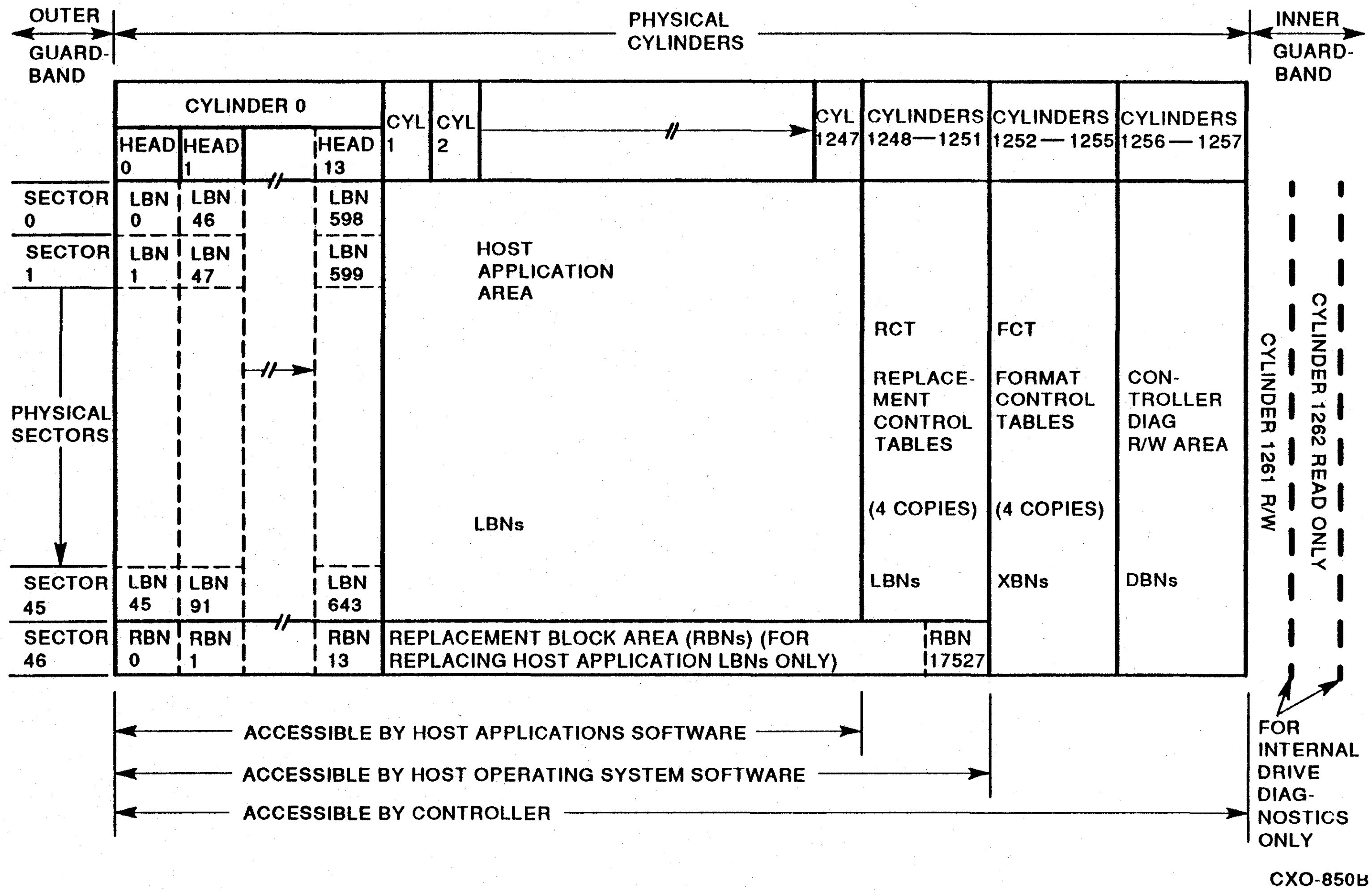


Figure 2–25: RA82 Topology – 16–Bit Format

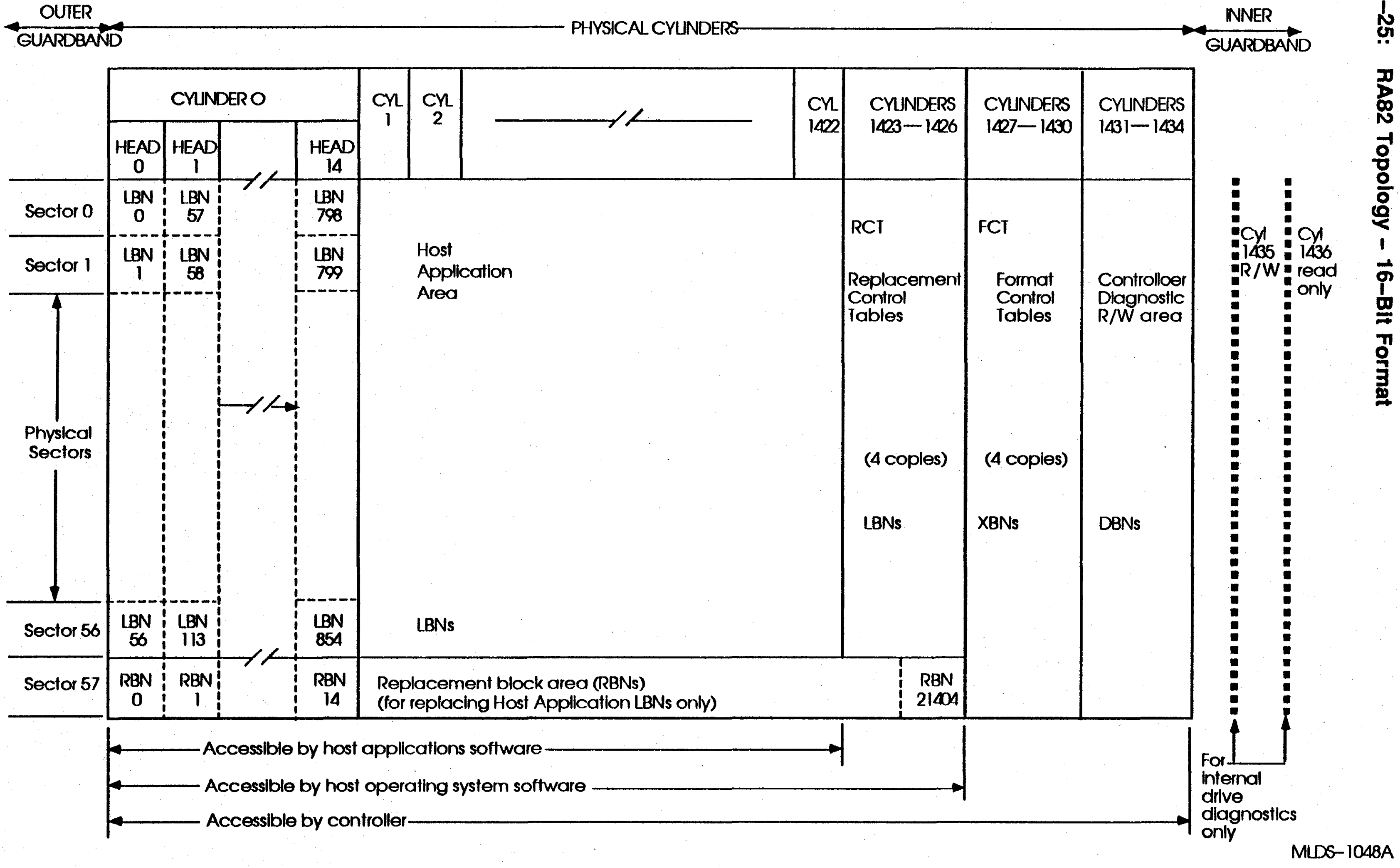


Figure 2–26: RA70 Topology

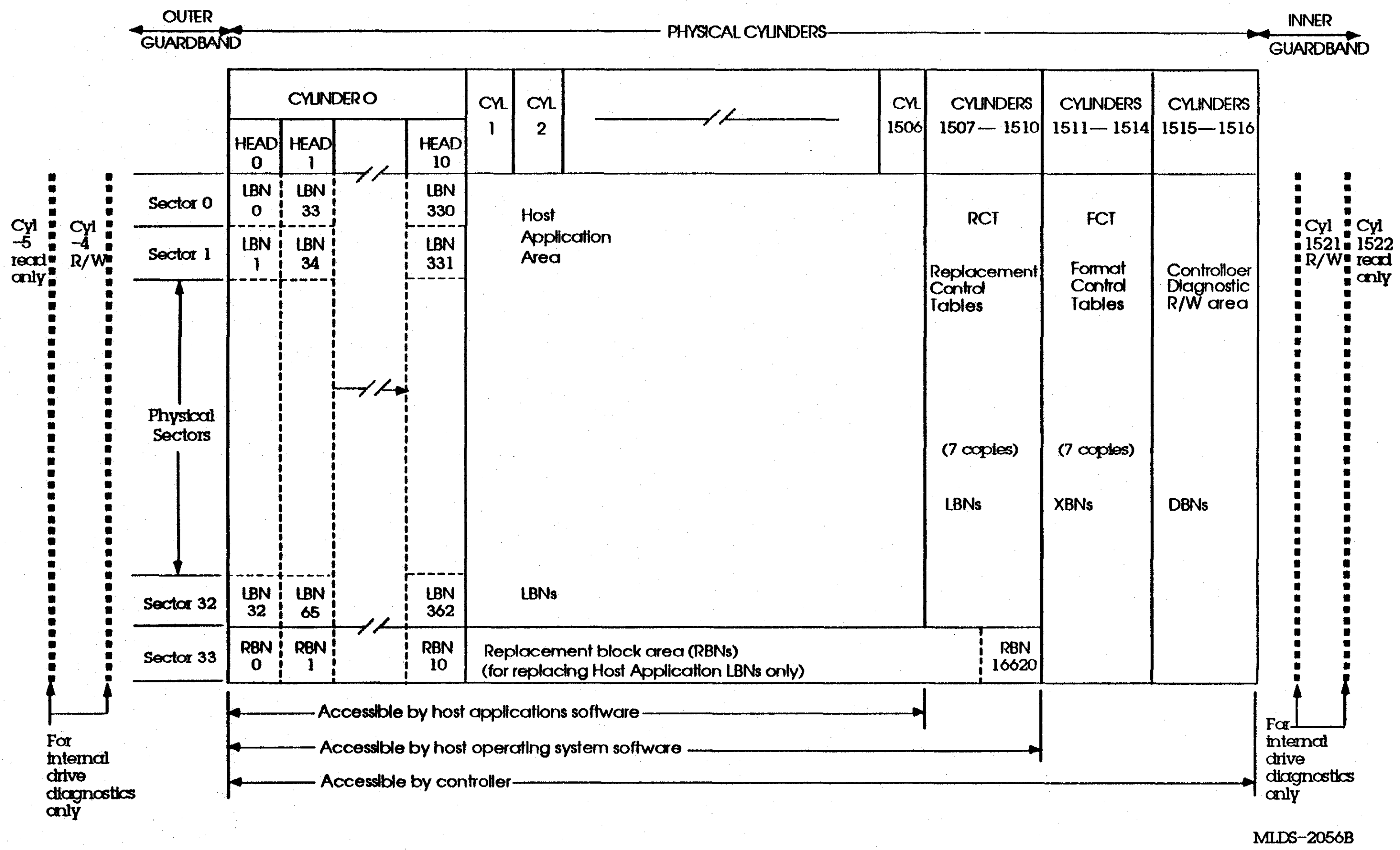
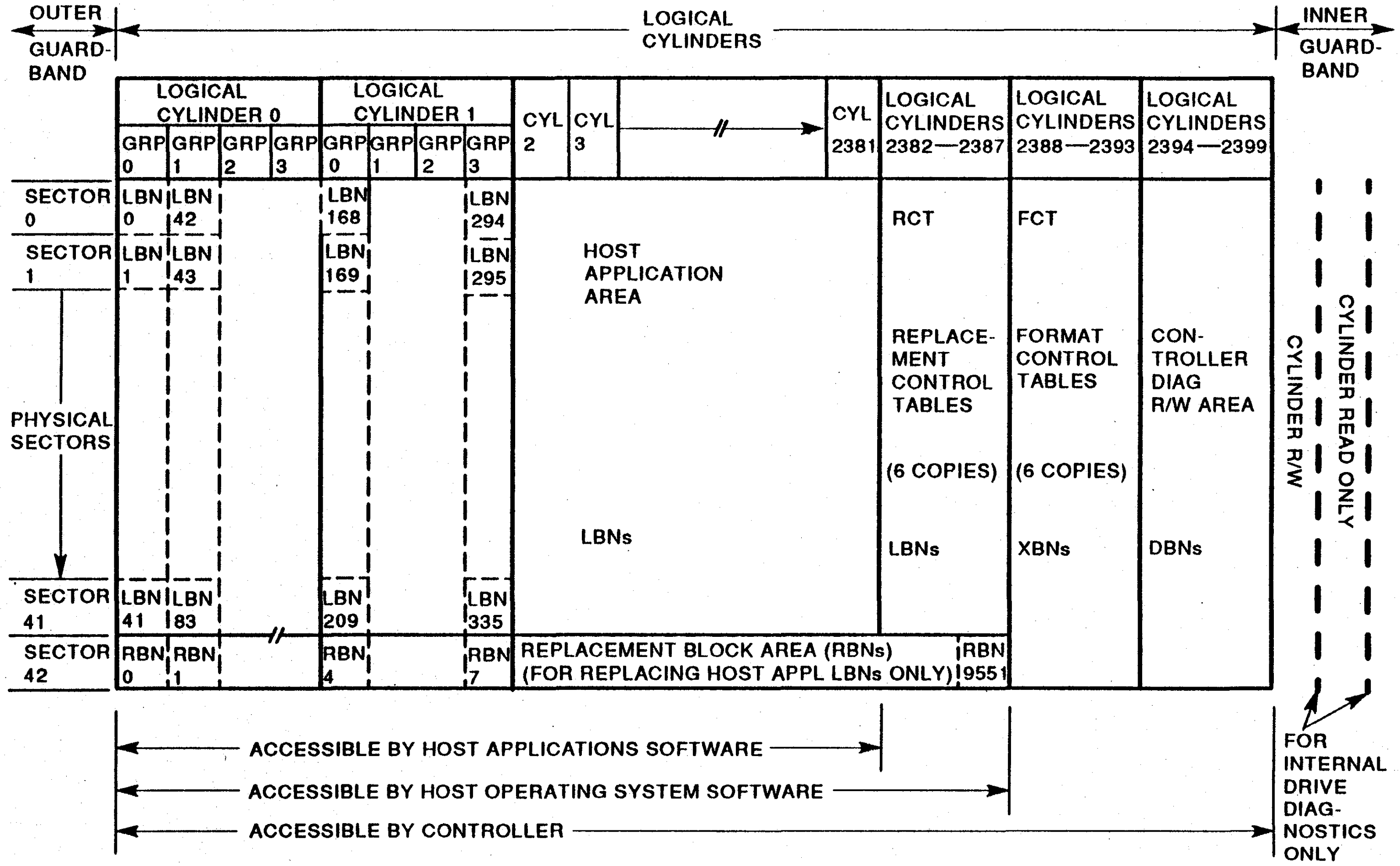
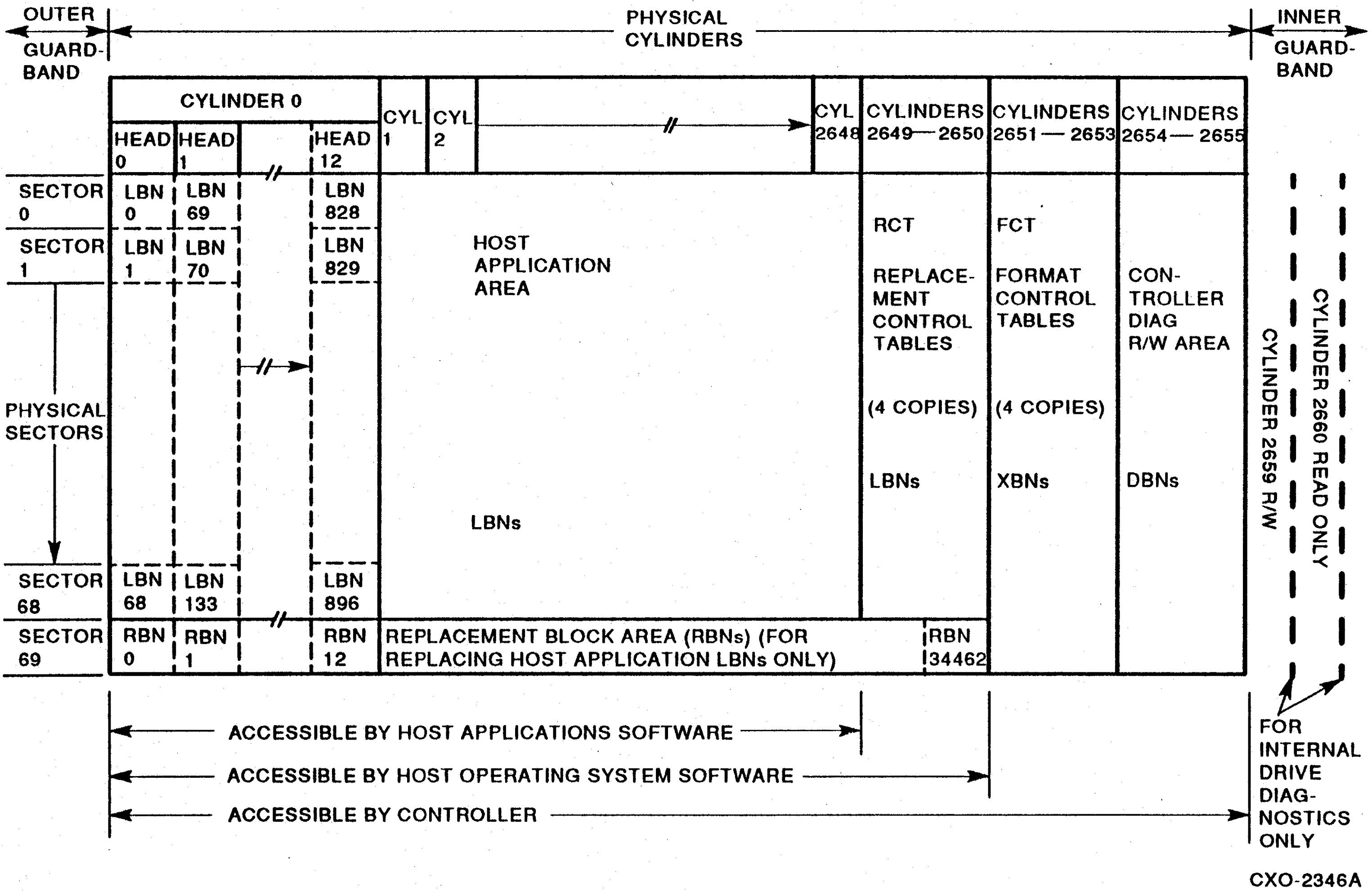


Figure 2–27: RA60 Topology



CXO-2343A

Figure 2–28: RA90 Topology – 16–Bit Format



2.10 DRIVE INTERNAL DIAGNOSTIC AREA

Many of the DSA drives provide special cylinders for use by the drive–specific internal diagnostics. These are also shown in Figure 2–21 through Figure 2–28 for the various drives.

For example, refer to Figure 2–25. The RA82 provides two additional physical cylinders located within the inner guard band area of the disk.

These additional cylinders are used for drive–internal diagnostics and are only available to the internal microcode of the drive itself. One cylinder is specially formatted and is used for internal read–only testing. A special utility resident within the RA82 permits reformatting the internal read–only cylinder should it become corrupt. The other cylinder is used for internal R/W testing. Notice in Figure 2–26 that the RA70 includes special diagnostic cylinders in both the inner and outer guard band areas of the disk.

Controller commands may invoke the drive internal diagnostics; however, drive internal R/W diagnostics are only performed on the specially allocated internal diagnostic cylinders on the inner guard band.

These internal diagnostic cylinders are not structured according to DSDF specifications. They do not contain special header codes, EDC, or ECC characters. They merely consist of sectors using special data patterns for internal drive testing purposes.

2.11 BAD BLOCK REPLACEMENT (BBR) and REVECTORING

Occasionally, defects in the disk storage media occur and cause sectors (blocks) to become bad. The header may become corrupt causing header–not–found or header–compare errors. The data may become corrupt causing ECC symbol errors. These conditions cause a block to become unusable and result in holes in the disk addressing space.

A technique known as Bad Block Replacement (BBR) was developed to permit replacing a bad logical block with a good replacement block. Once a block is replaced, further attempts to read or write to the bad block (sector) are transferred or *revectored* to the replacement block (sector). This revectoring process is automatic upon each access to the bad block. For this reason, the host always accesses data from a usable block. The disk drive appears to contain a set of contiguous, error–free blocks available to the users or the host.

- Bad block replacement is the process of moving data from a bad sector (block) to another good sector (block) and reassigning the block's address from the bad sector to the replacement sector (block).
- Revectoring is the process in which read data or write data operations to a block that is bad are rerouted to a replacement block during the read or write transfer operation.

2.11.1 Why is BBR Performed?

- To fill holes in the host applications area address space left by bad blocks (sectors).
- To reduce the risk of failure due to progressive deterioration of sectors that have a high ECC symbol error count.
- To improve the performance in applications where the error correction or error recovery mechanisms require more time than the revectoring mechanism.

In the current implementation of DSDF, only logical blocks in the host application area are replaced. Bad blocks in the replacement area can also be replaced by other good blocks in the replacement area. Bad blocks in the RCT, FCT, and DBN areas are not replaced. The RCT and FCT each contain multiple, redundant copies of information to provide protection in the event of detecting a bad block in these areas. The DBN area is currently not protected against bad block events.

2.11.2 When is BBR Invoked?

Bad Block Replacement (BBR) is invoked:

- When a header becomes corrupt resulting in header–compare or header–not–found errors.
- When ECC errors are detected and the number of symbol errors equals or exceeds the threshold defined by the disk drive. The RA81 threshold is currently set at 6 symbol errors.

  For example, if a read operation to an RA81 disk drive resulted in an ECC error with 6 or more symbols in error, BBR would be invoked. If, on the other hand, the same read operation resulted in 5 or less symbols in error, BBR would not be invoked, the data would merely be corrected, and the data error would not be reported to the host.
- When uncorrectable ECC errors occur. This occurs when the number of symbol errors exceeds the correction capability of the controller. UDA50, KDA/KDB50, HSC50, and HSC70 controllers, for example, can correct data with 1 to 8 ECC symbol errors maximum.

2.11.3 Who Detects and Performs BBR?

The controller is responsible for detecting ECC and header errors during read or write operations and, subsequently, setting the BBR flag. Here the term BBR flag means bad block request or bad block replacement request.

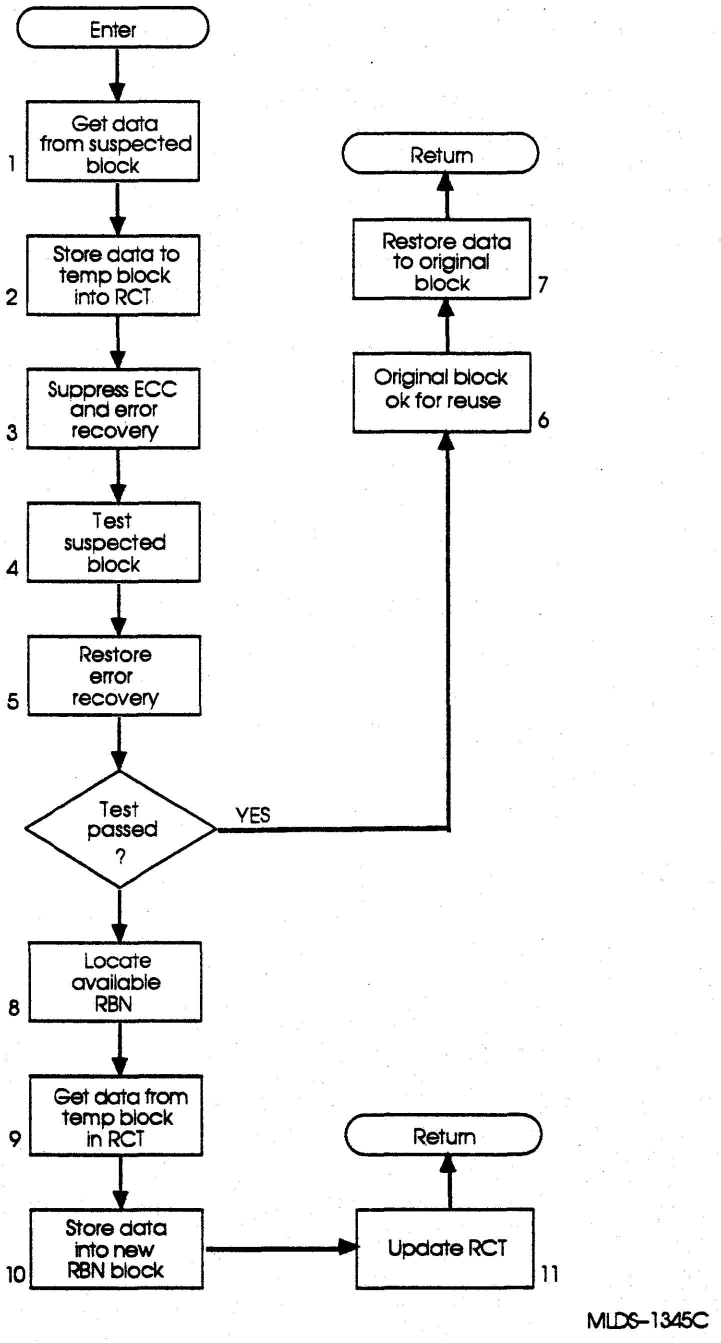
UDA50 and KDA/KDB50 controllers, for example, do not perform the actual replacement process but, instead, pass the BBR flag to the host. The host system operating software is then responsible for performing the actual block replacement tasks.

HSC50 controllers running microcode Version 200 (or higher) and HSC70 controllers set the BBR flag and also perform the actual block replacement tasks. The host is not burdened with the additional tasks required to accomplish bad block replacement.

2.11.4 How is BBR Performed?

Once a decision has been made to invoke Bad Block Replacement (BBR flag set), bad blocks in the host operating area are replaced using the procedure described here. Refer to Figure 2–29 for a simplified flow diagram and the following numbered steps for a simplified description.

Figure 2–29: Basic BBR Flow



Basic BBR Flow notes

1. The data is retrieved (reread) from the block suspected to be bad.

2. This data is temporarily stored (written) in a block in the RCT area.

3. The use of hardware–assisted error recovery and ECC correction is suppressed so that the suspect block is tested in its default state.

   NOTE

   Any ECC error reported in this state is considered uncorrectable.

4. The block suspected to be bad is tested by performing read and write data operations with user data and inverted user data to verify that the block is indeed bad.

5. Hardware error recovery and ECC correction capability is restored for use by subsequent read and write operations.

6. If the test(s) pass, then the block is considered reusable.

7. Since the block is considered reusable, the original data is retrieved from the RCT (see step #2) and rewritten to the original logical block (sector).

8. If the block failed the test from step #4, the block must be replaced and the original data moved to the replacement block. In this step, an available replacement block (RBN) in the replacement area is located using information found in the RCT.

9. The original data from the bad block is retrieved from the temporary storage block in the RCT (see step #2).

10. The original data is now written into the new replacement block (RBN).

11. Information in the RCT is updated to reflect the replacement process that has just occurred. Future access to the old bad block may require this information to find the new replacement block during revectoring. The new RBN is no longer available for replacement of other blocks.

Bad block replacement cannot be performed if the disk drive is write protected. If a bad block is detected on a disk drive that is write protected, the BBR functions fail with a write–protect error. Also, if the drive becomes write protected after BBR has started, incomplete replacement and possible loss of data could result.

2.11.5 Types of Replacement and Revectoring

There are two types of replacement currently implemented by the SDI:

1. **Primary Replacement**

   When the selected RBN resides on the same track as the block being replaced, the replacement is called **primary**. During replacement, the first priority for locating an available RBN is to attempt to locate a primary RBN. This way, subsequent revectoring requires the least amount of time. Refer to Figure 2–29, step #8.

2. **Non–Primary Replacement**

   When the selected RBN resides on a track other than that containing the block being replaced, the replacement is called **non–primary**. If the primary RBN is not available during bad block replacement, the closest available RBN to the track containing the bad block is selected. Refer to Figure 2–29, step #8. The intent is to minimize the time required to revector to the replacement block during subsequent read or write data operations.

   Some of the earlier documentation and utilities used the terms **secondary** and **tertiary**. Recent changes to the specifications made these terms obsolete. The current and proper term is **non–primary**.

2.12 HARDWARE ERROR RECOVERY

2.12.1 RA82 Error Recovery

The RA82 disk drive incorporates a feature known as hardware error recovery. This is implemented as part of the RA82 hardware circuitry. When activated, special circuits alter the characteristics of the read data circuits in the disk drive. Hardware error recovery is typically used to assist the controller during read operations when uncorrectable or unrecoverable errors are detected. This feature enhances the ability of a disk/controller subsystem to recover data that would otherwise be lost when specific media failures are detected.

2.12.2 What are the RA82 Error Recovery Circuits?

The RA82 hardware error recovery circuitry is currently divided into three functional areas. These are described as follows:

1. **Decrease read threshold**

   When activated, this circuitry decreases the threshold at which the read circuitry detects read pulses from the disk media. This makes the read circuits more sensitive to potentially weak signals from the HDA.

2. **Hold–over one–shot**

   When activated, this circuitry holds the VCO control voltage stable and prevents large phase errors from occurring due to a momentary loss of read pulses from the disk.

3. **Skew read gate**

   When enabled, this circuitry introduces a delay of one or two bytes of time between the moment the hybrid module receives the READ GATE signal from the SDI controller and the time the read/write module receives the READ GATE signal from the hybrid module. The amount of delay (skew) changes on each revolution of the disk when the index pulse is received. The skew time is one byte during the first revolution, two bytes during the second revolution, one byte during the third revolution, etc.

2.12.3 When are the Error Recovery Circuits Activated?

The RA82 error recovery circuits are activated only when the SDI controller issues an SDI ERROR RECOVERY command to the drive. When the controller issues the ERROR RECOVERY command to a disk, it also specifies an error recovery level number. This level number tells the disk which combination of error recovery circuits to activate. The controller is not aware of exactly what actions the disk will perform when the ERROR RECOVERY command is issued. It only knows that the disk will alter its R/W hardware characteristics.

The RA82 has seven different levels of error recovery. The circuits that are activated for each level are as follows:

LEVEL 7	DECREASE READ THRESHOLD (usually the first level tried by the controller)
LEVEL 6	HOLD–OVER ONE–SHOT
LEVEL 5	SKEW READ GATE
LEVEL 4	DECREASE READ THRESHOLD and HOLD–OVER ONE–SHOT
LEVEL 3	DECREASE READ THRESHOLD and SKEW READ GATE
LEVEL 2	HOLD–OVER ONE–SHOT and SKEW READ GATE
LEVEL 1	DECREASE READ THRESHOLD and HOLD–OVER ONE–SHOT and SKEW READ GATE (usually the last level tried by the controller)
LEVEL 0	NOP (This is the normal default state of the drive where none of the error recovery circuits are activated)

Different SDI disk types may have different levels depending upon the error recovery circuits available within the particular disk drive. The disk drive provides the number of error recovery levels it has to the SDI controller during the response to a GET COMMON CHARACTERISTICS command. The RA82 provides the value seven since it supports up to seven levels of error recovery. The RA60, RA80, and RA81, however, do not have error recovery circuits and, therefore, only support error recovery level zero.

2.12.4 RA70 Hardware Error Recovery

The RA70 also incorporates hardware error recovery circuits. Ten error recovery levels can be performed via the controller error recovery commands. Each error recovery level command to the RA70 changes only one recovery parameter of the drive. All other recovery parameters are returned to their normal condition. The default (normal) parameters of the circuits are as follows:

- Normal data read gate is delayed by 3 bytes
- PLO fast lock time is 6.36 microseconds
- Lockup is on the header preamble only
- Read threshold is 50%

The RA70 error recovery levels are divided into two major categories:

1. Drive logic recovery operations which change the electrical characteristics of the read/write path circuits.

Recovery Level	RA70 Operation Performed
10	Reduce read threshold to 25%
9	Shift/delay data read gate by 4 bytes
8	Shift/delay data read gate by 2 bytes
7	Shift/delay data read gate by 1 byte
6	Shift PLO fast lock time to 4.31 usec
5	Shift PLO fast lock time to 2.23 usec
4	Shift PLO fast lock time to 8.45 usec
3	Lockup on both header and data preambles

2. Drive servo error recovery operations which change the servo characteristics of the embedded servo centerline.

Recovery Level	RA70 Operation Performed
2	Shift the embedded centerline by −12%
1	Shift the embedded centerline by +12%
0	Return all error recovery to normal

2.12.5 How Is Error Recovery Used?

The following paragraphs explain how the error recovery feature is used in a disk subsystem during a read data operation. Refer to Figure 2–30.

Read/Write Error Recovery Flow Notes

First, the controller reads a block of data from the disk drive. If no ECC errors are detected, the data is sent back to the host operating system. If, however, ECC are errors detected, the controller determines if the number of ECC symbols in error equals or exceeds the recommended threshold supplied by the drive.

In the case of the RA82, for example, the threshold is 6 symbols. This means that if 5 or less ECC symbols were in error for this block, the controller would merely correct the data and send it to the host. If 6 or more ECC symbols were detected in error, the controller would send an error to the host error log and set the BBR (bad block replacement/request) flag. The BBR process is actually invoked at a later time.

Next, the controller determines if the data is correctable. This depends upon the correction capability and the maximum number of symbols that can be corrected by the particular controller. If the data is uncorrectable, the controller usually retries the read data operation. In most cases, the number of retries depends upon the retry count recommended by the drive characteristics. With the RA82, for example, the recommended retry count is 5.

If the data is uncorrectable after all retry operations have been exhausted, the next step is to determine if the particular drive has any hardware error recovery capabilities. For the RA82, 7 levels of error recovery are available. In this case, the controller issues an ERROR RECOVERY command and usually starts with level 7. This causes the RA82 to activate the R/W error recovery circuits corresponding to level 7. The controller now repeats the entire read data block process previously described, including additional retry operations as necessary.

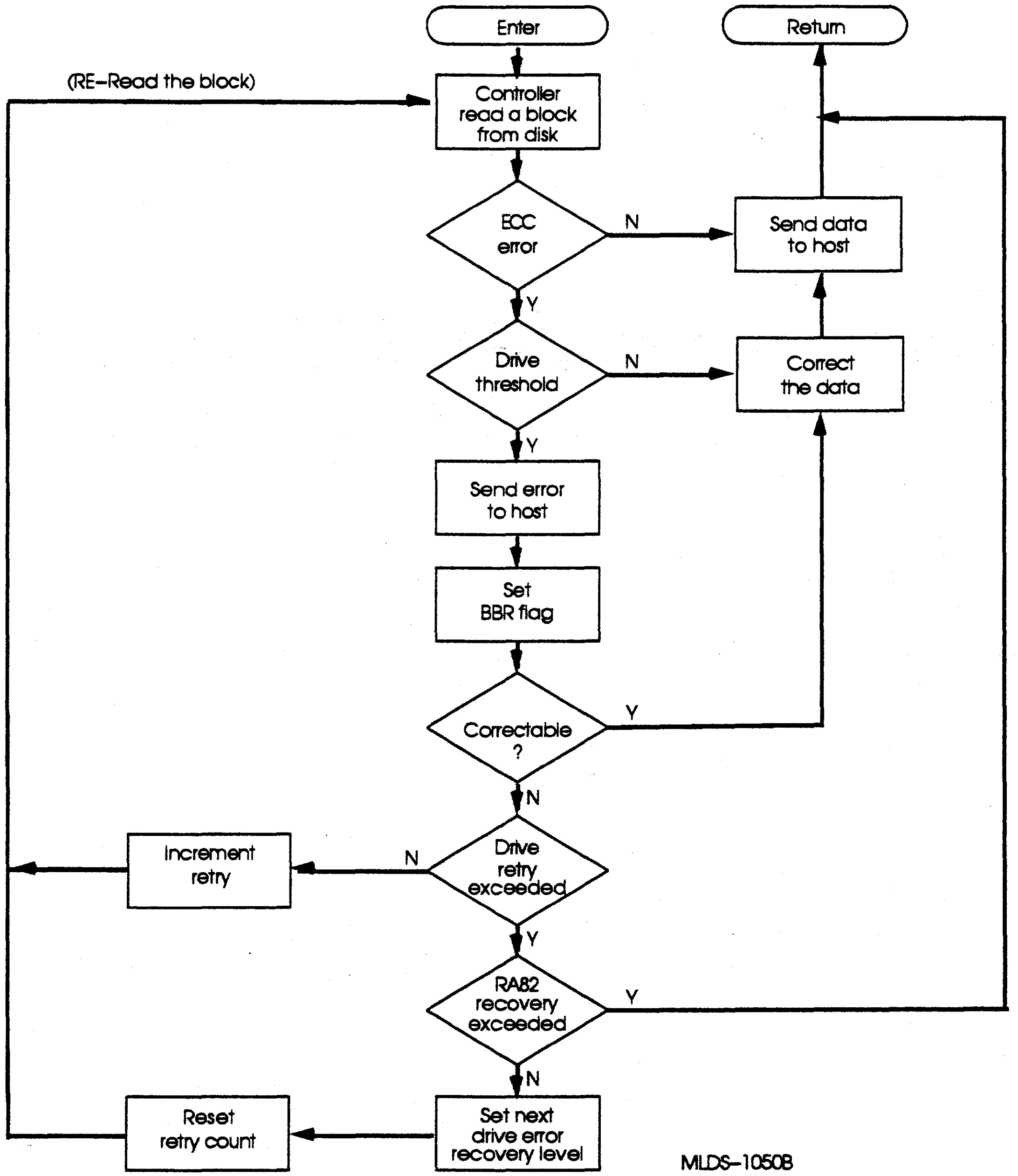
If the data block is still uncorrectable after all retries are exhausted during level 7 of error recovery, the controller issues another ERROR RECOVERY command and specifies the next lower level number, or level 6 in this example. Again, the entire operation is repeated.

This process continues with level 5, level 4, etc., until the data block is eventually read without an uncorrectable ECC error. If all levels of error recovery are exhausted and the data is still uncorrectable, the controller returns an error to the host.

For disk drives that do not support hardware error recovery, the operation is only performed to the point where all retry operations have been exhausted. At that point, the controller will also return an uncorrectable ECC error to the host.

This discussion on error recovery is a very simplified example of one way that this drive feature is used. Hardware error recovery is neither restricted nor limited to read errors due to ECC error detection. In fact, the controllers may also utilize drive hardware error recovery during read operations where header–related errors and other similar problems are detected during read operations.

Figure 2–30: Read/Write Error Recovery



2.13 FORCED ERROR

The forced error flag indicates to the host that incorrect data is correctly written into a sector.

When an uncorrectable ECC error is encountered in a block, several attempts are made to read and/or correct the data. If these attempts fail, the block causing the uncorrectable ECC error is assumed to be bad and becomes a candidate for replacement. During the replacement process, the bad block is read again (including retries and error recovery) in an attempt to extract the data for relocation to the replacement block.

If the data is still uncorrectable, the BBR process writes best–guess data into the replacement sector. The result is invalid data being correctly written to a good block. To inform the user that the data was at one time uncorrectable, the forced error flag is attached to the block. The actual mechanism used to indicate forced error is accomplished by inverting the EDC character during a write to a sector on the disk. Refer to Figure 2–6.

It is the responsibility of the host software or the user to take the necessary steps to correct or replace the data and clear the forced error indicator. The actual methods used depend upon the particular operating system, but the following points should be remembered:

1. Rewriting the block with any data will clear the forced error indicator.
2. Performing a simple read of the block with the forced error and then merely rewriting the data back to the block will result in clearing the forced error flag, but the data in the block will still be invalid. It is the responsibility of the user to insure that the data rewritten to the block is the desired data.

NOTE

The only reliable technique that should be used to recover from a forced error is to replace the file containing the forced error with a KNOWN GOOD COPY OF THAT FILE.

2.14 LOGICAL BLOCK NUMBER CONVERSION

Example 2–1: RA70/80/81/82/90 LBN to Physical and Logical Parameters

```
PC               LBN
Physical    =   ------                =   PC . PC_Rem
Cylinder         BPPC


PH               .PC_Rem * BPPC
Physical    =   ----------------      =   PH . PH_Rem
Head                 BPPT


GP               .PC_Rem * BPPC
Group       =   ----------------      =   GP . GP_Rem
(Logical)            BPG


TK               .GP_Rem * BPG
Track       =   ---------------       =   TK . TK_Rem
(Logical)            BPPT


S
Sector      =   .TK_Rem * BPPT        =   S     (Rounded to nearest
(Logical)                                        whole number)


SFI
Physical         (GP * GP_Offset) + S
Sector          ---------------------     =   X . SFI_Rem
from                 PSPT
Index                                         (discard X)

            =   SFI_Rem * PSPT            =   SFI

                                              (Rounded to nearest
                                               whole number)
```

Table 2–2: Values for RA70/80/81/82/90

	Disk	16–bit	18–bit
BPPC	RA70	363	–
	RA80	434	406
Blocks (LBNs) per	RA81	714	644
Physical cylinder	RA82	855	—
	RA90	897	—
BPPT	RA70	33	–
	RA80	31	28
Blocks (LBNs) per	RA81	51	46
Physical track	RA82	57	–
	RA90	69	–
BPG	RA70	33	—
	RA80	434	392
Blocks (LBNs) per	RA81	51	46
Group	RA82	57	—
	RA90	69	—
GP_offset	RA70	8	–
	RA80	16	16
Group offset	RA81	14	12
	RA82	11	–
	RA90	14	–
PSPT	RA70	34	–
	RA80	32	29
Physical sectors	RA81	52	47
Per track	RA82	58	–
	RA90	70	–

Example 2–2: RA81 16-Bit HDA LBN = 2498

```
PC                LBN        2498
Physical    =   ------  =  ------   =  3.498                        PC = 3
Cylinder          BPPC       714

PH                .PC_Rem * BPPC        0.498 * 714
Physical    =   ----------------   =  -------------   =  6.972      PH = 6
Head                 BPPT                  51

GP                .PC_Rem * BPPC       0.498 * 714
Group       =   ----------------   = ------------   =  6.972        GP = 6
(Logical)            BPG                   51

TK                .GP_Rem * BPG        0.972 * 51
Track       =   ---------------   =  ------------   =  0.972        TK = 0
(Logical)            BPPT                  51

S
Sector      =   .TK_Rem * BPPT = 0.972 * 51 = 49.572                S   = 50
Logical
                                                          (Rounded to nearest
                                                           whole number)

SFI
Physical          (GP * GP_Offset) + S        (6 * 14) + 50
Sector          ---------------------    =  --------------- = 2.576
from                 PSPT                          52
Index

     SFI    =   SFI_Rem * PSPT       = 0.576 * 52                   SFI = 30
                                     = 29.952
                                     = 30
                                                          (Rounded to nearest
                                                           whole number)

SUMMARY:    Physical Cylinder (PC) =  3
            Physical Head     (PH) =  6
            Group             (GP) =  6
            Track             (TK) =  0
            Logical Sector    (S)  = 50
            Phy Sector from Index  = 30
```

Example 2–3: RA81 16–Bit HDA LBN = 2499

```
PC               LBN        2499
Physical   =    ------  =  ------  =  3.5                            PC = 3
Cylinder         BPPC       714

PH               .PC_Rem * BPPC       0.5 * 714
Physical   =    ----------------  =  -------------  =  7.0          PH = 7
Head                 BPPT               51

GP               .PC_Rem * BPPC      0.5 * 714
Group      =    ----------------  = ------------  =  7.0            GP = 7
(Logical)            BPG                51

TK               .GP_Rem * BPG       0.0 * 51
Track      =    ---------------  =  ------------  =  0.0            TK = 0
(Logical)            BPPT               51

S
Sector     =    .TK_Rem * BPPT = 0.0 * 51 = 0.0                     S  = 0
Logical
                                                      (Rounded to nearest
                                                       whole number)

SFI
Physical         (GP * GP_Offset) + S       (7 * 14) + 0
Sector          ---------------------   =  --------------- = 1.884
from                 PSPT                        52
Index

     SFI   =    SFI_Rem * PSPT      = 0.884 * 52                    SFI = 46
                                    = 45.96
                                    = 46
                                                      (Rounded to nearest
                                                       whole number)

SUMMARY:   Physical Cylinder (PC) =  3
           Physical Head     (PH) =  7
           Group             (GP) =  7
           Track             (TK) =  0
           Logical Sector    (S)  =  0
           Phy Sector from Index  = 46
```

Example 2–4: RA60 LBN to Physical and Logical Parameters

```
LC               LBN
Logical     =   -----              =   LC . LC_Rem
Cylinder         BPLC

GP              .LC_Rem * BPLC
Group       =   --------------     =   GP . GP_Rem
                     BPG

TK               .GP_Rem * BPG
Track       =   ---------------    =   TK . TK_Rem
(Logical)            BPPT

S
Sector      =   .GP_Rem * BPT      (Result rounded to nearest
(Logical)                           whole number)

                   LBN
CYL60       =   --------           =   CYL60 . Remainder
                4 * BPPC                       (Discard)

Physical
Cylinder    =   (4 * CYL60) + GROUP

Physical        LBN - (CYL60 * 4 * BPPC)
Head        =   ------------------------  =    HEAD . Remainder
                         BPLC

SFI
Physical         (GP * GP_Offset) + S
Sector          ---------------------   =   X . SFI_Rem
from                 PSPT
Index                                       (discard X)

            =   SFI_Rem * PSPT          =   SFI

                                            (Rounded to nearest
                                             whole number)
```

	16-bit	18-bit	
BPLC	168	152	Blocks (LBN's) Per Logical Cylinder
BPPT	42	38	Blocks (LBN's) Per Physical Track
BPG	42	38	Blocks (LBN's) Per Group
BPPC	252	228	Blocks (LBN's) Per Physical Cylinder
PSPT	43	39	Physical Sectors Per Track
GP_Offset	16	15	Group Offset

Example 2–5: RA60 16-Bit HDA LBN = 6000

```
LC              LBN       6000
Logical    =   -----  =  ------  =  35.714                    LC = 35
Cylinder        BPLC      168


GP             .LC_Rem * BPLC      0.714 * 168
Group      =   --------------  =  -------------  =  2.856       GP = 2
                    BPG                42


TK              .GP_Rem * BPG      0.856 * 42
Track      =   ---------------  =  ----------  =  0.856         TK = 0
(Logical)           BPPT               42


S
Sector     =   .GP_Rem * BPT  =  0.856 * 42  =  35.952           S = 36
(Logical)
                                                   (Rounded to nearest
                                                    whole number)


                  LBN        6000
CYL60      =   --------  =  -------  =  5.952              CYL60 = 5
               4 * BPPC     4 * 252
                                                   (discard fraction)


Physical
Cylinder   =   (4 * CYL60) + GROUP  =  (4 * 5) + 2         Phy Cyl = 22
                                    =  22


Physical       LBN - (CYL60 * 4 * BPPC)      6000 - (5 * 4 * 252)
Head       =   ------------------------  =  ----------------------
                        BPLC                          168

                                         =  5.714

                          Physical Head  =  5
SFI
Physical        (GP * GP_Offset) + S      (2 * 16) + 36
Sector         ---------------------  =  ---------------
from               PSPT                         43
Index
                                      =  1.581


    SFI    =   SFI_Rem * PSPT     =  0.581 * 43                SFI = 25
                                  =  24.98
                                  =  25
                                        (Rounded to nearest
                                         whole number)
```

Example 2–5 Cont'd. on next page

Example 2–5 (Cont.): RA60 16-Bit HDA LBN = 6000

```
SUMMARY:    Physical Cylinder (PC) = 22
            Logical Cylinder       = 35
            Physical Head     (PH) =  5
            Group             (GP) =  2
            Track             (TK) =  0
            Logical Sector    (S)  = 36
            Phy Sector from Index  = 25
```

Example 2–6: Quick RA60 Head Algorithm

If you know the LBN (Logical Block Number), first determine the logical cylinder:

```
LBN
----  =  logical cylinder . fraction  (discard fraction)
BPLC


Logical Cylinder
----------------  =  XXX . YYY
    6 (heads)


PHYSICAL HEAD     =  6 * (.YYY)


BPLC = 168  (16-bit packs)       Blocks Per Logical Cylinder
       152  (18-bit packs)
```

Using "quick" RA60 head algorithm for the previous RA60 sample.

```
LBN
----  =  Logical Cylinder . Fraction  (discard fraction)
BPLC

         6000
      =  ----  =  35.714       LOGICAL CYLINDER = 35
         168


Logical Cylinder      35
----------------  =  ----  =  5.8333    (keep fraction)
    6 (heads)         6

PHYSICAL HEAD     =  6 *  0.8333  =  4.99 =  HEAD 5

BPLC = 168  (16-bit packs)       Blocks Per Logical Cylinder
       152  (18-bit packs)
```

2.15 EXERCISES

At this time complete the following exercises. You may use any reference material to answer the questions.

1. The term **sector** is used interchangeably with what other term?
   - **A.** GROUP
   - [✓ **B.**] BLOCK
   - **C.** TRACK
   - **D.** LOGICAL CYLINDER

2. Which of the following perform bad block replacement?
   - **A.** The host
   - **B.** The drive
   - **C.** The controller
   - [✓ **D.**] Either the host or the controller
   - **E.** Either the drive or the controller

3. What would cause bad block replacement (BBR) to be invoked?
   - ~~**A.**~~ ECC error
   - **B.** EDC error
   - **C.** Mis–seek error
   - **D.** Header error
   - [✓ **E.**] A and D
   - **F.** A and B

4. Where are the RBNs located on a disk?
   - **A.** On the last sector of each track in the FCT area.
   - **B.** In the outer guard band areas.
   - **C.** On the last sector of each track in the host area.
   - [✓ **D.**] On the last sector of each track in the host and RCT areas.

5. Replacement blocks (RBNs) are used to replace logical blocks from which areas?

[✓ **A.**] Defective blocks in the host applications area.

B. Defective blocks in the RCT area.

C. Defective blocks in the FCT area.

D. Defective blocks in the diagnostic block area.

E. A and B.

6. How many logical groups are there in a logical cylinder on an RA82 disk?

A. 8

[✓ **B.**] 15

C. 2

D. 1

7. What is the forced error flag used for?

A. Indicate that a block is bad.

[✓ **B.**] Indicate the data in a block was at one time uncorrectable.

C. Indicate the block is being tested with forced errors.

D. Indicate that the header in the block is bad.

E. A and B.

8. How is drive hardware error recovery activated?

A. By the drive when an ECC error is detected below drive threshold.

[✓ **B.**] By the controller when an uncorrectable error is detected after all retry operations have been attempted.

C. By the drive when an uncorrectable error is detected after all retry operations have been attempted.

D. By the controller when an ECC error is detected below drive threshold.

CHAPTER 3

DRIVE CHARACTERISTICS

Drive Compare Specifications

	RA90 16–bit	RA70 16–bit	RA82 16–bit	RA81 16–bit
Physical Specifications				
Heads per surface	1	1	2	2
Recording surfaces	13	11	7.5	7.0
Servo surfaces	1	1	0.5	0.5
Data heads	13	11	15	14
Blocks per track	70	34	58	52
Physical cylinders	2661	1517	1435	1258
Tracks per disk	37254	16687	21525	17612
Embedded servo	yes	yes	yes	yes
Data Specifications				
User blocks (host applications area)	2376153	547041	1216665	891072
Megabytes per disk (Host applications area)	1216	280	623	456
Logical cylinders	2661	1511	1435	1258
Tracks per logical group	1	1	1	1
Groups per logical cyl	13	11	15	14
Reserved Space				
Replacement blocks per disk	34463	16611	21405	17528
Tracks for replacement control table	39	44	60	56
Tracks for diagnostic use	26	22	60	28
Recording density				
Tracks per inch (TPI)	1750	1355	1063	960
Bits per inch (BPI)	22839	22437	12800	11400
Transfer rate				
Burst (MHz)	22.20	11.6	19.2	17.4

	RA90 16–bit	RA70 16–bit	RA82 16–bit	RA81 16–bit
Positioner access time				
Single track seek (ms.)	5.5	5.5	6.0	7.0
Average seek (ms.)	19.0	19.5	24.0	28.0
Total full stroke (ms.)	30.0	35.0	38.0	50.0
Head switch (ms.)	3.0	4.5	6.0	4

3.1 RA60 Common Characteristics

```
BYTE  Hex
#     byte
---------------------------------------------------------------------------

1     78    Response Opcode

2     33    (3)  SDI Version 3.0
            (3)  Short timeout = 8 seconds (2^3)

3     9E    Xfer Rate = 15.8 MHz  (9E hex = 158 Decimal)

4     F7    (5)   15 retries for data transfer operation
            (7)  128 second long timeout (2^7)

5     06    6 Copies FCT/RCT, 512 byte mode only (one copy/head
      86    6 Copies FCT/RCT, 512 or 576 byte mode (one copy/head)

6     00    Error Recovery Level = 0    (0 levels)

7     04    ECC Threshold = 4 (# of ECC symbol errors to consider
            abnormal)

8     xx    Microcode Revision Level

9     0x    (0) IE=0, No Special Internal Error Log Available
            (x) Hardware Rev from operator panel

10    xx    Lo -
11    xx        |
12    xx        |--- S/N of Drive
13    00        |
14    00        |
15    00    Hi -

16    04    Drive type Identifier per MSCP Spec  (RA60)

17    3C    60 Revs/Second  (3C hex = 60 decimal)

18    00    -
19    00     |
20    00     |--- Error Recovery Threshold (not used in RA60)
21    00     |
22    00     |
23    00    -
```

3.2 RA60 Subunit Characteristics

```
BYTE  Hex
#     byte
-----------------------------------------------------------------------
1     77    Response Opcode

2     54    -
3     09     |-- 00000954 hex = 2388 Decimal (LOGICAL) Cylinders in LBN
4     00     |   Space (Host Cyls + RCT Cyls, 0 thru 2387)
5     00    -


6     04    4 Groups/Cylinder

7     00    1st XBN = 0     1st LBN = 0

8     01    1 Track/Group

9     00    1st DBN = 0     1st RBN = 0

10    81    1 RBN/Track     (RM bit = 1, REMOVABLE Media)

11    00    Reserved

12    0D    13 Words DATA PREAMBLE   (for 512-byte mode, 43 sectors)

13    05    5 Words HEADER PREAMBLE (for 512-byte mode, 43 sectors)


14    3C    -
15    10     |--- Media Type Identifier  RA60-DJ
16    A4     |
17    22    -


18    AC    Lo   FCT Copy Size - XBNs  = 172
19    00    Hi
                ( 00AC hex = 172 Decimal= 4 x 43 )
```

```
BYTE  Hex
#     byte
______________________________________________________________________

********************* 512-BYTE MODE **************************

20    2A    42 LBN's/Track

21    10    Group Offset = 16 Decimal

22    30    -  # of HOST LBNs = 400,176 (= 00061B30 hex)
23    1B     |
24    06     |
25    00    -

26    A8    -  RCT Copy Size (LBNs) = 168 decimal  (4x42) = (00A8 hex)
27    00    -

********************* 576-BYTE MODE **************************

28    26    38  LBNs/Track

29    10    Group Offset = 16 decimal

30    50    -  # of HOST LBNs = 362,064  (= 00058650 hex)
31    86     |
32    05     |
33    00    -

34    98    -  RCT Copy Size - LBNs = 152  (4x38) = (0098 hex)
35    00    -

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

36    06    -  0006 =  2 LOGICAL Cylinders in XBN Space
37    00    -

38    02    Size of Diagnostic READ-ONLY DBN Area (Groups) = 2

39    06    6 LOGICAL Cylinders in DBN Area
```

3.3 RA70 Common Characteristics

BYTE #	Hex byte	
1	78	Response Opcode
2	43	(4) SDI Version 4.0 (3) Short timeout = 8 seconds (2^3)
3	74	Xfer Rate = 11.6 Mhz (4 hex = 116 Decimal)
4	57	(5) 5 retries for data transfer operation (7) 128 second long timeout (2^7)
5	07	7 Copies FCT/RCT, 512 byte mode only
6	0A	Error Recovery Level = 10 decimal (10 levels)
7	06	ECC Threshold = 6 (# of ECC symbol errors to consider abnormal)
8	xx	Microcode Revision Level
9	8x	(1) (Bit 7, IE=1), Special Internal Error Log Available (x) Hardware Rev from microcode
10	xx	Lo -
11	xx	|
12	xx	|--- S/N of Drive
13	00	|
14	00	|
15	00	Hi -
16	12	Drive type Identifier per MSCP Spec (18 Decimal, 22 Octal, RA70)
17	43	67 Revs/Second (43 hex = 67 decimal)
18	00	-
19	00	|
20	00	|--- Error Recovery Threshold (not used in RA70)
21	00	|
22	00	|
23	00	-

3.4 RA70 Subunit Characteristics

```
BYTE  Hex
#     byte
_________________________________________________________________________

1     77    Response Opcode

2     E7    -
3     05     |-- 00005E7 hex = 1511 Decimal (LOGICAL) Cylinders in LBN
4     00     |   Space (Host Cyls + RCT Cyls, 0 thru 1510)
5     00    -


6     0B    11 Groups/Cylinder

7     00    1st XBN = 0      1st LBN = 0

8     01    1  Track/Group

9     00    1st DBN = 0      1st RBN = 0

10    01    1 RBN/Track      (RM bit = 0, Non-removable Media)

11    00    Reserved

12    0E    14 words DATA PREAMBLE   (for 512-byte mode, 34 sectors)

13    09    9 Words HEADER PREAMBLE (for 512-byte mode, 34 sectors)


14    46    -
15    10     |--- Media Type Identifier
16    64     |
17    25    -


18    CC    Lo   FCT Copy Size - XBNs  = 204
19    00    Hi
                ( 00CC hex = 204 Decimal)
```

```
BYTE  Hex
#     byte
______________________________________________________________________

********************* 512-BYTE MODE ***************************

20    21    33 LBNs/Track

21    08    Group Offset = 8 Decimal

22    E1    -  # of HOST LBNs = 547,041 (= 000858E1 hex)
23    58     |
24    08     | (33 LBNs/Track x 11 Heads x 1507 LOGICAL Cylinders
25    00    -

26    C6    -  RCT Copy Size (LBNs) = 198 decimal
27    00    -  (198 decimal = 00C6 hex)

********************* 576-BYTE MODE ***************************

            18-Bit HDA's NOT SUPPORTED by RA70
28    00    LBNs/Track

29    00    Group Offset

30    00    -  # of HOST LBNs
31    00     |
32    00     |
33    00    -

34    00    -  RCT Copy Size
35    00    -

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

36    04    -  0004 =  4 Cylinders in XBN Space
37    00    -

38    0B    Size of Diagnostic READ-ONLY DBN Area (Groups) = 11 decimal

39    02    2 Cylinders in DBN Area
```

3.5 RA80 Common Characteristics

```
BYTE  Hex
#     byte
-----------------------------------------------------------------------

1     78    Response Opcode

2     33     (3)  SDI Version 3.0
             (3)  Short timeout = 8 seconds (2^3)

3     61    Xfer Rate = 9.7 Mhz  (61 hex = 97 Decimal)

4     57     (5)    5 retries for data transfer operation
             (7)  128 second long timeout (2^7)

5     04    4 Copies FCT/RCT, 512 byte mode only, (HDA jumper in)
      84    4 Copies FCT/RCT, 512 or 576 byte mode (HDA jumper out)

6     00    Error Recovery Level = 0   (0 levels)

7     02    ECC Threshold = 2 (# of ECC symbol errors to consider
            abnormal)

8     xx    Microcode Revision Level

9     0x    (0) IE=0, No Special Internal Error Log Available
            (x) Hardware Rev from operator panel

10    xx    Lo -
11    xx        |
12    xx        |--- S/N of Drive
13    00        |
14    00        |
15    00    Hi -

16    01    Drive type Identifier per MSCP Spec  (RA80)

17    3C    60 Revs/Second  (3C hex = 60 decimal)

18    00    -
19    00     |
20    00     |--- Error Recovery Threshold (not used in RA80)
21    00     |
22    00     |
23    00    -
```

3.6 RA80 Subunit Characteristics

```
BYTE  Hex
#     byte
______________________________________________________________________

1     77     Response Opcode

2     13     -
3     01      |-- 00000113 hex = 275 Decimal (LOGICAL) Cylinders in LBN
4     00      |    Space (Host Cyls + RCT Cyls, 0 thru 274 logical,
5     00     -     0 thru 558 physical)


6     02     2 Groups/Cylinder

7     00     1st XBN = 0     1st LBN = 0

8     0E     14 Tracks/Group

9     00     1st DBN = 0     1st RBN = 0

10    01     1 RBN/Track     (RM bit = 0, Non-removable Media)

11    00     Reserved

12    0B     11 Words DATA PREAMBLE   (for 512-byte mode, 32 sectors)

13    04     4 Words HEADER PREAMBLE (for 512-byte mode, 32 sectors)


14    50     -
15    10      |--- Media Type Identifier  DURA 80
16    64      |
17    25     -


18    E0     Lo    FCT Copy Size - XBNs  = 480
19    01     Hi
                ( 01E0 hex = 480 Decimal= 15 x 32 )
```

```
BYTE  Hex
#     byte

*********************** 512-BYTE MODE **************************

20    1F    31 LBNs/Track

21    10    Group Offset = 16 Decimal

22    9C    -  # of HOST LBNs = 237,212 (= 00039E9C hex)
23    9E     |
24    03     | (31 LBNs/Track x 14 Heads x 2 grps x 273 LOGICAL
25    00    -    Cylinders + 248 LBNs borrowed from RCT)

26    D1    -  RCT Copy Size (LBNs) = 465 decimal  (15x31) = (01D1
            hex)
27    01    -

********************* 576-BYTE MODE **************************

28    1C    28  LBNs/Track

29    10    Group Offset = 16 decimal

30    F0    -  # of HOST LBNs = 214,256  (= 000344F0 hex)
31    44     |
32    03     | (28 LBNs/Track x 14 Heads x 2 GRPS X 273 LOGICAL
33    00    -    Cylinders + 224 LBNs BORROWED FROM RCT)

34    A4    -  RCT Copy Size - LBNs = 420  (13x28)  = (1A4 hex)
35    01    -

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

36    02    -  0002 =  2 Cylinders in XBN Space
37    00    -

38    01    Size of Diagnostic READ-ONLY DBN Area (Groups) = 1

39    02    2 Cylinders in DBN Area
```

3.7 RA81 Common Characteristics

BYTE #	Hex byte	
1	78	Response Opcode
2	33	(3) SDI Version 3.0 (3) Short timeout = 8 seconds (2^3)
3	AE	Xfer Rate = 17.4 Mhz (AE hex = 174 Decimal)
4	57	(5) 5 retries for data transfer operation (7) 128 second long timeout (2^7)
5	04 84	4 Copies FCT/RCT, 512 byte mode only, (HDA jumper in) 4 Copies FCT/RCT, 512 or 576 byte mode (HDA jumper out)
6	00	Error Recovery Level = 0 (0 levels) No error recovery for this drive
7	06	ECC Threshold = 6 (# of ECC symbol errors to consider abnormal)
8	xx	Microcode Revision Level
9	0x	(0) IE=0, No Special Internal Error Log Available (x) Hardware Rev from operator panel
10	xx	Lo -
11	xx	|
12	xx	|--- S/N of Drive
13	00	|
14	00	|
15	00	Hi -
16	05	Drive type Identifier per MSCP Spec (RA81)
17	3C	60 Revs/Second (3C hex = 60 decimal)
18	00	-
19	00	|
20	00	|--- Error Recovery Threshold (not used in RA81)
21	00	|
22	00	|
23	00	-

3.8 RA81 Subunit Characteristics

```
BYTE   Hex
#      byte
______________________________________________________________________

1      77     Response Opcode

2      E4     -
3      04      |-- 000004E4 hex = 1252 Decimal (LOGICAL) Cylinders in LBN
4      00      |   Space (Host Cyls + RCT Cyls, 0 thru 1251)
5      00     -

6      14     14 Groups/Cylinder

7      00     1st XBN = 0     1st LBN = 0

8      01     1 Track/Group

9      00     1st DBN = 0     1st RBN = 0

10     01     1 RBN/Track     (RM bit = 0, Non-removable Media)

11     00     Reserved

12     13     19 Words DATA PREAMBLE   (for 512-byte mode, 52 sectors)

13     0C     12 Words HEADER PREAMBLE (for 512-byte mode, 52 sectors)


14     51     -
15     10      |--- Media Type Identifier  DURA 81
16     64      |
17     25     -


18     0C     Lo   FCT Copy Size - XBNs   = 780
19     03     Hi
                ( 030C hex = 780 Decimal= 15 x 52 )
```

```
BYTE  Hex
#     byte
_____________________________________________________________________

********************* 512-BYTE MODE *************************

20    33    51 LBNs/Track

21    0E    Group Offset = 14 Decimal

22    C0    -  # of HOST LBNs = 891,072  (= 000D98C0 hex)
23    98     |
24    0D     | (51 LBNs/Track x 14 Heads 1248 LOGICAL Cylinders
25    00    -

26    FD    -  RCT Copy Size (LBNs) = 765 decimal  (15x51) = (02FD
            hex)
27    02    -

********************* 576-BYTE MODE *************************

28    2E    46  LBNs/Track

29    0C    Group Offset = 12 decimal

30    80    -  # of HOST LBNs = 803,712  (= 000C4380 hex)
31    43     |
32    0C     | (46 LBNs/Track x 14 Heads X 1248 LOGICAL Cylinders
33    00    -

34    B2    -  RCT Copy Size - LBNs = 690  (15x46) = (2B2 hex)
35    02    -

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

36    04    -  0004 =  4 Cylinders in XBN Space
37    00    -

38    0E    Size of Diagnostic READ-ONLY DBN Area (Groups) = 14

39    02    2 Cylinders in DBN Area
```

3.9 RA82 Common Characteristics

BYTE #	Hex byte	
1	78	Response Opcode
2	43	(4) SDI Version 4.0 (3) Short timeout = 8 seconds (2^3)
3	C0	Xfer Rate = 19.2 Mhz (C0 hex = 192 Decimal)
4	57	(5) 5 retries for data transfer operation (7) 128 second long timeout (2^7) (max for SDI Spec.)
5	04	4 Copies FCT/RCT, 512 byte mode only, (HDA jumper in)
6	07	Error Recovery Level = 7 (7 levels)
7	06	ECC Threshold = 6 (# of ECC errors to consider abnormal)
8	xx	Microcode Revision Level
9	0x	(0) IE=0, No Special Internal Error Log Available (x) Hardware Rev from operator panel
10	xx	Lo -
11	xx	|
12	xx	|--- S/N of Drive
13	00	|
14	00	|
15	00	Hi -
16	0B	Drive type Identifier per MSCP Spec (11 decimal, 13 Octal, RA82)
17	3C	60 Revs/Second
18	00	-
19	00	|
20	00	|--- Error Recovery Threshold (not used in RA82)
21	00	|
22	00	|
23	00	-

3.10 RA82 Subunit Characteristics

```
BYTE  Hex
#     byte
____________________________________________________________________________

1     77     Response Opcode

2     93     -
3     05      |--- 00000593 hex = 1427 Decimal LOGICAL Cylinders in LBN
4     00      |     Space (Host Cyls + RCT Cyls, 0 thru 1426)
5     00     -

6     0F     15 Groups/Cylinder

7     00     1st XBN = 0      1st LBN = 0

8     01     1 Track/Group

9     00     1st DBN = 0      1st RBN = 0

10    01     1 RBN/Track      (RM bit = 0, Non-removable Media)

11    00     Reserved

12    12     18 Words DATA PREAMBLE   (for 512-byte mode, 58 sectors)

13    06     6 Words HEADER PREAMBLE (for 512-byte mode, 58 sectors)

14    52     -
15    10      |--- Media Type Identifier  DURA 82
16    64      |
17    25     -

18    A0     Lo   FCT Copy Size - XBNs  = 928
19    03     Hi
                ( 03A0 hex = 928 Decimal= 16 x 58 )
```

```
BYTE  Hex
#     byte
-------------------------------------------------------------------------

********************* 512-BYTE MODE **************************

20    39    57 LBNs/Track

21    0B    Group Offset = 11 Decimal

22    99    -  # of HOST LBNs = 1216665 (= 129099 hex)
23    90     |
24    12     | (57 LBNs/Track x 15 Heads x 1423 Cylinders)
25    00    -

26    90    -  RCT Copy Size (LBNs) = 912  (16x57) = (390 hex))
27    03    -

********************* 576-BYTE MODE **************************

            (There are no plans to implement actual 18-bit HDA's
             at the present.)
28    33    51 LBNs/Track

29    0E    Group Offset = 14

30    53    -  # of HOST LBNs = 1,088,595  (= 109C53 hex)
31    9C     |
32    10     | (51 LBNs/Track x 15 Heads x 1423 Cylinders)
33    00    -

34    30    -  RCT Copy Size - LBNs = 816  (16x51) = (330 hex)
35    03    -

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

36    04    -  0004 =  4 Cylinders in XBN Space
37    00    -

38    0F    Size of Diagnostic READ ONLY DBN Area (Groups) = 15

39    04    4 Cylinders in DBN Area
```

3.11 RA90 Common Characteristics

```
BYTE  Hex
#     byte
_____________________________________________________________________

1     78    Response Opcode

2     43    (4)  SDI Version 4.0
            (3)  Short timeout = 8 seconds (2^3)

3     DD    Xfer Rate = 22.198 Mhz  (DD hex = 221 Decimal)

4     57    (5)    5 retries for data transfer operation
            (7)  128 second long timeout (2^7)

5     04    4 Copies FCT/RCT, 512 byte mode only, (HDA jumper in)

6     0C *  Error Recovery Level = TBD (12 as implemented to date
            12-15-87, subject to change)

7     06    ECC Threshold = 6 (# of ECC symbol errors to consider
            abnormal)

8     xx    Microcode Revision Level

9     xx    Bit<0:3>  Hardware revision from switches
            Bit<4>    1 = Embedded Servo Enabled
            Bit<5:6>  HDA revision bits
            Bit<7>    IE=1, Special internal error log available

10    xx    Lo ---- S/N of Drive  (Drive Serial number is determined
11    xx        |   by switches on the flex cable, a part of the
12    xx        |   chassis assy. (item 7 of 70-22941-01) Decode:
13    00        |       Bit <19:18> = 00 = CXO mfg drive
14    00        |           Bit <17:00>  = 1 thru 262143
15    00    Hi -        Bit <19:18> = 01 = CXO mfg drive
                            Bit <17:00>  = 262144 thru 309,999
                        Bit <19:18> = 10 = KBO mfg drive
                            Bit <17:00>  = 1 thru 262143
                        Bit <19:18> = 11 = TBD plant mfg drive
                            Bit <17:00>  = 1 thru 262143

16    13    Drive type Identifier per MSCP Spec  (19 Decimal,
            23 Octal, RA90)

17    3C    60 Revs/Second  (3C hex = 60 decimal)

18    00    -
19    00     |
20    00     |--- Error Recovery Threshold (not used in RA90)
21    00     |
22    00     |
23    00    -
```

3.12 RA90 Subunit Characteristics

```
BYTE  Hex
#     byte
------------------------------------------------------------------------
1     77    Response Opcode

2     5B    -
3     0A     |-- 0000A5B hex = 2651 Decimal (LOGICAL) Cylinders in LBN
4     00     |   Space (Host Cyls + RCT Cyls, 0 thru 2650)
5     00    -

6     0D    13 Groups/Cylinder

7     00    1st XBN = 0     1st LBN = 0

8     01    1  Track/Group

9     00    1st DBN = 0     1st RBN = 0

10    01    1 RBN/Track      (RM bit = 0, Non-removable Media)

11    00    Reserved

12    0E    14 Words DATA PREAMBLE   (for 512-byte mode, 70 sectors)

13    05    5 Words HEADER PREAMBLE (for 512-byte mode, 70 sectors)

14    5A    -
15    10     |--- Media Type Identifier  DURA 90
16    64     |
17    25    -

18    76    Lo   FCT Copy Size - XBNs  = 630
19    02    Hi
                ( 0276 hex = 630 Decimal)
```

```
BYTE  Hex
#     byte
______________________________________________________________________

********************* 512-BYTE MODE **************************

20    45    69 LBNs/Track

21    0E    Group Offset = 14 Decimal

22    D9    -  # of HOST LBNs = 2,376,153 (= 002441D9 hex)
23    41     |
24    24     | (69 LBNs/Track x 13 Heads x 2649 LOGICAL Cylinders
25    00    -

26    9E    -  RCT Copy Size (LBNs) = 414 decimal  = (19E hex)
27    01    -

********************* 576-BYTE MODE **************************

            18-Bit HDA's NOT SUPPORTED by RA90
28    00    LBNs/Track

29    00    Group Offset

30    00    -  # of HOST LBNs
31    00     |
32    00     |
33    00    -

34    00    -  RCT Copy Size
35    00    -

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

36    03    -  0003 =  3 Cylinders in XBN Space
37    00    -

38    0D    Size of Diagnostic READ-ONLY DBN Area (Groups) = 13 decimal

39    02    2 Cylinders in DBN Area
```

DRIVE CHARACTERISTICS QUIZ

3.13 Student Exercises

1. **ECC error handling is performed by which of the following?**

   A. The drive
   [✓ B.] The controller
   C. The host
   D. The drive if the ECC threshold is exceeded

2. **The number of multiple copies of the RCT and the FCT for each drive is:**

   A. 4 copies of RCT/FCT
   B. 6 copies of RCT/FCT
   C. 7 copies of RCT/FCT
   [✓ D.] Depends upon drive type

3. **ECC threshold value is derived from which of the following?**

   [✓ A.] The drive
   B. The controller
   C. The host hardware
   D. The host software if BBR is supported

4. **The purpose of the ECC character is:**

   A. Only detect disk transfer errors.
   B. Only detect controller internal data path errors.
   [✓ C.] Detect disk transfer errors and provide for data correction.
   D. Detect controller internal data path errors and provide for data correction.

5. **The purpose of the EDC character is:**

   A. Only detect disk transfer errors.
   [✓ B.] Only detect controller internal data path errors.
   C. Detect disk transfer errors and provide for data correction.
   D. Detect controller internal data path errors and provide for data correction.

6. **Multiple copies of the RCT and FCT are located where?**

   A. Usually on the same cylinder.

   B. Usually on the same track.

   C. Usually distributed across the same media surface.

   [✓ **D.**] Usually distributed across different heads and cylinders.

7. **What are the addressing characteristics of the RA70?**

   [✓ **A.**] 1 track per group and 11 groups per logical cylinder.

   B. 11 tracks per group and 1 group per logical cylinder.

   C. 1 track per group and 2 groups per logical cylinder.

   D. 11 tracks per group and 11 groups per logical cylinder.

8. **A subsystem consisting of an HSC and an RA81 disk drive detects uncorrectable ECC errors. what is the error recovery technique used if the ECC error continues to be uncorrectable?**

   A. The controller will retry the operation at least 14 times.

   B. The drive will retry the operation at least 14 times.

   [✓ **C.**] The controller will retry the operation at least 5 times.

   D. The drive will retry the operation at least 5 times.

9. **Which of the following statements is false?**

   A. The ECC character protects both the data field and EDC character in a sector.

   B. The EDC character is used to detect internal controller parallel data path problems.

   C. The ECC character is used to detect serial read/write disk data path problems.

   [✓ **D.**] The ECC character only protects the data field of a sector. & EDC character too

10. **In an RA81 subsystem, a block of data is considered uncorrectable if the number of symbols in error exceed what?**

    [✓ **A.**] 8 symbols, the maximum correction capability of the controller.

    B. 8 symbols, the maximum threshold determined by the drive.

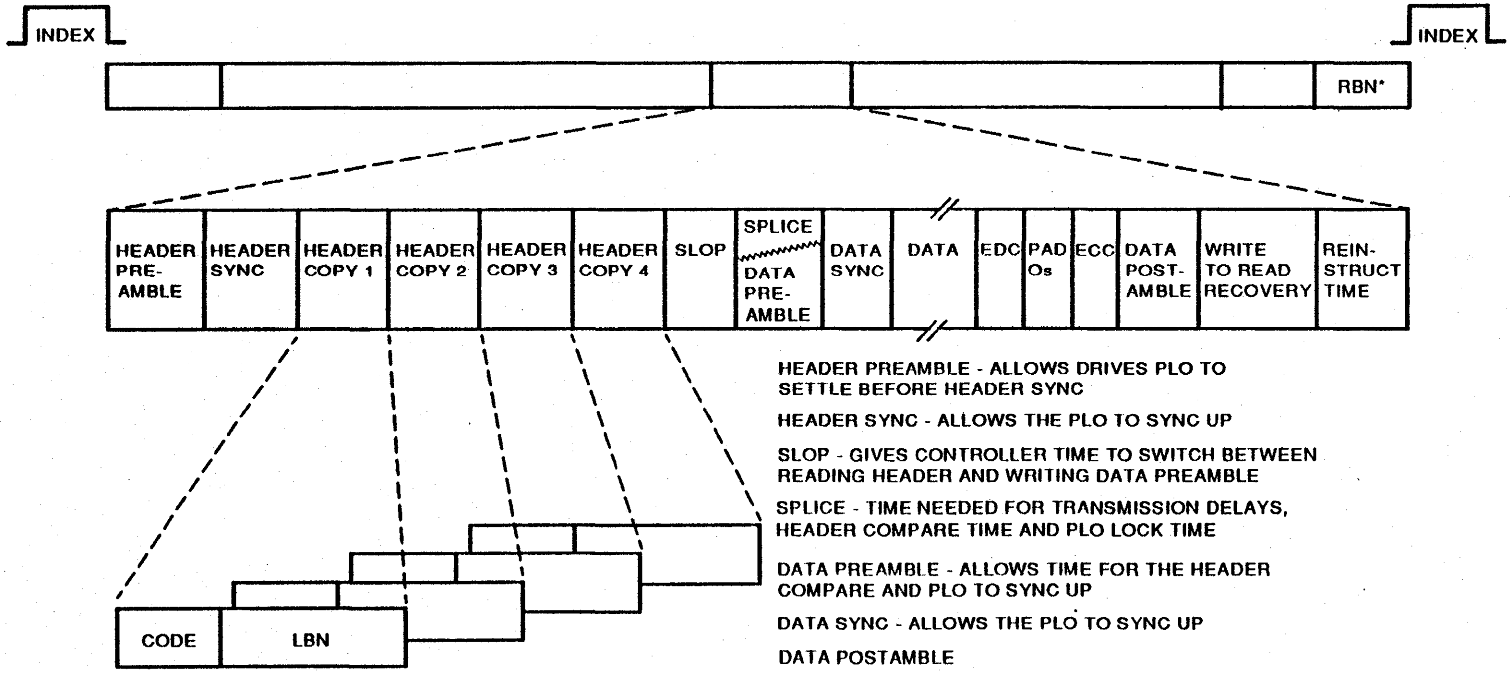
    C. 6 symbols, the maximum correction capability of the controller.

    D. 6 symbols, the maximum threshold determined by the drive

CHAPTER 4

BLOCKS AND HEADERS

Figure 4–1: LBN Sector



HEADER PREAMBLE - ALLOWS DRIVES PLO TO SETTLE BEFORE HEADER SYNC

HEADER SYNC - ALLOWS THE PLO TO SYNC UP

SLOP - GIVES CONTROLLER TIME TO SWITCH BETWEEN READING HEADER AND WRITING DATA PREAMBLE

SPLICE - TIME NEEDED FOR TRANSMISSION DELAYS, HEADER COMPARE TIME AND PLO LOCK TIME

DATA PREAMBLE - ALLOWS TIME FOR THE HEADER COMPARE AND PLO TO SYNC UP

DATA SYNC - ALLOWS THE PLO TO SYNC UP

DATA POSTAMBLE

WRITE TO READ RECOVERY - TIME NECESSARY FOR WRITE RECOVERY PLUS 48 STATE BIT TIMES

REINSTRUCT TIME - TIME ALLOTTED WHILE CONTROLLER IS CLEANING UP CURRENT SECTOR TRANSFER AND SENDING COMMAND FOR THE NEXT ONE

PAD = 8 DATA-BYTE ZEROS FOR 512-BYTE MODE
6 DATA-BYTE ZEROS FOR 576-BYTE MODE

LBN HEADER CODE: See P4-6

00 - USAGE LOGICAL SECTOR LBN (HOST AREA)
03 - UNUSABLE NON-PRIMARY REVECTORED LOGICAL SECTOR LBN (HOST AREA)
05 - UNUSABLE PRIMARY REVECTORED LOGICAL SECTOR LBN (HOST AREA)
11 - UNUSABLE LOGICAL SECTOR (RCT AREA)

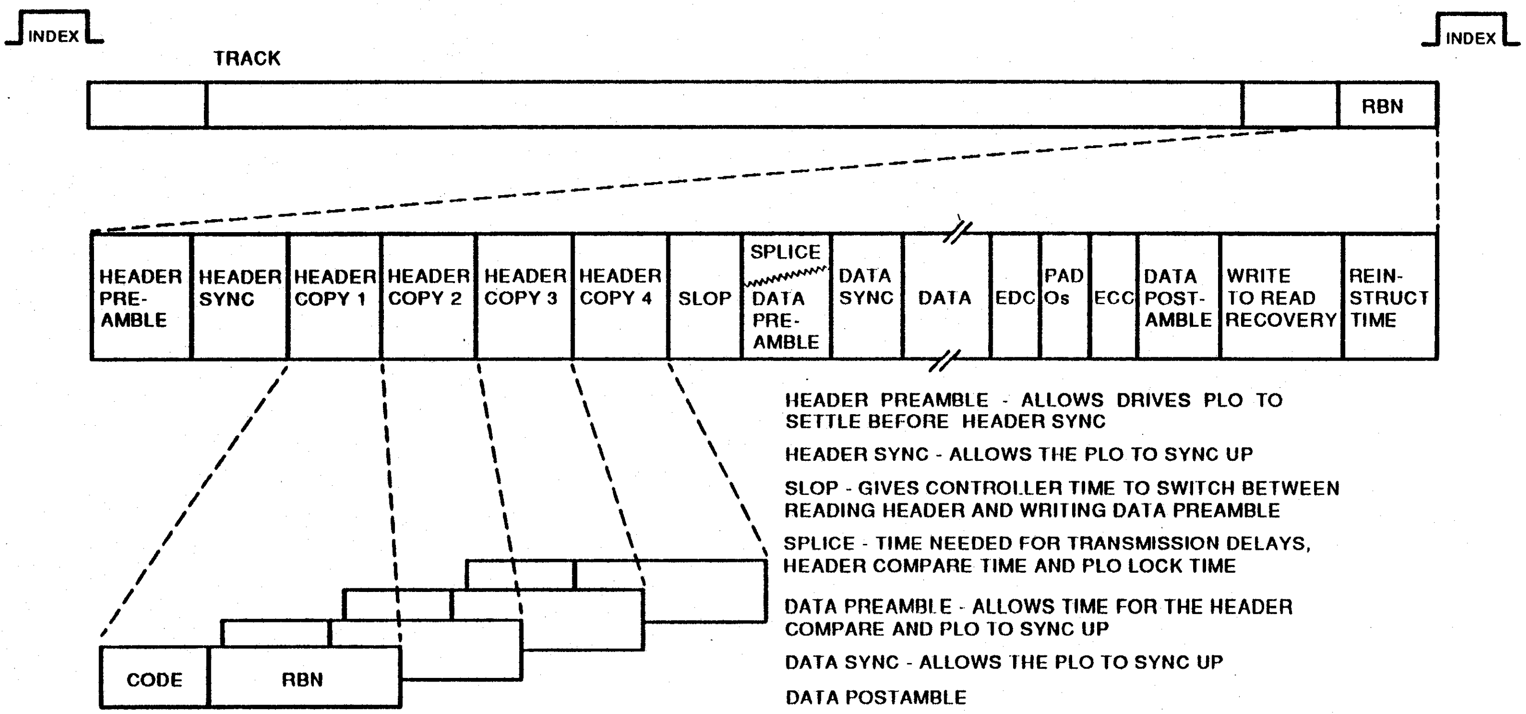
RBN HEADER CODE:

06 - USABLE RBN SECTOR
11 - UNUSABLE SECTOR

*NOTE: REFER TO RBN SECTOR LAYOUT FOR EXPANDED SECTOR.

CXO-2381A

Figure 4–2: RBN Sector



HEADER PREAMBLE - ALLOWS DRIVES PLO TO SETTLE BEFORE HEADER SYNC

HEADER SYNC - ALLOWS THE PLO TO SYNC UP

SLOP - GIVES CONTROLLER TIME TO SWITCH BETWEEN READING HEADER AND WRITING DATA PREAMBLE

SPLICE - TIME NEEDED FOR TRANSMISSION DELAYS, HEADER COMPARE TIME AND PLO LOCK TIME

DATA PREAMBLE - ALLOWS TIME FOR THE HEADER COMPARE AND PLO TO SYNC UP

DATA SYNC - ALLOWS THE PLO TO SYNC UP

DATA POSTAMBLE

WRITE TO READ RECOVERY - TIME NECESSARY FOR WRITE RECOVERY PLUS 48 STATE BIT TIMES

REINSTRUCT TIME - TIME ALLOTTED WHILE CONTROLLER IS CLEANING UP CURRENT SECTOR TRANSFER AND SENDING COMMAND FOR THE NEXT ONE

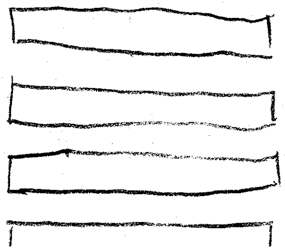
PAD = 8 DATA-BYTE ZEROS FOR 512-BYTE MODE
6 DATA-BYTE ZEROS FOR 576-BYTE MODE

CODE:
06 - USABLE PLACEMENT SECTOR RBN
11 - UNUSABLE SECTOR RBN

See P.4.6

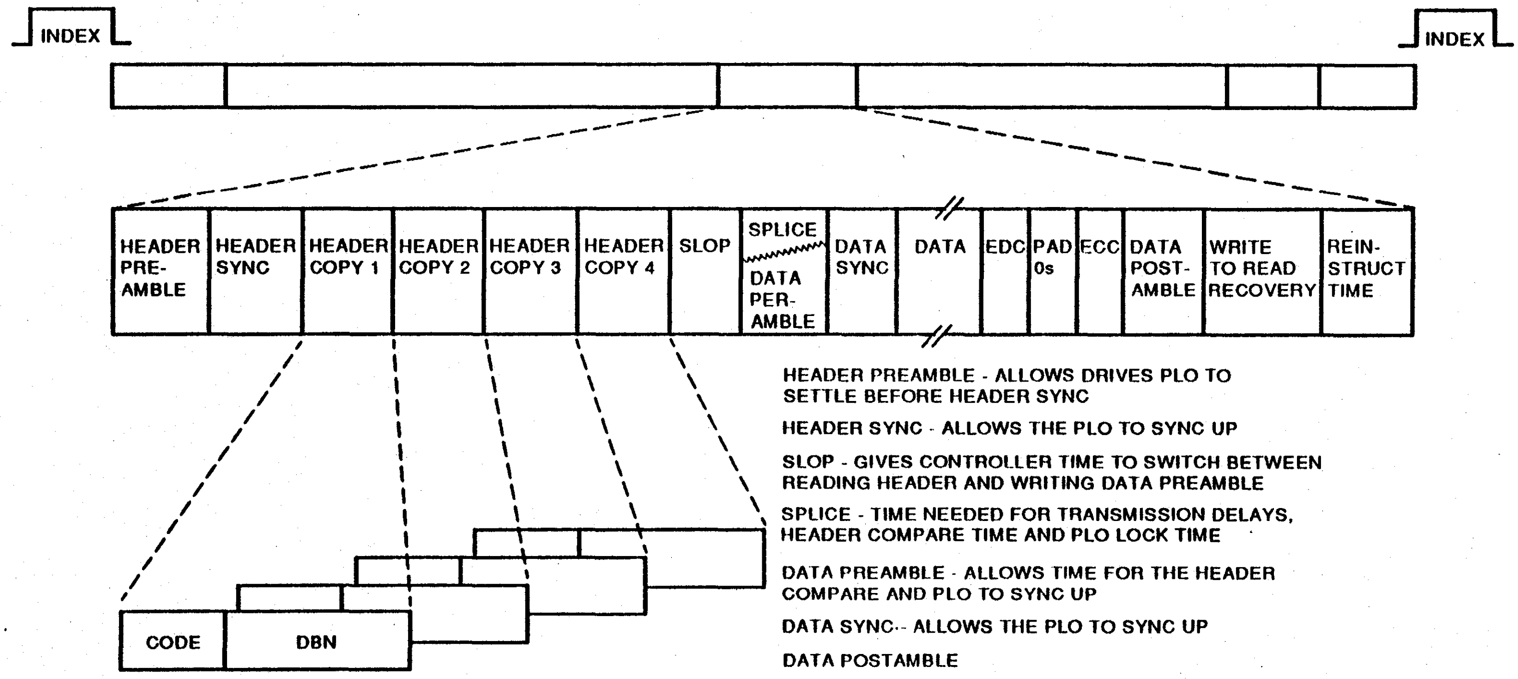
Headers broken down into 2 Parts ea
errors may be reported as read errors in different
Parts of the header Hi/Low ie Low header mismatch

4H 4Lo



CXO-680B

Figure 4–3: DBN Sector



HEADER PREAMBLE - ALLOWS DRIVES PLO TO SETTLE BEFORE HEADER SYNC

HEADER SYNC - ALLOWS THE PLO TO SYNC UP

SLOP - GIVES CONTROLLER TIME TO SWITCH BETWEEN READING HEADER AND WRITING DATA PREAMBLE

SPLICE - TIME NEEDED FOR TRANSMISSION DELAYS, HEADER COMPARE TIME AND PLO LOCK TIME

DATA PREAMBLE - ALLOWS TIME FOR THE HEADER COMPARE AND PLO TO SYNC UP

DATA SYNC - ALLOWS THE PLO TO SYNC UP

DATA POSTAMBLE

WRITE TO READ RECOVERY - TIME NECESSARY FOR WRITE RECOVERY PLUS 48 STATE BIT TIMES

REINSTRUCT TIME - TIME ALLOTTED WHILE CONTROLLER IS CLEANING UP CURRENT SECTOR TRANSFER AND SENDING COMMAND FOR THE NEXT ONE

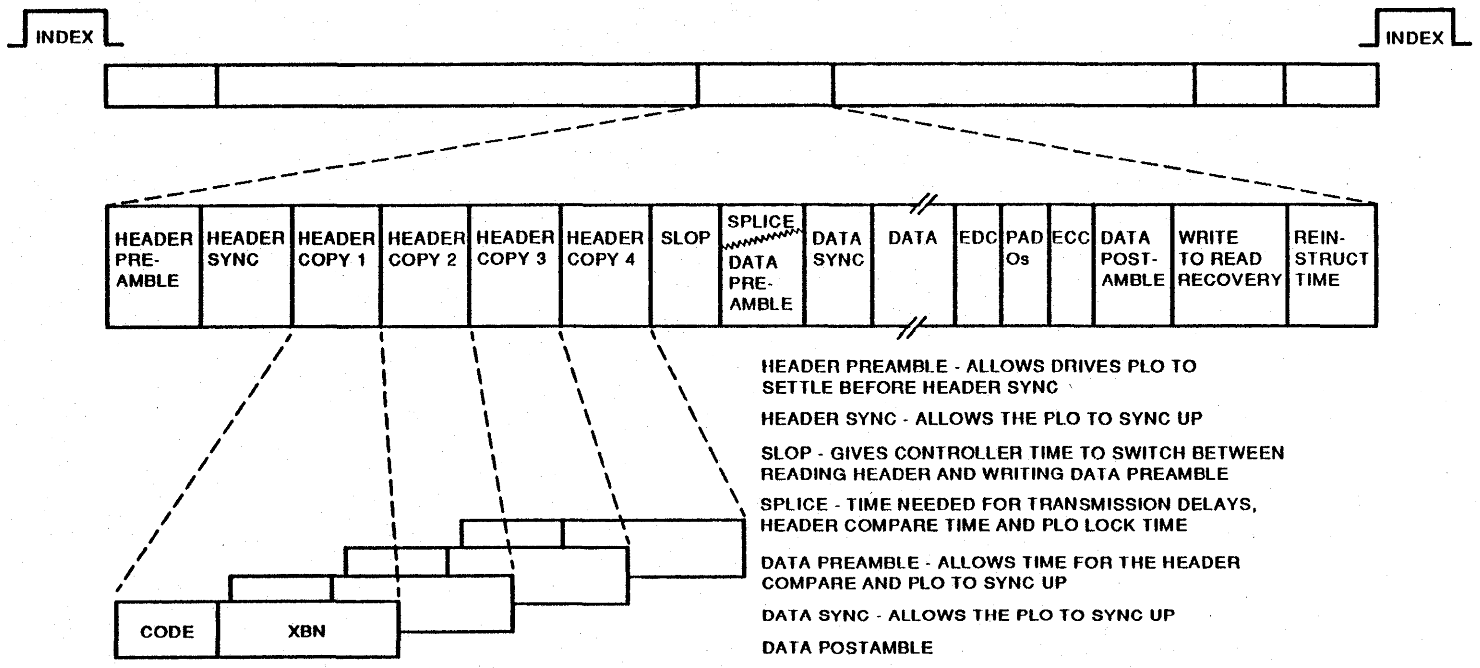
PAD = 8 DATA-BYTE ZEROS FOR 512-BYTE MODE
6 DATA-BYTE ZEROS FOR 576-BYTE MODE

CODE:
11 - UNUSABLE SECTOR DBN
14 - USABLE DIAGNOSTIC SECTOR DBN

See P4.6

CXO-2382A

Figure 4–4: XBN Sector



HEADER PREAMBLE - ALLOWS DRIVES PLO TO SETTLE BEFORE HEADER SYNC

HEADER SYNC - ALLOWS THE PLO TO SYNC UP

SLOP - GIVES CONTROLLER TIME TO SWITCH BETWEEN READING HEADER AND WRITING DATA PREAMBLE

SPLICE - TIME NEEDED FOR TRANSMISSION DELAYS, HEADER COMPARE TIME AND PLO LOCK TIME

DATA PREAMBLE - ALLOWS TIME FOR THE HEADER COMPARE AND PLO TO SYNC UP

DATA SYNC - ALLOWS THE PLO TO SYNC UP

DATA POSTAMBLE

WRITE TO READ RECOVERY - TIME NECESSARY FOR WRITE RECOVERY PLUS 48 STATE BIT TIMES

REINSTRUCT TIME - TIME ALLOTTED WHILE CONTROLLER IS CLEANING UP CURRENT SECTOR TRANSFER AND SENDING COMMAND FOR THE NEXT ONE

PAD = 8 DATA-BYTE ZEROS FOR 512-BYTE MODE
6 DATA-BYTE ZEROS FOR 576-BYTE MODE

CODE:
11 - UNUSABLE SECTOR XBN
12 - USABLE EXTERNAL SECTOR XBN see P.4.6

NOTE: ONLY THE FACTORY SCANNER CURRENTLY WRITES THESE HEADER CODES.

CXO-2383A

4.1 Simplified Summary of Header Codes

The controller reads disk headers when it is searching for a block of data. Each header contains a 4–bit code and a 28–bit block address field. The 4–bit code field contains information to tell the controller where the data can be found. It is the controller's responsibility to determine from the header code where the data resides and to retrieve the data. Since the disk is divided into different areas (HOST/RCT, RBN, DBN, XBN), codes for each area are used to protect against invalid access.

4–Bit Header Codes for LBN (LBNs in Host Applications Area)

00	This is a usable LBN. This code directs the controller to access the data following the header information just read.
03	This LBN is unusable and has been replaced by a non–primary RBN. This code indicates to the controller that the data following the header just read is invalid and directs the controller to retrieve the data from an RBN that is located on a different track than the track containing this LBN. The controller will use the RCT information to determine exactly what RBN was used.
05	This LBN is unusable and has been replaced by a primary RBN. This code indicates to the controller that the data following the header just read is invalid and directs the controller to retrieve the data from the RBN at the end of the current track.

4–Bit Header Codes for LBN (LBNs in RCT Area)

00	This is a usable LBN. This code directs the controller to access the data following the header information just read.
11	This is an unusable LBN (not replaced). This code indicates to the controller that the data following the header just read is invalid and directs the controller to retrieve the data from the next copy of the RCT. If all copies of the RCT are unreadable, an uncorrectable error is reported.

4–Bit Header Codes for RBN

06	This is a usable RBN. This code directs the controller to access the data following the header information just read.
11	This is an unusable RBN. This code indicates to the controller that the data following the header just read is invalid and directs the controller to retrieve the data from another RBN that is located on a different track than the track containing this RBN. The controller will use the RCT information to determine exactly what RBN was used.

4–Bit Header Codes for DBN

14	This is a usable DBN. This code directs the controller to access the data following the header information just read.
11	This is an unusable DBN. This code indicates to the controller that the data following the header just read is invalid. There is no multi–copy protection for DBNs, and DBNs are not replaced if they become defective. The controller will report an uncorrectable error if the data is not retrievable.

4–Bit Header Codes for XBN (In FCT Area)

12	This is a usable XBN. This code directs the controller to access the data following the header information just read.
11	This is an unusable XBN. This code indicates to the controller that the data following the header just read is invalid and directs the controller to retrieve the data from the next copy of the FCT. If all copies of the FCT are unreadable, an uncorrectable error is reported.

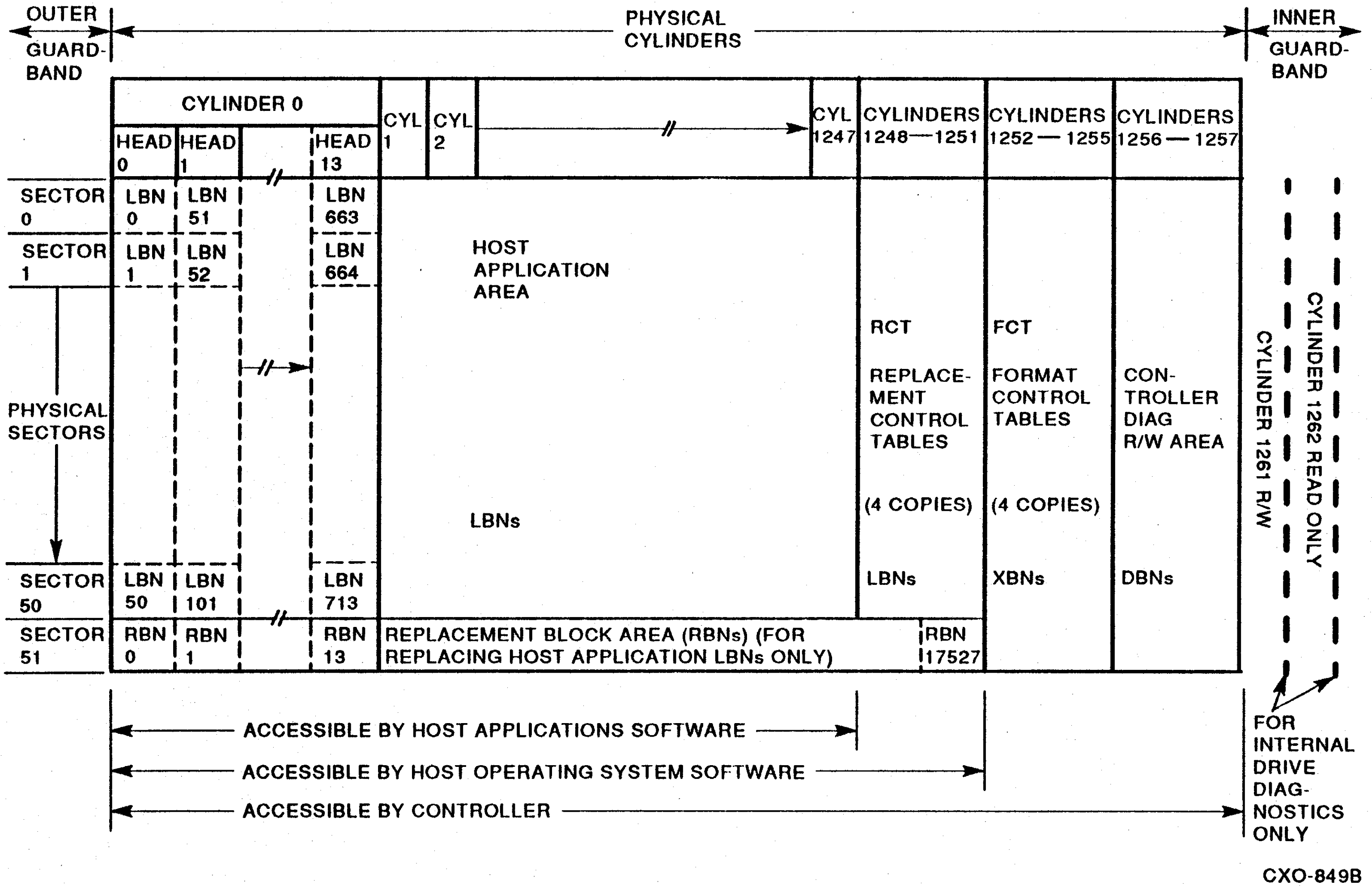
CHAPTER 5

DBN AREA AND R/W DATA PATHS

Figure 5–1: RA81 Topology – 16 Bit

Tests first by reading then Tries writing

Not Formatted to DSDF



CXO-849B

Figure 5–2: R/W Data Path External to Drive/RWDP

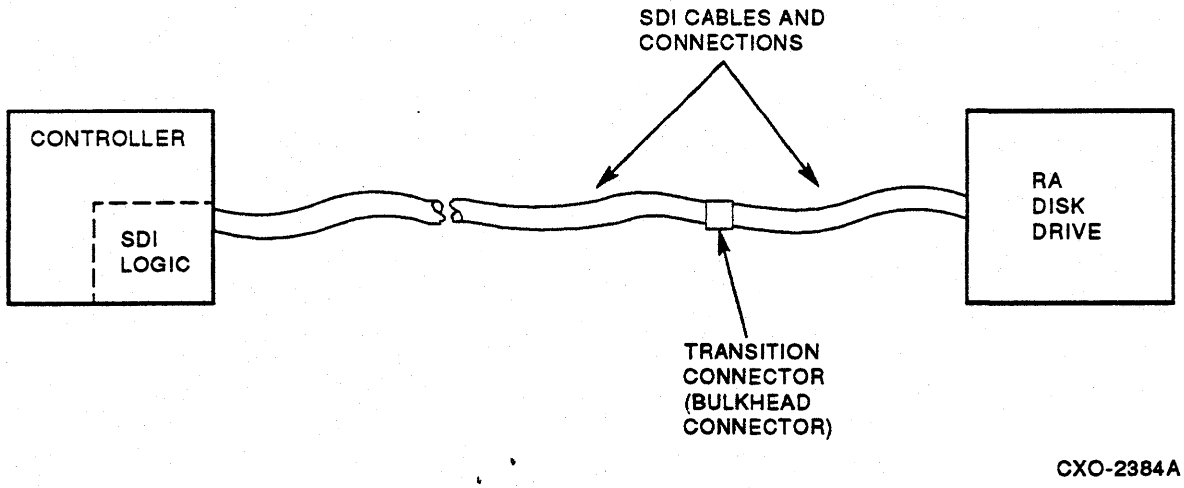
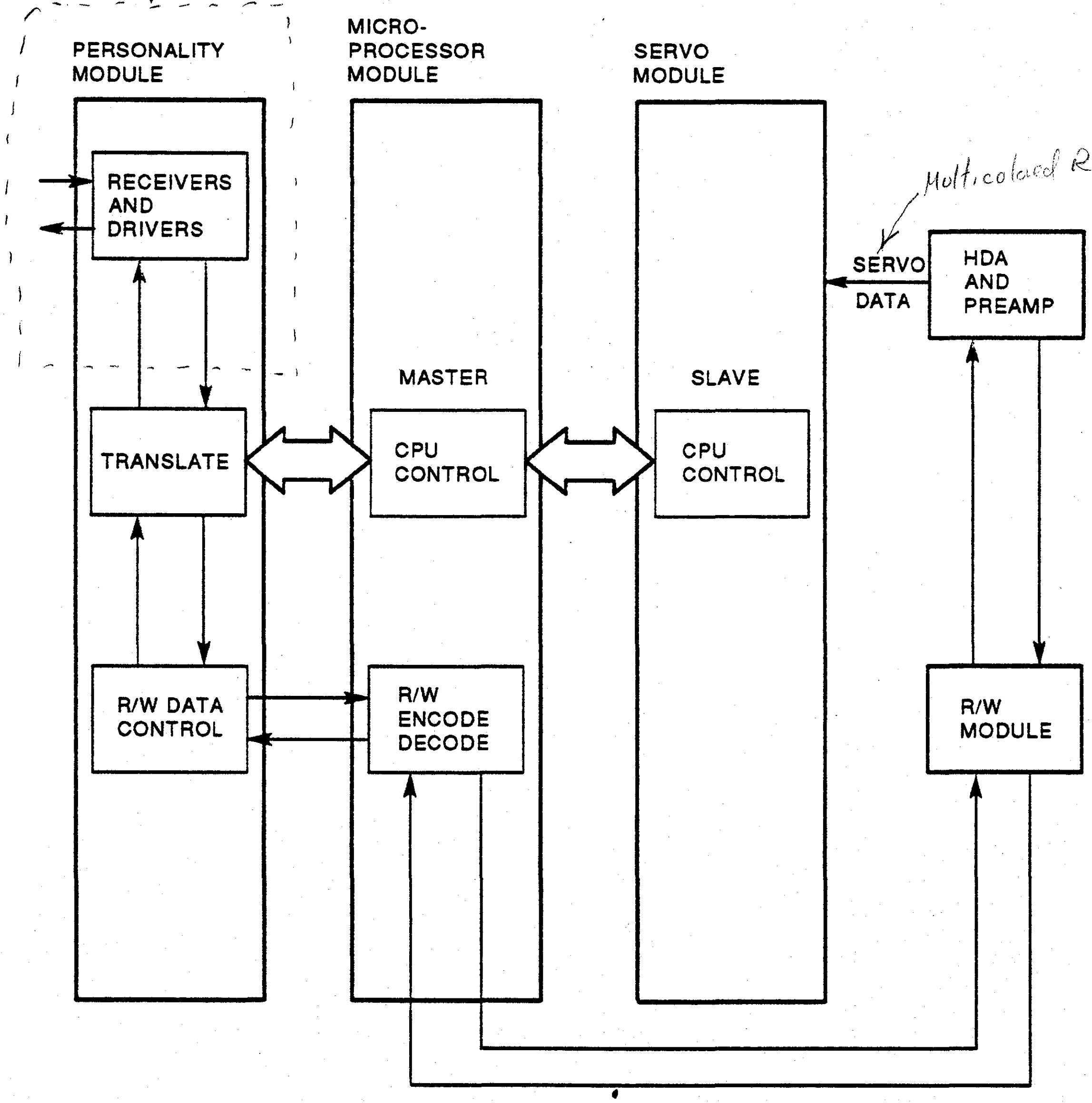


Figure 5–3: Simplified RA81 Block Diagram

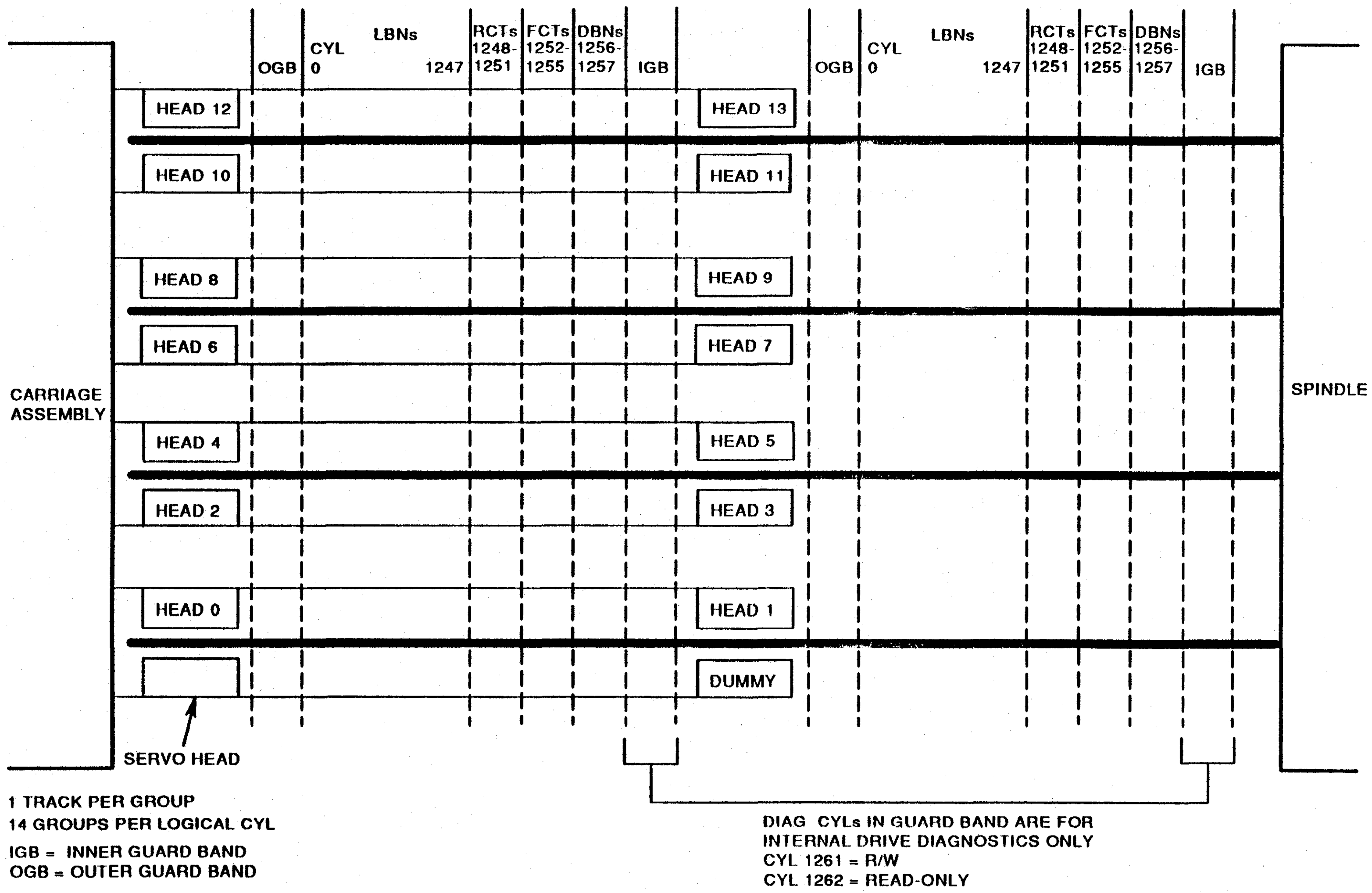
Fix Hardware problems first before fixing structure problem



Not Tested by InterNAL DIAGS

Multicolored Ribbon Cable

Figure 5–4: RA81 Topology and Physical Distribution



1 TRACK PER GROUP
14 GROUPS PER LOGICAL CYL

IGB = INNER GUARD BAND
OGB = OUTER GUARD BAND

BYTE: LOGICAL CYL = PHYSICAL CYL

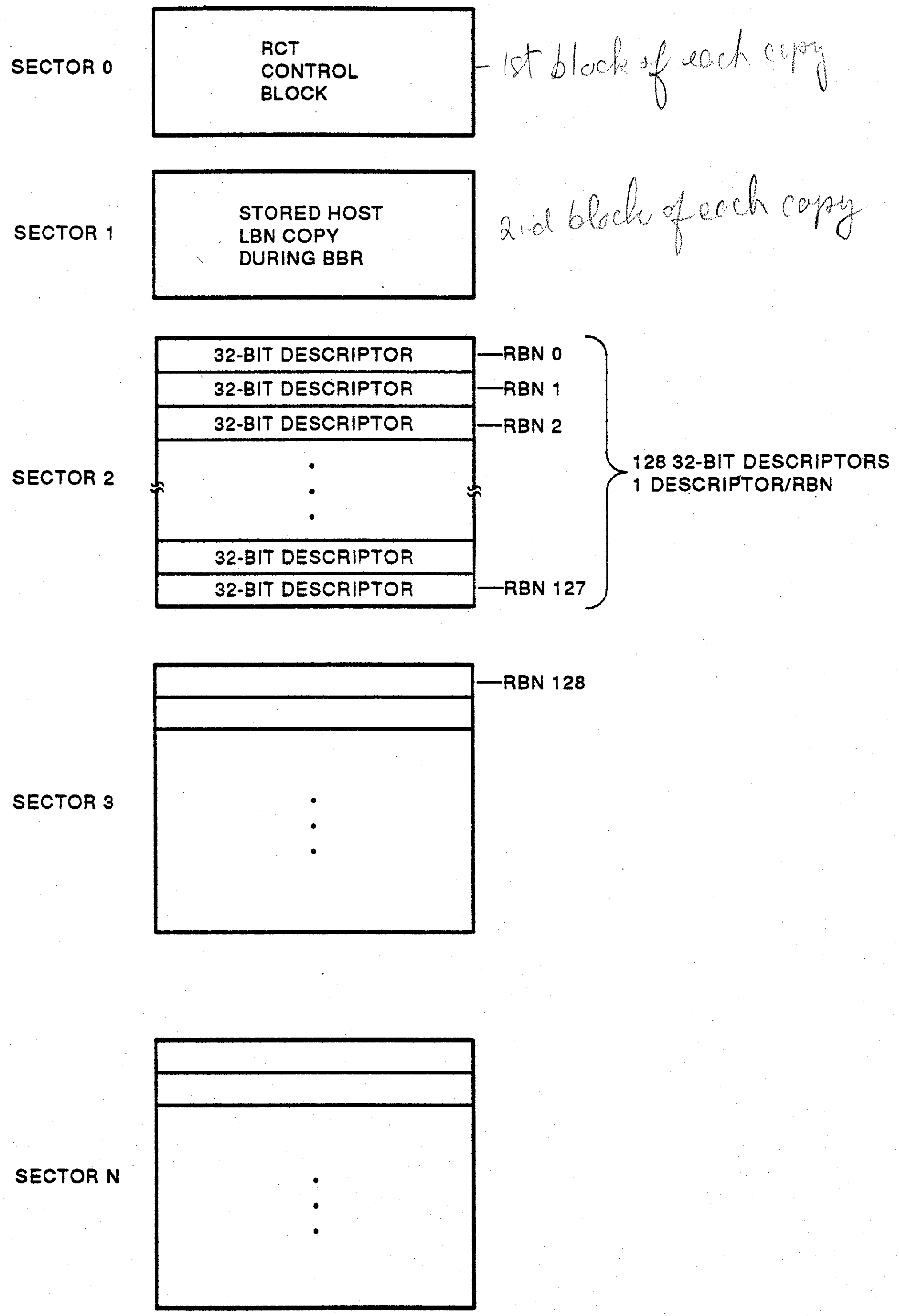
456 MBYTE
1258 CYL/HEAD
52 SECTORS

CXO-2379A

CHAPTER 6

REPLACEMENT CONTROL TABLE (RCT)

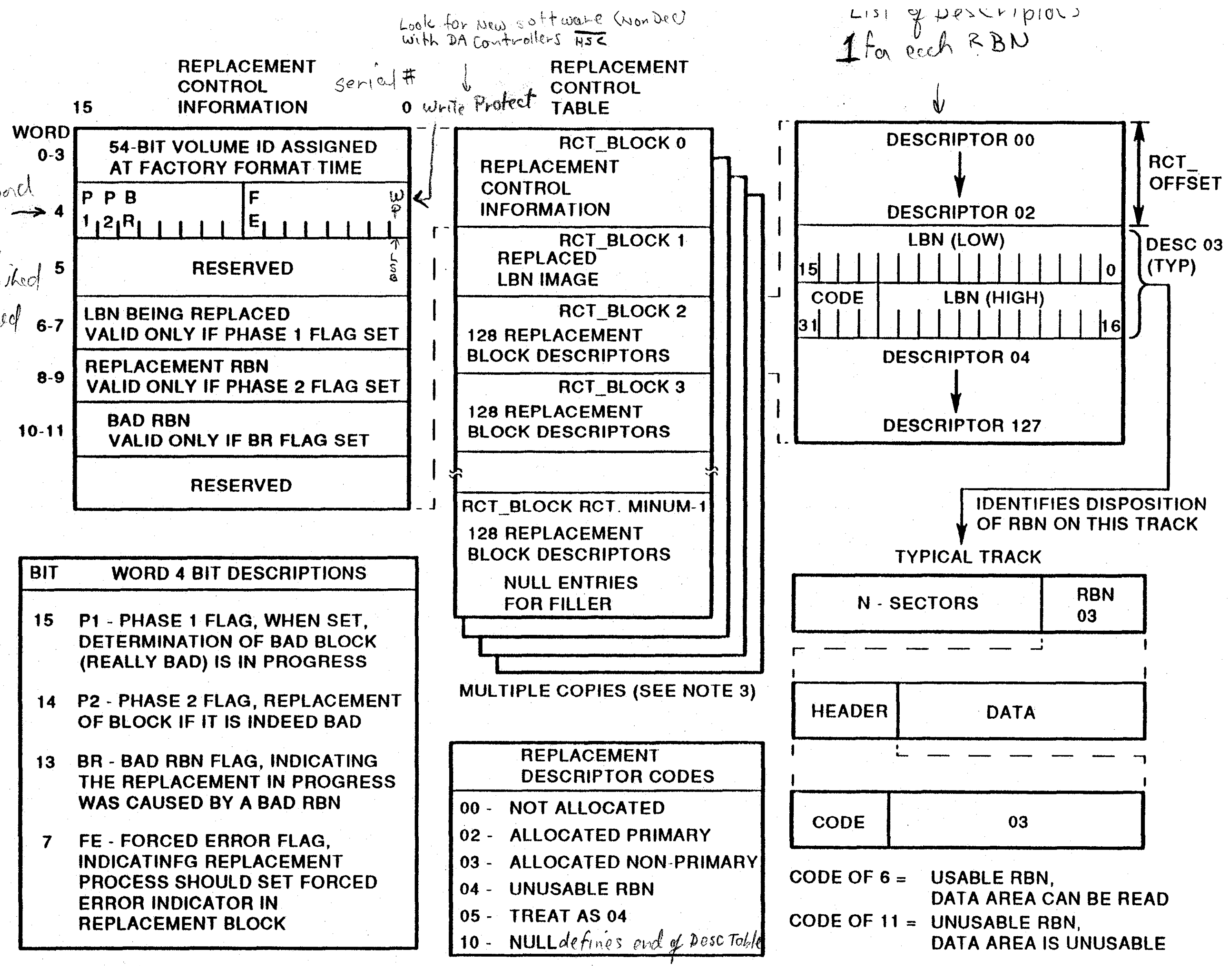
Figure 6–1: Simplified Replacement and Control Table



1st block of each copy

2nd block of each copy

CXO-2370A



Look for New software (WonDeC) with DA Controllers HSC

serial #

Write Protect

List of Descriptors
1 for each RBN

BBR control word

Everytime disk is mount system Looks here to see if unfinished BBR atmps to finished

BIT	WORD 4 BIT DESCRIPTIONS
15	P1 - PHASE 1 FLAG, WHEN SET, DETERMINATION OF BAD BLOCK (REALLY BAD) IS IN PROGRESS
14	P2 - PHASE 2 FLAG, REPLACEMENT OF BLOCK IF IT IS INDEED BAD
13	BR - BAD RBN FLAG, INDICATING THE REPLACEMENT IN PROGRESS WAS CAUSED BY A BAD RBN
7	FE - FORCED ERROR FLAG, INDICATINFG REPLACEMENT PROCESS SHOULD SET FORCED ERROR INDICATOR IN REPLACEMENT BLOCK

REPLACEMENT DESCRIPTOR CODES	
00 -	NOT ALLOCATED
02 -	ALLOCATED PRIMARY
03 -	ALLOCATED NON-PRIMARY
04 -	UNUSABLE RBN
05 -	TREAT AS 04
10 -	NULL defines end of Desc Table

NOTES:

1. RBN DESCRIPTOR CODES ARE DIFFERENT FROM HEADER CODES.
2. RBN HEADER CODES ARE DIFFERENT FROM LBN HEADER CODES.
3. EACH COPY IS PLACED ON DIFFERENT SURFACES AND DIFFERENT CYLINDERS FOR PROTECTION.

CXO-2366A

Figure 6–2: Detailed Replacement and Control Table

6.1 THE REPLACEMENT CONTROL TABLE

The replacement control table (RCT) records the status of each replacement block on the unit and the location of all revectored logical blocks. The RCT is a multi–copy structure. The subsystem provides the host with the number of copies of the RCT and an offset which enables the host to compute the location of the next copy of an RCT block.

The RCT is a two–part structure. The first part of the structure contains two blocks: a flags/control block and a temporary data storage block. The second part of the structure is an array of replacement block descriptors with an entry for each replacement block on the unit, and it is organized in ascending RBN order. There are as many sectors in the second half of the table as are required for replacement block descriptor storage.

There are n copies of the RCT in the RCT area, where n is a device characteristic. Each copy of the RCT is located "rct" LBNs from the previous copy. Copy 1 of the RCT is the base copy. The remaining copies provide individual backup blocks for the corresponding blocks in the base copy of the RCT. Both n, the number of RCT copies, and "rct," the offset to the next RCT copy, are passed to the host as unit characteristics in the response to the MSCP GET UNIT STATUS command.

While the size of the host application area is specified to the host, the size of the RCT area is not specified. The host is guaranteed that the RCT area will be at least large enough to contain n copies of the RCT. If any blocks in the RCT area are not actually used by an RCT copy, they are reserved and are not to be used by the host.

The following restrictions apply to RCT space access:

1. The subsystem must prohibit spiraling from the host application portion of LBN space into the RCT space.
2. I/O to the RCT must be a single block operation. This requirement does not have to be enforced by the subsystems, but it is required by the replacement algorithms. Transfers other than one block in length may have undefined results.
3. Any portion within the RCT space that is not used for a replacement control table is controller specific and must not be accessed by the host. Host access to any part of the RCT space other than within a replacement table may have undefined results.
4. Host write access to the RCT is prohibited during controller–initiated bad block replacement (BBR).
5. Controller write access to the RCT is prohibited during host–initiated BBR.

6.2 RBN DESCRIPTOR FORMAT

Each entry in the second part of the replacement control table points to a replacement block on the unit. The table is in ascending RBN order. Thus, the first entry corresponds to the RBN 0 on the unit, the second entry corresponds to the RBN 1 on the unit, etc. Entries that do not correspond to RBNs on the unit may be present to pad the RCT to a block boundary. Any entry which does not correspond to an RBN on the unit is called a null entry. There is always one null entry at the end of the RCT to demarcate the end the table. All other entries past this last null entry are undefined.

The format of a replacement block descriptor in the replacement control table is:

Figure 6–3: Replacement Block Descriptor

Right word is MSW
Left word is LSW

<table>
<tr><td colspan="2">LBN (LOW)</td></tr>
<tr><td>CODE</td><td>LBN (HIGH)</td></tr>
</table>

CXO-2367A

LBN is the logical block number of a revectored logical block.

CODE is one of the following octal values:

- 00 – Unallocated (empty) replacement block.
- 02 – Allocated replacement block – primary replacement block.
- 03 – Allocated replacement block – NON–primary replacement block.
- 04 – Unusable replacement block.
- 05 – Alternate unusable replacement block. Code 05 is reserved. Programs should treat this code as if it were code 04.
- 10 – Null entry – no corresponding replacement block.

For codes 00, 04, and 10 the LBN field is always zero.

6.3 PHYSICAL LAYOUT OF THE RCT

The n copies of the RCT are stored at the highest addresses of the LBN space. Each sector in the second part of the RCT contains 128 entries, regardless of the actual disk format (bytes 512 through 575 of 576 byte sectors are zero filled).

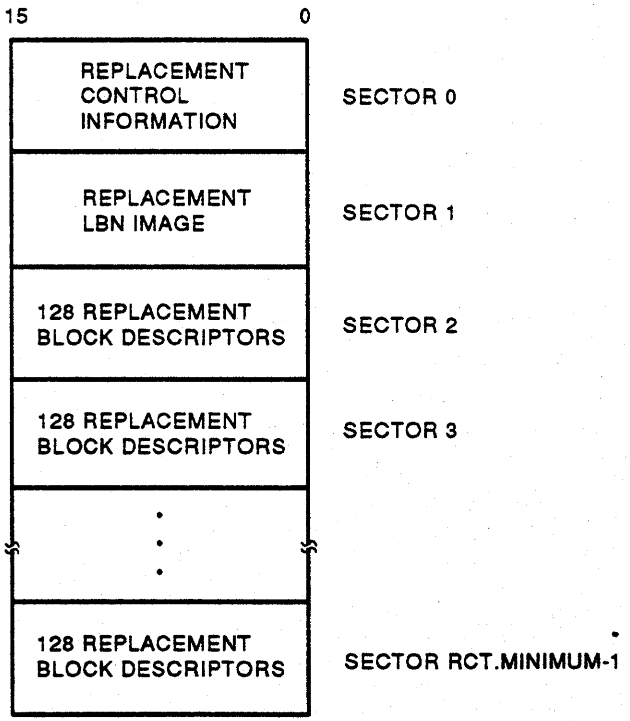
The size of the copies must be adjusted so that corresponding blocks of each copy are accessed using physically distinct components to the extent possible. This implies that:

- If the number of copies is less than or equal to the number of heads, then corresponding blocks of each copy must be accessed by different heads.
- If the number of copies is greater than the number of heads, then corresponding blocks of each copy must be distributed as evenly as possible across the heads.
- If a device uses a dedicated servo surface, then corresponding blocks of each copy must be located using different tracks of the servo surface.

The first sector in the RCT contains information about the state of any replacement operation that may be in progress. A copy of the volume serial number is contained in this sector to allow validation of the RCT by diagnostics. The second sector in each copy of the RCT is used by the bad block replacement algorithm. This sector is used to hold a copy of the data from the sector being replaced.

The remaining sectors each contain 128 32–bit replacement block descriptors. The RCT structure is shown in Figure 6–4. Sector 0 of the RCT is illustrated in Figure 6–5.

Figure 6–4: RCT Structure



CXO-2368A

Figure 6–5: RCT Sector 0

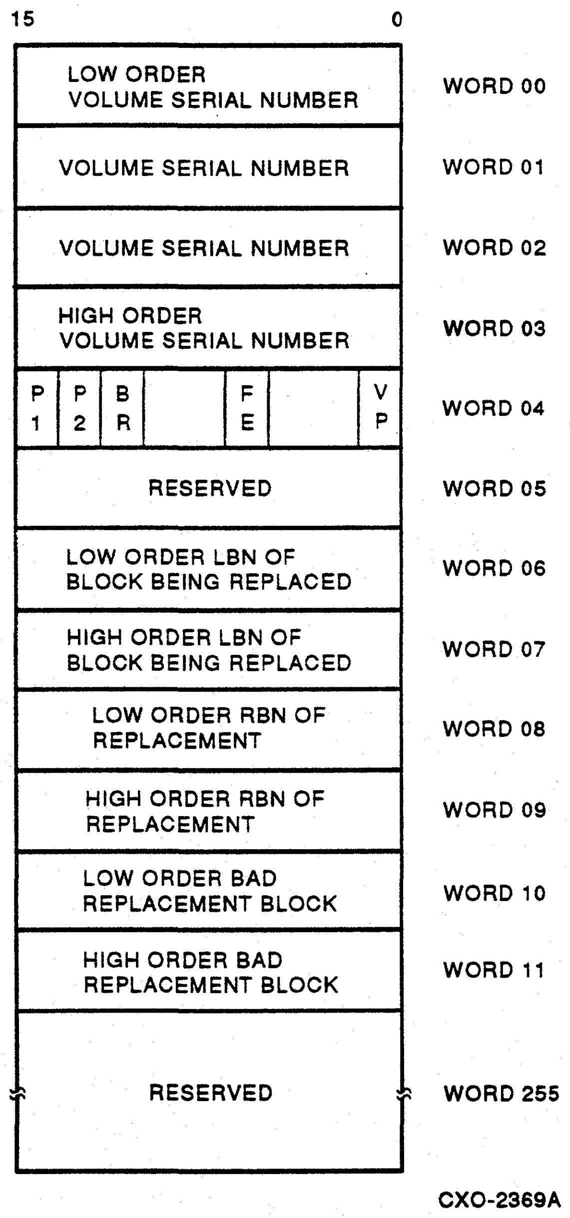


Table 6–1: RCT Block 0 Defined

WORD 0—WORD 3:	The 64–bit volume ID assigned during the factory formatting process. If the pack was formatted without the use of factory format information, a site volume ID must be input to the formatter for entry into this field. The low order 32 bits of this field are used as a volume ID in MSCP log packets.
WORD 4:	This word contains the status flags used during the bad block replacement process.
BIT 7: FE	The force error flag, indicating that the replacement process should set the forced error indicator in the target replacement block. This flag is reset when the replacement operation finishes. The flag is initially reset.
BIT 13: BR	The bad replacement block flag, indicating that the replacement in progress was caused by a bad replacement block. This flag is reset when the replacement operation finishes. The flag is initially reset.
BIT 14: P2	The phase 2 flag, indicating that the replacement process is in phase 2 of the replacement algorithm. If this flag is set when the unit comes on line, it indicates that a replacement was interrupted and must be completed. This flag is reset when replacement is completed. The flag is initially reset.
BIT 15: P1	The phase 1 flag, indicating that the replacement process is in phase 1 of the replacement algorithm. If this flag is set when the unit comes on–line, it indicates that a replacement was interrupted and must be completed. This flag is reset when phase 1 is completed. The flag is initially reset.

NOTE

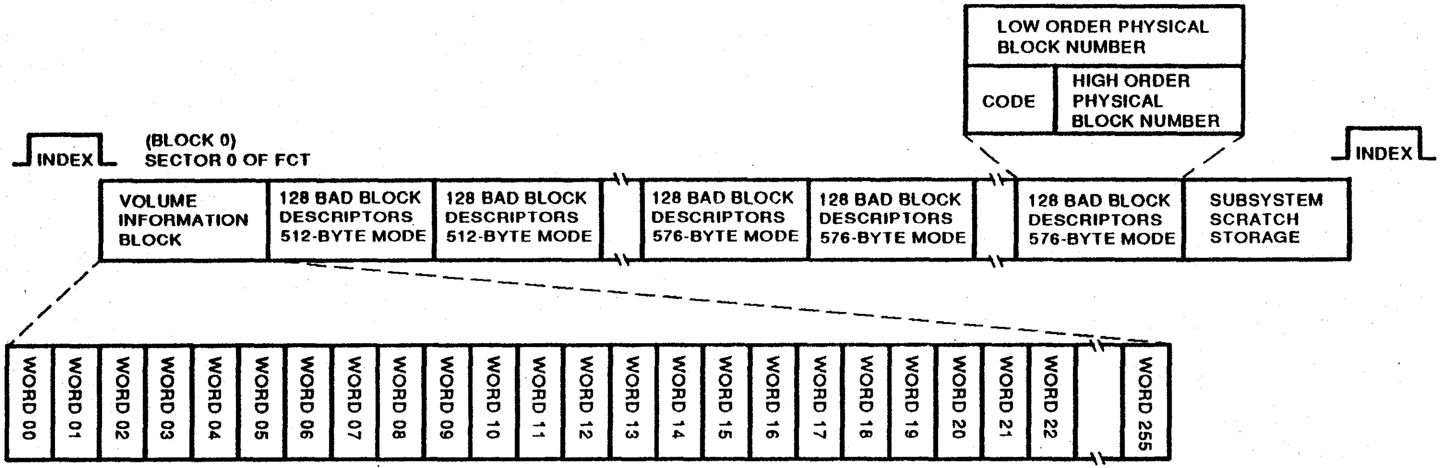
If any other bit in word 4 becomes set, whether deliberate or by accidental corruption, the controller usually considers the media to be VOLUME DATA SAFETY WRITE PROTECTED. The host operating system, in turn, is prevented from performing write data operations to the media.

WORD 6—WORD 7:	A copy of the LBN of the block being replaced, if a replacement operation is in progress. This field is invalid if the P1 flag is not set. This field is initialized to zero.
WORD 8—WORD 9:	A copy of the RBN of the block with which the LBN is being replaced, if a replacement operation is in progress. This field is invalid if the P2 flag is not set. This field is initialized to zero.
WORD 10—WORD 11:	The RBN of the bad replacement block being replaced. This field is invalid if the BBR flag is not set. This field is initialized to zero.

CHAPTER 7

FORMAT CONTROL TABLE (FCT)

Figure 7–1: FCT Layout



If 0 Formatter Started
+ Never finished
Complete format fixes this!

WORD:
0 - MEDIA MODE
    126736 = 512-BYTE MODE
    074161 = 576-BYTE MODE
1 - FORMATTING INSTANCE NUMBER
2-5 - VOLUME SERIAL NUMBER
    WORD 2 IS LEAST SIGNIFICANT
    WORD 5 IS MOST SIGNIFICANT
6-9 - DATE VOLUME FIRST FORMATTED
    WORD 6 IS LEAST SIGNIFICANT
    WORD 9 IS MOST SIGNIFICANT
10-13 - DATE OF MOST RECENT FORMATTING
    WORD 10 IS LEAST SIGNIFICANT
    WORD 13 IS MOST SIGNIFICANT
14-15 - NUMBER OF USED 512 TABLE ENTRIES
    WORD 14 IS LEAST SIGNIFICANT
    WORD 15 IS MOST SIGNIFICANT
16-17 - NUMBER OF USED 576 TABLE ENTRIES
18-19 - SCRATCH AREA ADDRESS
    WORD 18 IS LEAST SIGNIFICANT
    WORD 19 IS MOST SIGNIFICANT
20 - SIZE OF THE SCRATCH AREA IN THIS FCT COPY
21 - INCLUDES FK BIT (SEE NOTE)
22 - VERSION NO. OF THE FORMAT
23-UP - 0s

CODE:
X0 - UNUSED ENTRY
X1 - BAD HEADER
X2 - OTHER BAD FIELD
X4 - BAD DATA (INCLUDING EDC AND ECC FIELDS)
X = 1 IF A PBN IN HOST LBN AREA WAS PRIMARY REPLACEMENT BLOCK
X = 0 IF A PBN IN HOST LBN AREA WAS SECONDARY REPLACEMENT BLOCK
X = 1 IF A PBN MAPS INTO RBN, FCT, OR DBN AREAS
(PBN = PHYSICAL BLOCK NUMBER)

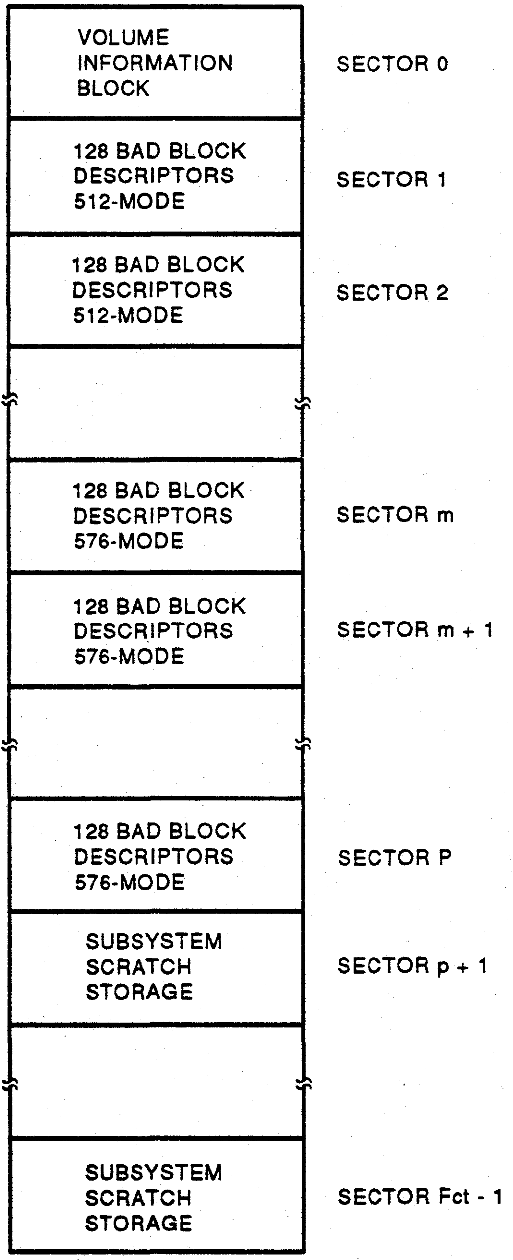
NOTE: THE FK BIT SET INDICATES THAT THIS IS A FAKE FCT AREA, AND THAT SECTOR 0 (BLOCK 0) OF THIS FCT AREA IS THE ONLY BLOCK WITH VALID INFORMATION. ALL REMAINING SECTORS IN THIS FCT AREA ARE MEANINGLESS AND CONTAIN NO USABLE INFORMATION. THE FK BIT IS MSB (BIT 15) OF WORD 21.

CXO-2371A

7.1 FCT STRUCTURE

Each copy of the FCT is composed of one volume information block, one 512–byte format table, one 576–byte format table, and one subsystem temporary storage area (distributed among the alignment pads). The 576–byte table is normally only supplied by manufacturing with 576–byte formatted media. An FCT copy has the format shown in Figure 7–2. Details of FCT block 0 (volume information block) are shown in Figure 7–3.

Figure 7–2: FCT Structure



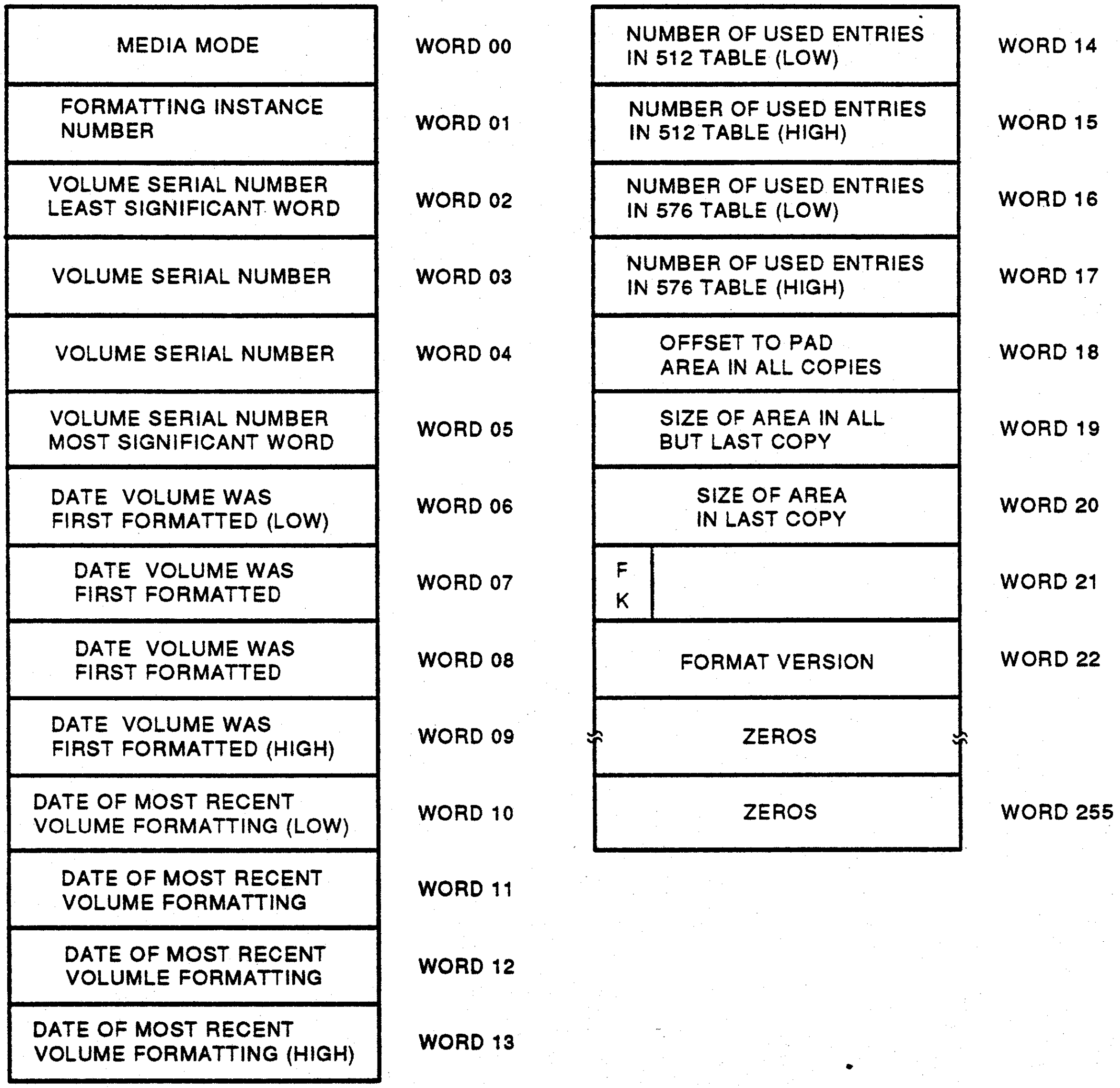
CXO-2372A

The XBN area itself is always formatted to contain 512–byte sectors.

Sector m is the first block of the table to store the factory detected bad blocks for 576–byte formatted disks. Sector p+1 is the first block of the subsystem scratch storage, reserved for use by controllers and formatting utilities, as needed. Sector FCT–1 marks the end of a copy of the FCT.

Sector 0 contains various volume identification and format information. The format is shown in Figure 7–3.

Figure 7–3: FCT Sector 0 – (Volume Information Block)



MEDIA MODE	WORD 00
FORMATTING INSTANCE NUMBER	WORD 01
VOLUME SERIAL NUMBER LEAST SIGNIFICANT WORD	WORD 02
VOLUME SERIAL NUMBER	WORD 03
VOLUME SERIAL NUMBER	WORD 04
VOLUME SERIAL NUMBER MOST SIGNIFICANT WORD	WORD 05
DATE VOLUME WAS FIRST FORMATTED (LOW)	WORD 06
DATE VOLUME WAS FIRST FORMATTED	WORD 07
DATE VOLUME WAS FIRST FORMATTED	WORD 08
DATE VOLUME WAS FIRST FORMATTED (HIGH)	WORD 09
DATE OF MOST RECENT VOLUME FORMATTING (LOW)	WORD 10
DATE OF MOST RECENT VOLUME FORMATTING	WORD 11
DATE OF MOST RECENT VOLUMLE FORMATTING	WORD 12
DATE OF MOST RECENT VOLUME FORMATTING (HIGH)	WORD 13
NUMBER OF USED ENTRIES IN 512 TABLE (LOW)	WORD 14
NUMBER OF USED ENTRIES IN 512 TABLE (HIGH)	WORD 15
NUMBER OF USED ENTRIES IN 576 TABLE (LOW)	WORD 16
NUMBER OF USED ENTRIES IN 576 TABLE (HIGH)	WORD 17
OFFSET TO PAD AREA IN ALL COPIES	WORD 18
SIZE OF AREA IN ALL BUT LAST COPY	WORD 19
SIZE OF AREA IN LAST COPY	WORD 20
F K	WORD 21
FORMAT VERSION	WORD 22
ZEROS	
ZEROS	WORD 255

CXO-2373A

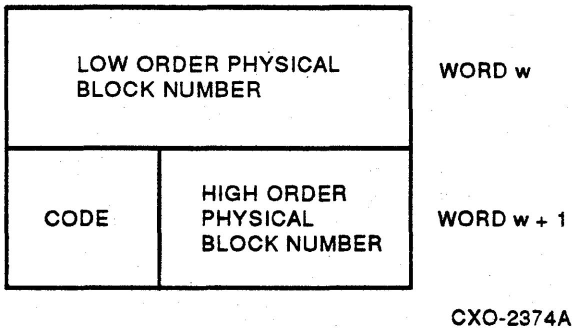
7.2 VOLUME INFORMATION BLOCK DETAILS

WORD 0:	**MEDIA MODE**—is 126736 for a 512–byte format and 074161 for a 576–byte format. During formatting the media mode word is set to zero.
WORD 1:	**Formatting Instance Number**—is a counter that is incremented each time the HDA or volume is formatted. Initialized to 1 at the factory.
WORD 2–5:	**Volume Serial Number**—is the HDA or volume identification.
WORD 6–9:	**Time and Date of the First Formatting** —is expressed in a quad–word field as the number of clunks since 00:00 o'clock, Nov. 17, 1858 (in the local time zone), or zero if the current time and date is not available. A clunk has a value of 100 nanoseconds. This is the standard VAX/VMS time and date format.
WORD 10–13:	**Time and Date of the Most Recent Formatting**—is the date that the media was last formatted.
WORD 14–15:	**The Number of Used 512 Table Entries**—indicates how many of the entries in the 512 byte table are used.
WORD 16–17:	**The Number of Used 576 Table Entries**—indicates how many of the entries in the 576 byte table are used.
WORD 18:	**Scratch Area Offset**—is the offset, in words, counted from the beginning of the FCT to the scratch area in all copies of the FCT.
WORD 19:	**Size of Area**—is the size of the scratch area in all copies of the FCT, except the last copy of the FCT.
WORD 20:	**Scratch Area Size of Last Copy** – is the size of the scratch area in the last copy of the FCT.
WORD 21:	**Bit FK**—is set if this is a fake FCT (i.e., only the first block exists). If this bit is set, then only the media mode, format instance number, serial number, date of last format, and pad area pointers (words 18–20) are valid. The format instance number will be 0. The contents of all other words in block 0 and all blocks following block 0 are undefined and, therefore, considered invalid.
WORD 22:	**Format Version**—is the version number of the format

Sectors 2 through m–1 contain the 512–byte mode bad block descriptors. Each descriptor describes the physical block (PBN) on the HDA or volume that is bad and the problem that has been detected. Additional information is contained in the code field for the use of the formatter in allocating primary and secondary replacements.

The format of a bad block descriptor is shown in Figure 7–4.

Figure 7–4: Bad Block Descriptor



Where:

PHYSICAL BLOCK NUMBER—is the relative position of the sector from the beginning of the HDA or volume.

CODE is an indication of the problem (reason) that caused the sector to be retired. The legal values for code are:

X0 – Unused entry.
X1 – Bad header.
X2 – Other bad field.
X4 – Bad data (including EDC and ECC fields).

The formatter uses bit 15 (X) of the code field to indicate that the bad PBN, if a non–RCT LBN, was replaced by a primary RBN, X=1, or a non–primary RBN, X=0. X will also equal 1 for those PBNs that map into RBNs, XBNs, DBNs, or LBNs in the RCT. These bits are set only in those formatting modes that result in the creation of (or re–creation of an invalid) FCT.

Bad block descriptors are sorted in descending track order within each sub–table (512 and 576). The entries are further sorted in ascending PBN order within each track.

A single unused entry is placed at the end of the sorted list in each table. The values in the remaining unused entries are undefined.

Sectors m through p contain the 576 byte mode bad block descriptors. The format of these descriptors is identical to that of the 512 byte descriptors.

CHAPTER 8

STANDARD DISK INTERFACE (SDI)

8.1 INTRODUCTION

This section describes the Standard Disk Interface (SDI) protocol that allows SDI controllers to communicate with SDI disks in the Digital Standard Architecture (DSA) disk subsystem.

This section stresses the Standard Disk Interface (SDI) characteristics that DSA disk drives implement. You will learn some SDI drive characteristics and how drive responsibilities differ from those of the SDI controllers.

8.2 OBJECTIVES

Upon completion of this discussion, you will be able to:

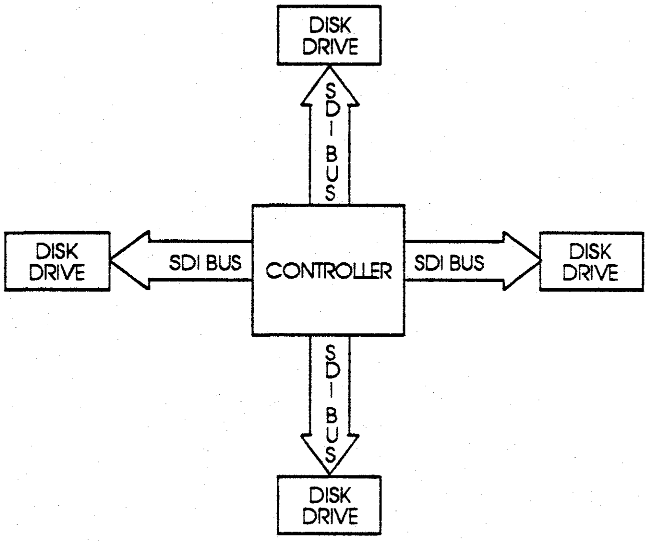
1. Identify and define the use of the four lines that comprise the SDI.
2. Identify and describe the use of the SDI bus encoding scheme.
3. Describe the format of the RTCS line and define the use of each signal.
4. Describe the format of the RTDS line and define the use of each signal.
5. Define the different drive states relative to the controller.
6. Describe the format used by the controller to send commands to the drive.
7. Describe the format used by the drive to send responses back to the controller.
8. Describe the events that occur between the controller and the drive during a seek operation.
9. Describe the events that occur between the controller and the drive during a read and a write operation.

8.3 SDI BUS

Disk drives and controllers within a DSA disk subsystem communicate with each other using a standard protocol. This protocol is transmitted over the Standard Disk Interconnect (SDI) bus.

A separate SDI bus connects the controller to each drive. Refer to Figure 8–1. This radial configuration allows simultaneous transactions to occur from more than one drive to the same controller. The radial bus allows you to disconnect a drive that is not being used while the controller continues to service the host on other drives.

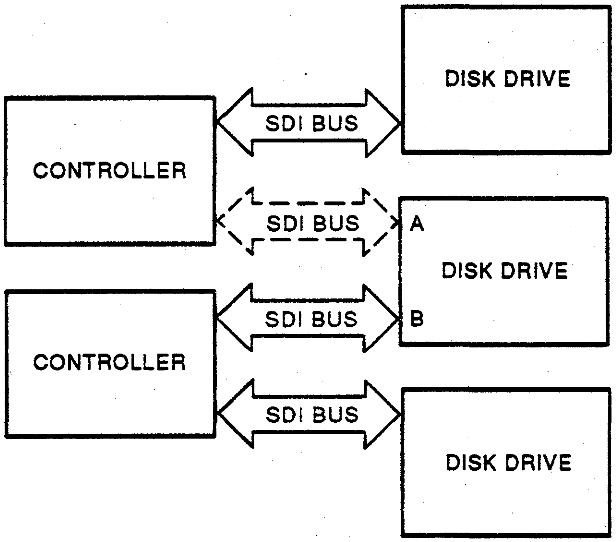
Figure 8–1: SDI Radial Bus



MLDS-1339A

In a dual–port configuration, the radial bus also shows it is possible to disconnect a controller from a drive port. Refer to Figure 8–2. The drive can continue to service the other controller on its other port.

Figure 8–2: SDI Dual Port

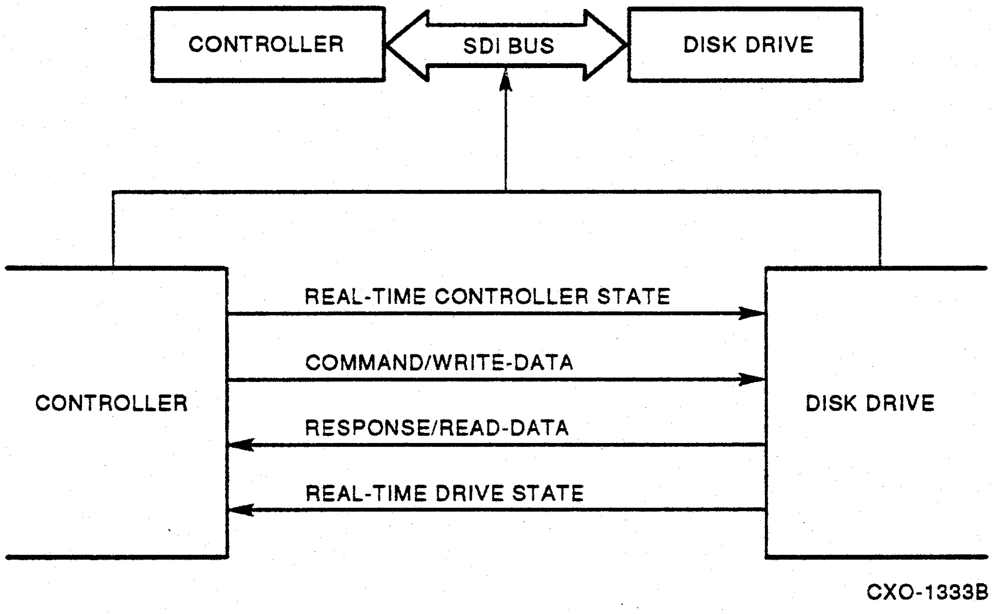


CXO-2348A

8.3.1 SDI Lines

The SDI bus consists of four high–speed, unidirectional lines. Each line transmits serial information in only one direction. See Figure 8–3.

Figure 8–3: SDI Bus



The Real Time Controller State (RTCS) line repeatedly transmits controller state information to the drive.

The Real Time Drive State (RTDS) line repeatedly sends drive state information to the controller.

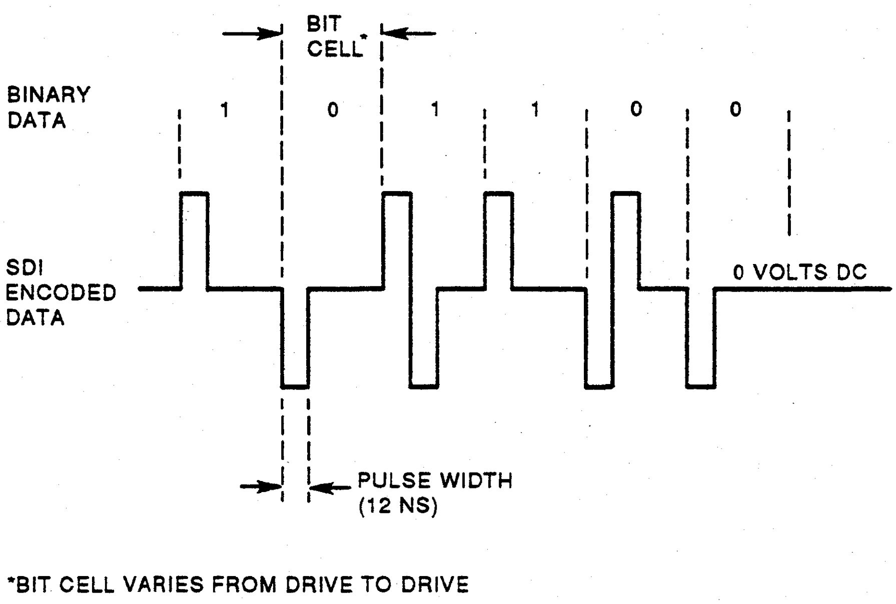
The Command/Write–Data line serves two purposes. It transmits commands and parameters to the drive. It also transmits write data to the recording surfaces in the drive. This line is also called the WRT/CMD line.

The Response/Read–Data line also performs two functions. It sends drive response messages to the controller. It also transmits read data from the recording surfaces in the drive to the controller. This line is also frequently called the Read/Response line.

8.3.2 SDI Bus Encoding

Each of the four SDI lines transmit serial ones and zeros using 12 nanosecond pulses occurring within bit cell times. Refer to Figure 8–4. The duration of each pulse is fixed at 12 nanoseconds. The bit cell time, however, is a function of the drive transfer rate. Drive transfer rates vary, depending on the disk speed and recording density of the drive.

Figure 8–4: SDI Bus Encode



CXO-2349A

The receiver separates the data in the following manner: A positive pulse at the beginning of a bit cell indicates a logical one. A negative pulse at the beginning of a bit cell indicates a logical zero. When a bit in the next cell is the same as the previous one, then a pulse of the opposite polarity is added immediately after the pulse for the previous bit.

In this manner, every pulse on an SDI line alternates polarity. This results in a net DC voltage of zero. Circuitry in the controller and in the drive detect any missing or additional, unwanted pulses that may occur.

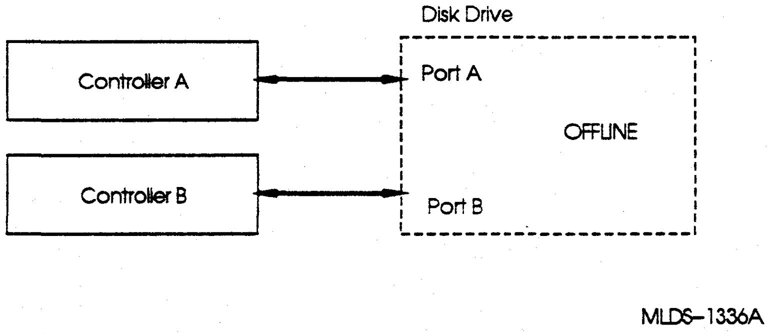
Missing or unwanted pulses represent transmission errors. When detected, they are usually entered in the system error log. These errors are referred to as pulse errors. For example, if a drive detected two sequential pulses of the same polarity while receiving information on the WRT/CMD data line, the error is referred to as a **write/command pulse error**.

8.4 DRIVE STATES

A drive can be in one of four different states relative to a controller.

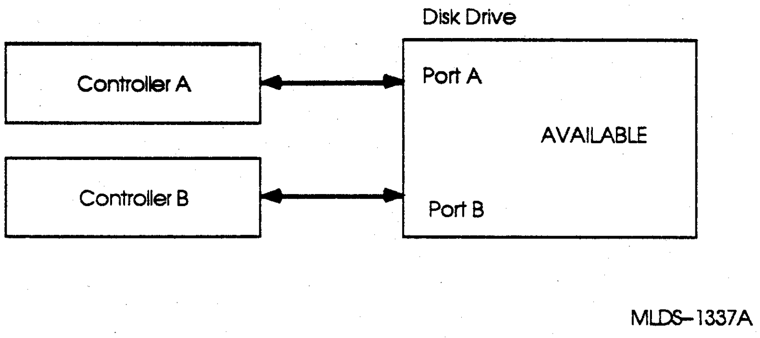
When the drive is not operational, it's in a state called "drive off line." See Figure 8–5. At this time, no communication takes place between the drive and either controller. A drive is off line to a controller when its port switch to that controller is released.

Figure 8–5: Drive Off Line



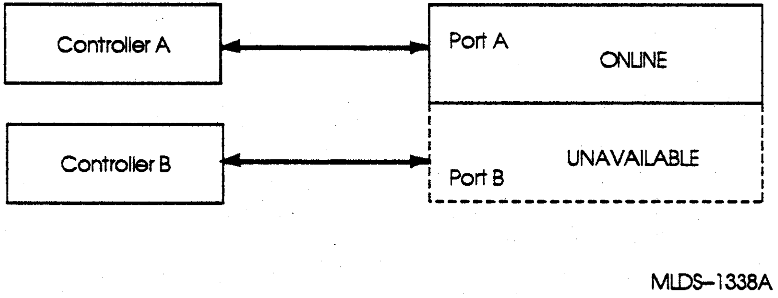
Now look at Figure 8–6. When the drive becomes operational, it enters a drive available state. This means that it is visible to, and capable of communicating with, either controller, providing the port switches are enabled. The term *operational* implies several conditions. The port switch(s) must be pressed to enable the communication paths between a drive and a controller. The drive must also be able to spin up; that is, no major drive problems prevent the drive spindle from spinning or prevent the drive from properly communicating with the controller.

Figure 8–6: Drive Available



When controller A wants to communicate with the drive, it must bring it to a state called "drive on line." This means that the drive, through Port A, becomes dedicated to the exclusive use of controller A. This is illustrated in Figure 8–7. During this time, the drive is visible but not available to controller B. Its state relative to controller B is called "drive unavailable."

Figure 8–7: Drive On Line

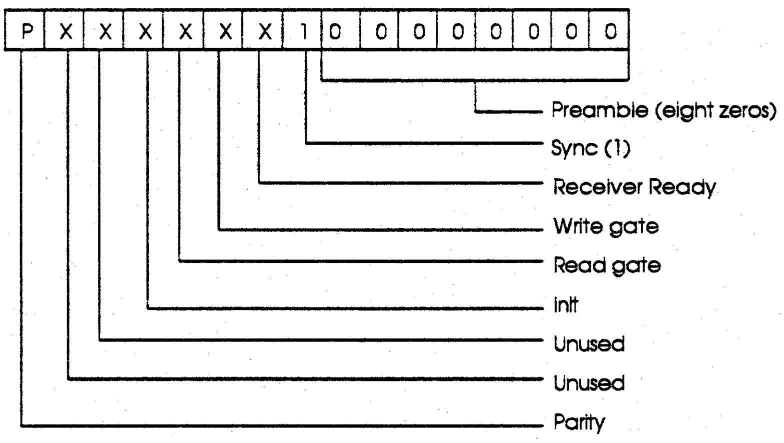


Since the drive can only communicate with one controller at a time, it must assume a "drive unavailable" state with one controller as the other controller brings it to a "drive on–line" state. When the controller which is communicating with the drive completes all of its activities and releases the drive on Port A, the drive will return to a "drive available" state relative to both ports.

8.5 RTCS FORMAT (Real Time Controller State)

The controller uses the Real–Time Controller State line to transmit a 16–bit pattern to the drive. See Figure 8–8. This pattern indicates the state of the controller and includes logical signals which are used to synchronize controller/drive operations. This pattern is repeatedly sent to the drive by the controller.

Figure 8–8: RTCS



MLDS–1334A

The first 8 zeros followed by a 1 bit constitute the sync character. Sensing a minimum of 7 zeros followed by a 1 bit also accomplishes synchronization.

The next four bits are logical signals required to synchronize controller and drive operations. They are used as follows:

RTCS RECEIVER READY

When asserted, this signal indicates that the controller is ready to receive a response from the drive on the Read/Response Data line.

RTCS WRITE GATE

During a write operation, the drive uses this signal to generate an internal signal that turns on the write current. The leading edge of this signal causes the drive to begin writing information from the WRT/CMD DATA line to the recording surfaces. The trailing edge of WRITE GATE indicates to the drive that the current WRITE command is finished.

RTCS READ GATE

During a read data operation, the drive uses this signal to enable a circuit that reads information from the recording surfaces and sends it to the controller on the SDI Read/Response line. The controller asserts READ GATE such that the leading edge of this signal occurs after the header field but before the data field of the sector. It remains asserted until after the ECC character has been read. The trailing edge of READ GATE indicates to the drive that the current data transfer command is finished.

RTCS INIT

This signal initializes the drive. The leading edge of this signal instructs the drive microprocessor to unconditionally go to a known memory location and execute the initialization sequence. This sequence aborts all operations in progress. The drive saves its status at the time of the interrupt and executes sufficient internal diagnostics to verify its processor and communication paths to the controller. Upon completion of the initialization sequence, the drive notifies the controller by asserting an appropriate signal on the RTDS line.

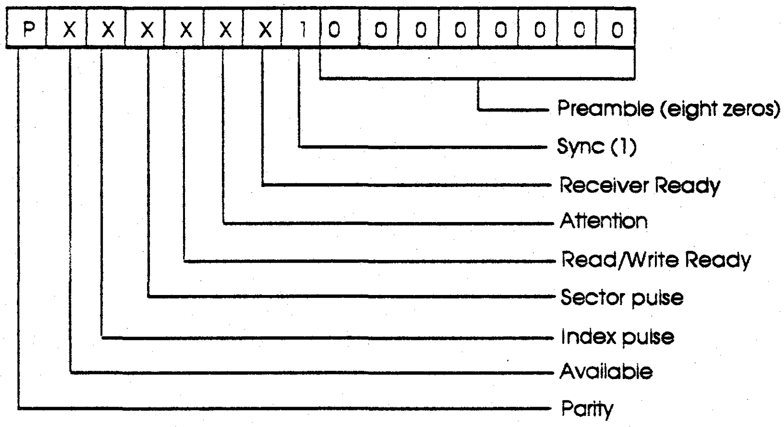
RTCS PARITY

The parity used is even over the entire 16 bits, including the SYNC bit. The parity is appended by the controller and used by the drive to further detect errors encountered when receiving information on this line. If parity errors occur, the information is ignored by the drive and the previous state of the controller is used until a valid update is received.

8.6 RTDS FORMAT (Real Time Drive State)

The drive uses the Real–Time Drive State line to transmit a 16–bit pattern to the controller. Refer to Figure 8–9. This pattern indicates the state of the drive and includes logical signals which are used to synchronize controller/drive operations. This pattern is sent continuously by available drives to all controllers for which drive port switches are pressed (enabled).

Figure 8–9: RTDS



MLDS–1335A

The first 8 zeros followed by a 1 bit constitute the sync character. Sensing a minimum of 7 zeros followed by a 1 bit also accomplishes synchronization.

The remaining bits are used as follows:

RTDS RECEIVER READY

This signal asserted indicates that the drive is ready to receive a command from the controller on the WRT/CMD line.

RTDS ATTENTION

This signal asserted notifies a controller that a potentially significant event has occurred and caused the status and/or state of the drive to change. The ATTENTION signal has no affect on any other activity on the SDI bus.

RTDS READ/WRITE READY

This signal indicates that the drive is capable of performing a data transfer to or from the disk surface. This signal is only asserted by drives in the on–line state when no condition prevents a transfer operation.

RTDS SECTOR PULSE

This signal marks the boundary between sectors. The leading edge of SECTOR PULSE may be used for rotational position sensing. The trailing edge of SECTOR PULSE marks the beginning of a sector.

RTDS INDEX PULSE

This signal is asserted once per revolution of the disk. The controller uses the leading edge of INDEX PULSE for rotational position sensing and the trailing edge to mark the beginning of the first sector after index.

RTDS AVAILABLE

This signal indicates to the controller that the drive is in the available state.

RTDS PARITY

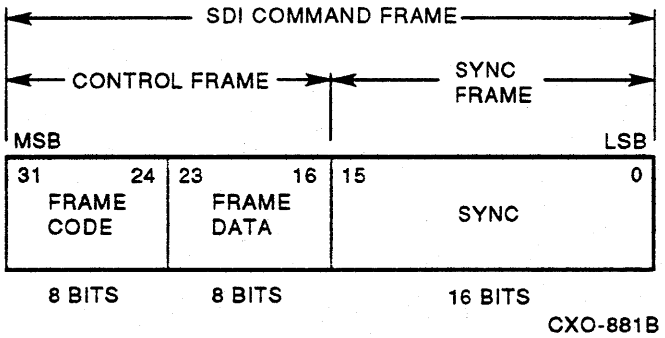
The parity used is even over the entire 16 bits, including the SYNC bit. This parity bit is appended by the drive and used by the controller to further detect errors encountered when receiving information on this line. If a parity error occurs during formatting, the operation is aborted. When a parity error occurs at other times, the information is ignored by the controller and the previous state is used until a valid update is received from the drive. In addition, an error log message is generated and sent to the host.

8.7 COMMAND FORMATS on the WRT/CMD LINE

The controller uses the WRT/CMD line to send write data to the drive. In addition, it uses this line to send commands and command messages or parameters to the drive.

Refer to Figure 8–10. Commands and messages are transmitted using a 32–bit SDI command frame. This command frame consists of a 16–bit sync frame followed by a 16–bit control frame.

Figure 8–10: SDI Command Frame



The sync frame portion is always sent first and is a special pattern used to synchronize the drive for receiving a command frame. The control frame consists of an 8–bit frame code and 8 bits of frame data.

There are two levels of commands transmitted on the Write Command/Data line.

8.8 LEVEL 1 COMMANDS

Refer to Figure 8–11. A level 1 command consists of one 32–bit SDI command frame. It begins with the sync frame which contains the sync character. The frame code field contains the opcode of the command. The frame data field contains the message data necessary to complete the level 1 command. There are six level 1 commands. Refer to Figure 8–12.

Figure 8–11: Level 1 Command Format

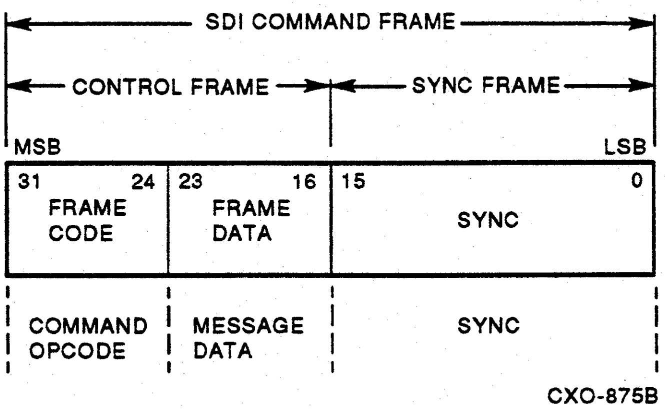
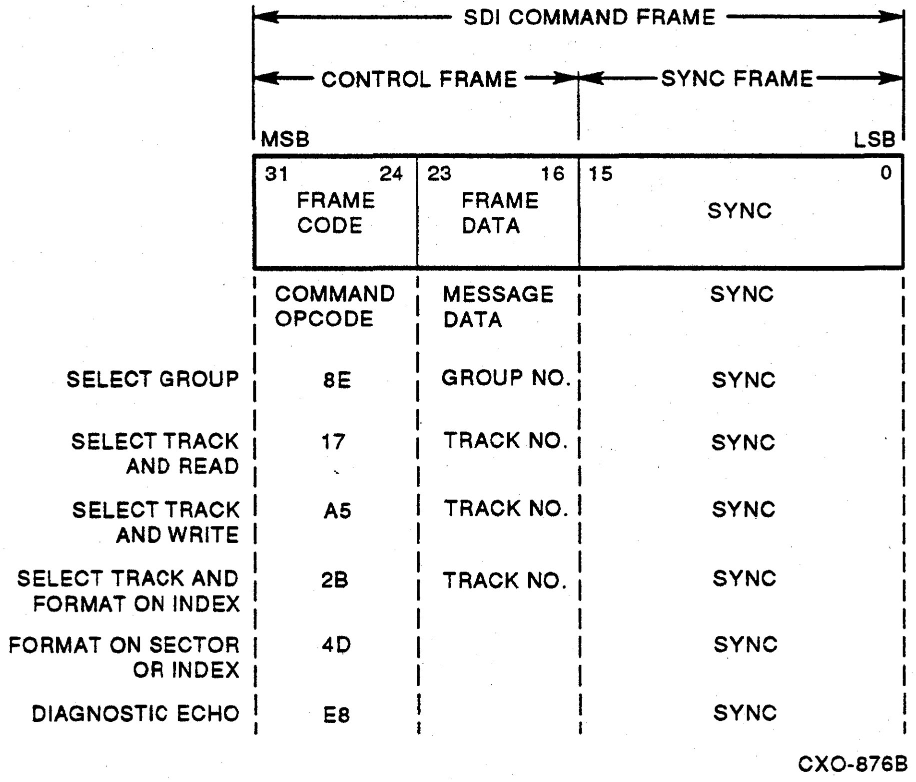


Figure 8–12: Level 1 Commands



	COMMAND OPCODE	MESSAGE DATA	SYNC
SELECT GROUP	8E	GROUP NO.	SYNC
SELECT TRACK AND READ	17	TRACK NO.	SYNC
SELECT TRACK AND WRITE	A5	TRACK NO.	SYNC
SELECT TRACK AND FORMAT ON INDEX	2B	TRACK NO.	SYNC
FORMAT ON SECTOR OR INDEX	4D		SYNC
DIAGNOSTIC ECHO	E8		SYNC

SELECT GROUP

This command has an opcode 8E in the frame code field. The group number to be selected is contained in the frame data field. This command causes the drive to clear R/W READY on the RTDS line, select the specified group, and set R/W READY when it is ready to perform another command or I/O operation.

SELECT TRACK AND READ

This command has an opcode of 17 in the frame code field and the track number in the frame data field. This command causes the drive to select the desired track and prepare for a read data operation. Read data operations are discussed in more detail later in the course.

SELECT TRACK AND WRITE

This command has an opcode of A5 in the frame code field and the track number in the frame data field. This command causes the drive to select the desired track and prepare for a write data operation. Write data operations are discussed in more detail later in the course.

SELECT TRACK AND FORMAT ON INDEX

This command causes the drive to select the desired track and prepare the necessary circuits to format the entire track.

FORMAT ON SECTOR OR INDEX

This command causes the drive to use the last selected track and prepare the necessary circuits to format one sector. Notice that this command does not require any further information in the frame data field.

DIAGNOSTIC ECHO

This command causes the drive to transmit diagnostic information back to the controller for testing purposes.

8.9 LEVEL 2 COMMANDS

8.9.1 Command Formats on the WRT/CMD Line

The basic characteristic of a level 1 command is that it only requires a single 32–bit SDI command frame to complete the entire command from the controller to the drive. Many commands, however, require much more information than could fit into a single 32–bit frame.

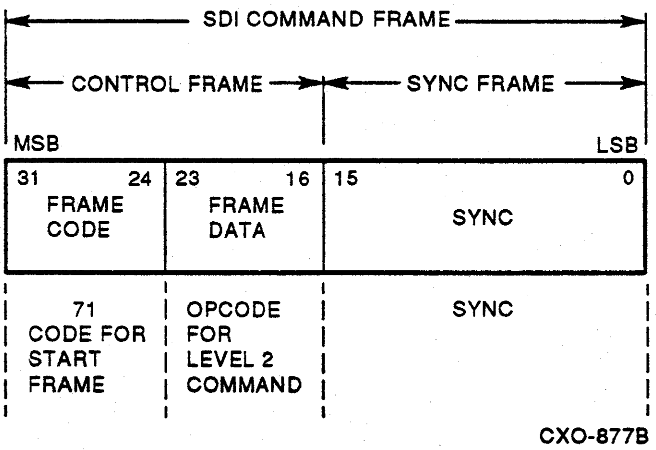
Level 2 commands, also transmitted on the WRT/CMD line, contain more than one 32–bit command frame. The actual number of frames sent for a level 2 command depends on the particular command. There are three types of level 2 command frames. They are:

START	Command Frame
CONTINUE	Command Frame
END	Command Frame

Like the level 1 command frame, each one begins with a sync frame.

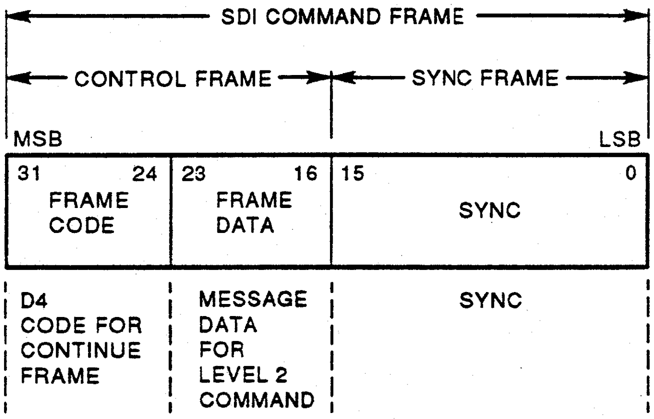
Refer to Figure 8–13. The frame code field of a START command frame contains a code of 71. This code indicates to the drive the beginning of a level 2 command. The frame data field contains the opcode for the level 2 command. This indicates to the drive the type of level 2 command to be performed.

Figure 8–13: Level 2 START Command Frame



Refer to Figure 8–14. The frame code field of a CONTINUE command frame contains a code of D4. This code indicates to the drive the continuation of a level 2 command. The frame data field contains further message data necessary to complete the particular level 2 command. Most, but not all, level 2 commands require at least one CONTINUE frame. This depends on the amount of information needed to complete the command.

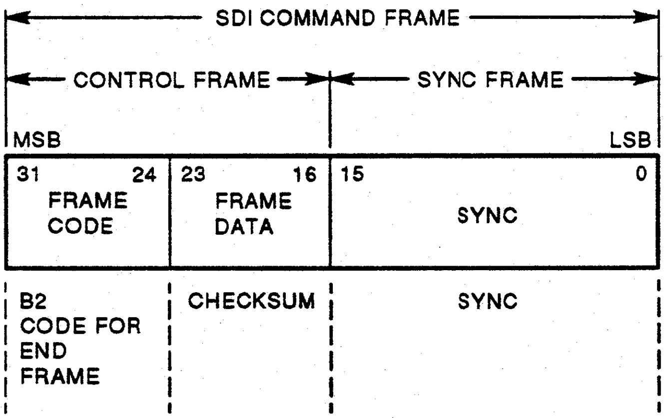
Figure 8–14: Level 2 CONTINUE Command Frame



CXO-878B

Refer to Figure 8–15. The frame code field of an END command frame contains a code of B2. This code indicates to the drive the end of a level 2 command. The frame data field contains a checksum character for the level 2 command. The checksum is used for error detection and is computed against all of the information transmitted in the 8–bit frame data fields for all of the 32–bit command frames sent during a level 2 command.

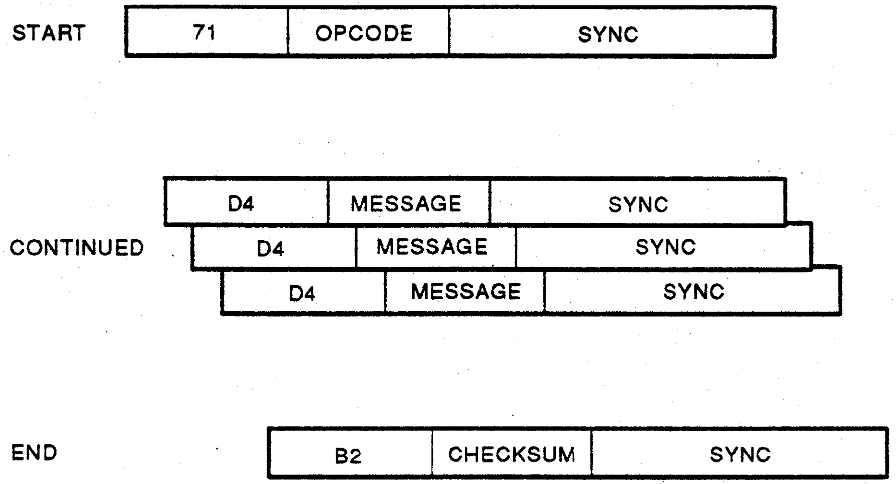
Figure 8–15: Level 2 END Command Frame



CXO-879B

Refer to Figure 8–16. In summary, all level 2 commands require a START frame, an END frame, and usually one or more CONTINUE frames. Some level 2 commands do not require a CONTINUE frame.

Figure 8–16: Level 2 Commands



CXO-880B

There are 16 level 2 commands. The number in parentheses indicates the number of CONTINUE frames needed to complete the command.

(2)	CHANGE MODE
(2)	CHANGE CONTROLLER FLAGS
(2)	DIAGNOSE
(1)	DISCONNECT
(1)	DRIVE CLEAR
(1)	ERROR RECOVERY
(0)	GET COMMON CHARACTERISTICS
(1)	GET SUBUNIT CHARACTERISTICS
(0)	GET STATUS
(5)	INITIATE SEEK
(1)	ON–LINE
(0)	RUN
(5)	READ MEMORY
(0)	RECALIBRATE
(1)	TOPOLOGY
(x)	WRITE MEMORY (the number of CONTINUE frames required will vary)

The following pages briefly describe the level 2 commands and the basic functions that will be performed by the drive when these commands are executed.

Level 2 Command	Command Opcode	Drive Function Performed
CHANGE MODE	81	Instructs the drive to alter its mode to the specified settings.
CHANGE CONTROLLER FLAGS	82	Directs the drive to change the specified bit(s) in its status "Controller Byte" to the specified settings.
DIAGNOSE	03	Directs the drive to execute the program which is resident in the specified drive memory region.
DISCONNECT	84	Directs an on-line drive (provided the Terminate Topology bit is clear) to enter the available state relative to all active ports.
DRIVE CLEAR	05	Directs the drive to clear the specified status bits in the error byte of the drive's generic status and attempt to clear the associated error condition.
ERROR RECOVERY	06	Directs the drive to invoke the specific hardware error recovery mechanism corresponding to the specified error recovery level.
GET COMMON CHARACTERISTICS	87	Requests the drive to send the controller a description of its common drive hardware characteristics.
GET SUBUNIT CHARACTERISTICS	88	Requests the drive to send the controller a description of characteristics, geometry, and topology of the specified subunit. For current DSA disk drives, there is only one subunit and it is equivalent to the HDA (or pack) that is installed.
GET STATUS	09	Request the drive to send all of its current status bytes to the controller.
INITIATE SEEK	0A	Directs the drive to initiate a seek to the specified group on the specified cylinder.
ONLINE	8B	Directs the drive to enter the on-line state relative to the controller that issued the ONLINE command.
RUN	0C	Directs the drive to spin up if the RUN/STOP switch is pressed.
READ MEMORY	8D	Directs the drive to fetch and send to the controller the specified number of bytes starting at the specified offset into the specified memory region of the drive. This command has no association with SELECT TRACK and READ commands or operations.
RECALIBRATE	8E	Directs the drive to perform a recalibration operation and to then seek to Cylinder 0.
TOPOLOGY	90	Instructs the drive to make itself available for limited dialogue to any and all controller(s) on enabled alternate ports.
WRITE MEMORY	0F	Directs the drive to load the supplied data into the indicated area of its memory. This command has no association with any SELECT TRACK and WRITE or FORMAT commands or operations.

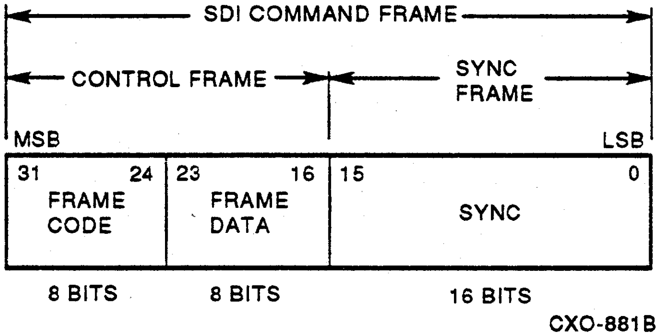
Some level 2 commands do not require CONTINUE frames. However, they are not sent as level 1 single frame commands because level 1 commands do not require any response from the drive. Level 2 commands do require a response from the drive.

8.9.2 Response Formats on the Read/Response Line

Each level 2 controller command requires a response from the drive. Responses are sent to the controller over the Read/Response line. They use the same protocol as command messages.

Refer to Figure 8–17. The sync frame is transmitted first, followed by the control frame. The control frame contains an 8–bit frame code and 8 bits of frame data.

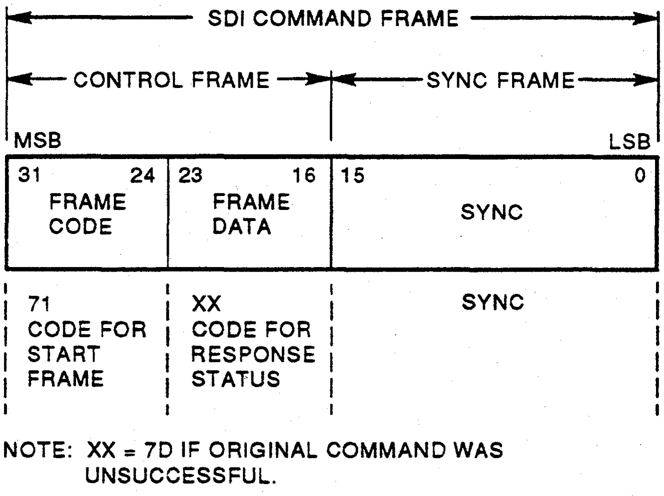
Figure 8–17: SDI Response Frame Format



The response protocol uses the same START, CONTINUE, and END frames as the command protocol. The frame codes for the START, CONTINUE, and END frames are also identical to those used in the command frame format.

See Figure 8–18. In a level 2 start response frame, the frame data field contains a response status code that indicates whether the original level 2 command was executed successfully or unsuccessfully. If the original level 2 command was successful, this field contains any of several success codes, depending on the particular level 2 command issued by the controller. The response also contains an end frame and usually some continue frames containing message data specific to the particular level 2 command that the controller previously issued.

Figure 8–18: Level 2 Response Start Frame



CXO-882B

If the original level 2 command is unsuccessful, the frame data field always contains the code 7D, indicating to the controller that a problem prevented the proper execution of the original level 2 command. When the drive provides an unsuccessful response, it automatically provides 14 additional CONTINUE frames containing drive status bytes and an END frame to complete the response. The controller uses the drive status bytes to determine the nature of the problem.

In addition, most of the drive status bytes are usually logged into the system error log. This information is quite useful when you are isolating drive problems or disk subsystem problems in the field. Tables of DSA drive status byte information are located in the *DSA Error Log Reference Manual* (EK–DSAEL–MN).

Like the level 2 command format, the END frame of a drive response contains the checksum character in the frame data field.

8.10 INITIATE SEEK COMMAND (Level 2 Command Example)

An INITIATE SEEK command directs a drive to initiate a positioning operation to a specified group within a specified cylinder. The drive response to this command indicates if the operation was successfully started or not.

Refer to Figure 8–19 to see how the INITIATE SEEK command starts. The controller transmits a start frame. The drive first sees the sync frame and then the control frame information. The frame code field contains a code 71, indicating that this is a start frame. The frame data field contains the opcode 0A which identifies it as an INITIATE SEEK command.

Figure 8–19: INITIATE SEEK Command

START	71	OA	SYNC
CONTINUE	D4	CYL ADDRESS	SYNC
CONTINUE	D4	CYL ADDRESS	SYNC
CONTINUE	D4	CYL ADDRESS	SYNC
CONTINUE	D4	GROUP NO.	SYNC
END	B2	CHECKSUM	SYNC

CXO-883B

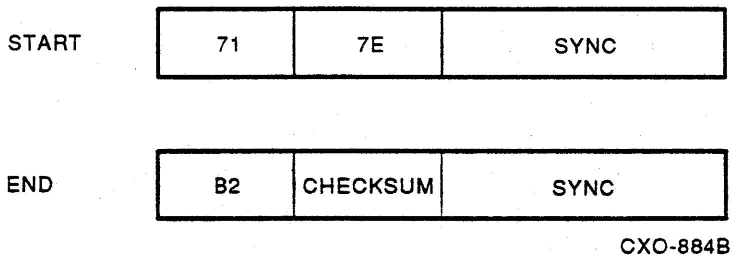
The next frames sent are CONTINUE frames, indicated by the D4 in the frame code field. The message data bytes of the CONTINUE frames contain the cylinder address and the group number for the positioning operation.

The last command frame sent is the END frame. This is indicated by the B2 in the frame code field. The message data field contains the checksum character which is the one's compliment of the sum of the six bytes transmitted in the message data field during the START frame and all CONTINUE frames.

This completes the transmission by the controller. It's now up to the drive to respond.

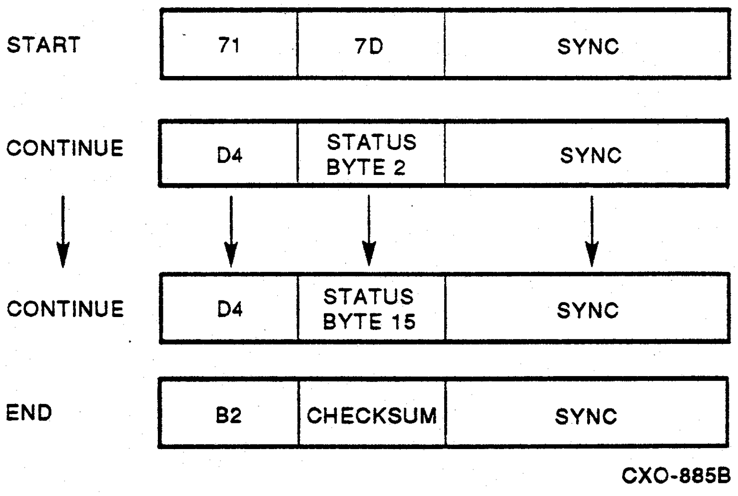
Figure 8–20 shows how a success response appears. The response contains a code 7E in the message data field of the start frame. This indicates success for an INITIATE SEEK command. The END frame contains the checksum. The success response for a level 2 INITIATE SEEK command informs the controller that the seek operation has started without error and is currently under way.

Figure 8–20: Successful Response for SEEK Command



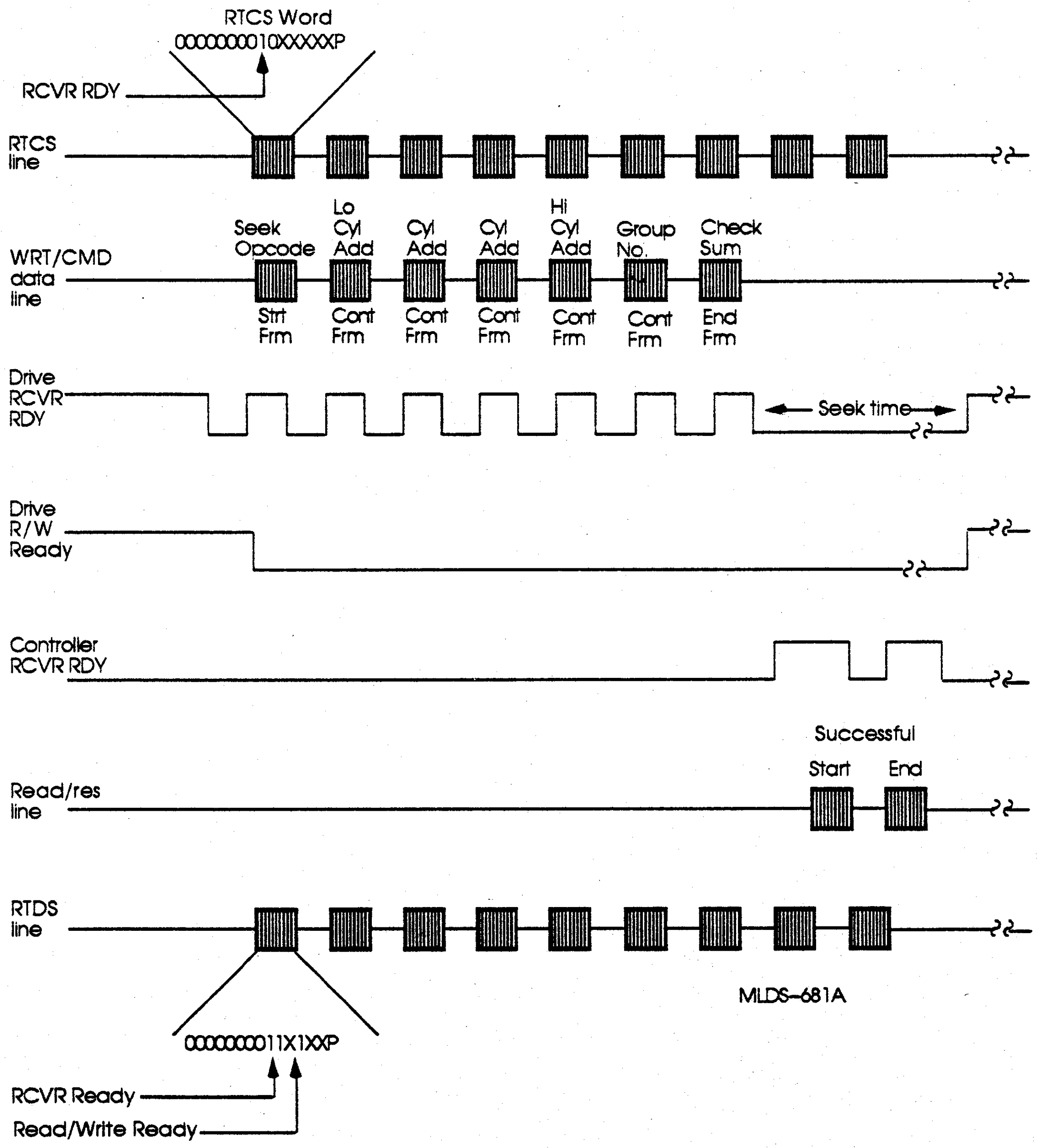
An unsuccessful response tells the controller that the operation could not be initiated. Refer to Figure 8–21. It consists of the unsuccessful code (7D) in the START frame, 14 bytes of drive status in the CONTINUE frames, and the checksum in the END frame. The error information in the drive status bytes allows the controller to develop error log information which it sends to the host.

Figure 8–21: Unsuccessful Response for SEEK Command



Now let's look at the timing associated with an INITIATE SEEK command that is sent over the Write/Command line. Refer to Figure 8–22.

Figure 8–22: Initiate Seek Simplified



Before the controller sends the SEEK command, the drive has both R/W READY and RECEIVER READY asserted on the RTDS line. This informs the controller that it is ready to receive a command and that no data transfer is currently under way. When the drive receives the seek opcode in the start frame on the Write/Command line, both RECEIVER READY and R/W READY are cleared.

R/W READY remains cleared until the seek is complete and indicates that data transfers must not occur to or from the recording surfaces on the WRT/CMD or Read/Response.

The drive then sets RECEIVER READY to receive the next frame. This continues until the drive receives the END frame. Then RECEIVER READY remains off until the seek is complete.

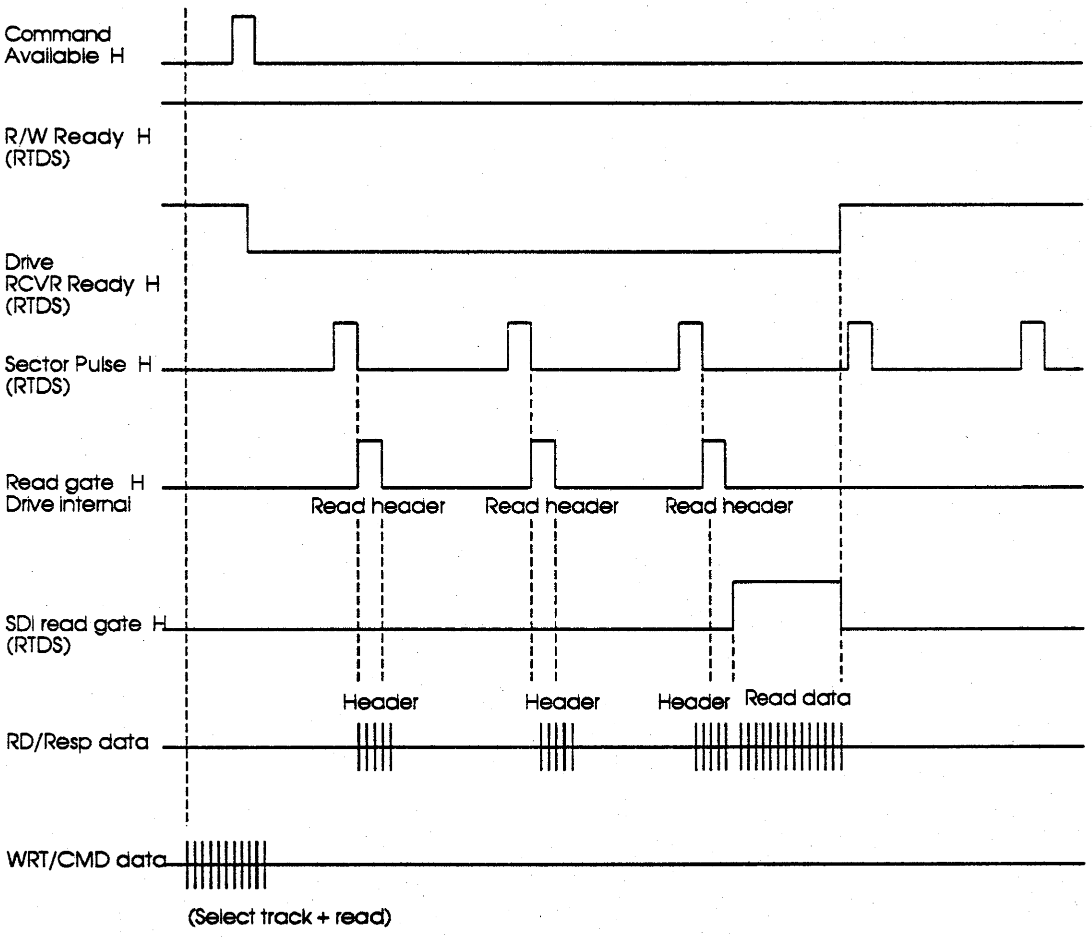
On the RTCS line, RECEIVER READY is asserted by the controller after it transmits the END frame. At this time, the drive is permitted to send the response on the Read/Response line. The controller sets RECEIVER READY to receive each frame of the response. This particular response indicates that the seek operation has successfully started.

When the seek is complete, the drive asserts R/W READY and RECEIVER READY on the RTDS line. This indicates to the controller that the entire seek operation has, in fact, been completed.

8.11 SDI READ OPERATION

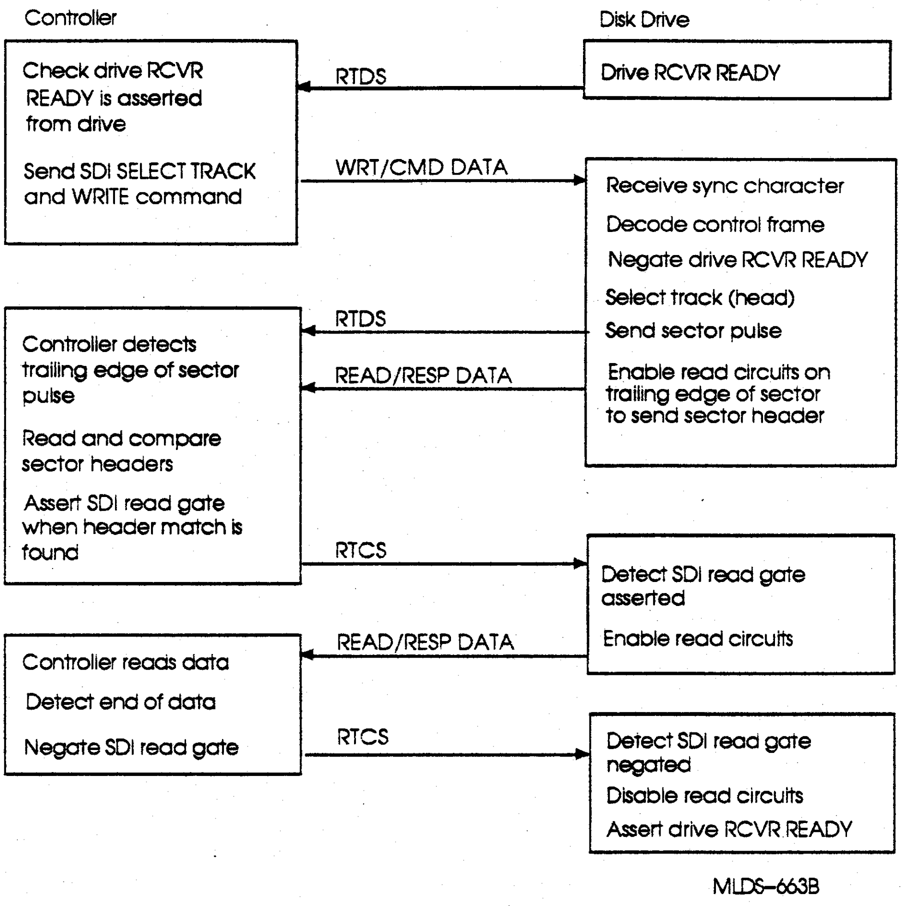
A read operation is initiated from the controller using a level 1 SELECT TRACK AND READ command. Figure 8–23 illustrates the timing between the controller and the drive on the SDI bus during this command. Figure 8–24 illustrates the basic flow of events that take place between the drive and the controller for this operation.

Figure 8–23: SDI Select Track and Read Timing



MLDS–662A

Figure 8–24: Select Track and Read Flow



Before the controller initiates a READ command, the drive has both R/W READY and RECEIVER READY asserted on the RTDS line. This informs the controller that the drive is ready to receive a command. When the drive receives the SELECT TRACK AND READ command, the drive internally generates a COMMAND AVAILABLE signal and decodes the command.

Since this is a read data operation, drive hardware generates an INTERNAL READ GATE that enables header information to be read from the recording surface and sent to the controller over the Read/Response Data line. Header information is sent for every sector on the currently selected track.

The controller is responsible for using the header information to determine when the desired sector is under the R/W head(s). When the controller determines that the correct sector is under the data head, it asserts the SDI READ GATE signal on the RTCS line after the desired header and before the data area of the desired sector. The leading edge of the SDI READ GATE causes the drive to read the data area of the sector and send the information to the controller, also over the SDI Read/Response Data line.

Different controllers use different techniques for determining the desired sector. In the UDA50, for example, once the SELECT TRACK AND READ command is sent, the controller reads each header until the target sector is found. The HSC50, however, uses the SECTOR PULSE signal from the RTDS line and keeps track of the sector count. The HSC50 then issues the SELECT TRACK AND READ command just prior to the target sector. In this way the target sector header is the first header read after issuing the SELECT TRACK AND READ command.

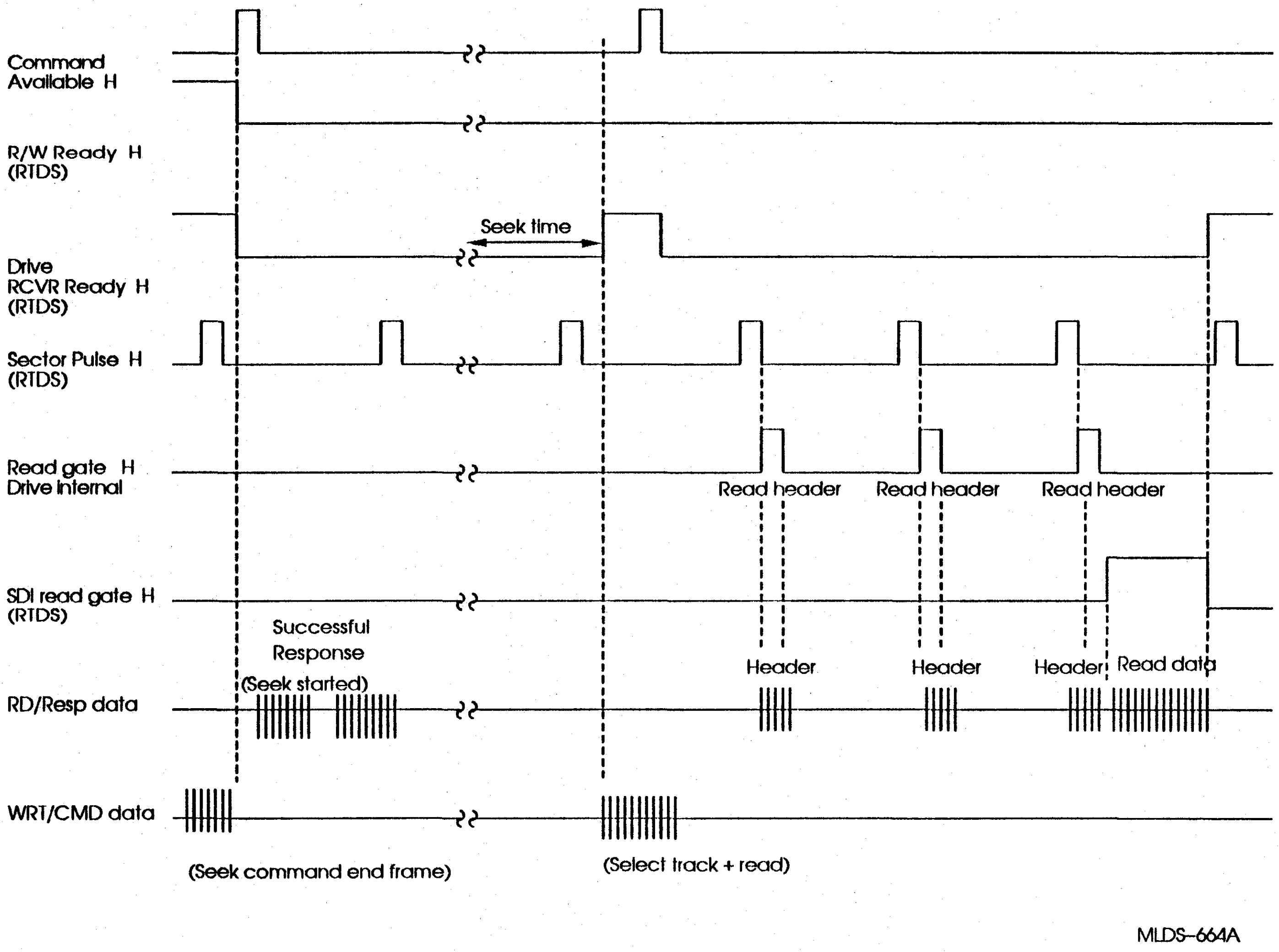
When the desired data has been read, the controller lowers (de–asserts) the SDI READ GATE SIGNAL on the RTCS line. The trailing edge of the SDI READ GATE notifies the drive to terminate the entire read operation. The drive then re–asserts the RECEIVER READY signal on the RTDS line to the controller in preparation for receiving another command.

Notice that the R/W READY signal remained asserted on the RTDS line throughout this operation. This is an indication to the controller that the Read/Response line was available for transferring actual data from the recording surface to the controller during this operation. As you can see, the controller is responsible for controlling most of the entire read data operation.

8.12 SEEK followed by a SELECT TRACK AND READ

Figure 8–25 shows the timing on the SDI for a seek operation followed by a SELECT TRACK AND READ command.

Figure 8–25: SEEK Command Followed by SELECT TRACK AND READ



8.13 SDI WRITE OPERATION

A write operation is initiated from the controller using a level 1 SELECT TRACK AND WRITE command. This operation is very similar to a READ operation. Figure 8–26 illustrates the timing between the controller and the drive on the SDI bus during this command. Figure 8–27 illustrates the basic flow of events that take place between the drive and the controller for this operation.

Figure 8–26: SDI Select Track and Write Timing

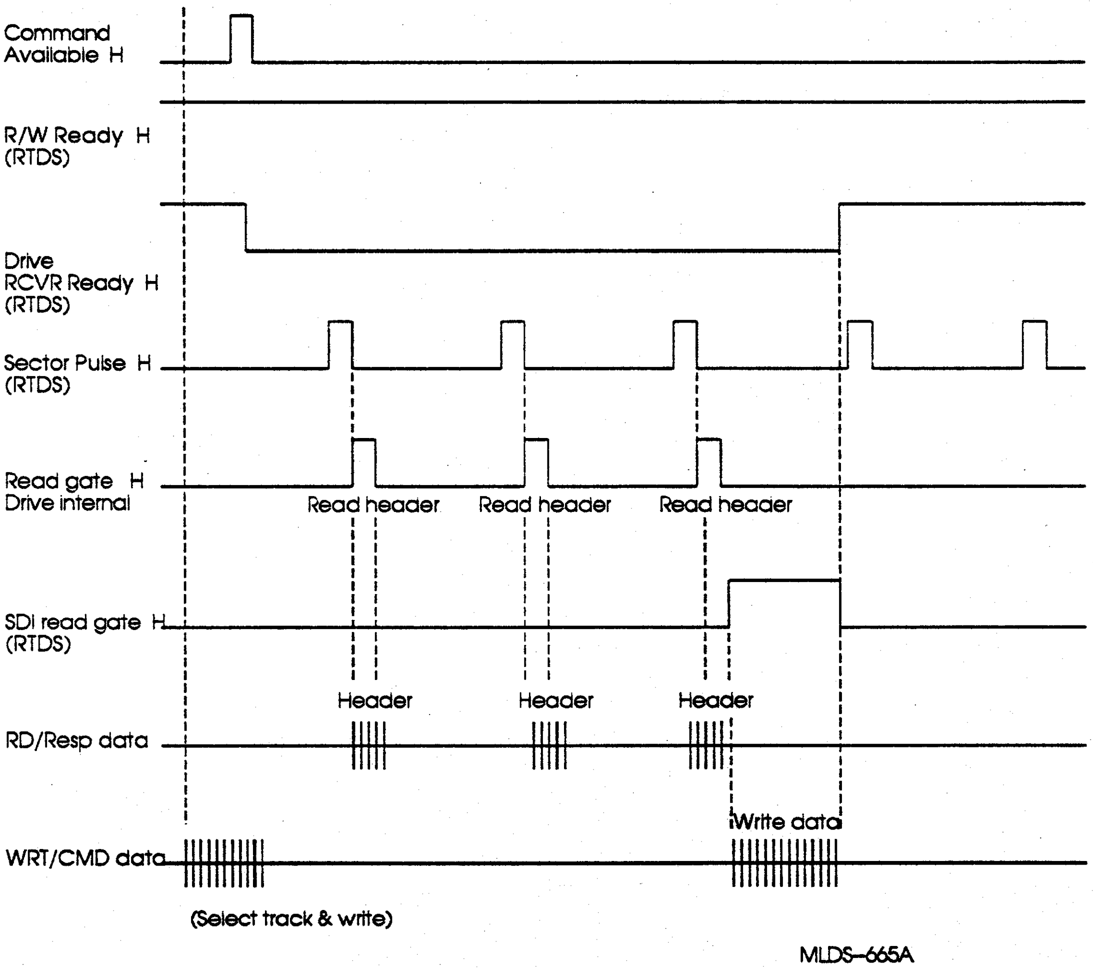
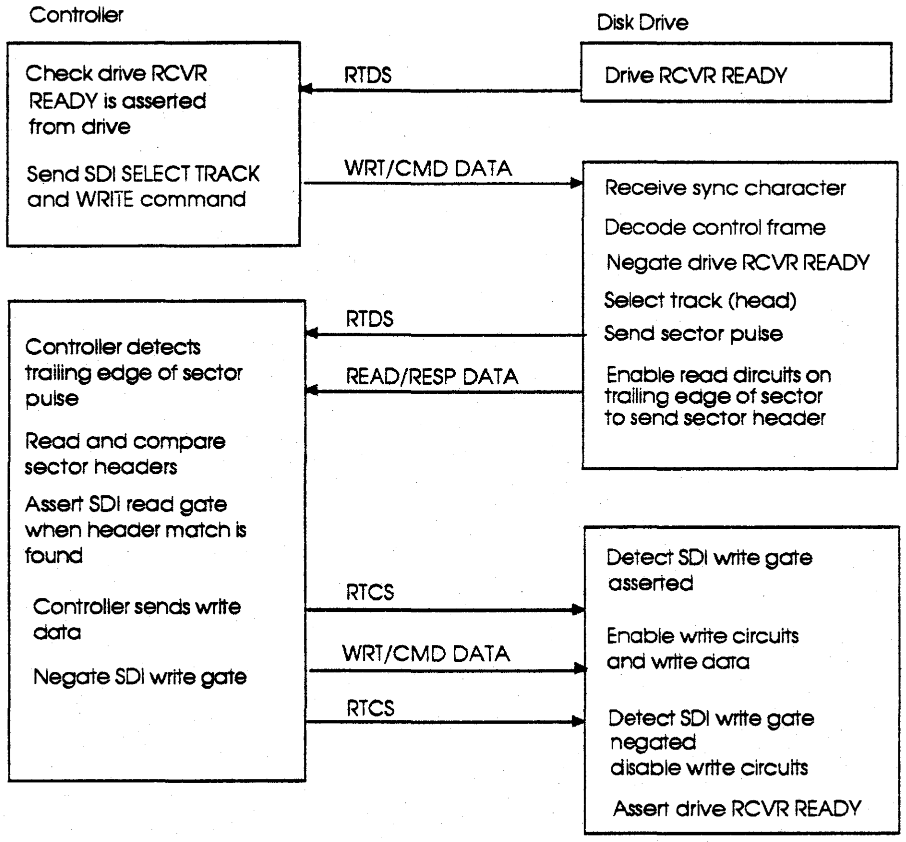


Figure 8–27: Select Track and Write Flow



MLDS–666B

Before the controller initiates a READ command, the drive again has both R/W READY and RECEIVER READY asserted on the RTDS line. This informs the controller that the drive is ready to receive a command. When the drive receives the SELECT TRACK AND WRITE command, the drive internally generates a COMMAND AVAILABLE signal and decodes the command.

Since this is a write data operation, drive hardware again generates an INTERNAL READ GATE that enables header information to be read from the recording surface and sent to the controller over the Read/Response Data line. Header information is sent for every sector on the currently selected track.

The controller is responsible for using the header information to determine when the desired sector is under the R/W head(s). When the controller determines that the correct sector is under the data head, it asserts the SDI WRITE GATE signal on the RTCS line after the desired header and before the data area of the desired sector and then begins sending data to the drive on the WRT/CMD Data line.

Different controllers use different techniques for determining which sector is the desired sector. In the UDA50, for example, once the SELECT TRACK AND WRITE command is sent, the controller reads each header until the target sector is found. The HSC50, however, uses the SECTOR PULSE signal from the RTDS line and keeps track of the sector count. The HSC50 then issues the SELECT TRACK AND WRITE command just prior to the target sector. In this way, the target sector header is the first header read after issuing the SELECT TRACK AND WRITE command.

The leading edge of the SDI WRITE GATE causes the drive to do two things. First, it de–asserts its internal read gate and discontinues sending header information to the controller. Then the drive turns on its write current and writes the data onto the data recording area of the sector. The data that it writes will be the data it receives on the WRT/CMD Data line from the controller.

When the desired information has been written, the controller lowers (de–asserts) the SDI WRITE GATE signal on the RTCS line. The trailing edge of the SDI WRITE GATE signal notifies the drive to terminate the entire write operation. The drive then re–asserts the RECEIVER READY signal on the RTDS line to the controller in preparation for receiving another command.

Like the read operation, the R/W READY signal remains asserted on the RTDS line throughout this operation. As you can see, the controller is also responsible for controlling most of the entire write data operation.

8.14 SEEK followed by SELECT TRACK AND WRITE

Now that you have seen the SDI timing for a seek and a write operation, let's put the two together. Figure 8–27 shows the timing on the SDI for a seek operation followed by a SELECT TRACK AND WRITE command.

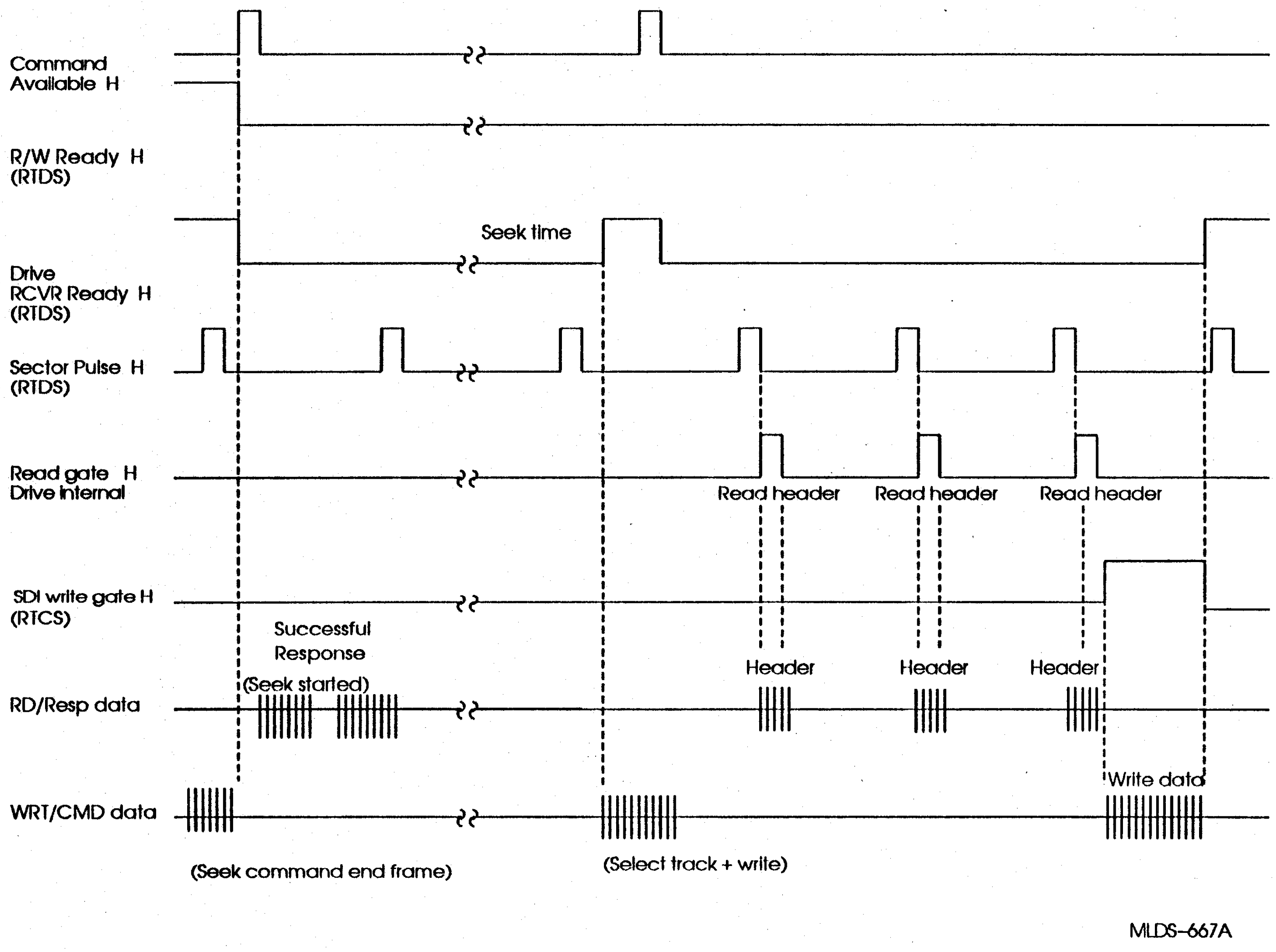


Figure 8–28: SEEK Command Followed by SELECT TRACK AND WRITE

At this time complete the following exercises.

8.15 EXERCISES

1. What are the four lines that comprise the SDI bus?

   A. RTDS, RTCS, READ GATE, WRITE GATE

   [✓ **B.**] RTCS, WRT/CMD, READ/RESPONSE, RTDS

   C. READ/RESPONSE, WRT/CMD, READ GATE, WRITE GATE

   D. R/W READY, READ/RESPONSE, RECEIVER READY, AVAILABLE

2. What are the states of a drive relative to a controller?

   A. Available, Operational, Off–Line, On–Line

   B. On–Line, Off–Line, Available, R/W Ready

   C. On–Line, Off–Line, Unavailable, R/W Ready

   [✓ **D.**] Available, Unavailable, On–line, Off–line

3. How does the controller know when a drive has completed a SEEK operation?

   [✓ **A.**] When the drive asserts both R/W Ready and Receiver Ready after receiving a SEEK command.

   B. When the controller receives a response to the SEEK command.

   C. When the drive asserts Read Gate after receiving a SEEK command.

   D. Both B and C.

4. What constitutes a level 2 command?

   A. At least one START frame and two or more CONTINUE frames.

   B. At least two START frames and one END frame.

-18

   [✓ **C.**] At least one START frame and one END frame.

   [✓ **D.**] At least one START frame, one CONTINUE frame, and one END frame.

5. During a SELECT TRACK AND READ command, when is the header information sent to the controller?

- **A.** When the controller asserts SDI READ GATE.
- **B.** When the drive asserts SDI READ GATE.
- **C.** When the controller asserts RECEIVER READY.
- [✓ **D.**] When the drive asserts its internal READ GATE.

6. At what time does the controller raise SDI READ GATE?

- **A.** As soon as the SDI drive RECEIVER READY negates.
- **B.** On the trailing edge of each sector pulse.
- [✓ **C.**] After a header match for the target block.
- **D.** On command from the drive.

7. When is write data sent to the drive from the controller?

- [✓ **A.**] After a header match on the target block.
- **B.** With the SELECT TRACK AND WRITE command.
- **C.** On the trailing edge of each sector pulse.
- **D.** Immediately before the SELECT TRACK AND WRITE command.

CHAPTER 9

LEVEL 2 SDI COMMANDS

9.1 INTRODUCTION

This section describes each of the level 2 SDI commands and relates to a variety of DSA disks. In some instances, specific disk drives interpret bits differently for unique reasons. Refer to the technical description manuals for specific disk drives for clarification of these matters.

9.2 CHANGE MODE Command

The CHANGE MODE command directs the drive to alter its mode to the specified settings illustrated in Figure 9–1. Only those bits in Byte 2 that have corresponding bits set in Byte 3 are altered. The remaining bits are unchanged. The following modes may be changed:

Figure 9–1: CHANGE MODE

	MSB							LSB	
BYTE 01	1	0	0	0	0	0	0	1	OPCODE = 81
BYTE 02	W4	W3	W2	W1	DD	FO	DB	S7	
BYTE 03	BIT MASK								

CXO-1324A

- **Write Protect (W1–Bit)**—The controller can request the drive write protect itself. If the drive is hardware write protected via the operator control panel and the controller attempts to write enable the drive, a specific drive–detected error occurs indicating "SDI write enable command while drive is hardware write–protected."

NOTE

Bits W2 and W3 are not used by current drives.

- **Write Protect (W4/ED–Bit)**—The W4 bit is not used by the RA60, RA80, RA81, or RA82. In the RA70 and the RA90, this bit has been redefined as ED (error log disable). The controller sets this bit during certain diagnostics to disable the special internal error logging features of the drive. This prevents the internal error log from over–writing its buffers with useless information while the controller is performing forced–fault and verify diagnostics to these drives.
- **Drive Disable (DD–Bit)**—The controller sets this bit under the following conditions:
  - If special diagnostics in the controller require the drive to be disabled.
  - If the DD bit gets set, the drive drops off line to host computers and spins down. The drive no longer responds to front panel activity until the DD bit is cleared or the drive is re–powered. The drive does not normally set the DD bit itself.
- **Format Operations (FO–Bit)**—The controller sets this bit prior to issuing any level 1 format commands to the drive. The drive cannot accept any level 1 format commands if this bit is not set. If the drive receives a level 1 format command and this bit is not set, the drive reports an error indicating "FORMAT attempted while format disabled."
- **DBN Access (DB–Bit)**—This bit must be set before the controller can access data in the DBN space (controller diagnostic blocks). If the drive receives a request to seek to the DBN space and this bit is not set, the drive reports an error indicating "Seek command contained an invalid cylinder address."
- **512/576 Byte Mode Select (S7–Bit)**—The controller sets this bit when the drive is to operate in the 576–byte mode. When cleared, the drive operates in 512–byte mode. Some drives do not support the 576–byte mode of operation and report an error.

9.3 CHANGE CONTROLLER FLAGS Command

The CHANGE CONTROLLER FLAGS command (Figure 9–2) instructs the drive to change the appropriate bits in Byte 7 (controller byte) of the status byte. Only those bits in Byte 2 that have corresponding bits set in Byte 3 are altered. The remaining bits are unchanged. Byte 7 of the status byte is part of the GET STATUS response the drive sends back to the controller after a GET STATUS command. Under normal circumstances, the bits in the controller byte are used only by controllers except as noted here.

Figure 9–2: CHANGE CONTROLLER FLAGS

	MSB							LSB	
BYTE 01	1	0	0	0	0	0	1	0	OPCODE = 82
BYTE 02	S4	S3	S2	S1	C1	C2	C3	C4	
BYTE 03	BIT MASK								

CXO-1347A

If any C bits are set (C1, C2, C3, or C4), most drives spin down and ignore any attempt to spin up using the front panel. Drives clear all the controller bits if any of the following conditions exist:

- The drive powers up.
- There is no unit select plug inserted in the front panel.
- The drive has a fault condition.
- Both port switches are disabled.

Table 9–1 lists the C–bit values and their interpretation.

Table 9–1: Byte 2 C–Bits

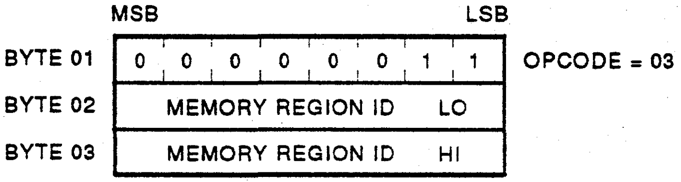
C1	C2	C3	C4	Indication
0	0	0	0	Normal drive operation.
1	0	0	0	Drive offline to hosts due to being under control of diagnostic.
1	0	0	1	Drive offline to hosts due to a duplicate drive unit number detected.

Most drives ignore the suppress attention bits (S1, S2, S3, and S4) under normal operating conditions. These bits can be set only by the controller. The drives may, however, clear the suppress attention bits in the same manner as they clear the C bits.

9.4 DIAGNOSE Command

The DIAGNOSE command instructs the drive to execute the diagnostic program resident in the specified drive memory. Figure 9–3 shows the format for the DIAGNOSE command. For most drives, the memory region specified in a DIAGNOSE command corresponds to the internal diagnostic test number. Refer to the specific drive service manuals for lists of the internal diagnostic test numbers.

Figure 9–3: DIAGNOSE Command



CXO-1348A

When the disk drive has executed the requested test, it sends a response to the controller specifying the region containing the results of the test. The controller then issues a READ MEMORY command to determine if the diagnostic passed or failed and what drive error code(s) were generated during the internal diagnostic.

When a controller issues a DIAGNOSE command, the drive initially provides one of the following responses:

- UNSUCCESSFUL—The DIAGNOSE command was incorrectly formatted, unintelligible, or specified incorrect information.
- SUCCESSFUL—The DIAGNOSE command was accepted, and the test was executed. The response also provides the drive memory region address containing the result of the diagnostic test.

If a DIAGNOSE command is received with an invalid test number, the drive reports an error indicating a specific error code or sets the PE (protocol error) bit in Byte 6 (error byte) of drive status. An invalid test number is one that exists within the drive but cannot be executed using the DIAGNOSE command. Invalid tests include:

- Front panel tests.
- SDI loopback tests.
- Format read–only cylinder test/utility.
- Tests requiring more than 128 seconds to complete.

If the drive receives a DIAGNOSE command with a nonexistent test number, the drive reports an error indicating a specific error code or sets the PE (protocol error) bit in Byte 6 (error byte) of drive status.

9.5 DISCONNECT Command

The DISCONNECT command (Figure 9–4) is used in a number of ways depending upon the assertion of the TT (terminate topology) and ST (stop/spin down) bits in Byte 2. Table 9–2 lists and describes each of the conditions.

Figure 9–4: DISCONNECT Command

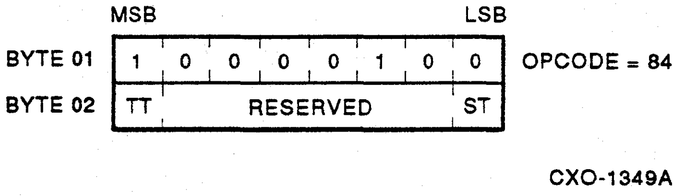


Table 9–2: DIAGNOSE Command TT/ST Bits

TT Bit	ST Bit	Description
0	0	Instructs an on–line drive to disconnect itself from the current controller and become available. The drive does not spin down after completing this command.
0	1	Instructs an on–line drive to disconnect itself from the current controller, become available, and spin down.
1	0	Informs an unavailable drive in the process of executing a TOPOLOGY command this controller is finished processing and the drive can return to the on–line port. Refer to the TOPOLOGY command for more detail.
1	1	This condition is invalid.

9.6 DRIVE CLEAR Command

The DRIVE CLEAR command instructs the drive to clear the bits specified in the ERROR BYTE of the drive status response. It also instructs the drive to attempt to clear the error condition. Cleared bits are set to 1 in the bit mask field (Figure 9–5). The drive sends a COMPLETED response to the controller if the specified bits were cleared.

If the error condition and error bits could not be cleared, the drive sends an UNSUCCESSFUL response to the controller. It also sends all of the status bytes to the controller to help the controller determine why the error could not be cleared. A separate section in your Student Reference Manual illustrates the drive status bytes in more detail.

Figure 9–5: DRIVE CLEAR Command

	MSB							LSB	
BYTE 01	0	0	0	0	0	1	0	1	OPCODE = 05
BYTE 02	BIT MASK								

CXO-1350A

9.7 ERROR RECOVERY Command

The ERROR RECOVERY command (Figure 9–6) instructs the drive to activate its error recovery circuits. The circuit activated depends upon the error recovery level specified in the command. RA60, RA80, and RA81 drives do not support controller–assisted hardware error recovery circuits. For drives that do support this feature (RA70, RA82, RA90, etc.), hardware error recovery information may be found in the drive technical description manuals.

Figure 9–6: ERROR RECOVERY Command

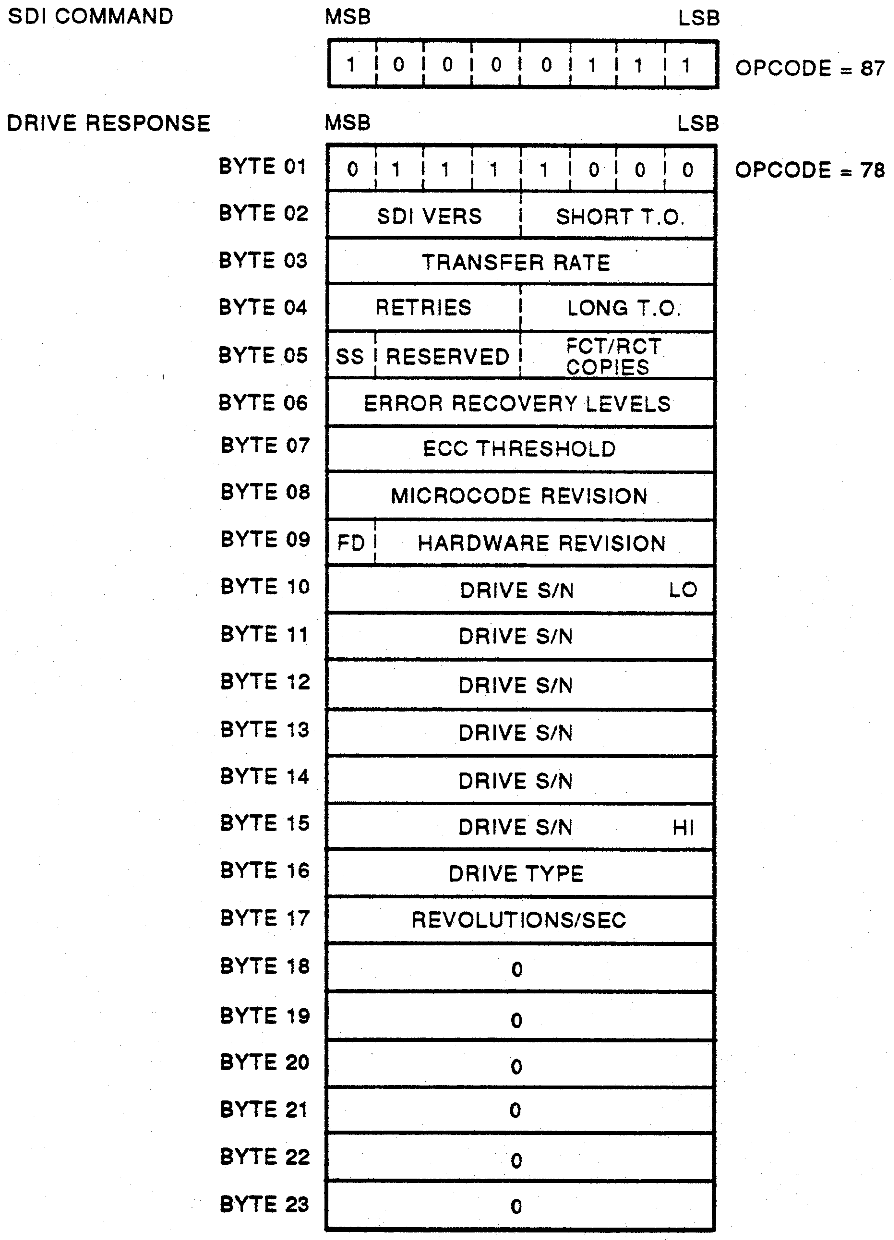
	MSB							LSB	
BYTE 01	0	0	0	0	0	1	1	0	OPCODE = 06
BYTE 02	ERROR RECOVERY LEVEL								

CXO-1351A

9.8 GET COMMON CHARACTERISTICS Command

The GET COMMON CHARACTERISTICS command instructs the drive to send the controller a description of its hardware characteristics common to all subunits of the drive. Figure 9–7 shows the hardware characteristics sent to the controller. The specific information that each drive type sends to the controller is listed in your Student Reference Manual.

Figure 9–7: GET COMMON CHARACTERISTICS Command and Response See Chap 3 for details



CXO-1509B

9.9 GET SUBUNIT CHARACTERISTICS Command

The GET SUBUNIT CHARACTERISTICS command instructs the drive to send the controller a description of hardware characteristics (Figure 9–8 of the subunit specified in the command. For most drives, there is only one subunit, and it is equivalent to the HDA installed. The specific information that each drive type sends to the controller is listed in your Student Reference Manual.

SDI COMMAND

MSB							LSB	
1	0	0	0	1	0	0	0	OPCODE = 88
SUBUNIT MASK				RESERVED				

DRIVE RESPONSE

	MSB			LSB	
BYTE 01	0 1 1 1 0 1 1 1				OPCODE = 77
BYTE 02	LBN SPACE IN CYL			LO	
BYTE 03	LBN SPACE IN CYL				
BYTE 04	LBN SPACE IN CYL				
BYTE 05	0	HI CYL NO.	LBN CYL	HI	
BYTE 06	GROUP/CYLINDER				
BYTE 07	HI STRT XBN	HI STRT LBN			
BYTE 08	TRACKS/GROUP				
BYTE 09	HI STRT XBN	HI STRT LBN			
BYTE 10	RM	RBNs/TRACK			
BYTE 11	RESERVED				
BYTE 12	DATA PREAMBLE (WORDS)				
BYTE 13	HEADER PREAMBLE (WORDS)				
BYTE 14	MEDIA TYPE			LO	
BYTE 15	MEDIA TYPE				
BYTE 16	MEDIA TYPE				
BYTE 17	MEDIA TYPE			HI	
BYTE 18	FCT COPY SIZE (XBNs)			LO	
BYTE 19	FCT COPY SIZE (XBNs)			HI	

Ⓐ

	MSB		LSB	
BYTE 20	LBNs/TRACK			512-BYTE FORMAT
BYTE 21	GROUP OFFSET			
BYTE 22	LBNs IN HOST AREA		LO	
BYTE 23	LBNs IN HOST AREA			
BYTE 24	LBNs IN HOST AREA			
BYTE 25	0 0 0 0	HOST LBNs	HI	
BYTE 26	RCT COPY SIZE (LBNs)		LO	
BYTE 27	RCT COPY SIZE (LBNs)		HI	
BYTE 28	LBNs/TRACK			576-BYTE FORMAT *
BYTE 29	GROUP OFFSET			
BYTE 30	LBNs IN HOST AREA		LO	
BYTE 31	LBNs IN HOST AREA			
BYTE 32	LBNs IN HOST AREA			
BYTE 33	0 0 0 0	HOST LBNs	HI	
BYTE 34	RCT COPY SIZE (LBNs)		LO	
BYTE 35	RCT COPY SIZE (LBNs)		HI	
BYTE 36	XBN SPACE IN CYL		LO	
BYTE 37	XBN SPACE IN CYL		HI	
BYTE 38	DIAG READ AREA (GROUPS)			
BYTE 39	DBN SPACE IN CYL			

* NOT USED IN RA82

CXO-1510A

Figure 9–8: GET SUBUNIT CHARACTERISTICS Command and Response

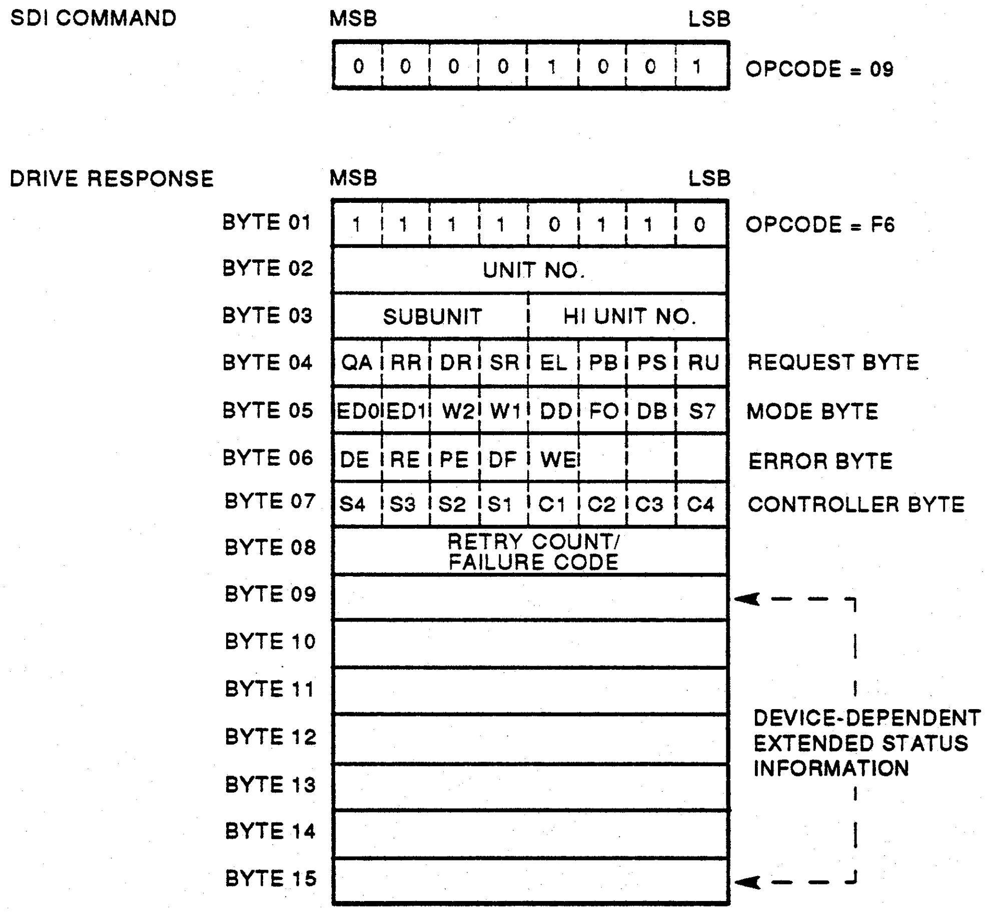
9.10 GET STATUS Command

The GET STATUS command instructs the drive to send the controller all of its current status bytes (Figure 9–9). More specific details about drive status and decoding for each of the drive types are discussed in your Student Reference Manual and will be covered later in this course.

NOTE

When a drive sends an UNSUCCESSFUL response for any level 2 command to the controller, GET STATUS bytes are always included in the response.

Figure 9–9: GET STATUS Command



SDI COMMAND

	MSB							LSB	
	0	0	0	0	1	0	0	1	OPCODE = 09

DRIVE RESPONSE

	MSB							LSB	
BYTE 01	1	1	1	1	0	1	1	0	OPCODE = F6
BYTE 02	UNIT NO.								
BYTE 03	SUBUNIT				HI UNIT NO.				
BYTE 04	QA	RR	DR	SR	EL	PB	PS	RU	REQUEST BYTE
BYTE 05	ED0	ED1	W2	W1	DD	FO	DB	S7	MODE BYTE
BYTE 06	DE	RE	PE	DF	WE				ERROR BYTE
BYTE 07	S4	S3	S2	S1	C1	C2	C3	C4	CONTROLLER BYTE
BYTE 08	RETRY COUNT/ FAILURE CODE								
BYTE 09									DEVICE-DEPENDENT EXTENDED STATUS INFORMATION
BYTE 10									
BYTE 11									
BYTE 12									
BYTE 13									
BYTE 14									
BYTE 15									

CXO-1639B

9.11 INITIATE SEEK Command

The INITIATE SEEK command instructs the drive to seek to the appropriate group and cylinder specified in the INITIATE SEEK command. Figure 9–10 shows the byte configuration for the INITIATE SEEK command.

Figure 9–10: INITIATE SEEK Command

	MSB							LSB	
BYTE 01	0	0	0	0	1	0	1	0	OPCODE = OA
BYTE 02	CYL ADDR LO								
BYTE 03	CYL ADDR								
BYTE 04	CYL ADDR								
BYTE 05	SC	CYL ADDR HI							
BYTE 06	GROUP NUMBER								

CXO-1352A

For the INITIATE SEEK command, the drive sends one of the following responses:

1. UNSUCCESSFUL—The seek operation could not be initiated.
2. SUCCESSFUL—The seek operation was initiated without errors and is currently executing.

9.12 ON–LINE Command

The ONLINE command instructs the drive to enter the on–line state relative to the controller that issued the ONLINE command. The ONLINE command includes a controller–timeout value expressed in seconds (Figure 9–11). DSA drives use this timeout value to monitor controller activity.

Figure 9–11: ONLINE Command

	MSB							LSB	
BYTE 01	1	0	0	0	1	0	1	1	OPCODE = 8B
BYTE 02	COMMAND TIMEOUT (SEC)								

CXO-1353A

DSA drives start the command timer (if the drive is on line) for the time specified in the command byte under the following conditions:

- The drive is ready to send a response to the controller.
- The drive is ready to assert the ATTENTION bit.
- The drive is completing a data transfer operation.
- The drive is waiting for another command (while on line).

The timer is started when the first response frame is ready to be transmitted. Then the controller can time the assertion of RECEIVER READY. CONTROLLER RECEIVER READY must be asserted before the drive can send a response to the controller. If the drive cannot send a response to the controller, it generates a drive–detected error indicating "response timeout error."

Whenever the drive receives a level 1 command or an END frame of a level 2 command from the controller, it cancels and resets the command timer. If the command timer expires, the drive sets the ATTENTION bit and resets and restarts the command timer. If the timer expires a second time (while the ATTENTION bit is set), the drive considers the controller to be off line (SDI cables disconnected, defective, etc.). The drive then disconnects itself from that controller and becomes AVAILABLE to any controller.

For troubleshooting reasons, the RA82 loads error code 1D into its drive internal error silo if the drive disconnects due to a controller timeout. In this case, error code 1D is not generated as a LED or front panel fault, nor is the error sent to the controller. (There is no controller to send it to.)

UDA & KDA 18 Sec Timeout
HSC ≤ V350 24 Sec
HSC V370 2 Sec
HSC V390 2 Sec

HSC V370 & beyond Timeout is programable
show Server will show you the Value
set Server Disk/Drive_timeout = xx 2-24 ! bigger means slower Failover
never set to 1 sec

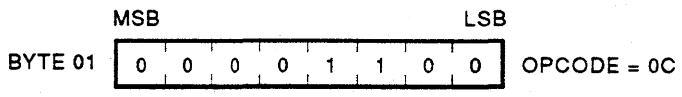
Problems with older U
Disk ucode
Change to update disk

To Make new Value apply must remount disks

9.13 RUN Command

The RUN command instructs the drive to perform a spin–up operation, provided the RUN/STOP switch is enabled. If the controller sends this command with the RUN/STOP switch disabled, most drives generate a drive–detected error indicating this condition. Figure 9–12 shows the command format for the RUN command.

Figure 9–12: RUN Command

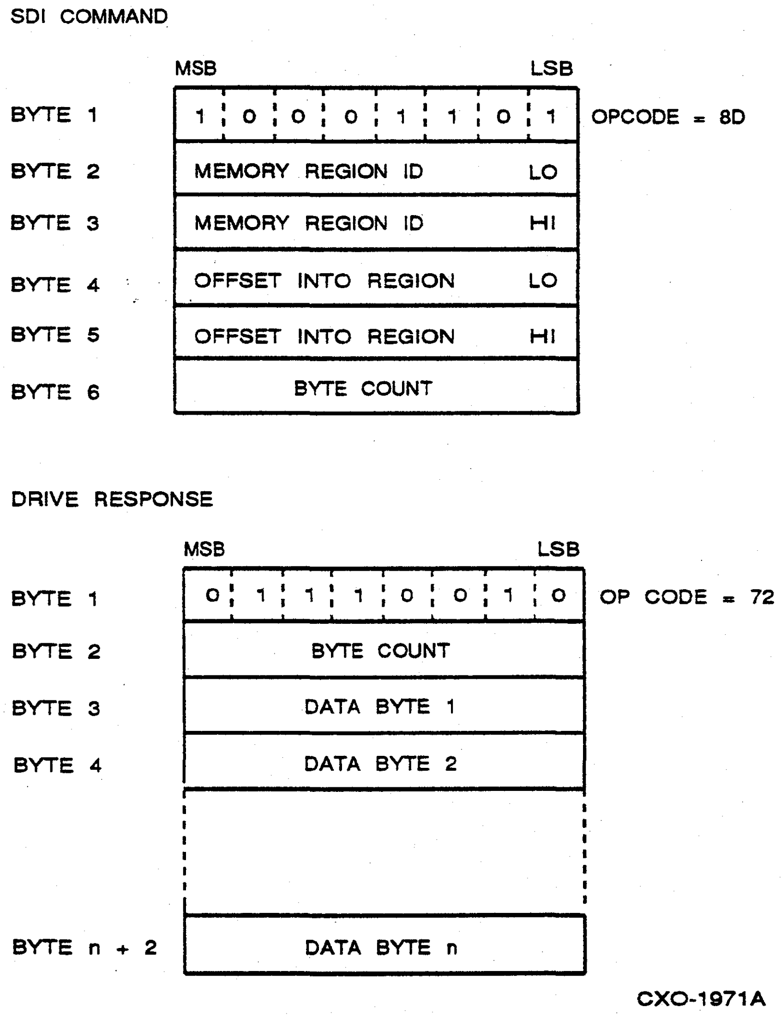


CXO-1354A

9.14 READ MEMORY Command

The READ MEMORY command instructs the drive to fetch and send a specified number of bytes to the controller. These bytes start at the specified offset and are read into the specified read memory region of the drive. Figure 9–13 shows the command format and the drive response for the READ MEMORY command.

Figure 9–13: READ MEMORY Command



The acceptable responses for the READ MEMORY command are as follows:

- UNSUCCESSFUL—The reason in the response data as shown in Figure 9–13.
- SUCCESSFUL—Contents of the requested memory locations as shown in Figure 9–13.

NOTE

Memory refers to the ROM/RAM areas of the disk drive and not disk media storage areas.

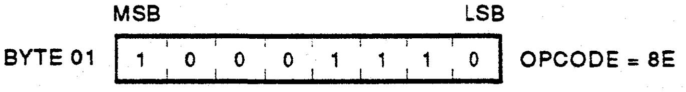
The various drive types provide different information during READ MEMORY REGION responses, depending upon the extent of the drive design. Some of the information provided includes:

- Error silo information
- Diagnostic parameters
- Diagnostic status and results
- Extended diagnostic status
- Extended drive status
- ROM revision and checksum field information
- RAM, SDI buffers and status words

9.15 RECALIBRATE Command

The RECALIBRATE command instructs the drive to perform a recalibrate operation and seek to cylinder 0. Figure 9–14 shows the format for the RECALIBRATE command. Some drives (RA81 and RA82, for example) also perform automatic internal servo adjustment routines during the RECALIBRATE command.

Figure 9–14: RECALIBRATE Command



CXO-1355A

Acceptable responses for the RECALIBRATE command are as follows:

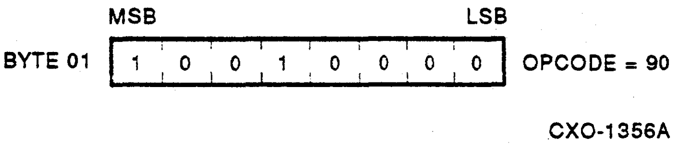
- **UNSUCCESSFUL**
- **SUCCESSFUL** —Recalibrate operation completed without errors.

9.16 TOPOLOGY Command

The TOPOLOGY command (Figure 9–15) instructs an on–line drive to make itself temporarily AVAILABLE for dialogue to any controller on an alternately enabled port (Figure 9–15). While communicating with a drive in topology mode, an alternate controller (controller B) can issue **only** the following level 2 commands:

- GET STATUS
- GET COMMON CHARACTERISTICS
- GET SUBUNIT CHARACTERISTICS
- CHANGE CONTROLLER FLAGS
- DISCONNECT (with TT bit set to terminate topology communication)

Figure 9–15: Topology Command



The acceptable responses for the TOPOLOGY command are as follows:

- **UNSUCCESSFUL**—As indicated by the drive status in the response data.
- **SUCCESSFUL** —Topology completed without errors.

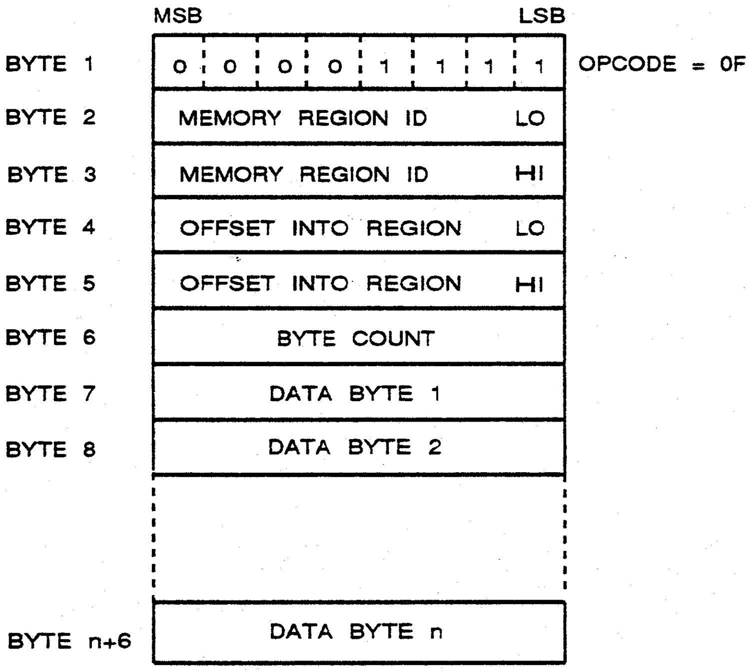
The drive sets the OA bit of status Byte 4 to indicate to controller B it is online to an alternate controller (controller A), and it is executing a TOPOLOGY command. This feature allows controller B to update its internal registers with status and drive characteristic information.

When the alternate controller completes its dialogue with the drive, it issues a DISCONNECT command with the TT (terminate topology) bit set. The drive then returns to its original on–line port and sends a COMPLETED response to that controller indicating the TOPOLOGY command is complete.

9.17 WRITE MEMORY Command

The WRITE MEMORY command instructs the drive to write the supplied data in the indicated memory regions. Figure 9–16 shows the command format for the WRITE MEMORY command.

Figure 9–16: WRITE MEMORY Command



	MSB							LSB	
BYTE 1	0	0	0	0	1	1	1	1	OPCODE = 0F
BYTE 2	MEMORY REGION ID LO								
BYTE 3	MEMORY REGION ID HI								
BYTE 4	OFFSET INTO REGION LO								
BYTE 5	OFFSET INTO REGION HI								
BYTE 6	BYTE COUNT								
BYTE 7	DATA BYTE 1								
BYTE 8	DATA BYTE 2								
BYTE n+6	DATA BYTE n								

CXO-1513A

Most drives support only one valid region for the WRITE MEMORY command, and it is listed below:

Region	Description	No. Bytes
FFFC	Diagnostic Parameters	6

NOTE

Memory refers to the RAM areas of the disk drive and not the disk media storage areas.

CHAPTER 10

DECODING DRIVE STATUS BYTES

10.1 RA60 DRIVE STATUS DECODE

The following pages describe decoding the RA60 status bytes. These bytes are passed from the drive to the controller when a GET STATUS command is issued to the RA60. The RA60 also provides these status bytes to the controller if a command from the controller fails to execute properly within the drive.

System error logs, controller error logs, and diagnostic utilities often display most of the drive status bytes. This decoding information is provided to help you understand the meaning of these bytes and/or the meaning of bits within any of these bytes.

Figure 10–1: Summary of RA60 Drive Status Codes

BYTE 01	RESPONSE OPCODE	
BYTE 02	UNIT SELECT (LOWER)	
BYTE 03	UNIT + SUBUNIT MASK	
BYTE 04	REQUEST BYTE	GENERIC DRIVE STATUS BYTE
BYTE 05	MODE BYTE	GENERIC DRIVE STATUS BYTE
BYTE 06	ERROR BYTE	GENERIC DRIVE STATUS BYTE
BYTE 07	CONTROLLER BYTE	
BYTE 08	RETRY COUNT/FAILURE	
BYTE 09	PREVIOUS CYL (LO)	EXTENDED DRIVE STATUS BYTE
BYTE 10	PREVIOUS CYL (HI)	EXTENDED DRIVE STATUS BYTE
BYTE 11	PREVIOUS HEAD	EXTENDED DRIVE STATUS BYTE
BYTE 12	CURRENT CYL (LO)	EXTENDED DRIVE STATUS BYTE
BYTE 13	CURRENT CYL (HI)	EXTENDED DRIVE STATUS BYTE
BYTE 14	CURRENT HEAD	EXTENDED DRIVE STATUS BYTE
BYTE 15	MASTER ERROR CODE	EXTENDED DRIVE STATUS BYTE

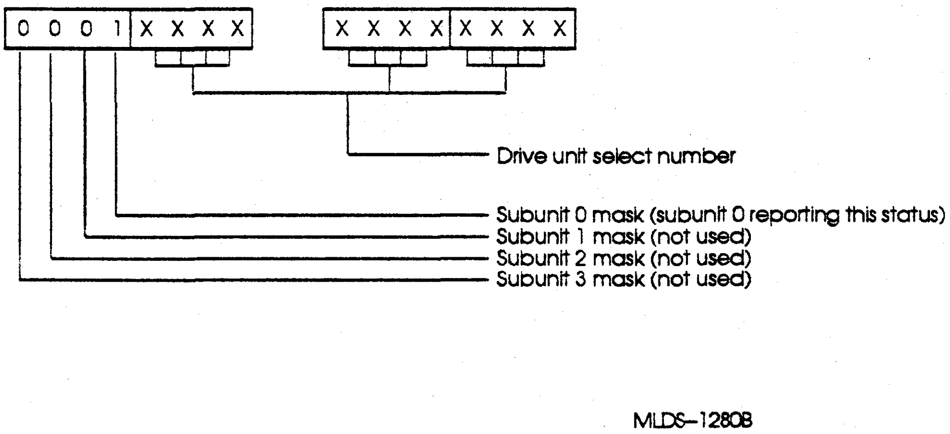
CXO-2363A

Bytes 2 through 15 are generally available from system error logs, the HSC console display, and various diagnostics that test RA–series drives. The format in which this information is displayed depends upon the specific type of system (VMS, RSTS, RSX, etc.) or the specific type of controller (HSC50, HSC70, etc.).

Figure 10–2: RA60 Byte 1

This is the RESPONSE opcode from the drive to the controller, but it is rarely displayed to a user. It indicates the success or non–success of the previous command sent from the controller to the drive.

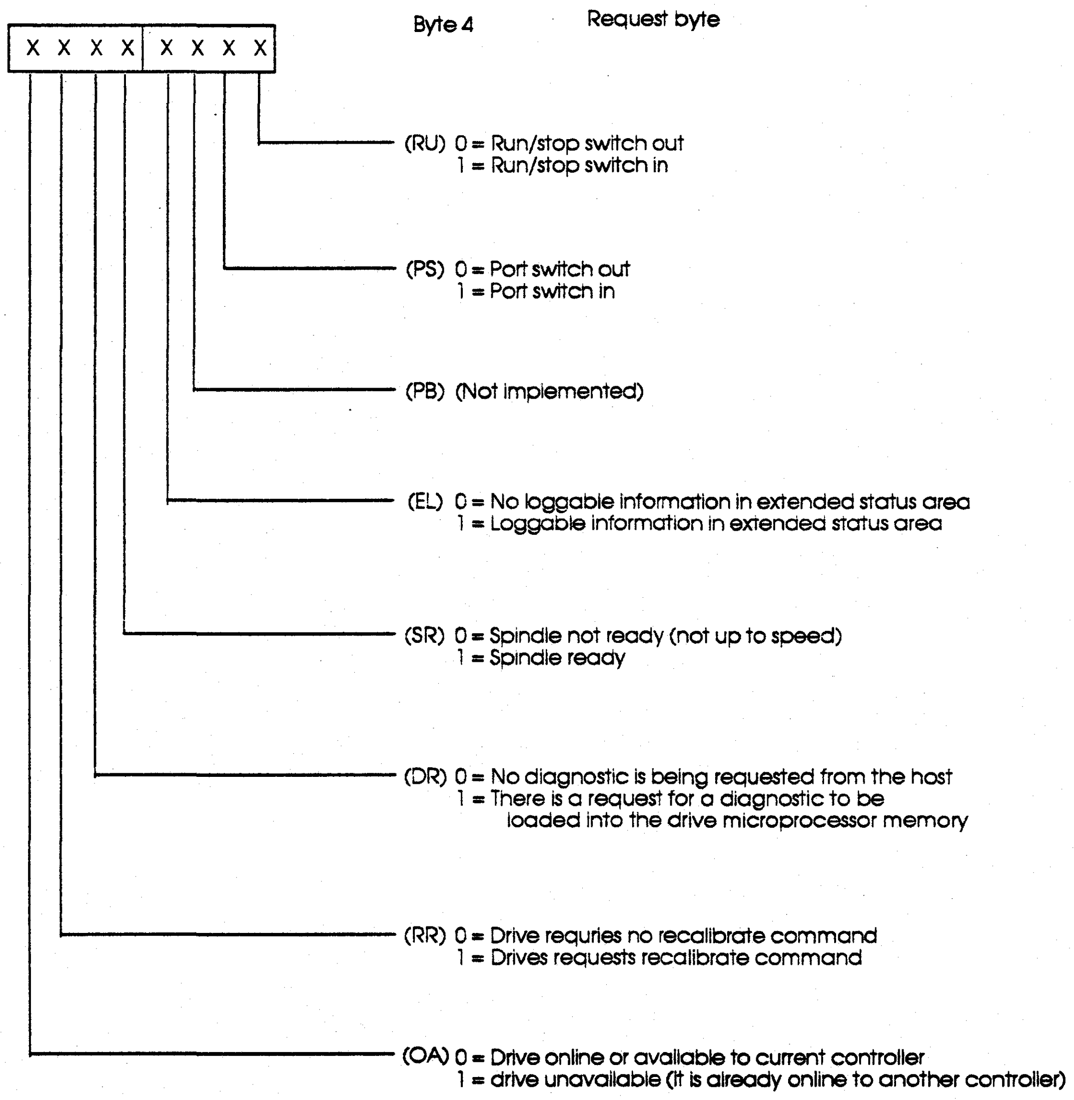
Figure 10–3: RA60 Bytes 2–3



NOTE

The RA60 has no multiple subunits and will always indicate Subunit 0.

Figure 10–4: RA60 Byte 4 Request Byte



(RU) 0 = Run/stop switch out
1 = Run/stop switch in

(PS) 0 = Port switch out
1 = Port switch in

(PB) (Not implemented)

(EL) 0 = No loggable information in extended status area
1 = Loggable information in extended status area

(SR) 0 = Spindle not ready (not up to speed)
1 = Spindle ready

(DR) 0 = No diagnostic is being requested from the host
1 = There is a request for a diagnostic to be loaded into the drive microprocessor memory

(RR) 0 = Drive requries no recalibrate command
1 = Drives requests recalibrate command

(OA) 0 = Drive online or available to current controller
1 = drive unavailable (It is already online to another controller)

MLDS–2350A

Figure 10–5: RA60 Byte 5 Mode Byte

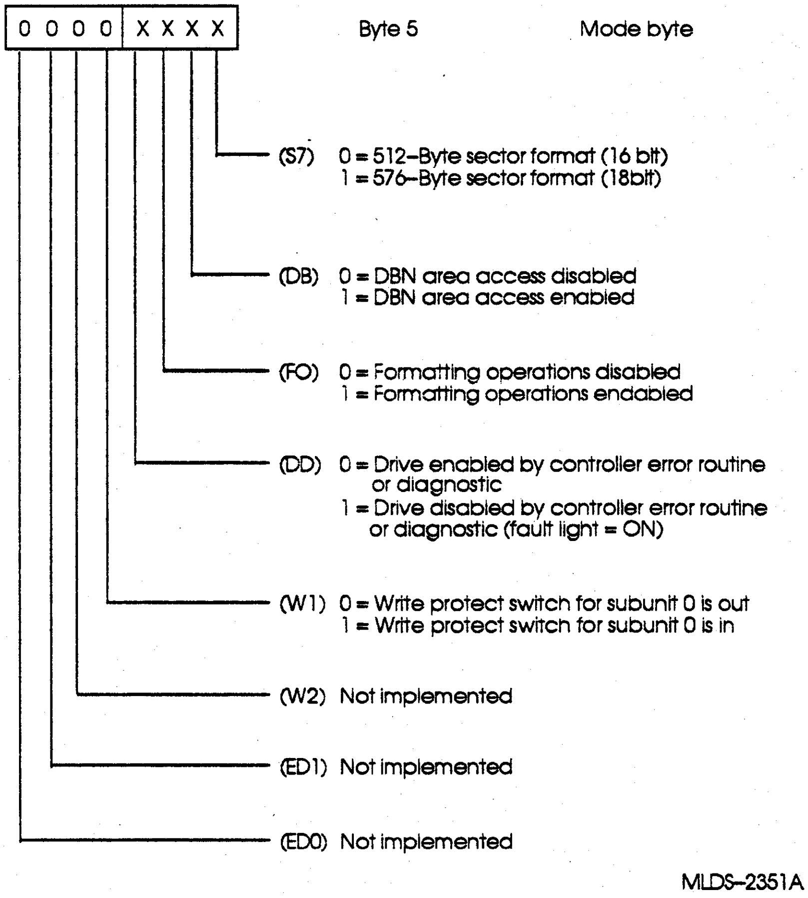


Figure 10–6: RA60 Byte 6 Error Byte

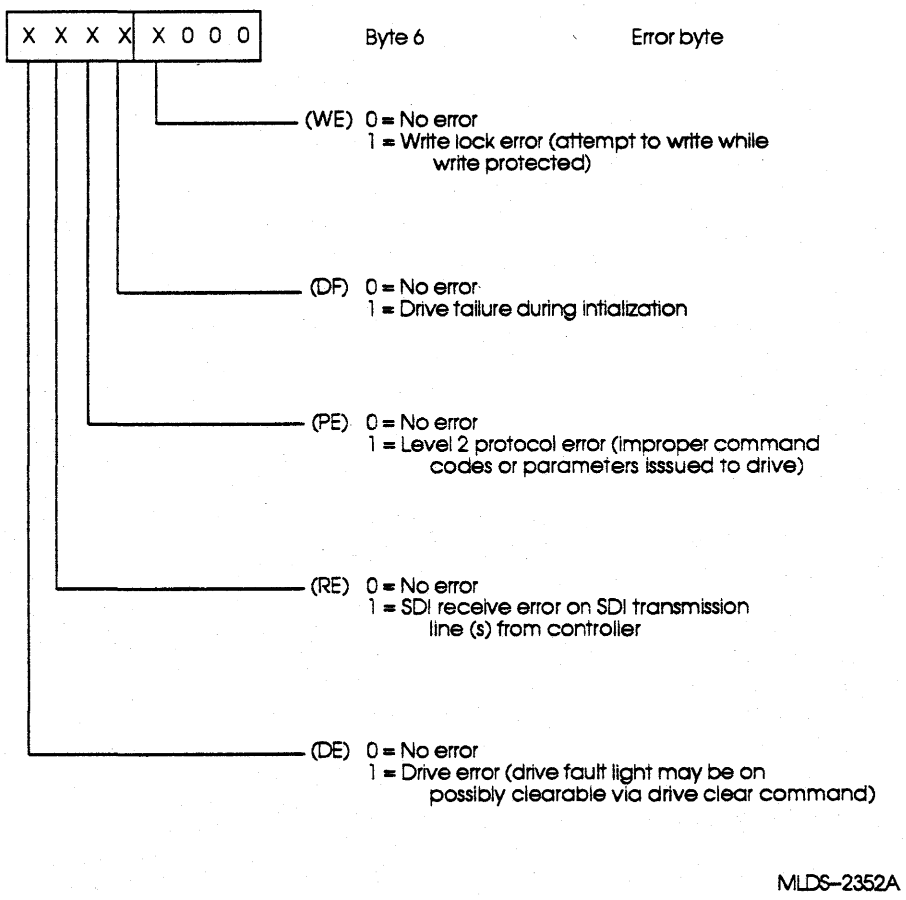
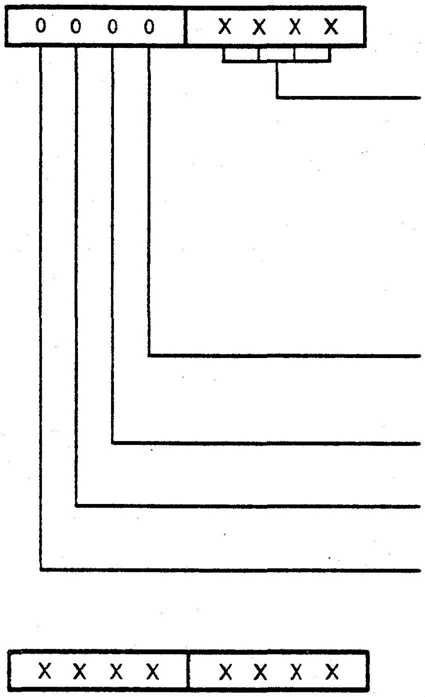


Figure 10–7: RA60 Bytes 7–8



BYTE 7 CONTROLLER BYTE

0000 = NORMAL DRIVE OPERATION

1000 = DRIVE OFFLINE DUE TO BEING UNDER CONTROL OF A DIAGNOSTIC

1001 = DRIVE OFFLINE DUE TO ANOTHER DRIVE HAVING THE SAME UNIT SELECT IDENTIFIER

(S1) 1 = SUBUNIT 0 ATTENTION AVAILABLE MESSAGES SUPPRESSED IN THE CONTROLLER

(S2) 1 = NOT USED

(S3) 1 = NOT USED

(S4) 1 = NOT USED

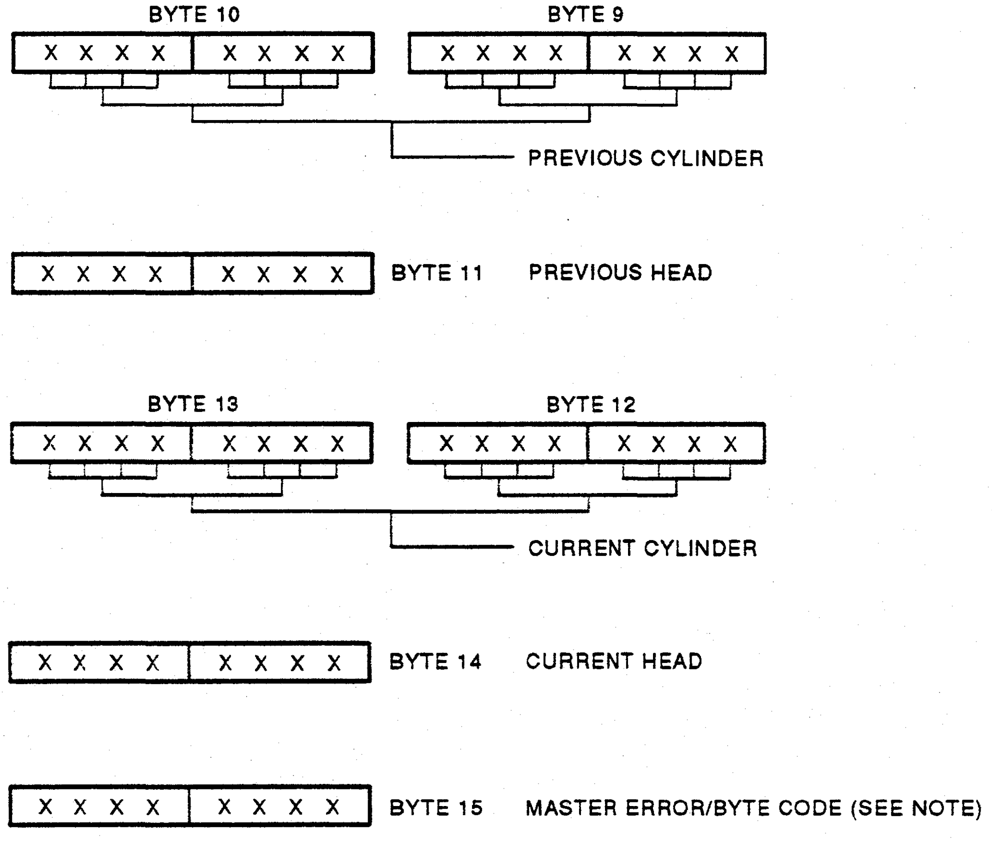
BYTE 8 RETRY COUNT/FAILURE CODE

CONTAINS DIFFERENT INFORMATION DEPENDING UPON CONDITION OF DF BIT IN BYTE 6

IF DF = 0	BYTE 8 CONTAINS RETRY COUNT FOR PREVIOUS COMMAND ISSUED (THE NUMBER OF TIMES COMMAND WAS RE-EXECUTED BEFORE SUCCESSFUL)
IF DF = 1	BYTE 8 CONTAINS SPECIFIC ERROR CODE RELATING TO INITIALIZATION FAILURE THAT CAUSED DF BIT TO SET (REFER TO RA60 ERROR CODE LIST IN SERVICE MANUAL)

CXO-2357A

Figure 10–8: RA60 Bytes 9–15



NOTE: REFER TO RA60 ERROR CODE LIST IN SERVICE MANUAL.

CXO-2358A

10.2 RA70 DRIVE STATUS DECODE

The following pages describe decoding the RA70 status bytes. These bytes are passed from the drive to the controller when a GET STATUS command is issued to the RA70. The RA70 also provides these status bytes to the controller if a command from the controller fails to execute properly within the drive.

System error logs, controller error logs, and diagnostic utilities often display most of the drive status bytes. This decoding information is provided here to help you understand the meaning of these bytes and/or the meaning of bits within any of these bytes.

Figure 10–9: Summary of RA70 Drive Status Codes

BYTE 01	RESPONSE OPCODE	
BYTE 02	UNIT SELECT (LOWER)	
BYTE 03	UNIT + SUBUNIT MASK	
BYTE 04	REQUEST BYTE	GENERIC DRIVE STATUS BYTE
BYTE 05	MODE BYTE	GENERIC DRIVE STATUS BYTE
BYTE 06	ERROR BYTE	GENERIC DRIVE STATUS BYTE
BYTE 07	CONTROLLER BYTE	
BYTE 08	RETRY COUNT/FAILURE	
BYTE 09	PREVIOUS CMD OPCODE	EXTENDED DRIVE STATUS BYTE
BYTE 10	DRIVE STATE FLAGS	EXTENDED DRIVE STATUS BYTE
BYTE 11	CYLINDER ADDR (LO)	EXTENDED DRIVE STATUS BYTE
BYTE 12	CYLINDER ADDR (HI)	EXTENDED DRIVE STATUS BYTE
BYTE 13	GROUP NO. (HEAD)	EXTENDED DRIVE STATUS BYTE
BYTE 14	DRIVE ERROR CODE	EXTENDED DRIVE STATUS BYTE
BYTE 15	MFG ERROR CODE	EXTENDED DRIVE STATUS BYTE

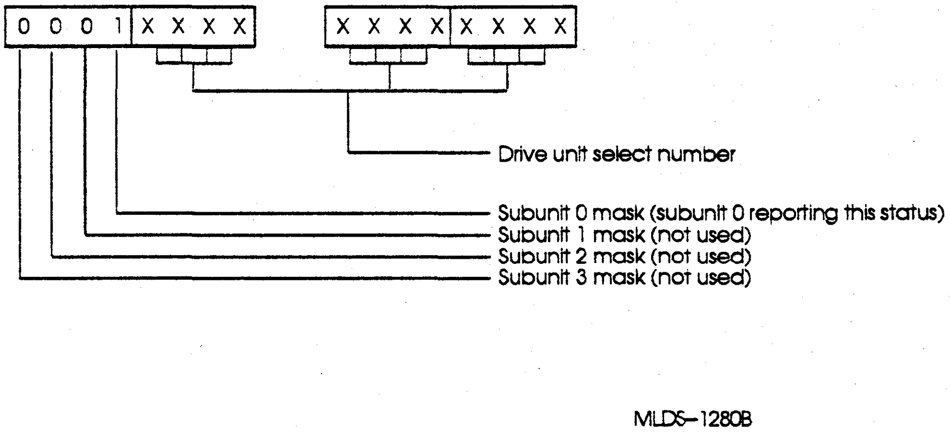
CXO-2364A

Bytes 2 through 15 are generally available from system error logs, the HSC console display, and various diagnostics that test RA–series drives. The format in which this information is displayed depends upon the specific type of system (VMS, RSTS, RSX, etc.) or the specific type of controller (HSC50, HSC70, etc.).

Figure 10–10: RA70 Byte 1

This is the RESPONSE opcode from the drive to the controller, but it is rarely displayed to a user. It indicates the success or non–success of the previous command sent from the controller to the drive.

Figure 10–11: RA70 Bytes 2–3



NOTE

The RA70 has no multiple subunits and will always indicate Subunit 0.

Figure 10–12: RA70 Byte 4 Request Byte

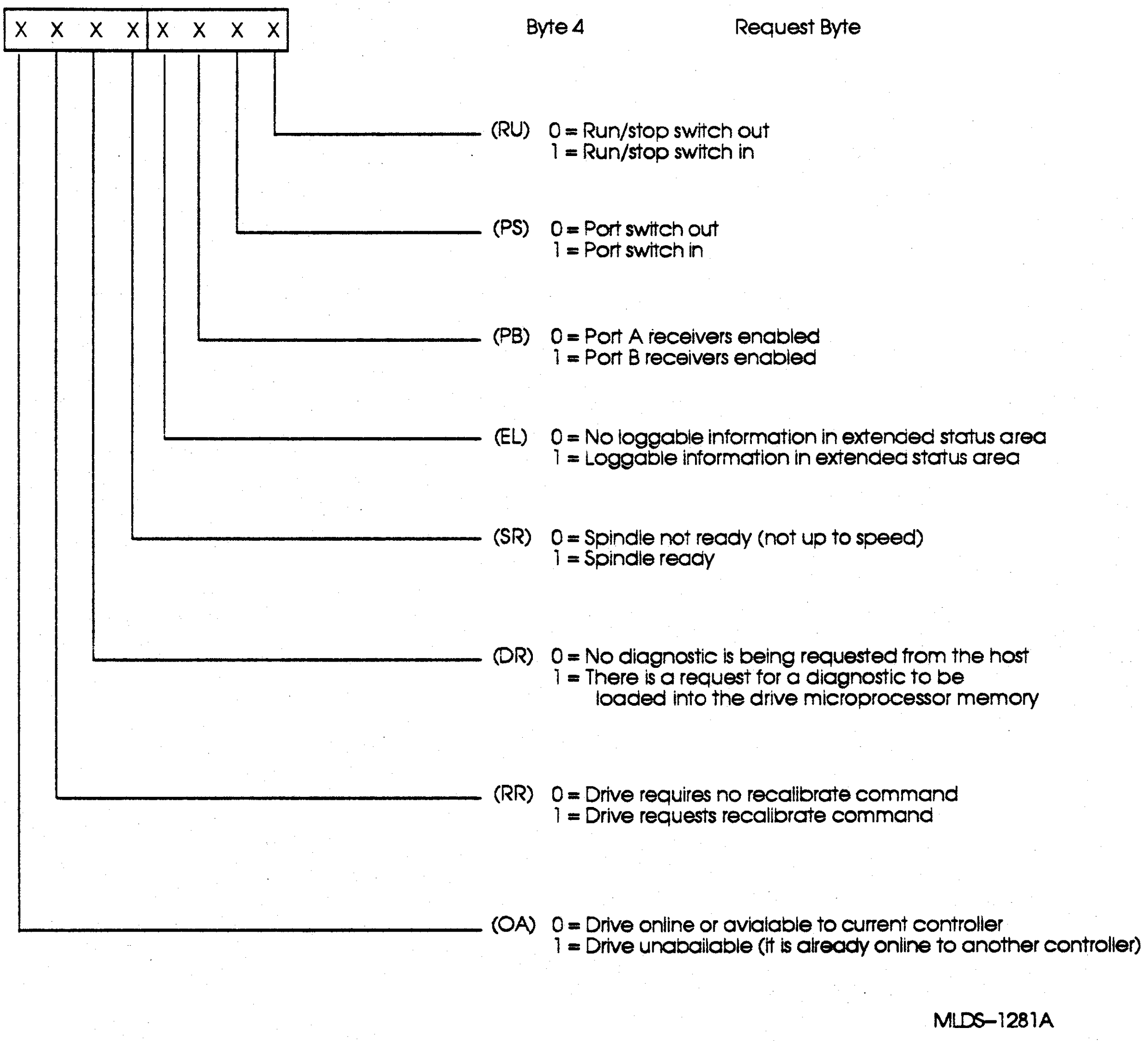


Figure 10–13: RA70 Byte 5 Mode Byte

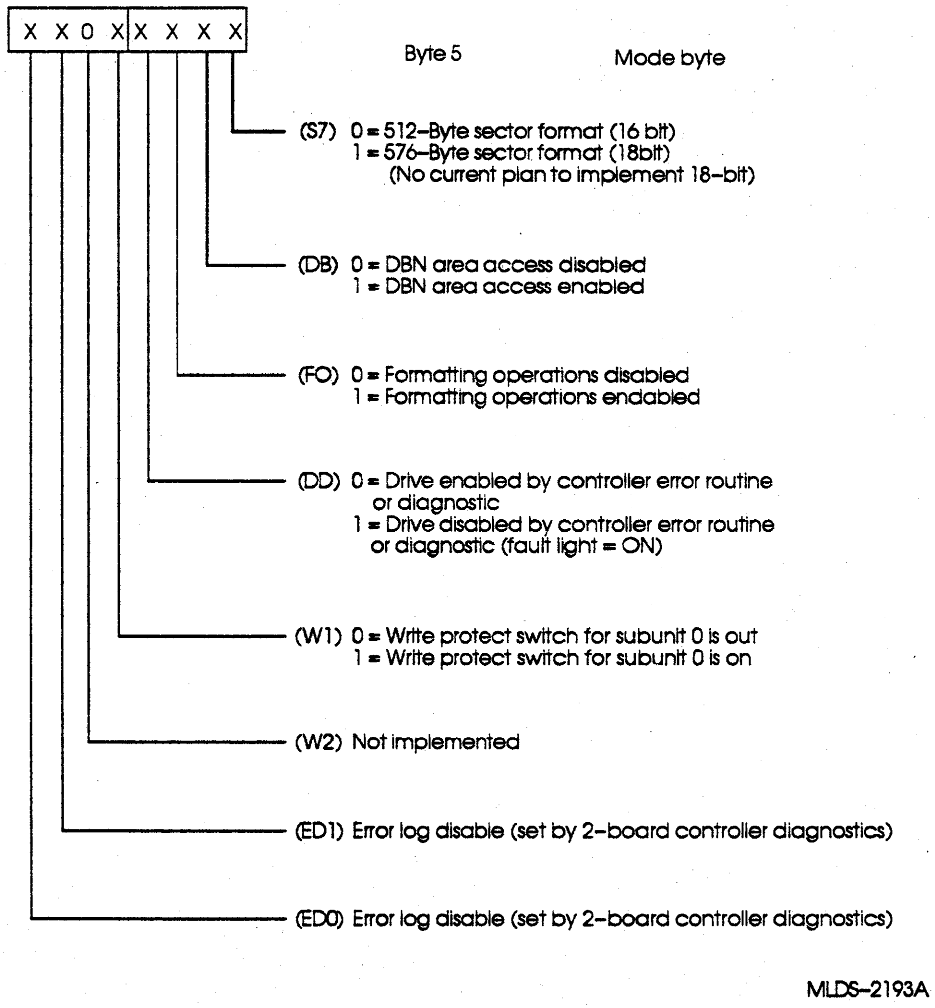


Figure 10–14: RA70 Byte 6 Error Byte

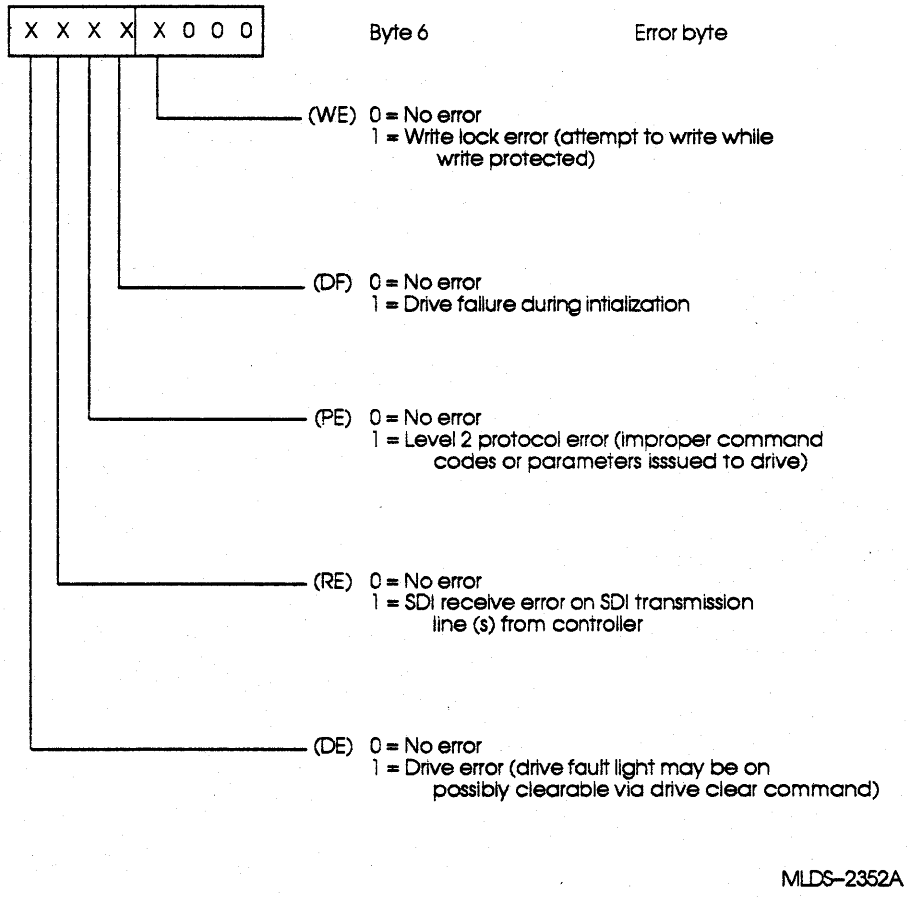
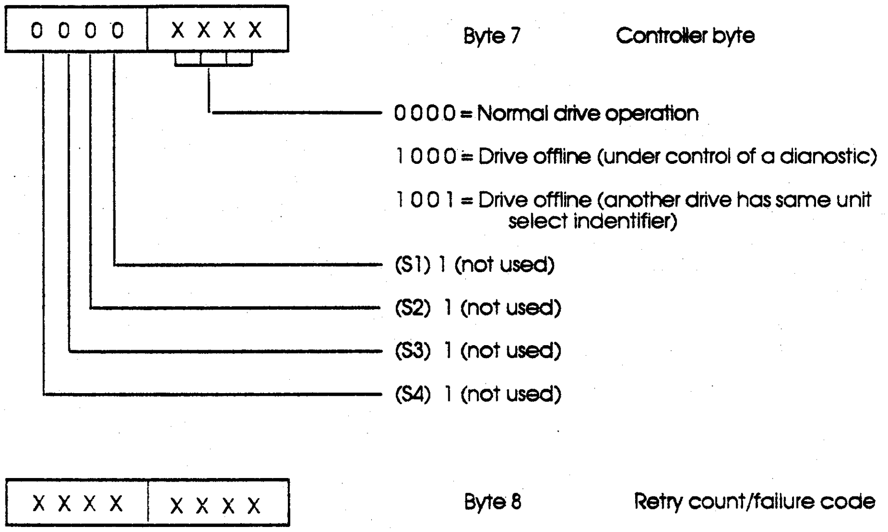


Figure 10–15: RA70 Bytes 7–8



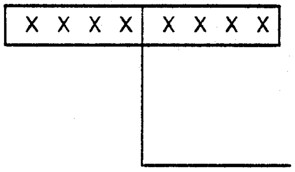
MLDS–1284B

NOTE

Byte 8; Retry count during the last Seek or Recalibration Command.

The number of times the command was re–tried internal to the RA70 in order to attempt successful completion of the SEEK or RECALIBRATE operation.

Figure 10–16: RA70 Byte 9 Last Opcode



Byte 9 Last opcode

(Extended drive status byte)

Opcode of the last previous level 2 drive command decoded by the drive (received from the SDI controller)

81 = Change mode
82 = Change controller flags
03 = Diagnose
84 = Disconnect (drive)
05 = Drive clear

06 = Error recovery
87 = Get common characteristics
88 = Get subunit characteristics
0A = Initiate seek
8B = Online

0C = Run
8D = Read memory
8E = Recalibrate
90 = Topology
0F = Write memory

FF = Select group (level 1 command – processed by firmware seek head select sub–routines

MLDS–1285B

Figure 10–17: RA70 Byte 10 Drive–Detected SDI Error

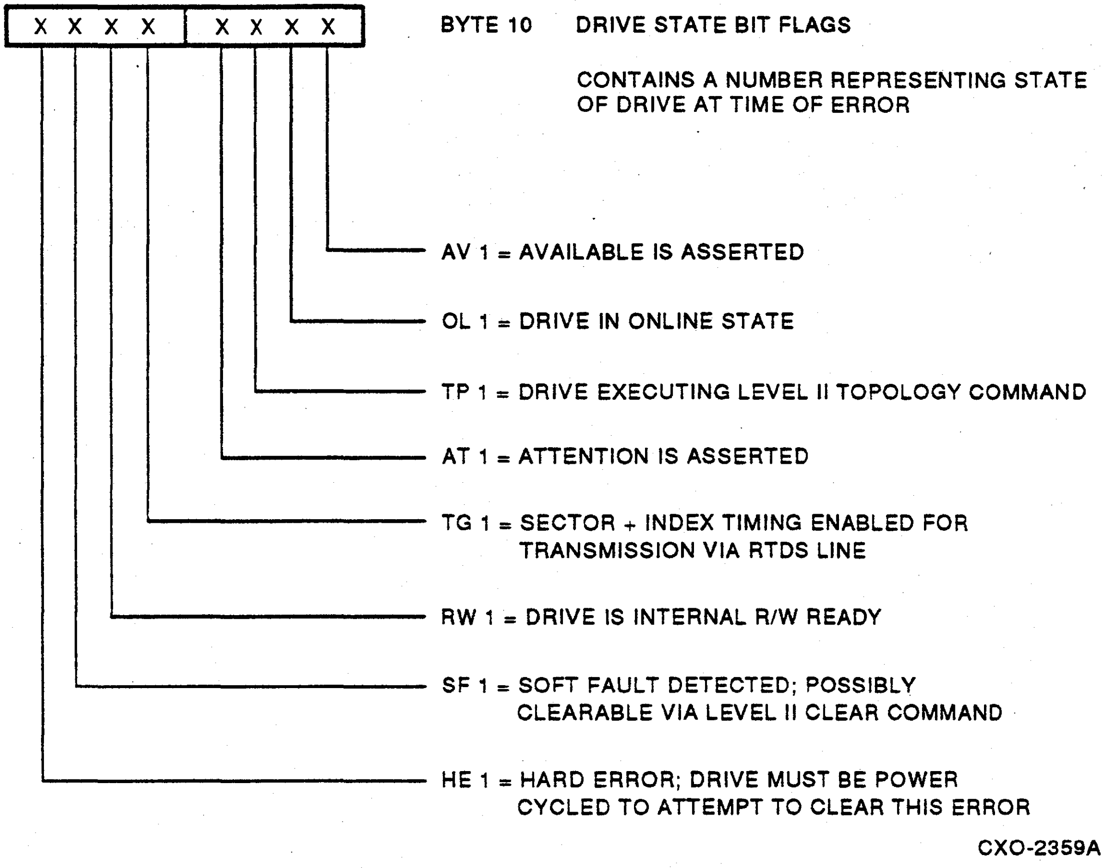
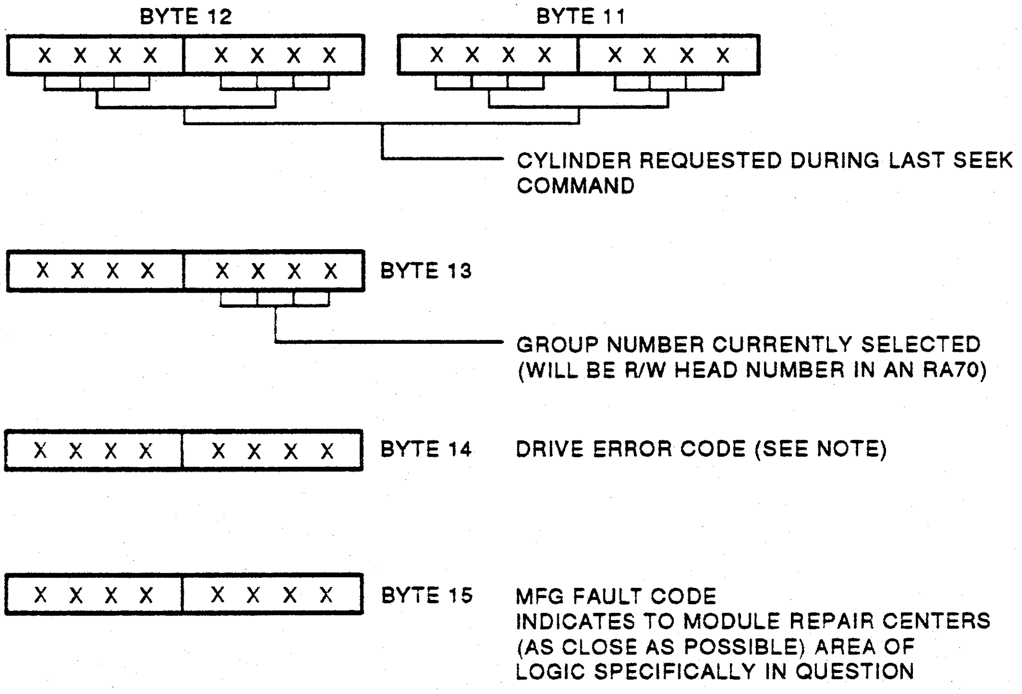


Figure 10–18: RA70 Bytes 11–15



NOTE: REFER TO RA70 ERROR CODE LIST IN SERVICE MANUAL.

CXO-2360A

10.3 RA80 DRIVE STATUS DECODE

The following pages describe decoding the RA80 status bytes. These bytes are passed from the drive to the controller when a GET STATUS command is issued to the RA80. The RA80 also provides these status bytes to the controller if a command from the controller fails to execute properly within the drive.

System error logs, controller error logs, and diagnostic utilities often display most of the drive status bytes. This decoding information is provided here to help you understand the meaning of these bytes and/or the meaning of bits within any of these bytes.

Figure 10–19: Summary of RA80 Drive Status Codes

BYTE 01	RESPONSE OPCODE	
BYTE 02	UNIT SELECT (LOWER)	
BYTE 03	UNIT + SUBUNIT MASK	
BYTE 04	REQUEST BYTE	GENERIC DRIVE STATUS BYTE
BYTE 05	MODE BYTE	GENERIC DRIVE STATUS BYTE
BYTE 06	ERROR BYTE	GENERIC DRIVE STATUS BYTE
BYTE 07	CONTROLLER BYTE	
BYTE 08	RETRY COUNT/FAILURE	
BYTE 09	PREVIOUS CMD OPCODE	EXTENDED DRIVE STATUS BYTE
BYTE 10	SDI ERROR BITS	EXTENDED DRIVE STATUS BYTE
BYTE 11	CYLINDER ADDR (LO)	EXTENDED DRIVE STATUS BYTE
BYTE 12	CYLINDER ADDR (HI)	EXTENDED DRIVE STATUS BYTE
BYTE 13	CURRENT GROUP NUMBER	EXTENDED DRIVE STATUS BYTE
BYTE 14	LED ERROR CODE	EXTENDED DRIVE STATUS BYTE
BYTE 15	F.P. ERROR CODE	EXTENDED DRIVE STATUS BYTE

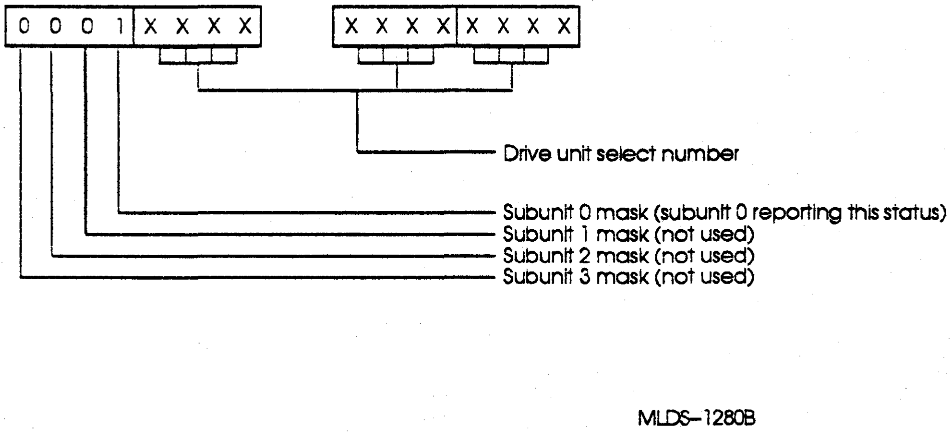
CXO-2365A

Bytes 2 through 15 are generally available from system error logs, the HSC console display, and various diagnostics that test RA–series drives. The format in which this information is displayed depends upon the specific type of system (VMS, RSTS, RSX, etc.) or the specific type of controller (HSC50, HSC70, etc.).

Figure 10–20: RA80 Byte 1

This is the RESPONSE opcode from the drive to the controller, but it is rarely displayed to a user. It indicates the success or non–success of the previous command sent from the controller to the drive.

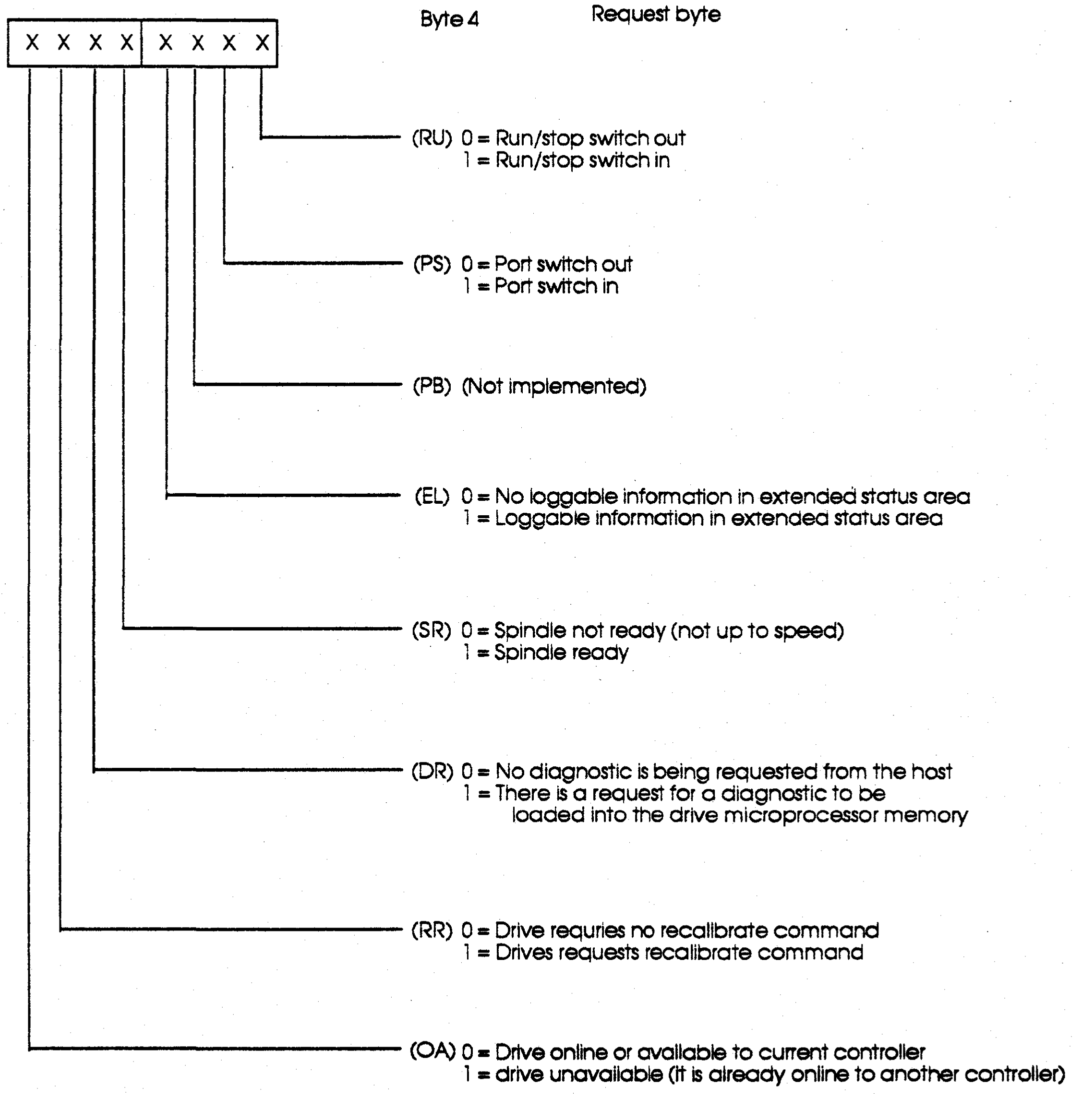
Figure 10–21: RA80 Bytes 2–3



NOTE

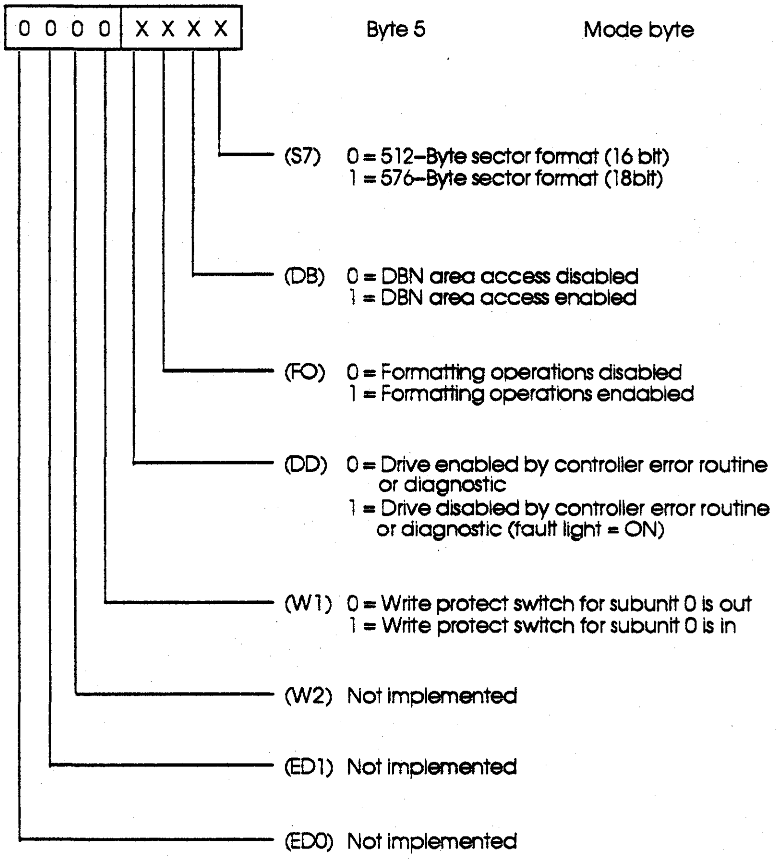
The RA80 has no multiple subunits and will always indicate Subunit 0.

Figure 10–22: RA80 Byte 4 Request Byte



MLDS-2350A

Figure 10–23: RA80 Byte 5 Mode Byte



MLDS–2351A

Figure 10–24: RA80 Byte 6 Error Byte

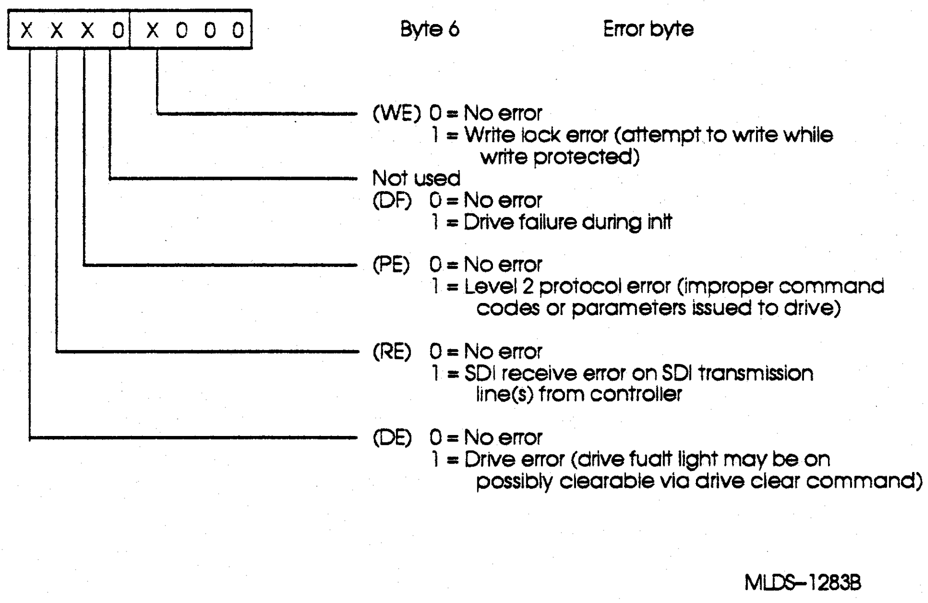
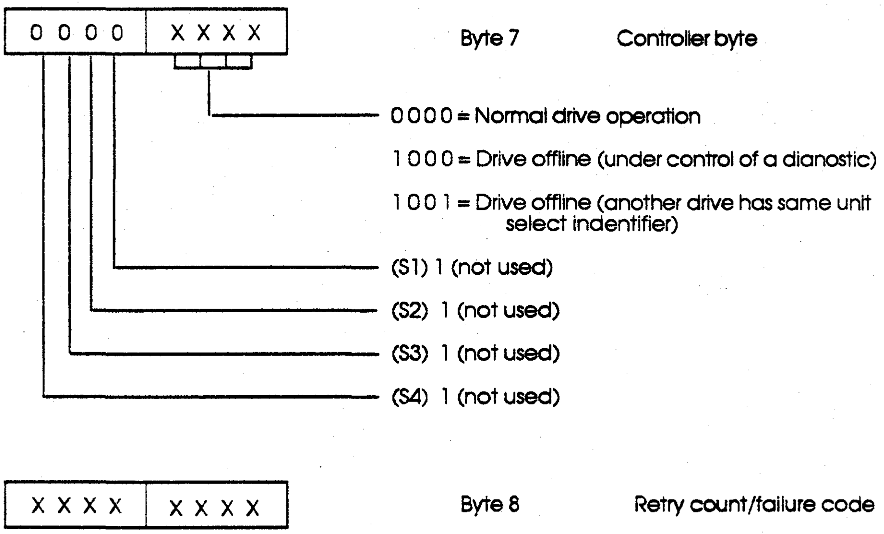


Figure 10–25: RA80 Bytes 7–8

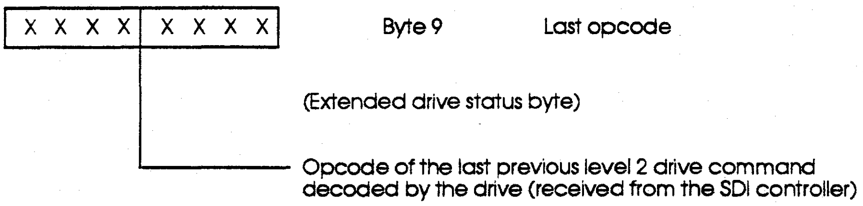


MLDS-1284B

NOTE

Byte 8: Retry count during the last SEEK or RECALIBRATION COMMAND (the number of times the command was re-tried before successful completion). Maximum allowed by microcode is 3 before the seek or recalibration will be aborted. A value of 0 indicates 3 retries completed since the counter decrements to 0 during each retry.

Figure 10–26: RA80 Byte 9 Last Opcode



81 = Change mode
82 = Change controller flags
03 = Diagnose
84 = Disconnect (drive)
05 = Drive clear

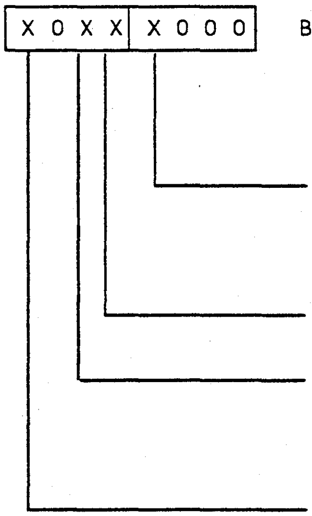
06 = Error recovery
87 = Get common characteristics
88 = Get subunit characteristics
0A = Initiate seek
8B = Online

0C = Run
8D = Read memory
8E = Recalibrate
90 = Topology
0F = Write memory

FF = Select group (level 1 command – processed by firmware
seek head select sub–routines

MLDS–1285B

Figure 10–27: RA80 Byte 10 Drive-Detected SDI Error



Byte 10 Drive–detected SDI error
Hard SDI transmission errors detected by drive receive logic circuitry

(OWE) 1 = R/W overrun/overwrite error

Sector pulse detected before finishing R/W transfer or read/write gate up too long, or "glitch" detected as a sector pulse.

(PAR) 1 = Parity error detected on the RTCS line from controller to drive.

(CPE) 1 = Control pulse error

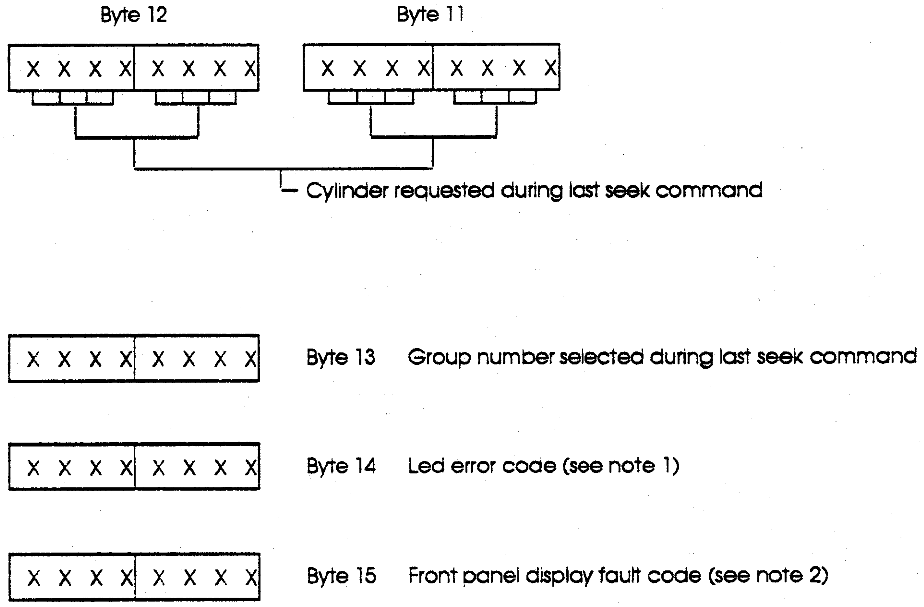
Two (2) or more pulses of same polarity detected on RTCS line from controller to drive

(DPE) 1 = Data pule error

Two (2) or more pulses of same polarity detected on CMD/WRT dta line from controller to drive.

MLDS–2353A

Figure 10–28: RA80 Bytes 11–15



NOTES:

1. Refer to LED error code list in service manual.
2. Refer to control panel fault code list in service manual.

MLDS–2354A

10.4 RA81 DRIVE STATUS DECODE

The following pages describe the decoding of the RA81 status bytes. These bytes are passed from the drive to the controller when a GET STATUS command is issued to the RA81. The RA81 also provides these status bytes to the controller if a command from the controller fails to execute properly within the drive.

System error logs, controller error logs, and diagnostic utilities often display most of the drive status bytes. This decoding information is provided to help you understand the meaning of these bytes and/or the meaning of bits within any of these bytes.

Figure 10–29: Summary of RA81 Drive Status Codes

BYTE 01	RESPONSE OPCODE	
BYTE 02	UNIT SELECT (LOWER)	
BYTE 03	UNIT + SUBUNIT MASK	
BYTE 04	REQUEST BYTE	GENERIC DRIVE STATUS BYTE
BYTE 05	MODE BYTE	GENERIC DRIVE STATUS BYTE
BYTE 06	ERROR BYTE	GENERIC DRIVE STATUS BYTE
BYTE 07	CONTROLLER BYTE	
BYTE 08	RETRY COUNT/FAILURE	
BYTE 09	PREVIOUS CMD OPCODE	EXTENDED DRIVE STATUS BYTE
BYTE 10	SDI ERROR BITS	EXTENDED DRIVE STATUS BYTE
BYTE 11	CYLINDER ADDR (LO)	EXTENDED DRIVE STATUS BYTE
BYTE 12	CYLINDER ADDR (HI)	EXTENDED DRIVE STATUS BYTE
BYTE 13	CURRENT GROUP NUMBER	EXTENDED DRIVE STATUS BYTE
BYTE 14	LED ERROR CODE	EXTENDED DRIVE STATUS BYTE
BYTE 15	F.P. ERROR CODE	EXTENDED DRIVE STATUS BYTE

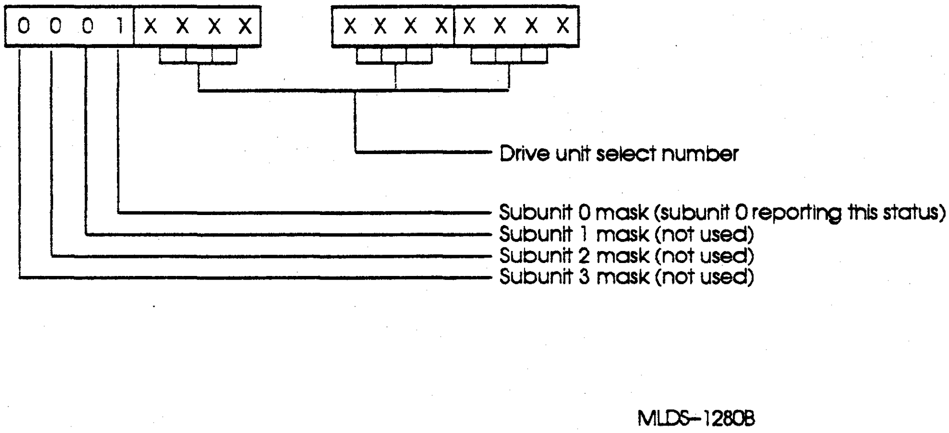
CXO-2365A

Bytes 2 thru 15 are generally available from system error logs, the HSC console display, and various diagnostics that test RA–series drives. The format in which this information is displayed depends upon the specific type of system (VMS, RSTS, RSX, etc.) or the specific type of controller (HSC50, HSC70, etc.).

Figure 10–30: RA81 Byte 1

This will be the RESPONSE opcode from the drive to the controller, but it is rarely displayed to a user. It indicates the success or non–success of the previous command sent from the controller to the drive.

Figure 10–31: RA81 Bytes 2–3



NOTE

The RA81 has no multiple subunits and will always indicate Subunit 0.

Figure 10–32: RA81 Byte 4 Request Byte

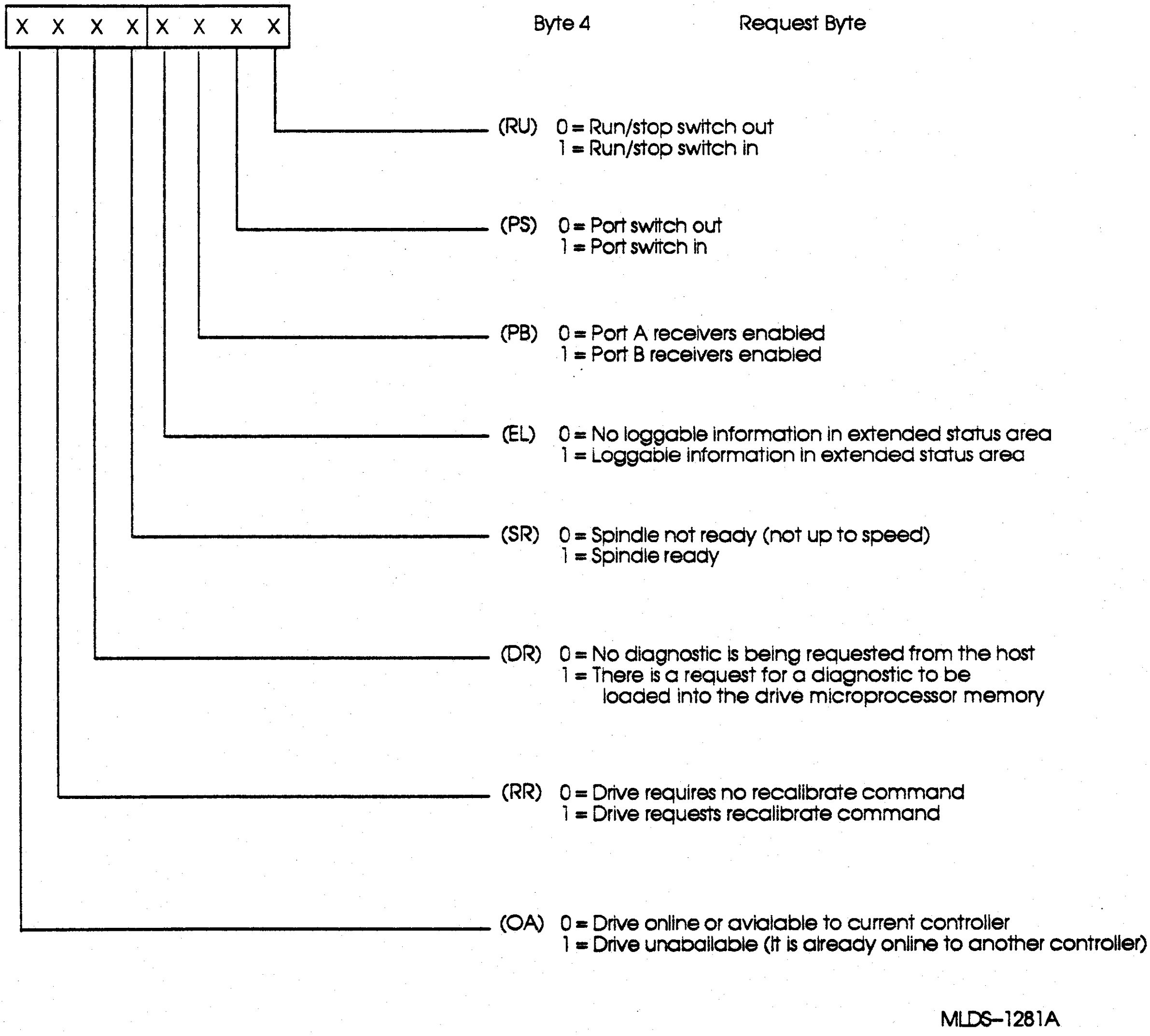


Figure 10–33: RA81 Byte 5 Mode Byte

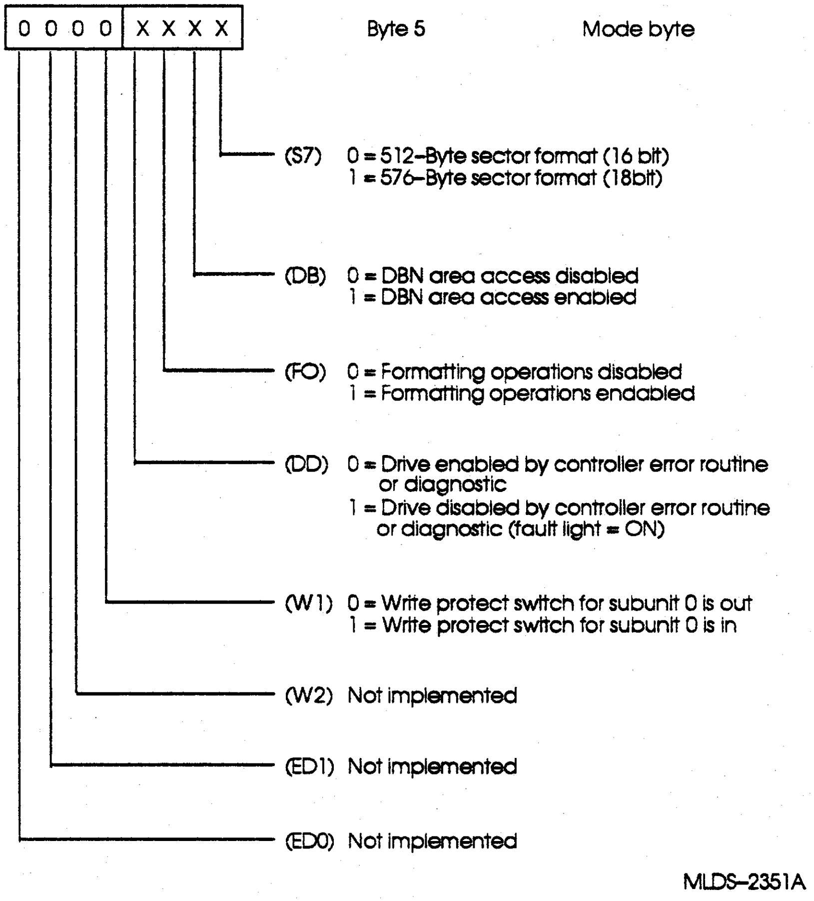


Figure 10–34: RA81 Byte 6 Error Byte

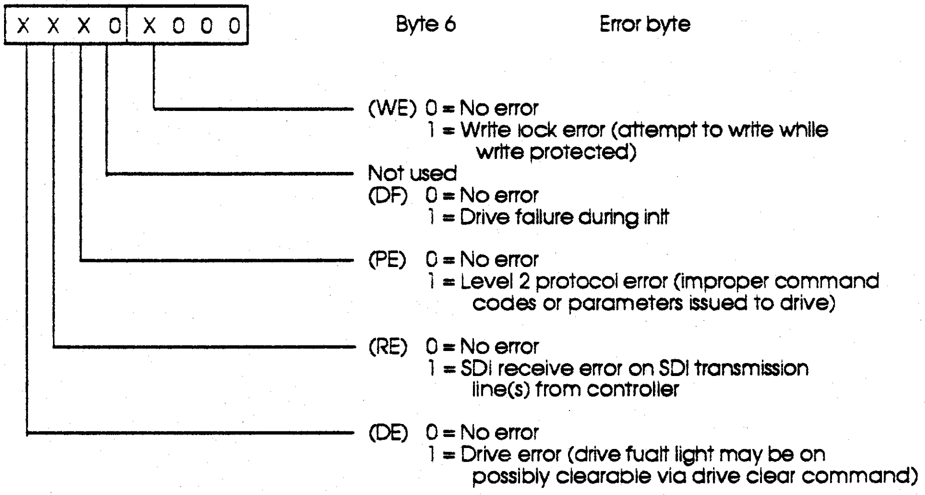
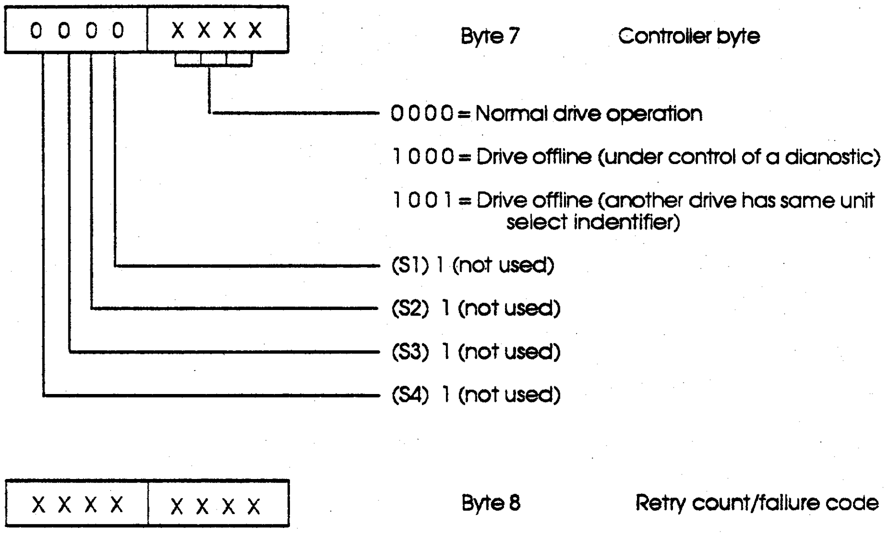


Figure 10–35: RA81 Bytes 7–8 Controller Byte

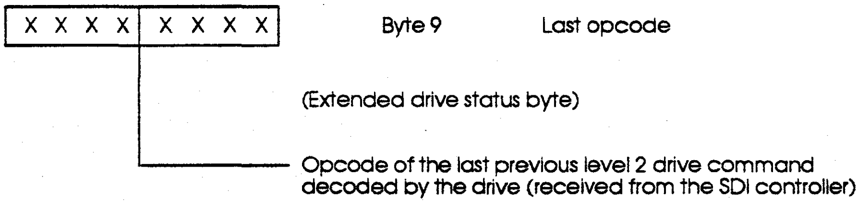


MLDS–1284B

NOTE

Byte 8; Retry count during the last SEEK or RECALIBRATION command (the number of times the command was re–tried before successful completion). Maximum allowed by microcode is 3 before the seek or recalibration will be aborted. A value of 0 indicates 3 retries completed since the counter decrements to 0 during each retry.

Figure 10–36: RA81 Byte 9 Last Opcode



81 = Change mode
82 = Change controller flags
03 = Diagnose
84 = Disconnect (drive)
05 = Drive clear

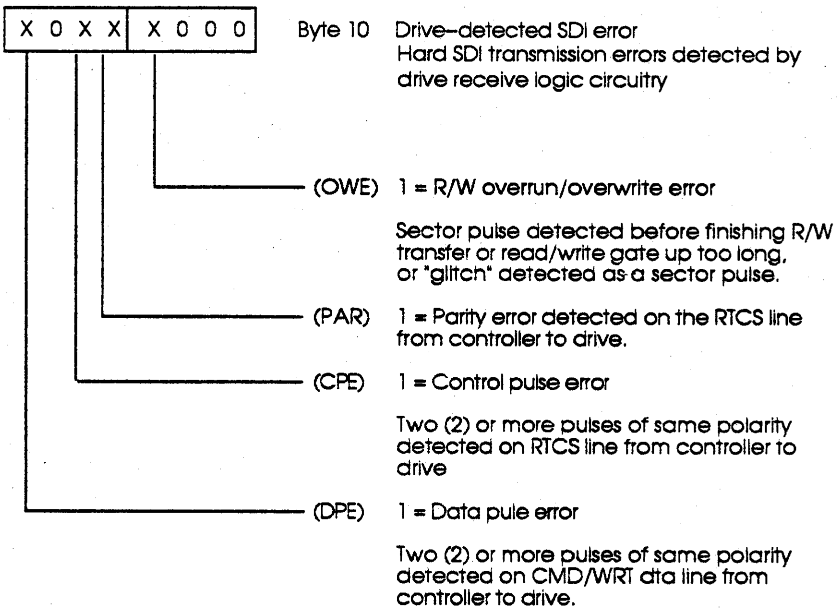
06 = Error recovery
87 = Get common characteristics
88 = Get subunit characteristics
0A = Initiate seek
8B = Online

0C = Run
8D = Read memory
8E = Recalibrate
90 = Topology
0F = Write memory

FF = Select group (level 1 command – processed by firmware
seek head select sub–routines

MLDS–1285B

Figure 10–37: RA81 Byte 10 Drive–Detected SDI Error



Drive–detected SDI error
Hard SDI transmission errors detected by drive receive logic circuitry

(OWE) 1 = R/W overrun/overwrite error

Sector pulse detected before finishing R/W transfer or read/write gate up too long, or "glitch" detected as a sector pulse.

(PAR) 1 = Parity error detected on the RTCS line from controller to drive.

(CPE) 1 = Control pulse error

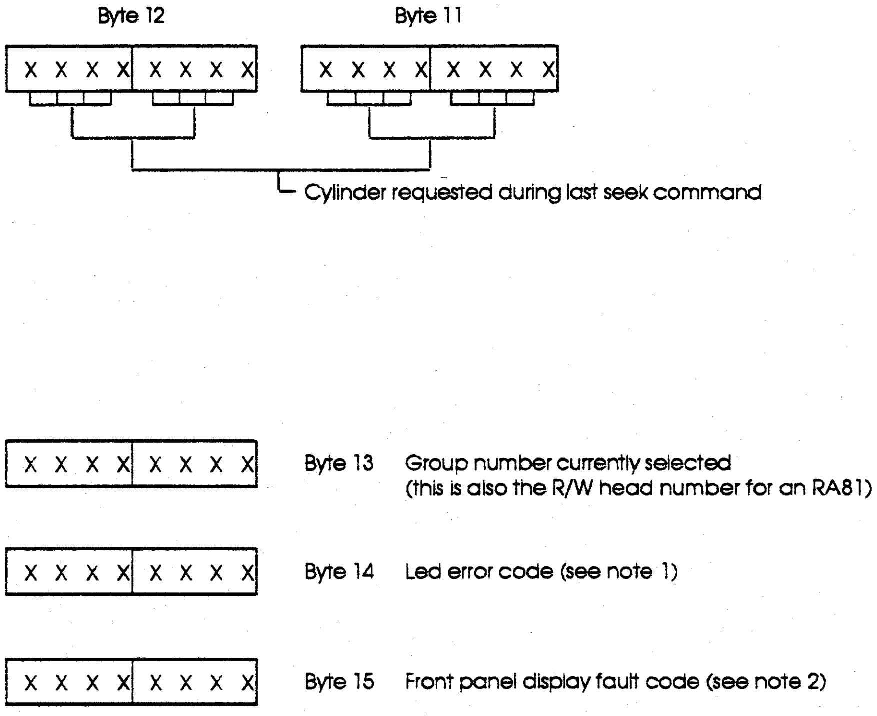
Two (2) or more pulses of same polarity detected on RTCS line from controller to drive

(DPE) 1 = Data pule error

Two (2) or more pulses of same polarity detected on CMD/WRT dta line from controller to drive.

MLDS–2353A

Figure 10–38: RA81 Bytes 11–15



NOTES:

1. Refer to LED error code list in service manual.
2. Refer to control panel fault code list in service manual.

MLDS–2355A

10.5 RA82 DRIVE STATUS DECODE

The following pages describe the decoding of the RA82 status bytes. These bytes are passed from the drive to the controller when a GET STATUS command is issued to the RA82. The RA82 also provides these status bytes to the controller if a command from the controller fails to execute properly within the drive.

System error logs, controller error logs, and diagnostic utilities often display most of the drive status bytes. This decoding information is provided to help you understand the meaning of these bytes and/or the meaning of bits within any of these bytes.

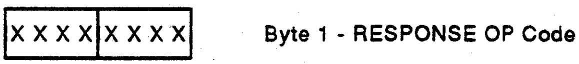
Figure 10–39: RA82 Drive Status Decode

BYTE 01	RESPONSE OPCODE		
BYTE 02	UNIT SELECT (LOWER)		
BYTE 03	UNIT + SUBUNIT MASK		← subunits were for mult HDA's
BYTE 04	REQUEST BYTE		GENERIC DRIVE STATUS BYTE
BYTE 05	MODE BYTE		GENERIC DRIVE STATUS BYTE
BYTE 06	ERROR BYTE		GENERIC DRIVE STATUS BYTE
BYTE 07	CONTROLLER BYTE		
BYTE 08	RETRY COUNT/FAILURE		
BYTE 09	PREVIOUS CMD OPCODE		EXTENDED DRIVE STATUS BYTE
BYTE 10	INTERNAL PORT REG		EXTENDED DRIVE STATUS BYTE
BYTE 11	CYLINDER ADDR (LO)		EXTENDED DRIVE STATUS BYTE
BYTE 12	CYLINDER ADDR (HI)		EXTENDED DRIVE STATUS BYTE
BYTE 13	RECOVERY	GROUP NO.	EXTENDED DRIVE STATUS BYTE
BYTE 14	LED ERROR CODE		EXTENDED DRIVE STATUS BYTE
BYTE 15	F.P. FAULT CODE		EXTENDED DRIVE STATUS BYTE

CXO-1279B

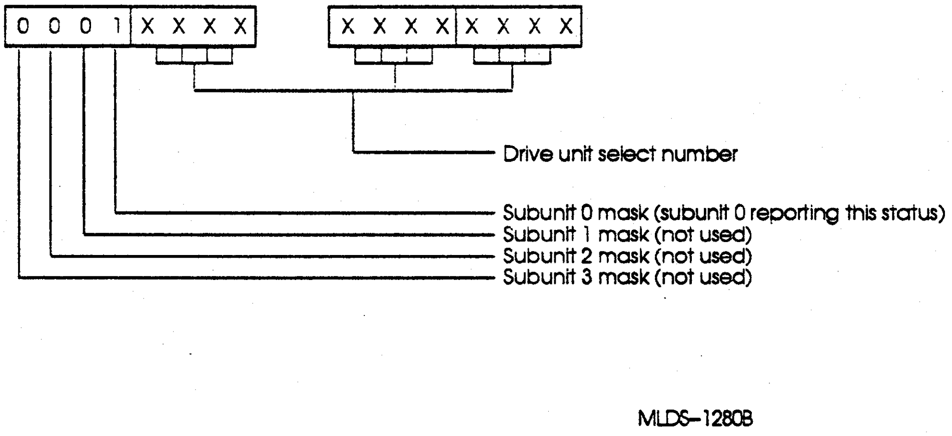
Bytes 2 thru 15 are generally available from system error logs, the HSC console display, and various diagnostics that test RA–series drives. The format in which this information is displayed depends upon the specific type of system (VMS, RSTS, RSX, etc.) or the specific type of controller (HSC50, HSC70, etc.).

Figure 10–40: RA82 Byte 1



This will be the RESPONSE opcode from the drive to the controller, but it is rarely displayed to a user. It indicates the success or non–success of the previous command sent from the controller to the drive.

Figure 10–41: RA82 Bytes 2–3

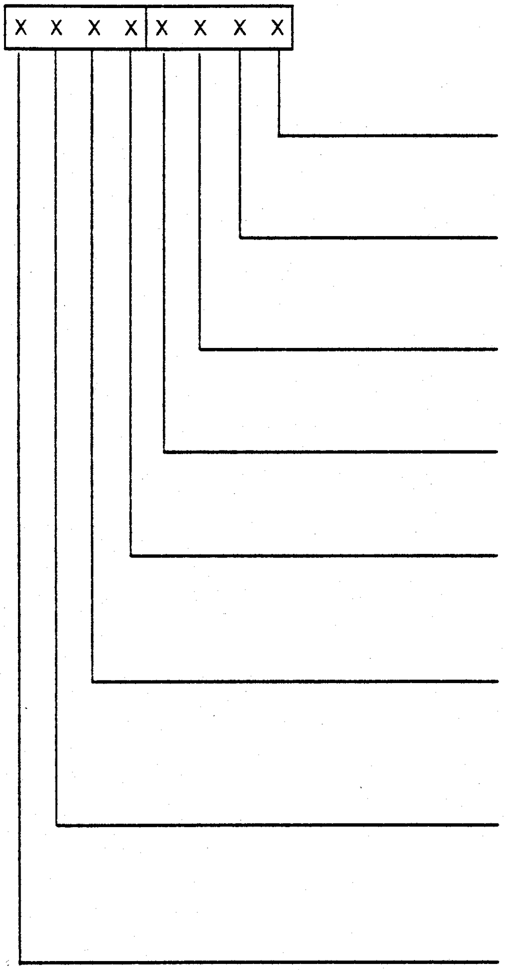


NOTE

The RA82 has no multiple subunits and will always indicate Subunit 0.

Figure 10–42: RA82 Byte 4 Request Byte

11SLG#Z - SDI



Byte 4 Request Byte

(RU) 0 = Run/stop switch out
1 = Run/stop switch in

(PS) 0 = Port switch out
1 = Port switch in

(PB) 0 = Port A receivers enabled
1 = Port B receivers enabled

} which port online

60+80 don't Support

(EL) 0 = No loggable information in extended status area
1 = Loggable information in extended status area

drive has valid Error info Validity bytes 9 thru 15 in status

(SR) 0 = Spindle not ready (not up to speed)
1 = Spindle ready

(DR) 0 = No diagnostic is being requested from the host
1 = There is a request for a diagnostic to be loaded into the drive microprocessor memory

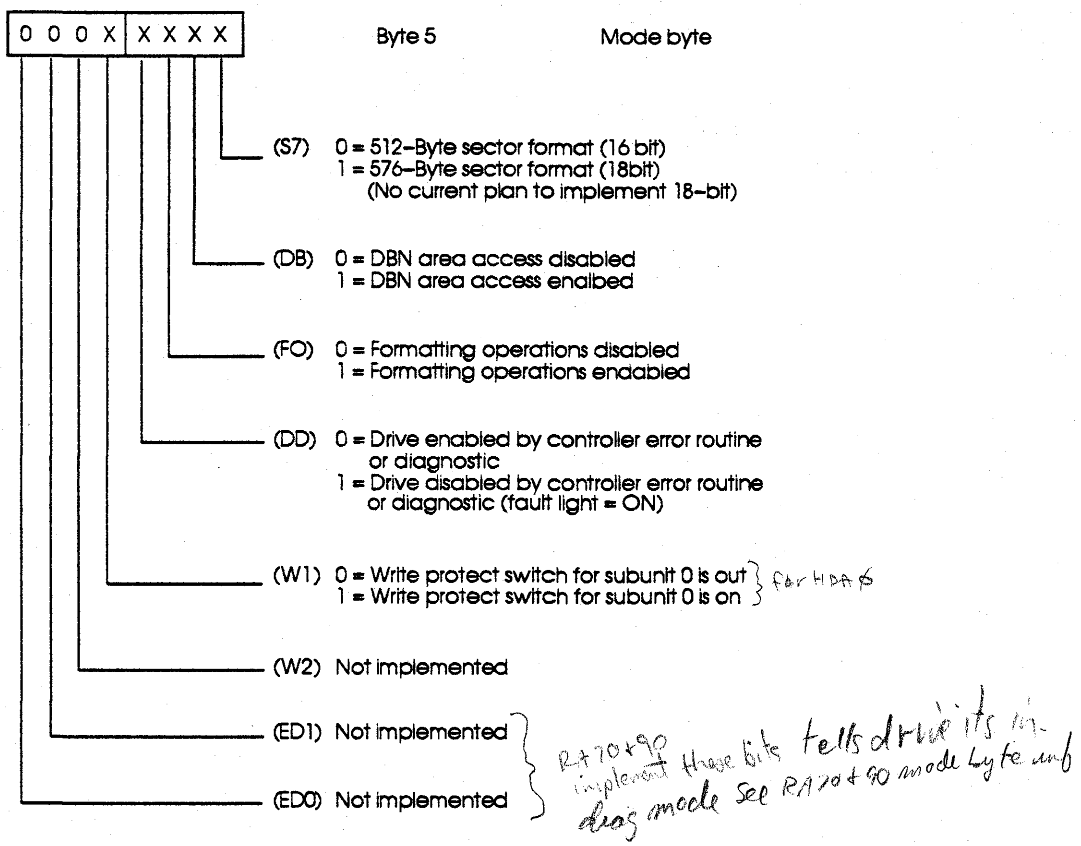
(RR) 0 = Drive requires no recalibrate command
1 = Drive requests recalibrate command

} not currently implemented

(OA) 0 = Drive online or avialable to current controller
1 = Drive unaballable (it is already online to another controller)

MLDS-1281A

Figure 10–43: RA82 Byte 5 Mode Byte



(S7) 0 = 512–Byte sector format (16 bit)
1 = 576–Byte sector format (18bit)
(No current plan to implement 18–bit)

(DB) 0 = DBN area access disabled
1 = DBN area access enalbed

(FO) 0 = Formatting operations disabled
1 = Formatting operations endabled

(DD) 0 = Drive enabled by controller error routine or diagnostic
1 = Drive disabled by controller error routine or diagnostic (fault light = ON)

(W1) 0 = Write protect switch for subunit 0 is out
1 = Write protect switch for subunit 0 is on
for HDA 0

(W2) Not implemented

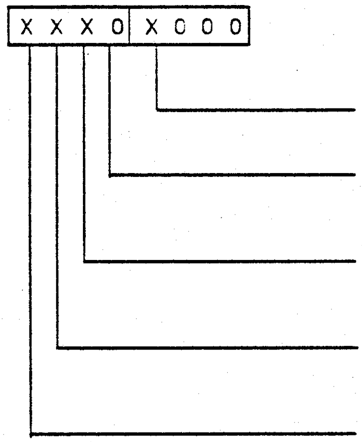
(ED1) Not implemented

(ED0) Not implemented

RA70+90 implement these bits tells drive its in diag mode See RA70+90 mode byte info

MLDS–1282B

Figure 10–44: RA82 Byte 6 Error Byte



(WE) 0 = No error
1 = Write lock error (attempt to write while write protected)

Not used

(DF) 0 = No error
1 = Drive failure during init

(PE) 0 = No error
1 = Level 2 protocol error (improper command codes or parameters issued to drive)

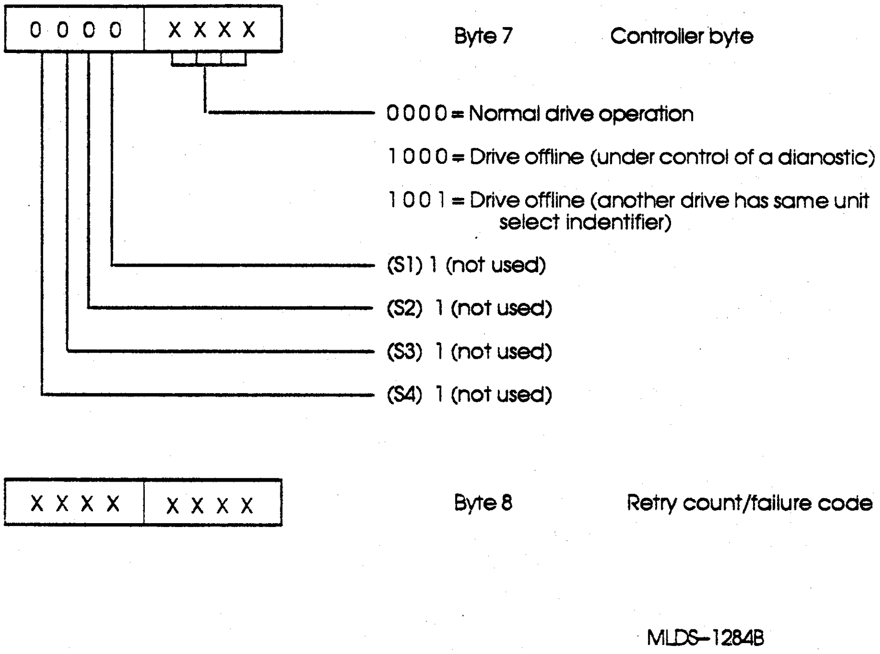
(RE) 0 = No error
1 = SDI receive error on SDI transmission line(s) from controller

CTL SDI transmission cables
SDI Rx Logic etc

(DE) 0 = No error
1 = Drive error (drive fualt light may be on possibly clearable via drive clear command)

MLDS–1283B

Figure 10–45: RA82 Bytes 7–8 Controller Byte

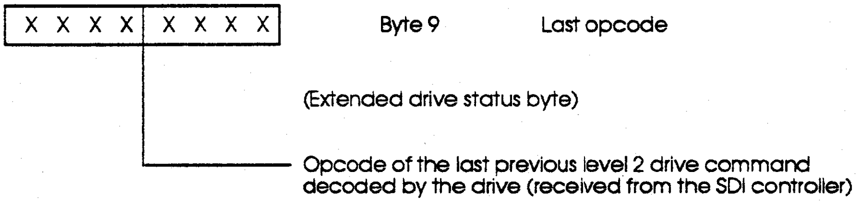


NOTE

Byte 8; Retry count during the last SEEK or RECALIBRATION command.

The number of times the command was re–tried internal to the RA82 in order to attempt successful completion of the SEEK or RECALIBRATE operation.

Figure 10–46: RA82 Byte 9 Last Opcode



81 = Change mode
82 = Change controller flags
03 = Diagnose
84 = Disconnect (drive)
05 = Drive clear

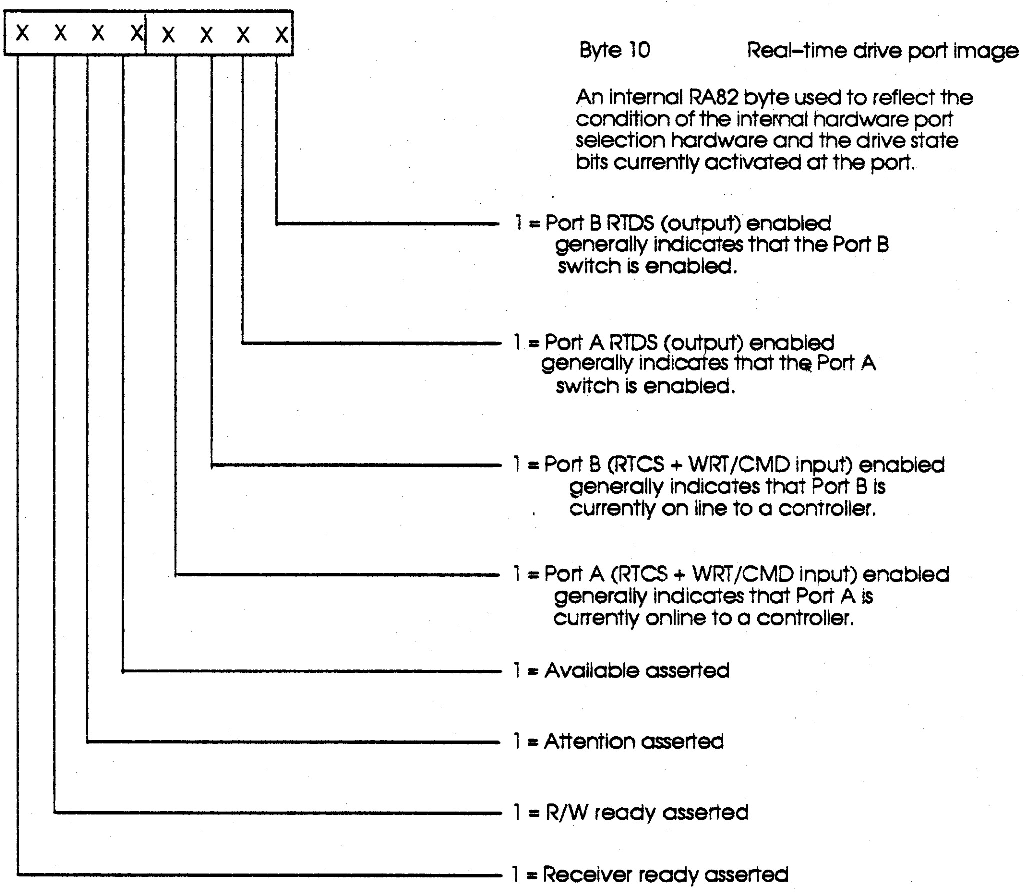
06 = Error recovery
87 = Get common characteristics
88 = Get subunit characteristics
0A = Initiate seek
8B = Online

0C = Run
8D = Read memory
8E = Recalibrate
90 = Topology
0F = Write memory

FF = Select group (level 1 command – processed by firmware seek head select sub–routines

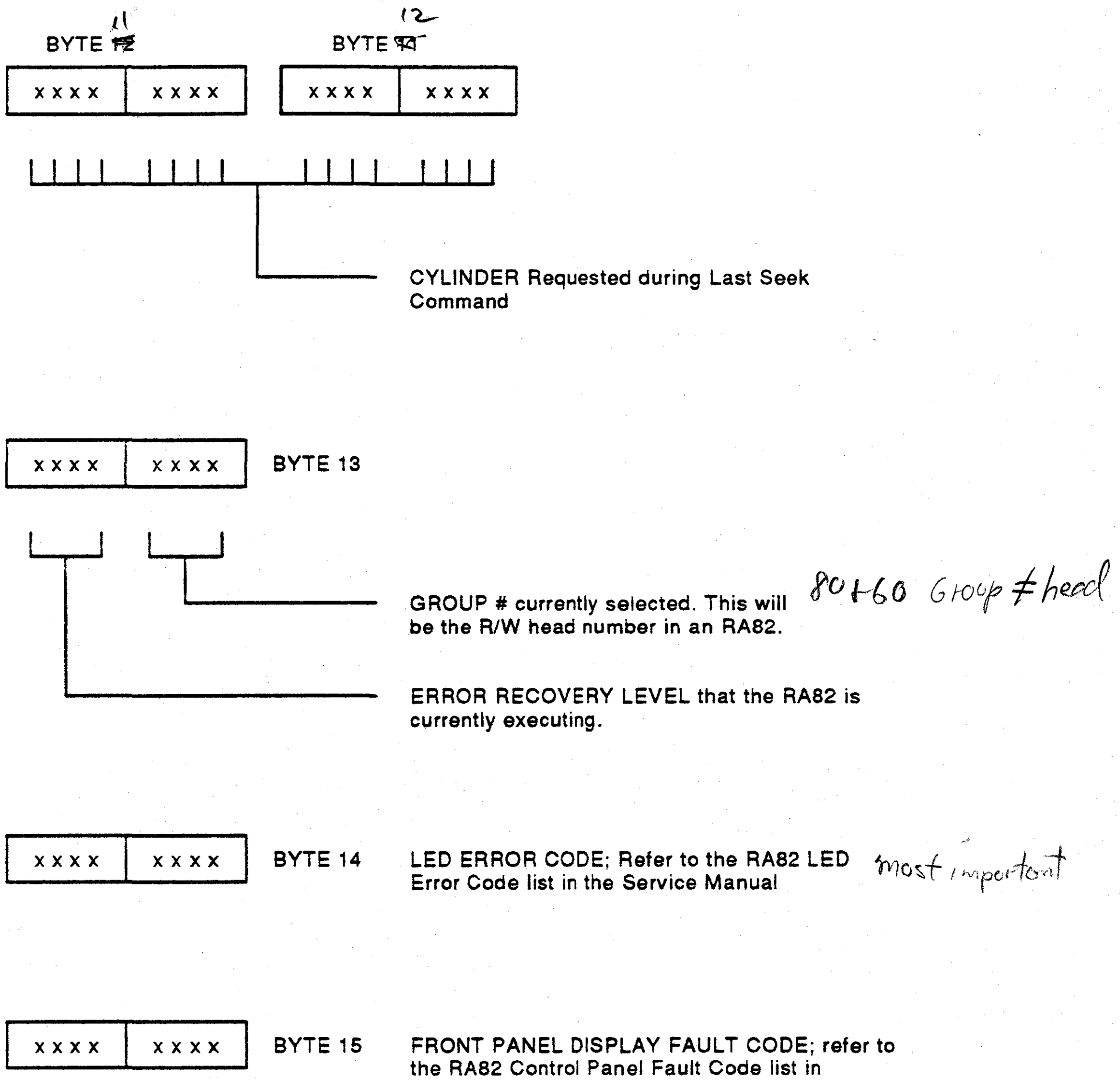
MLDS–1285B

Figure 10–47: RA82 Byte 10 Real–Time Drive Port Image



MLDS–2356A

Figure 10–48: RA82 Bytes 11–15



CYLINDER Requested during Last Seek Command

GROUP # currently selected. This will be the R/W head number in an RA82. 80+60 Group ≠ head

ERROR RECOVERY LEVEL that the RA82 is currently executing.

BYTE 14 LED ERROR CODE; Refer to the RA82 LED Error Code list in the Service Manual most important

BYTE 15 FRONT PANEL DISPLAY FAULT CODE; refer to the RA82 Control Panel Fault Code list in the Service manual.

10.6 RA90 DRIVE STATUS DECODE

The following pages describe the decoding of the RA90 status bytes. These bytes are passed from the drive to the controller when a GET STATUS command is issued to the RA90. The RA90 also provides these status bytes to the controller if a command from the controller fails to execute properly within the drive.

System error logs, controller error logs, and diagnostic utilities often display most of the drive status bytes. This decoding information is provided to help you understand the meaning of these bytes and/or the meaning of bits within any of these bytes.

Figure 10–49: Summary of RA90 Drive Status Codes

Byte 01	Response opcode	
Byte 02	Unit number low byte	
Byte 03	Subunit mask	
Byte 04	Request byte	Generic drive status byte
Byte 05	Mode byte	Generic drive status byte
Byte 06	Error byte	Generic drive status byte
Byte 07	Controller byte	Generic drive status byte
Byte 08	Retry count	
Byte 09	Previous command opcode	Extended drive status
Byte 10	HDA revision bits	Extended drive status
Byte 11	Cylinder address (lo)	Extended drive status
Byte 12	Cylnder address (hi)	Extended drive status
Byte 13	Recovery LVL Group No.	Extended drive status
Byte 14	Error code	Extended drive status
Byte 15	MFG fault code	Extended drive status

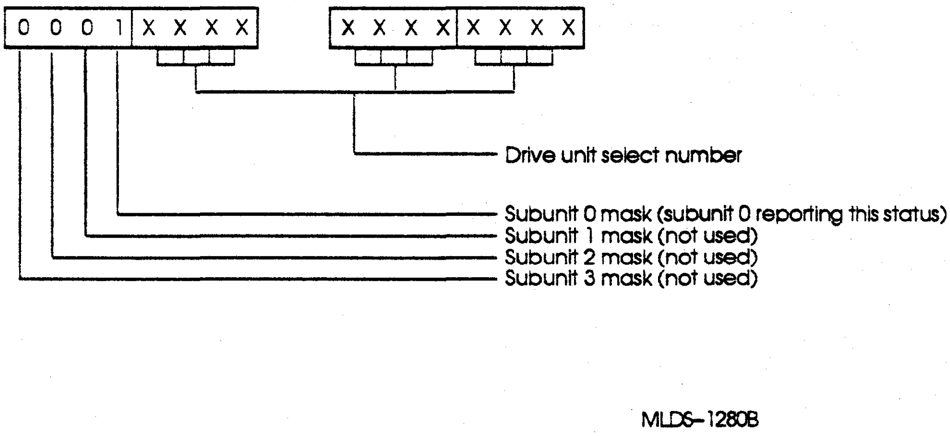
MLDS–0010A

Bytes 2 thru 15 are generally available from system error logs, the HSC console display, and various diagnostics that test RA–series drives. The format in which this information is displayed depends upon the specific type of system (VMS, RSTS, RSX, etc.) or the specific type of controller (HSC50, HSC70, etc.).

Figure 10–50: RA90 Byte 1

This will be the RESPONSE opcode from the drive to the controller, but it is rarely displayed to a user. It indicates the success or non–success of the previous command sent from the controller to the drive.

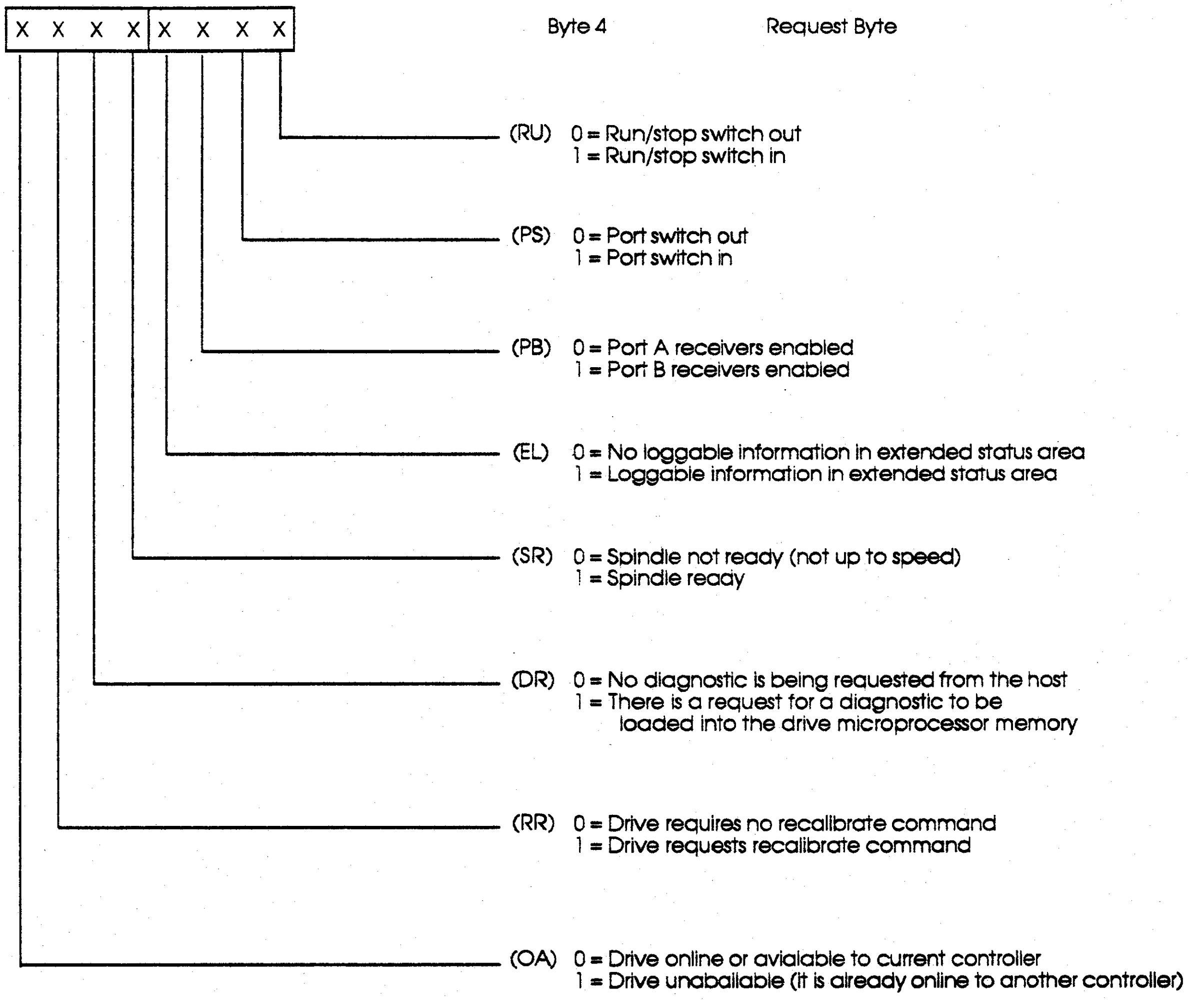
Figure 10–51: RA90 Bytes 2–3



NOTE

The RA90 has no multiple subunits and will always indicate Subunit 0.

Figure 10–52: RA90 Byte 4 Request Byte



Byte 4 Request Byte

(RU) 0 = Run/stop switch out
1 = Run/stop switch in

(PS) 0 = Port switch out
1 = Port switch in

(PB) 0 = Port A receivers enabled
1 = Port B receivers enabled

(EL) 0 = No loggable information in extended status area
1 = Loggable information in extended status area

(SR) 0 = Spindle not ready (not up to speed)
1 = Spindle ready

(DR) 0 = No diagnostic is being requested from the host
1 = There is a request for a diagnostic to be
loaded into the drive microprocessor memory

(RR) 0 = Drive requires no recalibrate command
1 = Drive requests recalibrate command

(OA) 0 = Drive online or avialable to current controller
1 = Drive unabaliable (It is already online to another controller)

MLDS-1281A

Figure 10–53: RA90 Byte 5 Mode Byte

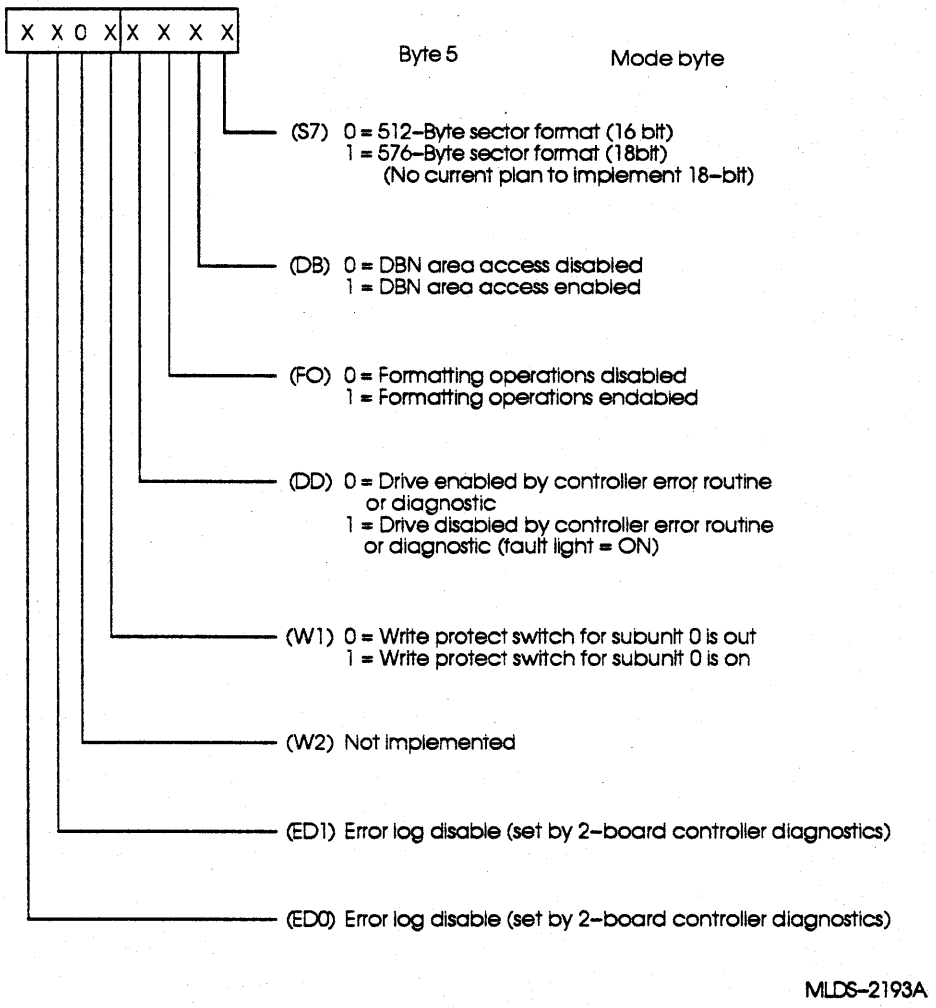


Figure 10–54: RA90 Byte 6 Error Byte

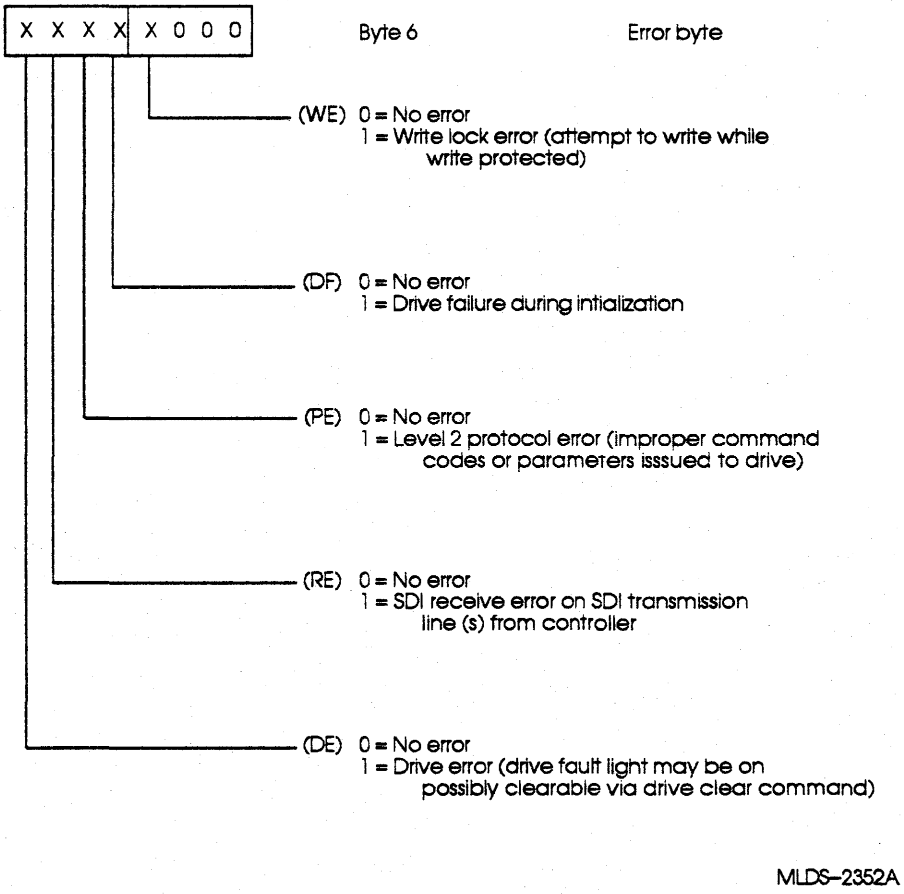
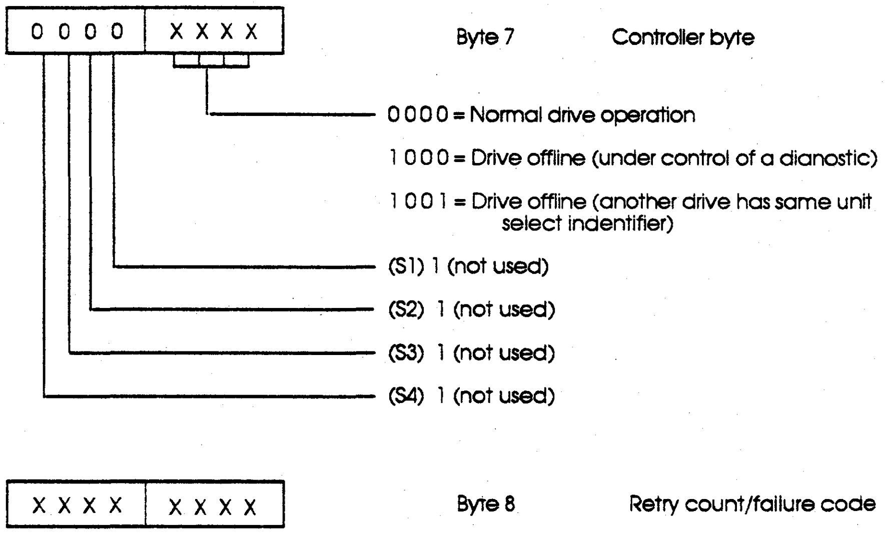


Figure 10–55: RA90 Bytes 7–8 Controller Byte



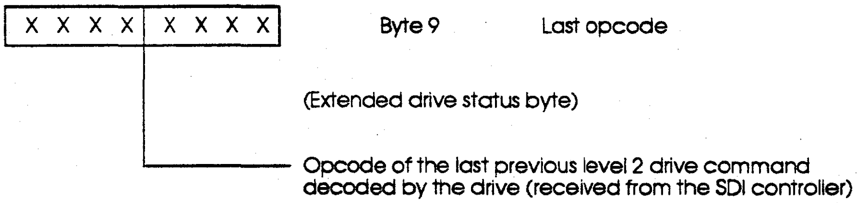
MLDS–1284B

NOTE

Byte 8; Retry count during the last SEEK or RECALIBRATION command.

The number of times the command was re–tried internal to the RA90 in order to attempt successful completion of the SEEK or RECALIBRATE operation.

Figure 10–56: RA90 Byte 9 Last Opcode



81 = Change mode
82 = Change controller flags
03 = Diagnose
84 = Disconnect (drive)
05 = Drive clear

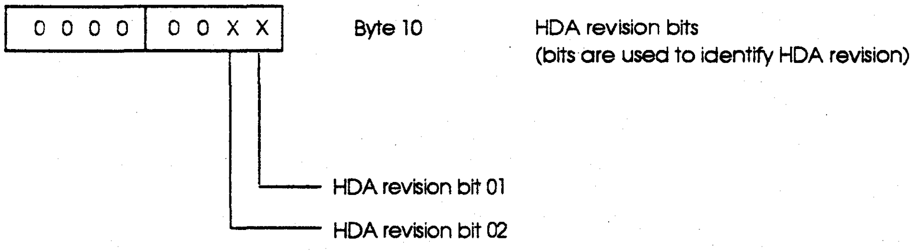
06 = Error recovery
87 = Get common characteristics
88 = Get subunit characteristics
0A = Initiate seek
8B = Online

0C = Run
8D = Read memory
8E = Recalibrate
90 = Topology
0F = Write memory

FF = Select group (level 1 command – processed by firmware
seek head select sub-routines

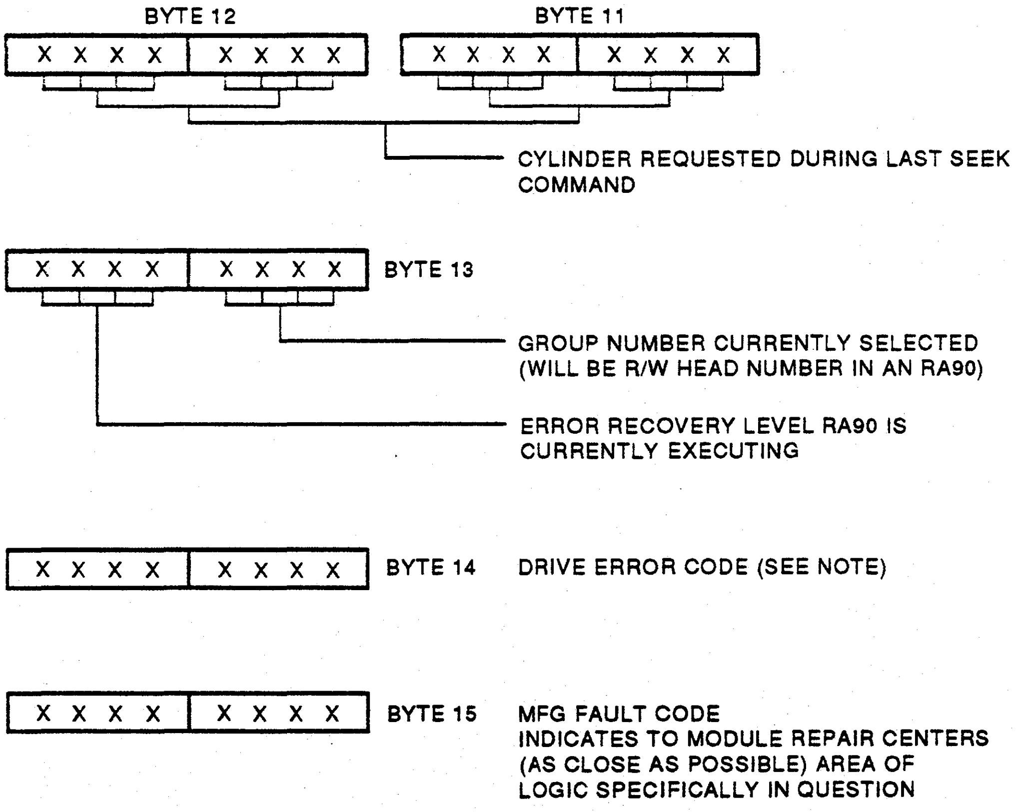
MLDS-1285B

Figure 10–57: RA90 Byte 10 HDA Revision Bits



MLDS–2362A

Figure 10–58: RA90 Bytes 11–15



NOTE: REFER TO RA90 ERROR CODE LIST IN SERVICE MANUAL.

CXO-2362A

Drive Status Decoding

SAMPLE 1

10.7 Status Error Decoding Sample 1

```
ERROR-E Drive detected error at 8-apr-1986 15:11:44.37
        Command Ref #   00000000
        RA82 unit #     66.
        Error Seq #     444.
        Error Flags     40
        Event           00EB  Drive Detected
        Request         1B    EL Bit set  — Very Imbortent next Look@ byte 14+15
        Mode            00
        Error           80  Drive
        Controller      00
        Retry/fail      00
        Extended Status9 0C  RUN Command  CTL → Driv "Spin Up"
                      10 0B  PortA    Bothswitches in
                      11 00 } seek =0  (Default)
                      12 00 }          (Default)
                      13 00 HD 0
                      14 C0 Fail Trk SERVO Timing } Servo Module
                      15 30 servo diag Failed     }
        Requestor #     7.
        Drive port #    0.
ERROR-I End of error.
```

```
V A X / V M S        SYSTEM ERROR REPORT      COMPILED  8-APR-1986 16:41
                                                               PAGE    2.

**************************** ENTRY        3. ****************************
ERROR SEQUENCE 83.                                LOGGED ON SID 01380A4F

ERL$LOGMESSAGE ENTRY   8-APR-1986 15:11:44.37
                       KA780  REV# 7.  SERIAL# 2639.  MFG PLANT 0.

I/O SUB-SYSTEM, UNIT _HSC007$DUA66:

      MESSAGE TYPE            0001
                                      DISK MSCP MESSAGE

      MSLG$L_CMD_REF      00000000
      MSLG$W_UNIT             0042
                                      UNIT #66.

      MSLG$W_SEQ_NUM          01BC
                                      SEQUENCE #444.

      MSLG$B_FORMAT             03
                                      "SDI" ERROR

      MSLG$B_FLAGS              40
                                      OPERATION CONTINUING

      MSLG$W_EVENT            00EB
                                      DRIVE ERROR
                                      DRIVE DETECTED ERROR

      MSLG$Q_CNT_ID       0000F807
                          01010000
                                      UNIQUE IDENTIFIER, 00000000F807
                                      MASS STORAGE CONTROLLER
                                      HSC70

      MSLG$B_CNT_SVR            02
                                      CONTROLLER SOFTWARE VERSION #2.

      MSLG$B_CNT_HVR            00
                                      CONTROLLER HARDWARE REVISION #0.

      MSLG$W_MULT_UNT         0050
      MSLG$Q_UNIT_ID      00000108
                          020B0000
                                      UNIQUE IDENTIFIER, 000000000108
                                      DISK CLASS DEVICE
                                      RA82
```

```
 V A X / V M S        SYSTEM ERROR REPORT      COMPILED  8-APR-1986 16:41
                                                                PAGE   3.

       MSLG$B_UNIT_SVR         01
                                       UNIT SOFTWARE VERSION #1.
       MSLG$B_UNIT_HVR         0F
                                       UNIT HARDWARE REVISION #15.
       MSLG$L_VOL_SER    03C769A2
                                       VOLUME SERIAL #63400354.
       MSLG$L_HEADER     00000000
                                       LBN #0.
                                       GOOD LOGICAL SECTOR

       MSLG$Z_SDI
         REQUEST               1B
                                       RUN/STOP SWITCH IN
                                       PORT SWITCH IN
                                       LOG INFORMATION IN EXTENDED AREA   EL Bit
                                       SPINDLE READY
                                       PORT A RECEIVERS ENABLED
         MODE                  00
                                       512-BYTE SECTOR FORMAT
         ERROR                 80
                                       DRIVE ERROR
         CONTROLLER            00
                                       NORMAL DRIVE OPERATION
         RETRY                 00
                                       0. RETRIES LEFT  not used
CONTROLLER OR DEVICE DEPENDENT INFORMATION
       LED CODE                C0
       PANEL CODE              30
       LAST OPCODE             0C
                                       RUN
       PORT IMAGE              0B
                                       PORT B RTDS ENABLED
                                       PORT A RTDS ENABLED
                                       PORT A ENABLED
```

```
 V A X / V M S        SYSTEM ERROR REPORT      COMPILED  8-APR-1986 16:41
                                                                PAGE   4.
       CUR CYLNDR            0000
                                       CURRENT CYLINDER, #0.
       CUR GROUP               00
                                       CURRENT GROUP, #0.
       REQUESTOR               07
                                       REQUESTOR #7.
       DRIVE PORT              00
                                       DRIVE PORT #0.
```

Drive Status Decoding

SAMPLE 2

10.8 Status Error Decoding Sample 2

```
ERROR-E SI Command Timeout  at 8-apr-1986 15:11:44.37
        Command Ref #   00000000
        RA82 unit #     66.
        Err Seq #       489.
        Error Flags     41
        Event           002B
        Request         13
        Mode            00
        Error           00
        Controller      00
        Retry/fail      00
        Extended Status 0A
                        0B
                        8F
                        05
                        00
                        00
                        00
        Requestor #     7.
        Drive port #    0. ERROR-I End of error.
```

init Seek" seekTimeout"

```
ERROR-E Drive Detected Error at 8-apr-1986 15:11:44.37
        Command Ref #   00000000
        RA82 unit #     66.
        Err Seq #       490.
        Error Flags     40
        Event           00EB
        Request         1B
        Mode            00
        Error           80
        Controller      00
        Retry/fail      00
        Extended Status 0A
                        0B
                        00
                        01
                        01
                        4D
                        28
        Requestor #     7.
        Drive port #    0. ERROR-I End of error.
                        .
```

-(EL Bit) Error info good

Seek

A ONLINE Both 11.

} cyl 1

group 1 = head

Need more data to check for different or Same HD

```
V A X / V M S       SYSTEM ERROR REPORT      COMPILED  8-APR-1986 16:41
                                                                PAGE   2.

*************************** ENTRY        5. ***************************
ERROR SEQUENCE 257.                                  LOGGED ON SID 01380A4F

ERL$LOGMESSAGE ENTRY   8-APR-1986 15:11:44.37
                       KA780  REV# 7.  SERIAL# 2639.  MFG PLANT 0.

I/O SUB-SYSTEM, UNIT _HSC007$DUA66:

      MESSAGE TYPE        0001
                                   DISK MSCP MESSAGE

      MSLG$L_CMD_REF  00000000
      MSLG$W_UNIT         0042
                                   UNIT #66.

      MSLG$W_SEQ_NUM      01E9
                                   SEQUENCE #489.

      MSLG$B_FORMAT         03
                                   "SDI" ERROR

      MSLG$B_FLAGS          41
                                   SEQUENCE NUMBER RESET
                                   OPERATION CONTINUING

      MSLG$W_EVENT        002B
                                   DRIVE ERROR
                                   DRIVE COMMAND TIMEOUT

      MSLG$Q_CNT_ID   0000F807
                      01010000
                                   UNIQUE IDENTIFIER, 00000000F807
                                   MASS STORAGE CONTROLLER
                                   HSC70

      MSLG$B_CNT_SVR        02
                                   CONTROLLER SOFTWARE VERSION #2.

      MSLG$B_CNT_HVR        00
                                   CONTROLLER HARDWARE REVISION #0.

      MSLG$W_MULT_UNT     0050
      MSLG$Q_UNIT_ID  00000108
                      020B0000
                                   UNIQUE IDENTIFIER, 000000000108
                                   DISK CLASS DEVICE
                                   RA82
```

```
V A X / V M S        SYSTEM ERROR REPORT      COMPILED  8-APR-1986 16:41
                                                               PAGE   3.
       MSLG$B_UNIT_SVR        01
                                         UNIT SOFTWARE VERSION #1.
       MSLG$B_UNIT_HVR        0F
                                         UNIT HARDWARE REVISION #15.
       MSLG$L_VOL_SER   03C769A2
                                         VOLUME SERIAL #63400354.
       MSLG$L_HEADER    00000000
                                         LBN #0.
                                         GOOD LOGICAL SECTOR

       MSLG$Z_SDI
         REQUEST              13
                                         RUN/STOP SWITCH IN
                                         PORT SWITCH IN
                                         SPINDLE READY
                                         PORT A RECEIVERS ENABLED
         MODE                 00
                                         512-BYTE SECTOR FORMAT
         ERROR                00
         CONTROLLER           00
                                         NORMAL DRIVE OPERATION
         RETRY                00
                                         0. RETRIES LEFT
 DEVICE DEPENDENT INFORMATION
       LONGWORD 1.      058F0B0A
                                         /..../
       LONGWORD 2.      07000000
                                         /..../
       LONGWORD 3.      00000000
                                         /..../
       LONGWORD 4.      00000000
                                         /..../
```

not decoded because EL Bit
not set VMS doesn't
care

```
V A X / V M S        SYSTEM ERROR REPORT      COMPILED  8-APR-1986 16:41
                                                              PAGE   4.

**************************** ENTRY       6. ****************************
ERROR SEQUENCE 258.                                 LOGGED ON SID 01380A4F

ERL$LOGMESSAGE ENTRY   8-APR-1986 15:11:44.37
                       KA780  REV# 7.  SERIAL# 2639.  MFG PLANT 0.

I/O SUB-SYSTEM, UNIT _HSC007$DUA66:

      MESSAGE TYPE         0001
                                        DISK MSCP MESSAGE
      MSLG$L_CMD_REF   00000000
      MSLG$W_UNIT          0042
                                        UNIT #66.
      MSLG$W_SEQ_NUM       01EA
                                        SEQUENCE #490.
      MSLG$B_FORMAT          03
                                        "SDI" ERROR
      MSLG$B_FLAGS           40
                                        OPERATION CONTINUING
      MSLG$W_EVENT         00EB
                                        DRIVE ERROR
                                        DRIVE DETECTED ERROR

      MSLG$Q_CNT_ID    0000F807

                       01010000
                                        UNIQUE IDENTIFIER, 00000000F807
                                        MASS STORAGE CONTROLLER
                                        HSC70
      MSLG$B_CNT_SVR         02
                                        CONTROLLER SOFTWARE VERSION #2.
      MSLG$B_CNT_HVR         00
                                        CONTROLLER HARDWARE REVISION #0.
      MSLG$W_MULT_UNT      0050
      MSLG$Q_UNIT_ID   00000108
                       020B0000
                                        UNIQUE IDENTIFIER, 000000000108
                                        DISK CLASS DEVICE
                                        RA82
```

Errors associated with a particular command Look for Same # here

```
 V A X / V M S        SYSTEM ERROR REPORT      COMPILED  8-APR-1986 16:41
                                                                PAGE    5.
       MSLG$B_UNIT_SVR       01
                                        UNIT SOFTWARE VERSION #1.
       MSLG$B_UNIT_HVR       0F
                                        UNIT HARDWARE REVISION #15.
       MSLG$L_VOL_SER  03C769A2
                                        VOLUME SERIAL #63400354.
       MSLG$L_HEADER   00000000
                                        LBN #0.
                                        GOOD LOGICAL SECTOR

       MSLG$Z_SDI
         REQUEST             1B
                                        RUN/STOP SWITCH IN
                                        PORT SWITCH IN
                                        LOG INFORMATION IN EXTENDED AREA
                                        SPINDLE READY
                                        PORT A RECEIVERS ENABLED
         MODE                00
                                        512-BYTE SECTOR FORMAT
         ERROR               80
                                        DRIVE ERROR
         CONTROLLER          00
                                        NORMAL DRIVE OPERATION
         RETRY               00
                                        0. RETRIES LEFT
CONTROLLER OR DEVICE DEPENDENT INFORMATION
       LED CODE              4D
       PANEL CODE            28
       LAST OPCODE           0A
                                        INITIATE SEEK
       PORT IMAGE            0B
                                        PORT B RTDS ENABLED
                                        PORT A RTDS ENABLED
                                        PORT A ENABLED
```

```
 V A X / V M S        SYSTEM ERROR REPORT      COMPILED  8-APR-1986 16:41
                                                                PAGE    6.
       CUR CYLNDR          0001
                                        CURRENT CYLINDER, #1.
       CUR GROUP             01
                                        CURRENT GROUP, #1.
       REQUESTOR             07
                                        REQUESTOR #7.
       DRIVE PORT            00
                                        DRIVE PORT #0.

                                .
                                .
                                .
```

Drive Status Decoding

SAMPLE 3

10.9 Status Error Decoding Sample 3

```
ERROR-E Drive detected error at 8-apr-1986 15:11:44.37
        Command Ref #    00000000
        RA82 unit #      66.
        Err Seq #        1.
        Error Flags      41
        Event            00EB
        Request          1B    EL
        Mode             00
        Error            40    SDI RX Error ? maybe not in drive
        Controller       00
        Retry/fail       00
        Extended Status  0A    seek
                         0B    A selected A+B in
                        ⇒00
                         65    CyL C5 Hex
                         0B    HD 13                          Controller TX Logic   K.SDI
                         4F                                   Cables
                         0C                                   Bulkheads
        Requestor #      7.                                   Hybrid RX Logic
        Drive port #     0.
ERROR-I End of error.
```

```
V A X / V M S       SYSTEM ERROR REPORT      COMPILED  8-APR-1986 16:41
                                                              PAGE   2.

**************************** ENTRY       3. ****************************
ERROR SEQUENCE 690.                                LOGGED ON SID 01380A4F

ERL$LOGMESSAGE ENTRY   8-APR-1986 15:11:44.37
                       KA780  REV# 7.  SERIAL# 2639.  MFG PLANT 0.

I/O SUB-SYSTEM, UNIT _HSC007$DUA66:

      MESSAGE TYPE         0001
                                    DISK MSCP MESSAGE

      MSLG$L_CMD_REF   00000000 —
      MSLG$W_UNIT          0042
                                    UNIT #66.

      MSLG$W_SEQ_NUM       0001
                                    SEQUENCE #1.

      MSLG$B_FORMAT          03
                                    "SDI" ERROR

      MSLG$B_FLAGS           41
                                    SEQUENCE NUMBER RESET
                                    OPERATION CONTINUING

      MSLG$W_EVENT         00EB
                                    DRIVE ERROR
                                    DRIVE DETECTED ERROR

      MSLG$Q_CNT_ID    0000F807
                       01010000
                                    UNIQUE IDENTIFIER, 00000000F807
                                    MASS STORAGE CONTROLLER
                                    HSC70

      MSLG$B_CNT_SVR         02
                                    CONTROLLER SOFTWARE VERSION #2.

      MSLG$B_CNT_HVR         00
                                    CONTROLLER HARDWARE REVISION #0.

      MSLG$W_MULT_UNT      0050
      MSLG$Q_UNIT_ID   00000108
                       020B0000
                                    UNIQUE IDENTIFIER, 000000000108
                                    DISK CLASS DEVICE
                                    RA82
```

```
 V A X / V M S        SYSTEM ERROR REPORT      COMPILED  8-APR-1986 16:41
                                                                PAGE   3.
       MSLG$B_UNIT_SVR       E5
                                       UNIT SOFTWARE VERSION #229.
       MSLG$B_UNIT_HVR       0F
                                       UNIT HARDWARE REVISION #15.
       MSLG$L_VOL_SER  03C769A2
                                       VOLUME SERIAL #63400354.
       MSLG$L_HEADER   00000000
                                       LBN #0.
                                       GOOD LOGICAL SECTOR

       MSLG$Z_SDI
         REQUEST             1B
                                       RUN/STOP SWITCH IN
                                       PORT SWITCH IN
                                       LOG INFORMATION IN EXTENDED AREA
                                       SPINDLE READY
                                       PORT A RECEIVERS ENABLED
         MODE                00
                                       512-BYTE SECTOR FORMAT
         ERROR               40
                                       SDI RECEIVE ERROR
         CONTROLLER          00
                                       NORMAL DRIVE OPERATION
         RETRY               00
                                       0. RETRIES LEFT
CONTROLLER OR DEVICE DEPENDENT INFORMATION
       LED CODE              4F
       PANEL CODE            0C
       LAST OPCODE           0A
                                       INITIATE SEEK
       PORT IMAGE            0B
                                       PORT B RTDS ENABLED
                                       PORT A RTDS ENABLED
                                       PORT A ENABLED
```

```
 V A X / V M S        SYSTEM ERROR REPORT      COMPILED  8-APR-1986 16:41
                                                                PAGE   4.
       CUR CYLNDR          0065
                                       CURRENT CYLINDER, #101.
       CUR GROUP             0B
                                       CURRENT GROUP, #11.
       REQUESTOR             07
                                       REQUESTOR #7.
       DRIVE PORT            00
                                       DRIVE PORT #0.
```

Drive Status Decoding

SAMPLE 4

10.10 Status Error Decoding Sample 4

event Log
Communications event

```
 ERROR-E SI Receiver Ready Collision at 8-apr-1986 15:11:44.37
        Command Ref #   00000000
        RA82 unit #     66.
        Err Seq #       430.
        Error Flags     41
        Event           01AB
        Request         13
        Mode            00
        Error           00
        Controller      00
        Retry/fail      00
        Extended Status 90
                        0B
                        C7
                        02
                        07
                        00
                        00
        Requestor #     7.
        Drive port #    0.
ERROR-I End of error.
```

only real info here -
This was a bad cable
"Crosstalk"

```
ERROR-E SI Receiver Ready Collision at 8-apr-1986 15:11:44.37
        Command Ref #   00000000
        RA82 unit #     66.
        Err Seq #       431.
        Error Flags     40
        Event           01AB
        Request         13
        Mode            00
        Error           00
        Controller      00
        Retry/fail      00
        Extended Status 90
                        0B
                        C7
                        02
                        07
                        00
                        00
        Requestor #     7.
        Drive port #    0.
ERROR-I End of error.
```

```
DISK-E Seq. 8. at 8-apr-1986 15:12:01.77
       Unrecoverable error on disk unit 66. Drive appears inoperative.
       intervention required.

ERROR-E SI Receiver Ready Collision at 8-apr-1986 15:12:01.77
        Command Ref #    00000000
        RA82 unit #      66.
        Err Seq #        432.
        Error Flags      00
        Event            01AB
        Request          13
        Mode             00
        Error            00
        Controller       00
        Retry/fail       00
        Extended Status  90
                         0B
                         C7
                         02
                         07
                         00
                         00
        Requestor #      7.
        Drive port #     0.
ERROR-I End of error.

ERROR-E Position or Unintelligible Header Error at 8-apr-1986 15:12:01.77
        Command Ref #    130E0006
        RA82 unit #      66.
        Err Seq #        433.
        Error Flags      00
        Event            006B
        Recovery level   7.
        Recovery count   0.
        LBN              608485.
        Orig err flags   014000
        Recovery flags   000002
        Lvl A retry cnt  3.
        Lvl B retry cnt  0.
        Buffer addr      141706
        Source Req.      7.
        Detecting Req.   7.
ERROR-I End of error
```

```
V A X / V M S       SYSTEM ERROR REPORT      COMPILED  8-APR-1986 16:41
                                                              PAGE   2.

**************************** ENTRY       6. ****************************
ERROR SEQUENCE 26.                                 LOGGED ON SID 01380A4F

ERL$LOGMESSAGE ENTRY   8-APR-1986 15:11:44.37
                       KA780  REV# 7.  SERIAL# 2639.  MFG PLANT 0.

I/O SUB-SYSTEM, UNIT _HSC007$DUA66:

      MESSAGE TYPE        0001
                                   DISK MSCP MESSAGE

      MSLG$L_CMD_REF  00000000
      MSLG$W_UNIT         0042
                                   UNIT #66.

      MSLG$W_SEQ_NUM      01AE
                                   SEQUENCE #430.

      MSLG$B_FORMAT         03
                                   "SDI" ERROR

      MSLG$B_FLAGS          41
                                   SEQUENCE NUMBER RESET
                                   OPERATION CONTINUING

      MSLG$W_EVENT        01AB
                                   DRIVE ERROR
                                   RECEIVER READY COLLISION

      MSLG$Q_CNT_ID   0000F807
                      01010000
                                   UNIQUE IDENTIFIER, 00000000F807
                                   MASS STORAGE CONTROLLER
                                   HSC70

      MSLG$B_CNT_SVR        02
                                   CONTROLLER SOFTWARE VERSION #2.

      MSLG$B_CNT_HVR        00
                                   CONTROLLER HARDWARE REVISION #0.

      MSLG$W_MULT_UNT     0050
      MSLG$Q_UNIT_ID  00000108
                      020B0000
                                   UNIQUE IDENTIFIER, 000000000108
                                   DISK CLASS DEVICE
                                   RA82
```

```
V A X / V M S        SYSTEM ERROR REPORT      COMPILED  8-APR-1986 16:41
                                                                PAGE   3.

      MSLG$B_UNIT_SVR        01
                                         UNIT SOFTWARE VERSION #1.

      MSLG$B_UNIT_HVR        0F
                                         UNIT HARDWARE REVISION #15.

      MSLG$L_VOL_SER   03C769A2
                                         VOLUME SERIAL #63400354.

      MSLG$L_HEADER    00000000
                                         LBN #0.
                                         GOOD LOGICAL SECTOR

      MSLG$Z_SDI
        REQUEST              13
                                         RUN/STOP SWITCH IN
                                         PORT SWITCH IN
                                         SPINDLE READY
                                         PORT A RECEIVERS ENABLED
        MODE                 00
                                         512-BYTE SECTOR FORMAT

        ERROR                00
        CONTROLLER           00
                                         NORMAL DRIVE OPERATION

        RETRY                00
                                         0. RETRIES LEFT

DEVICE DEPENDENT INFORMATION
      LONGWORD 1.      02C70B90
                                         /..../

      LONGWORD 2.      07000007
                                         /..../

      LONGWORD 3.      00000000
                                         /..../

      LONGWORD 4.      00000000
                                         /..../
```

```
V A X / V M S       SYSTEM ERROR REPORT      COMPILED  8-APR-1986 16:41
                                                             PAGE   4.

**************************** ENTRY       7. ****************************
ERROR SEQUENCE 27.                               LOGGED ON SID 01380A4F

ERL$LOGMESSAGE ENTRY   8-APR-1986 15:11:44.37
                       KA780  REV# 7.  SERIAL# 2639.  MFG PLANT 0.

I/O SUB-SYSTEM, UNIT _HSC007$DUA66:

      MESSAGE TYPE        0001
                                       DISK MSCP MESSAGE

      MSLG$L_CMD_REF  00000000
      MSLG$W_UNIT         0042
                                       UNIT #66.

      MSLG$W_SEQ_NUM      01AF
                                       SEQUENCE #431.

      MSLG$B_FORMAT         03
                                       "SDI" ERROR

      MSLG$B_FLAGS          40
                                       OPERATION CONTINUING

      MSLG$W_EVENT        01AB
                                       DRIVE ERROR
                                       RECEIVER READY COLLISION

      MSLG$Q_CNT_ID   0000F807
                      01010000
                                       UNIQUE IDENTIFIER, 00000000F807
                                       MASS STORAGE CONTROLLER
                                       HSC70

      MSLG$B_CNT_SVR        02
                                       CONTROLLER SOFTWARE VERSION #2.

      MSLG$B_CNT_HVR        00
                                       CONTROLLER HARDWARE REVISION #0.

      MSLG$W_MULT_UNT     0050
      MSLG$Q_UNIT_ID  00000108
                      020B0000
                                       UNIQUE IDENTIFIER, 000000000108
                                       DISK CLASS DEVICE

                                       RA82
```

```
V A X / V M S        SYSTEM ERROR REPORT      COMPILED  8-APR-1986 16:41
                                                                 PAGE   5.
      MSLG$B_UNIT_SVR         01
                                          UNIT SOFTWARE VERSION #1.
      MSLG$B_UNIT_HVR         0F
                                          UNIT HARDWARE REVISION #15.
      MSLG$L_VOL_SER    03C769A2
                                          VOLUME SERIAL #63400354.
      MSLG$L_HEADER     00000000
                                          LBN #0.
                                          GOOD LOGICAL SECTOR

      MSLG$Z_SDI
        REQUEST               13
                                          RUN/STOP SWITCH IN
                                          PORT SWITCH IN
                                          SPINDLE READY
                                          PORT A RECEIVERS ENABLED
        MODE                  00
                                          512-BYTE SECTOR FORMAT
        ERROR                 00
        CONTROLLER            00
                                          NORMAL DRIVE OPERATION
        RETRY                 00
                                          0. RETRIES LEFT
DEVICE DEPENDENT INFORMATION
      LONGWORD 1.       02C70B90
                                          /..../
      LONGWORD 2.       07000007
                                          /..../
      LONGWORD 3.       00000000
                                          /..../
      LONGWORD 4.       00000000
                                          /..../
************************************************************************
      1. INTERVENING RECORD(S) WILL BE PRINTED AT INPUT FILE "<EOF>"
```

70-19074-01 Loopback 2
82 Kit Should
have them
70-18340-02 Bulkhead Assy

VMS V4.4 ERROR LOG ENTRY FORMATTER

Problem with RA Disks on UDA/KDA/KDB50 Controllers

10.11 VMS V4.4 ERROR LOG ENTRY FORMATTER – Problem with RA Disks on UDA/KDA/KDB50 Controllers

10.11.1 Drive-Detected Errors

A problem was introduced in ANALYZE/ERROR_LOG in VMS V4.4. It prevents bytes 9 through 15 of the extended status for RA-series disk drives connected to UDA, KDA, and KDB controllers from being displayed in the output file. An example of a faulty VMS error log report output is illustrated at the end of this chapter.

Unfortunately, the extended status data contains the DISCRETE FAULT CODE which is used to determine the repair action for drive detected errors. This problem will be corrected in VMS V4.5.

In the interim, use the following procedure to view the extended status data:

The error log information resides in a binary file ERRLOG.SYS which resides in the SYS$ERRORLOG directory on the system disk. (If the file was renamed, use the new file name.) The extended status information can be decoded by using the VMS DUMP utility and the service manual for the appropriate RA-drive type.

10.11.2 How to use the Dump Utility

STEP 1:

```
$ ANAL/ERR/INCLUDE=$5$DUA0 SYS$ERRORLOG:ERRLOG.SYS
```

In this step we will retrieve all 5DUA0 entries from the ERRLOG.SYS file and display them on the terminal. While inspecting the error log entries for fault indicators, you should note the 'ENTRY number value' that resides in the first line output for each individual entry. You will need this number if there are any SDI errors that have a drive-detected error event code. (See sample ANALYZE/ERROR_LOG output.)

For example purposes, assume entries 16 through 21 were drive-detected errors that require inspection of the extended status data.

STEP 2:

```
$ DUMP/RECORD=(START:16,END:21) SYS$ERRORLOG:ERRLOG.SYS
```

In this step we will dump the contents of SYS$ERRORLOG:ERRLOG.SYS in a long word format as indicated in the following example. In this case, entries 16 through 21 were drive-detected errors which were indicated by the output of STEP 1.

```
                   OUTPUT OF STEP 2
     (edited for 80 column, DUMP output is 132 column)

Dump of file SYS$SYSROOT:[SYSERR]ERRLOG.SYS;1 on  9-JUL-1986 16:34:30.00
File ID (27633,4,0)   End of file block 8 / Allocated 9

Record number 16 (00000010), 94 (005E) bytes

 4F415544 244E4946 46554D0A 00001E01 0448008F 19D6CF26 18E00064 01380A4F
 07DC0002 00040106 008141C3 4D4400EB 41030000 00000000 00000001 492D5453
     0000 0100010B 81000000 04130000 000003D3 A10A0000 01010205 00000000

Record number 17 (00000011), 94 (005E) bytes

 4D415544 244E4946 46554D0A 00001E01 0449008F 19D6CF26 18E00064 01380A4F
 07DC0002 00040106 008141C3 4D4400EB 41030000 00000000 00000001 49464655
 1425 0900300B 0A000080 040B0000 000003D3 A10A0000 01010205 00000000

     .
     .
     .

Record number 21 (00000015), 94 (005E) bytes

 48415544 244E4946 46554D0A 00001E01 044D008F 19D6DD58 BDA00064 01380A4F
 07DC0002 00040106 008141C3 4D4400EB 41030000 00000000 00000001 30304353
     0000 0900303B 05000000 000B0000 000003D3 A10A0000 01010205 00000000
```

STEP 3:

In this step we will retrieve the DISCRETE FAULT CODE from the output of the DUMP utility.

The DISCRETE FAULT CODE is in different places for the RA8x and RA60 disks:

```
o RA80/RA81/RA82  = BYTE 14
o RA60  = BYTE 15.
```

The output of a DUMPed record is read from right to left (the starting byte for record number 17 contains the value 4F). The extended status data that is not displayed in the output of ANALYZE/ERROR_LOG starts at byte 88 as indicated by the dashed line in the example below (byte 88 of the error log record contains byte 9 of the extended status data). As the example below is for an RA81 (see ANALYZE/ERROR_LOG output), the discrete fault code is Byte 14 which contains a 25 (hexadecimal). At this point we can go to the *RA81 Service Manual* to translate the fault code.

```
NOTE that the values in the DUMPed record are HEXADECIMAL.             ||
                                                                       ||
Record number 17 (00000011), 94 (005E) bytes                           ||
                                                                       vv
 4D415544 244E4946 46554D0A 00001E01 0449008F 19D6CF26 18E00064 01380A4F
 07DC0002 00040106 008141C3 4D4400EB 41030000 00000000 00000001 49464655
     1425 0900300B 0A000080 040B0000 000003D3 A10A0000 01010205 00000000
     ---- -------- -------- ----
     |  |  | | | |  |
Byte 15 |  | | | |  |  Byte 15 = RA8x Front Panel Display Fault Code
        |  | | | |  |          = RA60 Master Error Fault Code
Byte 14-   | | | |  |  Byte 14 = RA8x Led Error Code
           | | | |  |          = RA60 Current Head
Byte 13----  | | |  |  Byte 13 = RA81/82 SEE NOTE, RA80 Group Number
             | | |  |              = RA60 High Byte of Current Cylinder
Byte 12------  | |  |  Byte 12 = RA8x High Byte of Cylinder for last Seek
               | |  |          = RA60 Low Byte of Current Cylinder
Byte 11--------  |  |  Byte 11 = RA8x Low Byte of Cylinder for last Seek
                 |  |          = RA60 Previous Head
Byte 10----------   |  Byte 10 = RA8x Drive Detected SDI Error
                    |          = RA60 High Byte of Previous Cylinder
Byte  9------------    Byte  9 = RA8x Last Opcode
                               = RA60 Low Byte of Previous Cylinder
```

NOTE

RA81/RA82 low nibble is equal to head. RA82 only high nibble is equal to SDI RECOVERY LEVEL used at time of this error.

10.11.3 Summary

- Problem is on RA disk drives on UDA/KDA/KDB controllers and the entries that are drive–detected errors only.
- It requires a three–step process to get the missing bytes.
- The missing bytes are bytes 9 through 15 of Extended Status.
- Refer to RA80/RA81/RA82/RA60 service manuals for descriptions of EXTENDED STATUS information.
- There are HELP files for DUMP and ANALYZE utilities under VMS to help you further understand the three–step process.
- An example of an ANALYZE/ERROR_LOG output for ENTRY 17 is found on the next page.

10.11.4 Example of an ANALYZE/ERROR_LOG Output for Entry 17

```
*************************** ENTRY      17. ****************************
ERROR SEQUENCE 1097.                                LOGGED ON SID 01380A4F
ERL$LOGMESSAGE ENTRY   8-JUL-1986 14:20:05.87
                       KA780  REV# 7.  SERIAL# 2639.  MFG PLANT 0.

I/O SUB-SYSTEM, UNIT _MUFFIN$DUA0:

      MESSAGE TYPE        0001
                                  DISK MSCP MESSAGE

      MSLG$L_CMD_REF  00000000
      MSLG$W_UNIT         0000
                                  UNIT #0.

      MSLG$W_SEQ_NUM      0000
                                  SEQUENCE #0.

      MSLG$B_FORMAT         03
                                  "SDI" ERROR

      MSLG$B_FLAGS          41
                                  SEQUENCE NUMBER RESET
                                  OPERATION CONTINUING

      MSLG$W_EVENT        00EB
                                  DRIVE ERROR
                                  DRIVE DETECTED ERROR

      MSLG$Q_CNT_ID   41C34D44
                      01060081
                                  UNIQUE IDENTIFIER, 008141C34D44
                                  MASS STORAGE CONTROLLER
                                  UDA50A

      MSLG$B_CNT_SVR        04
                                  CONTROLLER SOFTWARE VERSION #4.

      MSLG$B_CNT_HVR        00
                                  CONTROLLER HARDWARE REVISION #0.

      MSLG$W_MULT_UNT     0002
      MSLG$Q_UNIT_ID  000007DC
                      02050000
                                  UNIQUE IDENTIFIER, 0000000007DC
                                  DISK CLASS DEVICE
                                  RA81

      MSLG$B_UNIT_SVR       01
                                  UNIT SOFTWARE VERSION #1.

      MSLG$B_UNIT_HVR       01
                                  UNIT HARDWARE REVISION #1.

      MSLG$L_VOL_SER  03D3A10A
                                  VOLUME SERIAL #64200970.

      MSLG$L_HEADER   00000000
                                  LBN #0.
                                  GOOD LOGICAL SECTOR



                  (CONTINUED .....)
```

```
MSLG$Z_SDI
  REQUEST             0B
                               RUN/STOP SWITCH IN
                               PORT SWITCH IN
                               LOG INFORMATION IN EXTENDED AREA
                               PORT A RECEIVERS ENABLED
  MODE                04
                               FORMAT OPERATIONS ENABLED
                               512-BYTE SECTOR FORMAT
  ERROR               80

                               DRIVE ERROR
  CONTROLLER          00
                               NORMAL DRIVE OPERATION
  RETRY               00
                               0. RETRIES LEFT

  xxxxxxxxxx          xx       xxxxxxxxxxxxxxxxxxxxxxxxxxxxx
  xxxxxxxxxx          xx       xxxxxxxxxxxxxxxxxxxxxxxxxxxxx
  xxxxxxxxxx          xx       xxxxxxxxxxxxxxxxxxxxxxxxxxxxx
  xxxxxxxxxx          xx       xxxxxxxxxxxxxxxxxxxxxxxxxxxxx
  xxxxxxxxxx          xx       xxxxxxxxxxxxxxxxxxxxxxxxxxxxx
  xx                                                    xx
  xx   THIS IS WHERE THE EXTENDED DRIVE STATUS BYTES    xx
  xx          SHOULD HAVE BEEN DISPLAYED.               xx
  xx                                                    xx
  xxxxxxxxxx          xx       xxxxxxxxxxxxxxxxxxxxxxxxxxxxx
  xxxxxxxxxx          xx       xxxxxxxxxxxxxxxxxxxxxxxxxxxxx
  xxxxxxxxxx          xx       xxxxxxxxxxxxxxxxxxxxxxxxxxxxx
  xxxxxxxxxx          xx       xxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

CHAPTER 11

VAXSIMPLUS

11.1 VAXsimPLUS OVERVIEW

SYMPTOM DIRECTED DIAGNOSIS

- **ARTIFICIAL INTELLIGENCE (AI) SERVICE TECHNOLOGY**

- **SDD TOOLS KIT:**
  - SPEAR Basic
  - RFTS
  - VAXsim–PLUS

- **DEC PROPRIETARY – NOT FOR SALE**

VAXSIMPLUS

- **THE PLUS FEATURE**
  - Predictive Analysis
  - Automatic Disk Substitution

- **BENEFITS TO THE CUSTOMER**
  - Enhanced Data Integrity
  - Higher System Availability
  - Perceived Higher Reliability
  - Problem Notification

- **BENEFITS TO FIELD SERVICE**
  - Automatic FRU Analysis
  - Automatic Symptom Directed Diagnosis
  - Formatted Evidence Data

11.2 VAXsimPLUS PHONE NUMBERS

VAXSIMPLUS Service Delivery

VAXsimPLUS notifies customers of impending problems and supplies a DIGITAL phone number.

PL01 CSC/AT (Atlanta)	1-800-241-2546
PL31 CSC/CX (Colorado)	1-800-525-6570

The CSC will log the call, event code (VAXsim Theory Number), etc.

The CSC will diagnose the problem and take appropriate actions.

11.3 VAXsimPLUS RESOURCES

RESOURCES

Part Number	Component	Qty
EY-7687-P0-001	VAXsimPLUS Training Course	
QLX07-RW	Entire SDD TOOLS KIT:	
AA-KN79A-TE	Getting Started with VAXsimPLUS	25
AA-KN80A-TE	VAXsimPLUS User's Guide	25
AB-KN81A-TE	SDD Tools Kit Installation Guide	25
AA-KN82A-TE	VAXsimPLUS Field Service Guide	5
AA-J917B-RE	VAX SPEAR Manual	5
AA-J917B-R1	VAX SPEAR Manual Update #1	5
AV-M381B-RE	VAX SPEAR Reference Card	5
AV-P012A-TK	Guide to Measuring Up Time	25
AV-KQ93A-TE	Field Service Tools Cover Letter	1
AV-KQ94A-TE	FS SDD Tools T+C Amendment, Part 1	25
AV-KV74A-TE	FS SDD Tools T+C Amendment, Part 2	25
99-07862-01	Binder	1
AV-KY93A-TE	VAXsimPLUS Spine Insert	25
AQ-KQ91A-RE	FS SDD Tool V1.0 B in TK50	1
BB-KQ92A-RE	FS SDD Tool V1.0 B in 16MT9	1

The distribution of the SDD TOOL KIT will be automatic to unit managers on the Control Diagnostics Distribution List.

VAXsimPLUS MESSAGE TYPES

- **MEDIA Error Messages**

- **SDI Error Messages**

- **DRIVE-DETECTED (Non-Media) Error Messages**

11.4 VAXsimPLUS MESSAGE EXAMPLES

VAXsimPLUS Message Examples

The following pages contain examples of messages generated by VAXsimPLUS. These messages were generated as a result of disk subsystem errors detected by VAXsimPLUS.

VAXsimPLUS RA Disk Notification Message

```
VAXsimPLUS has detected that the following device needs attention:

          $3$DUA161 (RA70 S/N:18CB)    EVENT CODE: [xx.xx.xx.xx]

    * Autocopy was not started
      -- (The autocopy switch is turned off)

There were 11 total media related events for this drive.

    Event Type          Number
    ----------------    ------
    Soft                    11
    Hard                     0

Suggested recovery procedure (A):

  1. Start appropriate backup or copy procedures for your site.

  2. Notify Digital Field Service (include event code in service call info).
     Field service phone: 1-800-224-1900
```

VAXsimPLUS RA Disk Notification Message

```
VAXsimPLUS has detected that the following device needs attention:

          $3$DUA50 (RA81 S/N:2C95E)   EVENT CODE: [xx.xx.xx.xx]

    * Autocopy was not started
      -- (There are too many hard errors)

There were 19 total media related events for this drive.

    Event Type          Number
    ----------------    ------
    Soft                    17
    Hard                     2

Suggested recovery procedure (B):

  1. Notify Digital Field Service (include event code in service call info).
     Field service phone: 1-800-224-1900

  2. Continued use of this drive may result in more hard errors
     occurring.  Take this into account in determining if you wish
     to  continue  using  this drive or start a backup or copy
     operation.
```

VAXsimPLUS RA Disk Notification Message

```
VAXsimPLUS has detected that the following device needs attention:

          $6$DUA62 (RA81 S/N:95E01)   EVENT CODE: [xx.xx.xx.xx]

    * Autocopy was not started
      -- (There are too many hard errors)

There were 22 total media related events for this drive.

    Event Type          Number
    ----------------    ------
    Soft                    13
    Hard                     9

Suggested recovery procedure (C):

  1. This  message is to notify you that one or more hard errors
     have  been  detected  and the errors did not fall into one of
     the  failure  modes.  You may want to determine what file the
     hard  error(s)  occurred  on  and  do the appropriate action
     based on that.

     Field service phone: 1-800-224-1900
```

VAXsimPLUS RA Disk Notification Message

```
VAXsimPLUS has detected that the following device needs attention:

          $3$DUA151 (RA70 S/N:17CA)    EVENT CODE: [xx.xx.xx.xx]

    * Autocopy was not started
      -- (The autocopy switch is turned off)

There were 47 total non-media events for this drive.

Suggested recovery procedure (D):

  1. Notify Digital Field Service (include event code in service call info).
     Field service phone: 1-800-224-1900
```

VAXsimPLUS RA Disk Notification Message

```
VAXsimPLUS has detected that the following device needs attention:

          $3$DUA151 (RA70 S/N:17CA)   EVENT CODE: [xx.xx.xx.xx]

    * Autocopy was  started

There were 47 total non-media events for this drive.

Suggested recovery procedure (E):

  1. Notify Digital Field Service (include event code in service call info).
     Field service phone: 1-800-224-1900

  2. Do a VAXsim/DISMOUNT after autocopy is finished.
```

```
From:   FELIX::SYSTEM
To:     NODE::NORMAN
Subj:   BUSY::RA81 S/N:F4AC

Attn:       Field Service

Device:     FATCAT$DUA4  (RA81 S/N:F4AC)
            MIDCAT$DUA4

Theory:     [xx.xx.xx.xx]

Evidence (All results are in decimal except LED Code):
Total errors on drive: 17

          Sector                Volume
     Phys. From  Soft  Hard     Serial
Head Cyl.  Index Count Count    Number          Error Type (Led Code is in hex)
---- ----- ----- ----- ----- ---------- ---------------------------------------
   8   255     3     1     0      20728 Lost Read/Write Ready
   8   255 <unk>     1     0      20728 LED 39, Write and Off Track
   8   257 <unk>     1     0      20728 LED 39, Write and Off Track
   8   310 <unk>     1     0      20728 LED 39, Write and Off Track
   8   338 <unk>     1     0      20728 LED 39, Write and Off Track
   8   461     4     1     0      20728 Lost Read/Write Ready
   8   461 <unk>     1     0      20728 LED 39, Write and Off Track
   8   462     4     1     0      20728 Lost Read/Write Ready
   8   462 <unk>     1     0      20728 LED 39, Write and Off Track
   8   621 <unk>     1     0      20728 LED 39, Write and Off Track
   8   885 <unk>     1     0      20728 LED 39, Write and Off Track
   8  1009 <unk>     4     0      20728 LED 39, Write and Off Track
   8  1083 <unk>     1     0      20728 LED 39, Write and Off Track
   9  1133 <unk>     1     0      20728 LED 39, Write and Off Track
-------------------------------------------------------------------------------
```

2 theory codes

Definitly a bad head
Possibly a bad head Do more analysis
with RAUTIL + more disk activity

```
From:   GRAMPS::SYSTEM        3-NOV-1987 21:48
To:     DETMAC::NICHOLS
Subj:   GRAMPS::GRAMPS$DUA2 analysis

Attn:       Field Service
Device:     GRAMPS$DUA2 (RA81 S/N:A352)
Theory:     [xx.xx.xx.xx]  This would be a head

Evidence (All results are in decimal except LED Code):
Total errors on drive: 10

           Sector              Volume
     Phys. From  Soft  Hard    Serial
Head Cyl.  Index Count Count   Number         Error Type (Led Code is in hex)
---- ----- ----- ----- ----- ---------- ----------------------------------------
   1   760    3     1     0     116949 ECC Error
   2   451   11     1     0     116949 ECC Error
   2   451   22     1     0     116949 ECC Error
   2   629   34     1     0     116949 ECC Error
   2   637   31     1     0     116949 ECC Error
   2   662    9     1     0     116949 ECC Error
   2   665    5     1     0     116949 ECC Error
   2   749   21     1     0     116949 ECC Error
   2   760   29     1     0     116949 ECC Error
   2   761   44     1     0     116949 ECC Error

Time:  1-NOV 23:45:07  TO   3-NOV 23:38:47   Span:  47:54:39
```

```
From:   DTHSTR::SYSTEM       17-AUG-1987 20:34
To:     NODE::NORMAN
Subj:   DTHSTR::$3$DUA9,LUKE$DUS9 analysis

Attn:       Field Service
Device:     LUKE$DUA9 (RA82 S/N:22C3)
            LUKE$DUS9

Theory:     [xx.xx.xx.xx]

NOTE:       There were 1 hard errors recorded for this device.
            A hard error is defined as an error on a block in which BBR
            was invoked and the data was replaced with 'Force Error'.

Evidence (All results are in decimal except LED Code):
Total errors on drive: 1

          Sector              Volume
     Phys. From  Soft  Hard   Serial
Head Cyl.  Index Count Count  Number        Error Type (Led Code is in hex)
---- ----- ----- ----- ----- ---------- ---------------------------------------
  11   432     3     0     1  72109769 Positioning Error

Time of Error: 11-AUG 09:29:32
```

EVENT 6B

Do more I/O to this disk
Anal/disk Readcheck

```
From:    DTHSTR::SYSTEM        17-NOV-1987 23:07
To:      NODE::JEFFRY
Subj:    DIMILO::$3$DUA50 analysis

Attn:       Field Service
Device:     OBIWAN$DUA50 (RA81 S/N:2C95E)-

Theory:     [xx.xx.xx.xx]

NOTE:       There were 1 hard errors recorded for this device.
            A hard error is defined as an error on a block in which BBR
            was invoked and the data was replaced with 'Force Error'.

Evidence (All results are in decimal except LED Code):
Total errors on drive: 5

            Sector           Volume
     Phys.  From  Soft  Hard   Serial
Head Cyl.   Index Count Count  Number        Error Type (Led Code is in hex)
---- -----  ----- ----- ----- ----------- --------------------------------------
   0   299    47      0     1     142538 ECC Error
   0   302    18      1     0     142538 Lost Read/Write Ready
   0   302  <unk>     1     0     142538 LED 39, Write and Off Track
   0   306  <unk>     1     0     142538 LED 39, Write and Off Track
   0   314    18      1     0     142538 ECC Error

Time: 17-NOV 20:49:41  TO  17-NOV 22:59:28   Span:   2:10:47
```

} probably same commo

```
From:   DTHSTR::SYSTEM        4-NOV-1987 04:45
To:     NODE::MSTONE
Subj:   GUITAR::$3$DUA191 analysis

Attn:       Field Service
Device:     C3P0$DUA191 (RA82 S/N:57A2)

Theory:     [xx.xx.xx.xx]

Evidence:   18 SDI Communication Errors

              Event-
       Count  Status     Translation
       -----  ------     -----------
          4.  (  4B) Controller Detected Transmission Errors
         14.  ( 10B) Controller Detected Pulse or State Parity Errors

    Since there were SDI communication errors which could have caused media
    related  transfer  errors to occur, the following is a summary of those
    media errors:

Evidence (All results are in decimal):
Total media errors on drive: 2

           Sector              Volume
           From  Soft  Hard    Serial
Head  Cyl. Index Count Count   Number         Error Type (Led Code is in hex)
---- ----- ----- ----- ----- ---------- ----------------------------------------
  3      3    6      1     0    101859 ECC Error
  4     23    0      1     0    101859 ECC Error

Time:  4-NOV 01:45:11  TO   4-NOV 04:22:13   Span:    2:37:01
```

-Protocal error fix com first

```
From:    PICKUP::SYSTEM        22-JUL-1987 14:00
To:      NODE::NORMAN
Subj:    PICKUP::$1$DUA15,$1$DUS52 analysis

Attn:        Field Service
Device:      HSC015$DUA15 (RA80 S/N:163A)
             HSC015$DUS52

Theory:      [xx.xx.xx.xx]

NOTE:        There were 2 hard errors recorded for this device.
             A hard error is defined as an error on a block in which BBR
             was invoked and the data was replaced with 'Force Error'.

Evidence (All results are in decimal except LED Code):
Total errors on drive: 18

            Sector              Volume
      Phys. From  Soft  Hard    Serial
Head  Cyl.  Index Count Count   Number        Error Type (Led Code is in hex)
---- ------ ----- ----- ----- ----------- ---------------------------------------
   0   252    30     1     0       17758 Lost Read/Write Ready
   0   510    20     0     1       17758 ECC Error
   0   510    21     1     0       17758 Lost Read/Write Ready
   0   510    24     1     0       17758 Positioning Error
   3     3    23     1     0       17758 Lost Read/Write Ready
   4   250    28     1     0       17758 ECC Error
   5   252    12     1     0       17758 Lost Read/Write Ready
   6   530    26     1     0       17758 Lost Read/Write Ready
   6   530    29     1     0       17758 Positioning Error
   9   540    12     1     0       17758 Lost Read/Write Ready
   9   540    14     1     0       17758 Positioning Error
  10   517    11     0     1       17758 ECC Error
  10   517    12     1     0       17758 Lost Read/Write Ready
  10   517    20     1     0       17758 Positioning Error
  11     2     7     1     0       17758 Lost Read/Write Ready
  11     2    25     1     0       17758 ECC Error
  11    71    20     1     0       17758 Lost Read/Write Ready
  11    71    26     1     0       17758 Lost Read/Write Ready

Time: 22-JUL 13:43:16  TO  22-JUL 14:01:51   Span:   0:19:34
```

DATA Path
Spindle gnd
R/W
Servo
Belts
Cables
P.S.
Isolate by Stimulating I/o
Fix HDWR & Reformat.
Replace BLK

```
From:    GRAMPS::SYSTEM        5-NOV-1987 10:14
To:      FREDH
Subj:    GRAMPS::GRAMPS$DUA2 analysis

Attn:       Field Service
Device:     GRAMPS$DUA2 (RA81 S/N:A352)

Theory:     [xx.xx.xx.xx]

NOTE:       There were 3 hard errors recorded for this device.
            A hard error is defined as an error on a block in which BBR
            was invoked and the data was replaced with 'Force Error'.

Evidence (All results are in decimal except LED Code):
Total errors on drive: 728

           Sector             Volume
     Phys. From  Soft  Hard   Serial
Head Cyl.  Index Count Count  Number         Error Type (Led Code is in hex)
---- ----- ----- ----- ----- ---------- --------------------------------------
   1   189   31      1     0     116949 ECC Error
   1   275   23      1     0     116949 ECC Error
   1   415    1      1     0     116949 ECC Error
   1   435   17      1     0     116949 ECC Error
   1   444   46      1     0     116949 ECC Error
   1   449    3      1     0     116949 ECC Error
   1   479   24      1     0     116949 ECC Error
   1   483   51      1     0     116949 ECC Error
   1   513   50      1     0     116949 ECC Error
   1   514   12      1     0     116949 ECC Error
   1   516   34      1     0     116949 ECC Error
   1   529   17      1     0     116949 ECC Error
   1   624   14      1     0     116949 Invalid Header Error
   1   624   50      1     0     116949 ECC Error
   1   636   40      1     0     116949 ECC Error
   1   711   50      1     0     116949 ECC Error
   2   363   47      1     0     116949 ECC Error
   2   415    9      1     0     116949 ECC Error
   2   415   40      1     0     116949 ECC Error
   2   419    0      1     0     116949 ECC Error
   2   419   35      1     0     116949 ECC Error
   2   419   39      1     0     116949 ECC Error
   2   420    4      1     0     116949 ECC Error
   2   420   41      1     0     116949 ECC Error
   2   425   37      1     0     116949 ECC Error
   2   425   47      1     0     116949 ECC Error
   2   429   22      1     0     116949 ECC Error
   3  1251   43      2     0     116949 ECC Error
   6   670   51      1     0     116949 ECC Error
   6   775   22      1     0     116949 ECC Error
   6   967   35      1     0     116949 ECC Error
   6  1127    1      1     0     116949 ECC Error

Time:  4-NOV 16:55:52  TO   5-NOV 12:01:02   Span:  19:05:09
 (Truncated)
```

```
From:   DTHSTR::SYSTEM       12-OCT-1987 12:24
To:     FREDH
Subj:   GUITAR::$3$DUA35 analysis

Attn:        Field Service
Device:      HAN$DUA35 (RA81 S/N:2AD2B)

Theory:      [xx.xx.xx.xx]

NOTE:        There were 2 hard errors recorded for this device.
             A hard error is defined as an error on a block in which BBR
             was invoked and the data was replaced with 'Force Error'.

Evidence (All results are in decimal except LED Code):
Total errors on drive: 6

           Sector             Volume
     Phys. From  Soft  Hard   Serial
Head Cyl.  Index Count Count  Number         Error Type (Led Code is in hex)
---- ----- ----- ----- ----- ---------- ---------------------------------------
   3    44    13     2     0     135694 Lost Read/Write Ready
   3    45    13     2     0     135694 Lost Read/Write Ready
   5    54    13     0     1     135694 ECC Error
  13    44    13     0     1     135694 ECC Error

Time: 12-OCT 10:31:10  TO  12-OCT 10:31:33   Span:   0:00:22
```

Head Slap

Invoke more disk Act to Mon for growing proble

Replaced? Run RAUT

```
From:   DTHSTR::SYSTEM       13-OCT-1987 04:12
To:     8672::SMITHJ
Subj:   OBOE::$3$DUA143 analysis

Attn:       Field Service
Device:     C3P0$DUA143 (RA70 S/N:0)
            LUKE$DUA143

Theory:      [xx.xx.xx.xx]

Evidence (All results are in decimal except LED Code):
Total errors on drive: 52

           Sector              Volume
     Phys. From  Soft  Hard    Serial
Head Cyl.  Index Count Count   Number        Error Type (Led Code is in hex)
---- ----- ----- ----- ----- ---------- --------------------------------------
   0     2 <unk>     3     0      60032 LED 39, Write and Off Track
   0     4 <unk>     3     0      60032 LED 39, Write and Off Track
   0    12 <unk>     1     0      60032 LED 39, Write and Off Track
   0    37 <unk>     1     0      60032 LED 39, Write and Off Track
   0    45 <unk>     1     0      60032 LED 39, Write and Off Track
   0   386 <unk>     1     0      60032 LED 39, Write and Off Track
   0   417 <unk>     1     0      60032 LED 39, Write and Off Track
   4   369 <unk>     1     0      60032 LED 39, Write and Off Track
   4   387 <unk>     1     0      60032 LED 39, Write and Off Track
   4   409 <unk>     2     0      60032 LED 39, Write and Off Track
   6   205     2     1     0      60032 Lost Read/Write Ready
   6   205 <unk>     2     0      60032 LED 39, Write and Off Track
   6   365 <unk>     1     0      60032 LED 39, Write and Off Track
   6   368 <unk>     1     0      60032 LED 39, Write and Off Track
   6   373 <unk>     1     0      60032 LED 39, Write and Off Track
   6   405 <unk>     1     0      60032 LED 39, Write and Off Track
   7   371 <unk>     1     0      60032 LED 39, Write and Off Track
   8   405 <unk>     1     0      60032 LED 39, Write and Off Track
   8   406 <unk>     2     0      60032 LED 39, Write and Off Track
   8   409 <unk>     1     0      60032 LED 39, Write and Off Track
   9   365 <unk>     1     0      60032 LED 39, Write and Off Track
   9   370 <unk>     1     0      60032 LED 39, Write and Off Track
   9   372 <unk>     1     0      60032 LED 39, Write and Off Track
   9   373 <unk>     1     0      60032 LED 39, Write and Off Track
   9   374 <unk>     1     0      60032 LED 39, Write and Off Track
   9   381 <unk>     1     0      60032 LED 39, Write and Off Track
   9   391 <unk>     1     0      60032 LED 39, Write and Off Track
   9   405 <unk>     1     0      60032 LED 39, Write and Off Track
   9   410 <unk>     1     0      60032 LED 39, Write and Off Track
   9   411 <unk>     2     0      60032 LED 39, Write and Off Track
   9   416 <unk>     1     0      60032 LED 39, Write and Off Track
  10   183    10     1     0      60032 Lost Read/Write Ready
  10   183 <unk>     1     0      60032 LED 39, Write and Off Track
  10   372 <unk>     1     0      60032 LED 39, Write and Off Track
  10   384 <unk>     1     0      60032 LED 39, Write and Off Track
  10   385 <unk>     1     0      60032 LED 39, Write and Off Track
  10   393 <unk>     2     0      60032 LED 39, Write and Off Track
  10   397 <unk>     1     0      60032 LED 39, Write and Off Track
  10   403 <unk>     1     0      60032 LED 39, Write and Off Track
  10   405 <unk>     1     0      60032 LED 39, Write and Off Track

Time: 10-OCT 10:17:21  TO  13-OCT 03:54:53   Span:  65:38:31
```

Data Path?
Servo Related?
-HDA Thermo Arm
shift?

```
From:    COOKIE::SYSTEM        27-OCT-1987 01:10
To:      GENRAL::USER1
Subj:    COOKIE::RA82 S/N:59

Attn:       Field Service

Device:     CAROB$DUA120 (RA82 S/N:59)

Theory:     [xx.xx.xx.xx]

Evidence (All results are in decimal except LED Code):
Total errors on drive: 6

           Sector             Volume
     Phys. From  Soft  Hard   Serial
Head Cyl.  Index Count Count  Number         Error Type (Led Code is in hex)
---- ----- ----- ----- ----- ---------- --------------------------------------
   7   711   49      5     0  62600173 Positioning Error
   7   711 <unk>     1     0  62600173 LED 4D, Bad Embedded Servo During Write

Time: 26-OCT 14:20:16  TO  26-OCT 16:55:44   Span:   2:35:27
```

Manually replace to fix

```
From:    USMRM9::SYSTEM          15-DEC-1987 09:36
To:      GENRAL::USER4
Subj:    USMRM6::$1$DUA34 analysis

Attn:        Field Service
Device:      HSC001$DUA34 (RA81 S/N:122C3)

Theory:      [xx.xx.xx.xx]

NOTE:        There were 4 hard errors recorded for this device.
             A hard error is defined as an error on a block in which BBR
             was invoked and the data was replaced with 'Force Error'.

Evidence (All results are in decimal except LED Code):
Total errors on drive: 8

             Sector                Volume
      Phys.  From  Soft  Hard      Serial
Head  Cyl.   Index Count Count     Number         Error Type (Led Code is in hex)
---- ----- ----- ----- ----- ---------- ---------------------------------------
   2     2 <unk>     1     0      58142 LED 25, Servo Check
   2    67    44     0     1      58142 ECC Error
   2    67    45     0     1      58142 ECC Error
   2    67    46     0     1      58142 Positioning Error
   2    67    47     0     1      58142 Positioning Error
   8   259 <unk>     1     0      58142 LED 25, Servo Check
   8  1224    31     1     0      58142 ECC Error
  11     1 <unk>     1     0      58142 LED 25, Servo Check

Time: 15-DEC 01:39:30  TO  15-DEC 11:36:31   Span:   9:57:01
```

scratch

Servo
R/W
HDA
Belts
Cables
P.S.

```
From:   DTHSTR::SYSTEM        28-JUL-1987 12:26
To:     NODE::NORMAN
Subj:   DTHSTR::$3$DUA77 analysis

Attn:       Field Service
Device:     LUKE$DUA77 (RA81 S/N:151E)

Theory:     [xx.xx.xx.xx]

Evidence (All results are in decimal except LED Code):
Total errors on drive: 14

          Sector              Volume
     Phys. From  Soft  Hard   Serial
Head Cyl.  Index Count Count  Number          Error Type (Led Code is in hex)
---- ----- ----- ----- ----- ---------- --------------------------------------
   9    33     2     1     0      81052 Lost Read/Write Ready
   9    33     3     1     0      81052 Lost Read/Write Ready
   9    33     5     1     0      81052 Lost Read/Write Ready
   9    33     8     1     0      81052 Lost Read/Write Ready
   9    33     9     1     0      81052 Lost Read/Write Ready
   9    33    18     1     0      81052 Lost Read/Write Ready
   9    33 <unk>     8     0      81052 LED 4D, Bad Embedded Servo During Write

Time: 28-JUL 12:20:40  TO  28-JUL 12:21:30   Span:   0:01:50
```

Probably a
Scratch or a erasure
Data Path R/W ?
Scrub/Scan growing?

This is e

```
From:   COOKIE::SYSTEM       System Management" 14-DEC-1987 15:53
To:     GENRAL::JACKN
Subj:   COOKIE::$3$DUA119 analysis

Attn:       Field Service
Device:     SPICE$DUA119 (RA82 S/N:47)

Theory:     [xx.xx.xx.xx]

Evidence (All results are in decimal except LED Code):
Total errors on drive: 40

           Sector              Volume
     Phys. From  Soft  Hard    Serial
Head Cyl.  Index Count Count   Number         Error Type (Led Code is in hex)
---- ----- ----- ----- ----- ---------- -------------------------------------
  13   641     2     1     0   64301171 Invalid Header Error
  13   641     4     1     0   64301171 Positioning Error
  13   641     5     1     0   64301171 Positioning Error
  13   641     6     1     0   64301171 Invalid Header Error
  13   641     7     1     0   64301171 Positioning Error
  13   641     8     1     0   64301171 Positioning Error
  13   641    11     1     0   64301171 Positioning Error
  13   641    12     1     0   64301171 Positioning Error
  13   641    13     1     0   64301171 Invalid Header Error
  13   641    14     1     0   64301171 Positioning Error
  13   641    15     1     0   64301171 Positioning Error
  13   641    16     1     0   64301171 Positioning Error
  13   641    17     1     0   64301171 Invalid Header Error
  13   641    18     1     0   64301171 Positioning Error
  13   641    19     1     0   64301171 Positioning Error
  13   641    23     1     0   64301171 Positioning Error
  13   641    24     1     0   64301171 Positioning Error
  13   641    25     1     0   64301171 Invalid Header Error
  13   641    27     1     0   64301171 Positioning Error
  13   641    28     1     0   64301171 Positioning Error
  13   641    29     1     0   64301171 Positioning Error
  13   641    30     1     0   64301171 Positioning Error
  13   641    32     1     0   64301171 Positioning Error
  13   641    33     1     0   64301171 Positioning Error
  13   641    34     1     0   64301171 Positioning Error
  13   641    35     1     0   64301171 Positioning Error
  13   641    37     1     0   64301171 Positioning Error
  13   641    39     1     0   64301171 Positioning Error
  13   641    44     1     0   64301171 Positioning Error
  13   641    45     1     0   64301171 Positioning Error
  13   641    47     1     0   64301171 Positioning Error
  13   641    50     1     0   64301171 Positioning Error
  13   641    51     1     0   64301171 Positioning Error
  13   641    52     1     0   64301171 Invalid Header Error
  13   641    56     1     0   64301171 Positioning Error
  13   641    57     1     0   64301171 Positioning Error

Time: 14-DEC 15:44:16  TO  14-DEC 15:45:27   Span:   0:01:11
```

HDA.
Probable erasure or Scratch
Embedded servo probably Bad

-smashed header

```
From:    WIMPY::SYSTEM        6-AUG-1987 15:17
To:      NODE::JACKSN
Subj:    WIMPY::RA82 S/N:3A

Attn:        Field Service

Device:      HSC010$DUA44 (RA82 S/N:3A)
             HSC010$DUS1

Theory:      [xx.xx.xx.xx]

Evidence (All results are in decimal except LED Code):
Total errors on drive: 8

           Sector            Volume
     Phys. From  Soft  Hard   Serial
Head Cyl.  Index Count Count  Number        Error Type (Led Code is in hex)
---- ----- ----- ----- ----- ---------- -------------------------------------
   1  1421   46      1     0         44 Positioning Error
   3   270   26      1     0         44 ECC Error
   3   272   26      1     0         44 ECC Error
   3   273   26      1     0         44 Positioning Error
   3   275   26      1     0         44 ECC Error
   3   276   26      1     0         44 ECC Error
   4   216   20      1     0         44 ECC Error
   4   217   20      1     0         44 ECC Error
-----------------------------------------------------------------------------

Time: 31-JUL 01:34:44  TO   6-AUG 15:16:59   Span: 157:42:14
```

Stimulate I/O
growing?
HDA
Look at RCT
Scandisk

check error log

```
From:   DTHSTR::SYSTEM       17-FEB-1988 02:47
To:     NODE::NORMAN
Subj:   TUBA::$3$DUA41 analysis

Attn:       Field Service
Device:     SAX$DUA41 (RA82 S/N:835A)

Theory:     [xx.xx.xx.xx]

Evidence (All results are in decimal except LED Code):
Total errors on drive: 12

           Sector                Volume
     Phys. From  Soft  Hard      Serial
Head Cyl.  Index Soft  Count     Number         Error Type (Led Code is in hex)
---- ----- ----- ----- ----- ---------- --------------------------------------
   2     9   36      5     0     105021 Positioning Error
   2    11   36      7     0     105021 Positioning Error

Time: 10-FEB 21:05:09  TO  11-FEB 06:18:40    Span:   9:13:30
```

Scratch ? not enough info
Do I/o get more' info
Do scrub Look For Forced error

```
From:    USMRM5::SYSTEM          15-FEB-1988 06:43
To:      GENRAL::FIELD
Subj:    USMRM5::$1$DUA113 analysis

Attn:        Field Service
Device:      HSC011$DUA113  (RA81 S/N:8D6B)

Theory:      [xx.xx.xx.xx]

Evidence:

          LED
          CODE
Count     (HEX)    Translation of LED CODE
-----     ----     -----------------------
   4.      00      Undefined LED code; no translation available (HARDCORE)
  20.      01      Spindle motor speed transducer timeout XDUCR Pulses missing | Spin up p
  11.      F1      Slave load timeout                                                 not right

    The drive detected errors may have been responsible for the
    following 1 SI events:

Evidence : 1 SDI Communication Transfer Errors

            Event-
    Count   Status   Translation
    -----   ------   -----------
       1.   ( 1AB) Receiver Ready Collision Errors

Time: 13-FEB 18:14:13  TO  15-FEB 08:40:31   Span:  38:26:18
```

```
From:    ZEPHYR::SYSTEM        21-NOV-1987 14:32
To:      FIELDS
Subj:    ZEPHYR::$5$DUA1 analysis

Attn:        Field Service
Device:      ZEPHYR$DUA1 (RA81 S/N:0)

Theory:      [xx.xx.xx.xx]

Evidence:

         LED
         CODE
Count    (HEX)    Translation of LED CODE
-----    ----     -----------------------
  28.     F1      Slave load timeout
   2.     F8      Slave seek timeout

    The following drive-detected errors may have occurred as a result of the
    above  errors.  These  errors do not have any extended status, hence no
    valid led code was available.

Evidence:    2 Errors

               Request    Mode    Error    Controller
    Count        Byte     Byte     Byte       Byte
    -----      -------- -------- -------- ----------
       2.         13       00       80         00

Time: 21-NOV 15:15:53  TO  21-NOV 16:27:19   Span:   1:11:26
```

Probably Servo/Head lock lever/PS. etc.
Heads aren't moving

```
From:   TUBA::SYSTEM       20-OCT-1987 15:48
To:     NODE::NORMAN
Subj:   TUBA::$3$DUA77 analysis

Attn:       Field Service
Device:     HAN$DUA77 (RA82 S/N:C4E)

Theory:     [xx.xx.xx.xx]

Evidence:

         LED
         CODE
Count    (HEX)    Translation of LED CODE
-----    ----     -----------------------
 213.     0B      INVALID LEVEL 2 SDI COMMAND OPCODE. Comm problem
                  The  command opcode within a level 2 command from a SDI
                  controller  was  received  with good parity but did not
                  match  any  of  the  valid  SDI level 2 command opcodes
                  known by the drive.
   1.     1F      R/W SECTOR OVERRUN ERROR. Communications related event
                  The  drive  detected  READ or WRITE gate asserted while
                  simultaneously detecting the presence of a sector pulse
                  or an index pulse.

    Since there were SDI communication errors which could have caused media
    related  transfer  errors to occur, the following is a summary of those
    media errors:

Evidence (All results are in decimal):
Total media errors on drive: 195

           Sector             Volume
           From  Soft  Hard   Serial
Head  Cyl. Index Count Count  Number        Error Type (Led Code is in hex)
---- ----- ----- ----- ----- ---------- -------------------------------------
  0    313     0     1     0        504 ECC Error
  0    331    13     1     0        504 ECC Error      No media error
  0    316    20     1     0        504 ECC Error
  5    333    30     1     0        504 ECC Error      Hybrid/Cable/Bulkheads
  5    318    52     1     0        504 ECC Error
  6    329     0     1     0        504 ECC Error
  6    324    10     1     0        504 ECC Error      Controller (fix first)
  6    326    14     1     0        504 ECC Error
  8    331    17     1     0        504 ECC Error
  8    321    30     1     0        504 ECC Error
 10    711     3     3     0        504 ECC Error
 10    711     4     2     0        504 ECC Error
 10    327     7     1     0        504 ECC Error
 10    311    17     1     0        504 ECC Error
 12    320     9     1     0        504 ECC Error
 12    321     9     1     0        504 ECC Error
 12    315    12     1     0        504 ECC Error
 13    328     1     1     0        504 ECC Error
 14    322     2     1     0        504 ECC Error
 14    316     9     1     0        504 ECC Error

Time: 16-OCT 14:04:42  TO  19-OCT 16:57:45   Span:  74:53:03
         (Truncated) Lots more
```

```
From:   DTHSTR::SYSTEM       11-OCT-1987 23:15
To:     NODE::NORMAN
Subj:   DRUM::$3$DUA143 analysis

Attn:       Field Service
Device:     C3P0$DUA143 (RA70 S/N:0)

Theory:     [xx.xx.xx.xx]

Evidence (All results are in decimal except LED Code):
Total errors on drive: 29

            Sector             Volume
     Phys.  From  Soft  Hard   Serial
Head Cyl.   Index Count Count  Number        Error Type (Led Code is in hex)
---- -----  ----- ----- ----- ---------- ------------------------------------
   0   386 <unk>      1     0      60032 LED 39, Write and Off Track
   0   417 <unk>      1     0      60032 LED 39, Write and Off Track
   4   369 <unk>      1     0      60032 LED 39, Write and Off Track
   4   387 <unk>      2     0      60032 LED 39, Write and Off Track
   4   409 <unk>      2     0      60032 LED 39, Write and Off Track
   6   365 <unk>      1     0      60032 LED 39, Write and Off Track
   6   368 <unk>      1     0      60032 LED 39, Write and Off Track
   6   373 <unk>      1     0      60032 LED 39, Write and Off Track
   8   405 <unk>      1     0      60032 LED 39, Write and Off Track
   9   365 <unk>      1     0      60032 LED 39, Write and Off Track
   9   370 <unk>      1     0      60032 LED 39, Write and Off Track
   9   374 <unk>      1     0      60032 LED 39, Write and Off Track
   9   375 <unk>      1     0      60032 LED 39, Write and Off Track
   9   381 <unk>      1     0      60032 LED 39, Write and Off Track
   9   391 <unk>      2     0      60032 LED 39, Write and Off Track
   9   405 <unk>      1     0      60032 LED 39, Write and Off Track
   9   410 <unk>      1     0      60032 LED 39, Write and Off Track
   9   411 <unk>      2     0      60032 LED 39, Write and Off Track
   9   416 <unk>      1     0      60032 LED 39, Write and Off Track
  10   372 <unk>      1     0      60032 LED 39, Write and Off Track
  10   393 <unk>      2     0      60032 LED 39, Write and Off Track
  10   397 <unk>      1     0      60032 LED 39, Write and Off Track
  10   403 <unk>      2     0      60032 LED 39, Write and Off Track

Time: 10-OCT 10:17:21  TO  11-OCT 22:00:26   Span:  35:43:04
```

firmware fault

ECM

```
From:   DTHSTR::SYSTEM       19-NOV-1987 09:34
To:     JACKN
Subj:   GUITAR::$3$DUA553 analysis

Attn:       Field Service
Device:     LANDO$DUA553 (RA90 S/N:225)

Theory:     [xx.xx.xx.xx]

Evidence:

         Error
         Code
Count    (HEX)    Translation of Error Code
-----    -----    -------------------------
  20.     21      SDI Pulse Error
```

Comm problem TS Comm (direction)

```
    The drive detected errors may have been responsible for the
    following 14 SI events:

Evidence : 14 SDI Communication Transfer Errors

           Event-
    Count  Status   Translation
    -----  ------   -----------
       1.  (  2B) SDI Drive Command Timeout Errors
      11.  (  CB) Lost Receiver Ready for Transfer Errors
       2.  ( 16B) Drive Failed Initialization Errors
```

```
From:   COOKIE::SYSTEM       System Management" 14-DEC-1987 15:53
To:     GENRAL::JACKN
Subj:   LANDO$DUA169 analysis

Attn:       Field Service
Device:     LANDO$DUA169 (RA70 S/N:0)

Theory:     [xx.xx.xx.xx]

Evidence (All results are in decimal except LED Code):
Total errors on drive: 143

           Sector             Volume
     Phys. From  Soft  Hard   Serial
Head Cyl.  Index Count Count  Number         Error Type (Led Code is in hex)
---- ----- ----- ----- ----- ---------- --------------------------------------
   7   158     8     1     0  172500556 Positioning Error
   7   158     9     2     0  172500556 Positioning Error
   7   158    10     1     0  172500556 Positioning Error
   7   158    11     1     0  172500556 Positioning Error
   7   158    12     2     0  172500556 Positioning Error
   7   158    13     1     0  172500556 Positioning Error
   7   158    14     1     0  172500556 Positioning Error
   7   158    15     2     0  172500556 Positioning Error
   7   158    16     1     0  172500556 Positioning Error
   7   158    17     1     0  172500556 Positioning Error
   7   158    18     2     0  172500556 Positioning Error
   7   158    19     1     0  172500556 Positioning Error
   7   158    20     1     0  172500556 Positioning Error
   7   753    15    29     0  172500556 Positioning Error
   8   158     0     1     0  172500556 Positioning Error
   8   158     1     1     0  172500556 Positioning Error
   8   158     2     2     0  172500556 Positioning Error
   8   158     3     1     0  172500556 Positioning Error
   8   158     4     1     0  172500556 Positioning Error
   8   158     5     2     0  172500556 Positioning Error
   8   158     6     1     0  172500556 Positioning Error
   9   158     0     2     0  172500556 Positioning Error
   9   158     1     1     0  172500556 Positioning Error
   9   158     2     1     0  172500556 Positioning Error
   9   158     4     2     0  172500556 Positioning Error
   9   158     5     1     0  172500556 Positioning Error
   9   158     6     1     0  172500556 Positioning Error
   9   158    33     1     0  172500556 Positioning Error
  10   158    12     2     0  172500556 Positioning Error
  10   158    13     1     0  172500556 Positioning Error
  10   158    14     1     0  172500556 Positioning Error
  10   158    15     2     0  172500556 Positioning Error
  10   158    16     1     0  172500556 Positioning Error
  10   158    17     1     0  172500556 Positioning Error
  10   158    18     1     0  172500556 Positioning Error

Time:  5-NOV 14:46:53  TO   5-NOV 14:56:17   Span:   0:09:24
        (truncated)
```

Servo Circomphrential Scratch
or Erasure

(HDA)

```
From:    COOKIE::SYSTEM       14-DEC-1987 15:53
To:      GENRAL::JACKN
Subj:    FATCAT$DUA5 analysis

Attn:         Field Service
Device:       FATCAT$DUA5 (RA81 S/N:F885)
              MIDCAT$DUA5

Theory:       [xx.xx.xx.xx]

Evidence (All results are in decimal except LED Code):
Total errors on drive: 68

               Sector                Volume
      Phys. From  Soft  Hard     Serial
Head Cyl.  Index Count Count     Number         Error Type (Led Code is in hex)
---- ----- ----- ----- ----- ---------- --------------------------------------
  12     9     6     1     0      46726 Lost Read/Write Ready
  12   356    45     1     0      46726 ECC Error
  12   428 <unk>     1     0      46726 LED 39, Write and Off Track
  12  1160     0     1     0      46726 ECC Error
  12  1160    15     1     0      46726 Lost Read/Write Ready
  12  1160    28     1     0      46726 ECC Error
  12  1160 <unk>     1     0      46726 LED 39, Write and Off Track
  12  1174     8     1     0      46726 ECC Error
  12  1174    46     1     0      46726 Positioning Error
  12  1221    31     1     0      46726 ECC Error
  12  1226    10     1     0      46726 ECC Error
  12  1236     3     1     0      46726 ECC Error
  12  1238    33     1     0      46726 ECC Error
  12  1241     8     1     0      46726 ECC Error
  13   318    24     1     0      46726 ECC Error
  13   342    16     1     0      46726 ECC Error
  13   396    15     1     0      46726 ECC Error
  13   442    46     1     0      46726 Positioning Error
  13   445    14     1     0      46726 ECC Error
  13   445    50     1     0      46726 Positioning Error
  13   958    34     1     0      46726 ECC Error
  13   980    17     1     0      46726 ECC Error
  13  1055    39     1     0      46726 ECC Error
  13  1070    51     1     0      46726 Positioning Error
  13  1071    46     1     0      46726 ECC Error
  13  1119    15     1     0      46726 ECC Error
  13  1136    47     1     0      46726 ECC Error
  13  1174    16     1     0      46726 Invalid Header Error
  13  1174    24     1     0      46726 Positioning Error
  13  1174    36     1     0      46726 Positioning Error
  13  1175    11     1     0      46726 ECC Error
  13  1176    18     1     0      46726 Positioning Error
  13  1178    11     1     0      46726 ECC Error
  13  1223    23     1     0      46726 ECC Error
  13  1223    23     1     0      46726 Invalid Header Error

Time: 24-JUL 10:42:56  TO  31-JUL 08:49:55   Span: 166:07:58
      <literal>(<)truncated>
```

(HDA)

Same Surface
-Spindl GND-

```
From:    PICKUP::SYSTEM       22-JUL-1987 14:00
To:      NODE::NORMAN
Subj:    PICKUP$DUA71 analysis

Attn:       Field Service
Device:     PICKUP$DUA71 (RA80 S/N:0)

Theory:     [xx.xx.xx.xx]

NOTE:       There were 1 hard errors recorded for this device.
            A hard error is defined as an error on a block in which BBR
            was invoked and the data was replaced with 'Force Error'.

Evidence (All results are in decimal except LED Code):
Total errors on drive: 6

          Sector              Volume
     Phys. From  Soft  Hard   Serial
Head Cyl.  Index Count Count  Number         Error Type (Led Code is in hex)
---- ----- ----- ----- ----- ---------- -------------------------------------
   4    72     1     1     0          1 Positioning Error
   4    82     8     0     1          1 ECC Error
   4   123    18     1     0          1 Invalid Header Error
   6     2     5     1     0          1 Positioning Error
   6    42     4     1     0          1 ECC Error
   6   119    27     1     0          1 Positioning Error

Time:  2-MAR 15:10:33   TO   5-MAR 15:25:12    Span:  72:15:38
```

Oposing Heads (diffrent sur
Heed inteferes crud from 6
Falls to 4 Look for
5+3 as stuff gets
distributed!

Probably HDA
will probably spread

```
From:   MUFFIN::SYSTEM        7-MAY-1988 10:23
To:     GENRAL::HOLMES
Subj:   MUFFIN::$5$DUA1 analysis

Attn:       Field Service
Device:     GREASY$DUA1 (RA81 S/N:20D9)

Theory:     [xx.xx.xx.xx]

Evidence:   18 SDI Communication Errors

              Event-
        Count  Status     Translation
        -----  ------     ------------
          18.  ( 10B) Controller Detected Pulse or State Parity Errors

Time:   7-MAY 10:02:45  TO   7-MAY 10:18:16   Span:   0:16:31
```

- Run Drive Loopba
Then Determine Dr

```
From:   PICKUP::SYSTEM       25-FEB-1988 14:35
To:     REEMAN
Subj:   PICKUP::$1$DUA252 analysis

Attn:       Field Service
Device:     HSC015$DUA252 (RA90 S/N:20B)

Theory:     [xx.xx.xx.xx]

Evidence:   19 SDI Communication Errors

                    Event-
              Count  Status     Translation
              -----  ------     -----------
              - 1.   (  2B) SDI Drive Command Timeout Errors
out of Spec  — 18.   ( 1AB) Receiver Ready Collision Errors

     The following 1 drive detected errors may be related to the above
     SDI errors:

Evidence:

       Error  Mfg.
       Code   Code
Count  (HEX)  (HEX) Translation of Error Code
-----  -----  ----- --------------------------
  1.    1F     2D    Sector Overrun Error

Time: 24-FEB 15:37:50  TO  25-FEB 14:32:50   Span:  22:55:00
```

3 use other port on same K.SDI.
2 - DRIVE Loopback multiple Passes
1 - Drive on another port first
3 - Reseat ECM

Probably K.SDI
Cable
ECM

SDI Problem

```
From:    HICKUP::SYSTEM       30-MAR-1988 16:28
To:      GENRAL::HIMES
Subj:    HICKUP::$5$DUA231 analysis

Attn:        Field Service
Device:      WHEEZY$DUA231 (RA81 S/N:2ED92)

Theory:      [xx.xx.xx.xx]

Evidence:
```

Com problem Probably
Check out disk Scrund
Scan disk Exercise

```
          LED
          CODE
Count     (HEX)    Translation of LED Code
-----     ----     -----------------------
   1.      0D      Status error byte non-zero while attempting to execute a
                   command
 452.      21      Two or more pulses of the same polarity are detected on
                   the controller real-time state line (control pulse error)
   8.      22      Two or more pulses of the same polarity are detected on
                   the controller write command data line (data pulse error).
   1.      41      SDI controller response time out
```

Personality

Personality Module timeout

Pers d
CNTLR d
Recom

```
        The drive detected errors may have been responsible for the
        following 405 SI events:

Evidence : 405 SDI Communication Transfer Errors

              Event-
        Count Status   Translation
        ----- ------   -----------
           3. (  2B) SDI Drive Command Timeout Errors
         396. (  CB) Lost Receiver Ready for Transfer Errors
           6. ( 16B) Drive Failed Initialization Errors

        Since there were SDI communication errors which could have caused media
        related  transfer  errors to occur, the following is a summary of those
        media errors:

Evidence (All results are in decimal, except LED code):
Total media errors on drive: 438

              Sector                Volume
              From  Soft  Hard      Serial
Head   Cyl.   Index Count Count     Number          Error Type (Led Code is in hex)
---- ------ ----- ----- ----- ----------- ----------------------------------------
  0    102    2      1     0      152034 Lost Read/Write Ready
  0    236    3      1     0      152034 Positioning Error
  0    627   11      1     0      152034 Lost Read/Write Ready
  0    451   18      1     0      152034 Lost Read/Write Ready

  "     "    "       "     "        "      "      "      "      "
  "     "    "       "     "        "      "      "      "      "

 13    279   43      1     0      152034 Lost Read/Write Ready
 13    418   44      1     0      152034 Lost Read/Write Ready
 13    280   50      1     0      152034 Lost Read/Write Ready
 13    546   50      1     0      152034 Lost Read/Write Ready

Time: 25-MAR 17:06:32  TO  30-MAR 15:27:47   Span: 118:21:14
```

```
From:   HICKUP::SYSTEM        30-MAR-1988 10:50
To:     USER5
Subj:   HICKUP::$5$DUA232 analysis

Attn:       Field Service
Device:     WHEEZY$DUA232 (RA81 S/N:2ED7C)

Theory:     [xx.xx.xx.xx]

Evidence:                                                                    Com problem
          LED
          CODE                                                               Pers/sw proc
Count     (HEX)    Translation of LED Code
-----     ----     -----------------------                                   & cable
1 Drive  { 4.      21     Two or more pulses of the same polarity are detected on
         }                the controller real-time state line (control pulse error)
         { 56.     22     Two or more pulses of the same polarity are detected on
                          the controller write command data line (data pulse error)
    The drive detected errors may have been responsible for the
    following 6 SI events:

Evidence : 6 SDI Communication Transfer Errors

              Event-
    Count     Status    Translation
    -----     ------    -----------
       2.     (  2B) SDI Drive Command Timeout Errors      } from Controller
       4.     ( 16B) Drive Failed Initialization Errors    }

    Since there were SDI communication errors which could have caused media
    related transfer errors to occur, the following is a summary of those
    media errors:

Evidence (All results are in decimal, except LED code):
Total media errors on drive: 2

              Sector                   Volume
              From   Soft   Hard       Serial
Head   Cyl.   Index  Count  Count      Number          Error Type (Led Code is in hex)
----   -----  -----  -----  -----  ----------  ---------------------------------------
  0    1248     1      0      1        152788  Media Format Error } fix this too    Uncorr err in RCT
  0     624     3      1      0        152788  Positioning Error  } Probably all copies of Blk in Rct Bad
NOTE:          There  were  1  hard  errors  recorded for this device.
               A hard error is defined as an error on a block in which BBR
               was  invoked  and the data was replaced with 'Force Error',
               or any Media Format Errors.

Time: 30-MAR 09:31:31  TO  30-MAR 10:34:22   Span:   1:03:51
```

```
From:    MUFFIN::SYSTEM      17-MAY-1988 15:09
To:      SERVICE
Subj:    MUFFIN::$5$DUA253 analysis

Attn:        Field Service
Device:      SLEAZY$DUA253 (RA90 S/N:1FF)

Theory:      [xx.xx.xx.xx]

NOTE:        There were 11 hard errors recorded for this device.
             A hard error is defined as an error on a block in which BBR
             was invoked and the data was replaced with 'Force Error'.

Evidence (All results are in decimal except LED Code):
Total errors on drive: 56

           Sector             Volume
     Phys. From  Soft  Hard   Serial
Head Cyl.  Index Count Count  Number        Error Type (LED Code is in hex)
---- ----- ----- ----- ----- ---------- -------------------------------------
   0    40    21     1     0         28 ECC Error
   0    40    24     1     0         28 ECC Error
   0    40    30     1     0         28 ECC Error
   0   528     7     1     0         28 Positioning Error
   0   528     8     1     0         28 Positioning Error
   0   528    15     1     0         28 ECC Error
   0   528    16     1     0         28 ECC Error
   0   528    17     1     0         28 ECC Error
   0   528    18     1     0         28 ECC Error
   0   528    19     1     0         28 ECC Error
   0   528    22     1     0         28 Positioning Error
   0   528    24     0     1         28 ECC Error
   0   528    31     1     0         28 ECC Error
   0   914     6     1     0         28 ECC Error
   0   914     9     0     1         28 ECC Error
   0   914    10     1     0         28 Positioning Error
   0   914    11     1     0         28 ECC Error
   0   914    12     0     1         28 ECC Error
   0   914    17     0     1         28 ECC Error
   0   914    20     0     1         28 ECC Error
   0   914    21     0     1         28 ECC Error
   0   914    23     1     0         28 ECC Error
   0  1181    14     1     0         28 ECC Error
   0  1181    19     1     0         28 ECC Error
   0  1181    20     1     0         28 ECC Error
   0  2416    39     1     0         28 ECC Error
   0  2416    51     1     0         28 ECC Error
   0  2416    52     1     0         28 ECC Error
   0  2416    53     1     0         28 ECC Error
   1   181    35     1     0         28 ECC Error
   1   181    50     1     0         28 ECC Error
   1   181    51     1     0         28 ECC Error

Time: 16-MAY 08:32:30  TO  16-MAY 15:46:41   Span:   7:14:11
      <truncated>
```

growing ?

HDA

```
From:    HICKUP::SYSTEM        12-MAY-1988 15:53
To:      GENRAL::FHARDY
Subj:    HICKUP::$5$DUA253 analysis

Attn:       Field Service
Device:     GREASY$DUA254 (RA90 S/N:20A)

Theory:     [xx.xx.xx.xx]

Evidence (All results are in decimal except LED Code):
Total errors on drive: 16

           Sector            Volume
     Phys. From  Soft  Hard   Serial
Head Cyl.  Index Count Count  Number         Error Type (LED Code is in hex)
---- ----- ----- ----- ----- ---------- --------------------------------------
   3    27   13      1     0        521 ECC Error
   3    49   29      1     0        521 ECC Error
   4  2403   25      1     0        521 ECC Error
   4  2420    6      1     0        521 ECC Error
   4  2453   31      1     0        521 ECC Error
   4  2453   33      1     0        521 ECC Error
   4  2453   37      1     0        521 ECC Error
   4  2562   44      1     0        521 ECC Error
   6  2607   22      1     0        521 Positioning Error
   8  2507   30      1     0        521 ECC Error
   9  2317   21      1     0        521 ECC Error
   9  2357   29      1     0        521 ECC Error
   9  2463   18      1     0        521 ECC Error
  11    37   41      1     0        521 Positioning Error
  11    40    9      1     0        521 ECC Error
  11    45   50      1     0        521 ECC Error

Time: 12-MAY 14:48:51  TO  12-MAY 15:48:32   Span:   0:60:40
```

Data Path problem
Controller → Disk

```
From:    MUFFIN::SYSTEM        29-MAR-1988 14:25
To:      GENRAL::FMILER
Subj:    MUFFIN::$5$DUA7 analysis

Attn:        Field Service
Device:      GRANPA$DUA7 (RA81 S/N:12D1)

Theory:      [xx.xx.xx.xx]

Evidence:

         LED
         CODE
Count    (HEX)    Translation of LED Code
-----    ----     ------------------------
  45.     41      SDI controller response time out

    The drive detected errors may have been responsible for the
    following 2 SI events:

Evidence : 2 SDI Communication Transfer Errors

           Event-
    Count  Status   Translation
    -----  ------   -----------
       2.  ( 1AB) Receiver Ready Collision Errors

Time: 29-MAR 14:15:23  TO  29-MAR 14:16:43   Span:   0:01:20
```

Probably in drive
Microprocessor/Personal
Controller uses disk T
ie: servo clock
Drive Spun down Ca
SDI Level 2 com

```
From:    MUFFIN::SYSTEM        21-MAR-1988 15:45
To:      GENRAL::HIMES
Subj:    MUFFIN::$5$DUA7 analysis

Attn:        Field Service
Device:      WHEEZY$DUA7 (RA81 S/N:ED9)

Theory:      [xx.xx.xx.xx]

Evidence:

          LED
          CODE
Count     (HEX)    Translation of LED Code
-----     ----     -----------------------
   1.      00      Undefined LED code; no translation available (HARDCORE)
   4.      23      Spindle motor interlock broken (belt tension lever is
                   released)
   2.      34      Read data separator/encoder error
   1.      41      SDI controller response time out
   4.      46      R/W safety interrupt occurred with no cause bits set
   6.      F1      Slave load timeout

    The drive-detected errors may have been responsible for the
    following 5 SI events:

Evidence : 5 SDI Communication Transfer Errors

            Event-
    Count   Status   Translation
    -----   ------   -----------
       3.   (  2B) SDI Drive Command Timeout Errors
       2.   ( 16B) Drive Failed Initialization Errors

Time: 21-MAR 14:44:04  TO  21-MAR 15:38:00   Span:   0:54:56
```

Interlocks tied together
@ micro

micro

→ micro

```
From:    MUFFIN::SYSTEM        30-MAR-1988 16:48
To:      GENRAL::HIMES
Subj:    MUFFIN::$5$DUA35 analysis

Attn:        Field Service
Device:      WHEEZY$DUA35 (RA81 S/N:2ED93)

Theory:      [xx.xx.xx.xx]

Evidence:    19 SDI Communication Errors
                Event-
       Count  Status     Translation
       -----  ------     -----------
          4.  (  2B) SDI Drive Command Timeout Errors
          6.  (  4B) Controller Detected Transmission Errors
          1.  ( 10B) Controller Detected Pulse or State Parity Errors
          8.  ( 16B) Drive Failed Initialization Errors

    The following 2 drive detected errors may be related to the above
    SDI errors:

Evidence:

         LED
         CODE
Count    (HEX)     Translation of LED Code
-----    ----      -----------------------
   1.     1F       A sector pulse is detected during the execution of
                   a read or write of a sector
   1.     21       Two or more pulses of the same polarity are detected on
                   the controller real-time state line (control pulse error)

Time: 26-MAR 09:21:57  TO  30-MAR 16:30:37   Span: 103:09:39
```

Comm problems

```
From:   PICKUP::SYSTEM       25-FEB-1988 14:49
To:     NEWTON
Subj:   PICKUP::$1$DUA252 analysis

Attn:       Field Service
Device:     HSC015$DUA252 (RA90 S/N:20B)

Theory:     [xx.xx.xx.xx]

Evidence (All results are in decimal except LED Code):
Total errors on drive: 14

           Sector              Volume
     Phys. From  Soft  Hard    Serial
Head Cyl.  Index Count Count   Number         Error Type (LED Code is in hex)
---- ----- ----- ----- ----- ---------- ---------------------------------------
   7  2370 <unk>    14     0         26 LED 4B, Index Error

Time: 25-FEB 14:46:38  TO  25-FEB 14:47:41   Span:   0:01:03
```

- Vaxsim says —
Scratch on Servo surface
Do more I/O

```
From:    MUFFIN::SYSTEM       12-MAY-1988 15:33
To:      GENRAL::JACKN
Subj:    MUFFIN::$5$DUA253 analysis

Attn:        Field Service
Device:      SLEAZY$DUA253 (RA90 S/N:1FF)

Theory:      [xx.xx.xx.xx]

Evidence:

          Error
          Code
Count     (HEX)     Translation of Error Code
-----     -----     -------------------------
   1.      07       SDI Frame Sequence Failure
   1.      13       Spindle Motor Control Fault
 407.      21       SDI Transfer Error (Pulse Error)
  94.      60       Read/Write Head Select failure
  25.      EB       Unknown Error Code
  41.      F3       Servo Spinup Failed

    The following drive detected errors may have occurred as a result of the
    above  errors.  These  errors do not have any extended status, hence no
    valid led code was available.

Evidence:    1 Errors

              Request   Mode    Error   Controller
    Count       Byte    Byte     Byte      Byte
    -----     -------- -------- -------- ----------
       1.        03      00       00        00

    The drive detected errors may have been responsible for the
    following 307 SI events:

Evidence : 307 SDI Communication Transfer Errors

           Event-
    Count  Status   Translation
    -----  ------   -----------
       8.  (  2B) SDI Drive Command Timeout Errors
     293.  (  CB) Lost Receiver Ready for Transfer Errors
       5.  ( 16B) Drive Failed Initialization Errors
       1.  ( 1AB) Receiver Ready Collision Errors
```

— Comm problem try another po to Isolate pro

..........CONTINUED

```
    Since there were SDI communication errors which could have caused media
    related  transfer  errors to occur, the following is a summary of those
    media errors:

Evidence (All results are in decimal, except LED code):
Total media errors on drive: 341

           Sector              Volume
           From  Soft  Hard    Serial
Head  Cyl. Index Count Count   Number         Error Type (Led Code is in hex)
---- ----- ----- ----- ----- ---------- ----------------------------------------
  0   1179    0      1     0         28 Lost Read/Write Ready
  0   1388    0      1     0         28 Lost Read/Write Ready
  0    408    1      1     0         28 Lost Read/Write Ready
  0   1804    2      1     0         28 Lost Read/Write Ready
  0    326    6      1     0         28 Lost Read/Write Ready
  0   1892    9      1     0         28 Lost Read/Write Ready
  0   2221    9      1     0         28 Lost Read/Write Ready
  0   2226    9      1     0         28 Lost Read/Write Ready
  0      2   12      1     0         28 Lost Read/Write Ready
  0    545   17      1     0         28 Lost Read/Write Ready

  "     "    "       "     "         "   "    "     "     "
  "     "    "       "     "         "   "    "     "     "
  "     "    "       "     "         "   "    "     "     "
  "     "    "       "     "         "   "    "     "     "

 12   2214   19      1     0         28 Lost Read/Write Ready
 12   1418   28      1     0         28 Lost Read/Write Ready
 12   1459   34      1     0         28 Lost Read/Write Ready
 12    333   36      1     0         28 Lost Read/Write Ready
 12   2107   36      1     0         28 Lost Read/Write Ready
 12   1914   37      1     0         28 Lost Read/Write Ready
 12   2256   38      1     0         28 Lost Read/Write Ready
 12   1023   45      1     0         28 Lost Read/Write Ready
 12   1967   50      1     0         28 Lost Read/Write Ready
 12   1585   52      1     0         28 Lost Read/Write Ready
 12   2109   52      1     0         28 Lost Read/Write Ready
 12    821   58      1     0         28 Lost Read/Write Ready

Time:  9-MAY 14:14:11  TO  12-MAY 15:19:51   Span:  73:06:40
```

CHAPTER 12

DSA DSDF/BBR

12.1 INTRODUCTION

This section is intended to introduce some of the features in the the Digital Storage Architecture (DSA) which are different from other storage subsystems. The major topic is Bad Block Replacement (BBR). To help describe BBR, disk addressing, ECC detection and correction, and ECC thresholding are also covered. One section contains answers to often asked questions.

Throughout this document are references to blocks and sectors. They are interchangeable. Also, the term *controller* refers to the UDA50, KDA50, KDB50, and HSC50/70.

12.2 OVERVIEW MATERIAL for UNDERSTANDING BBR

The following sections cover the terminology used in BBR and REVECTOR topics described later in this document. These new DSA features are prerequisite to understanding BBR and REVECTOR.

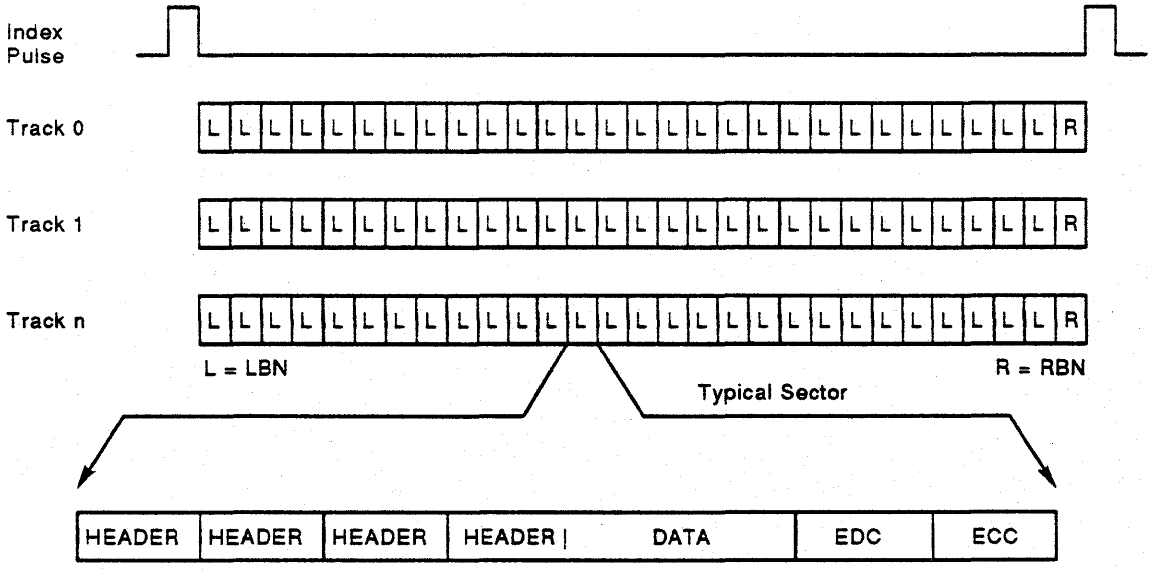
12.2.1 LBN and RBN Association (Disk Organization for BBR)

The organization of logical blocks (LBNs) and replacement blocks (RBNs) on the RA–series drives is defined in such a way that the user area of the disk remains constant regardless of the number of blocks that go bad. The constant number of blocks is maintained by supplying RBNs at the end of each track on the disk which can be substituted for a bad block. A track on the disk contains many LBNs. The number of LBNs and RBNs depends on the device type. At this time, there is only one RBN per disk track. The BBR algorithm selects a replacement RBN by finding the first available, good RBN that is closest to the bad block.

A typical sector contains a header (replicated four times), a data area which can be 512 or 576 bytes in length, an EDC field, and an ECC field. The header, EDC field, and ECC field are described later.

A track on a disk is circular, starts at index, and continues until index is again reached. For the sake of illustration Figure 12–1 depicts a track in a straight line (rather than circular as it really is on the disk platter) to show the relationship between LBNs, RBNs, and the typical sector.

Figure 12–1: Disk Track and Sector Organization



12.2.2 Disk Addressing

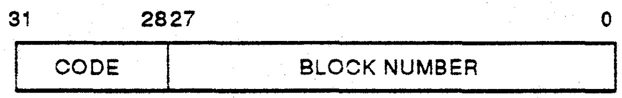
The DSA disk addressing and disk header concept is very different from the traditional schemes. First, the disk header does not contain the traditional cylinder, track, and sector information. The DSA header contains 32 bits. The upper 4 bits contain a header code, and the lower 28 bits contain a block number (see Figure 12–2). There are four copies of the 32–bit header for integrity. The code field of the header defines whether a block is a logical block (LBN), a replacement block (RBN), a diagnostic block (DBN), or an external block (XBN). If the block is an LBN, the header code also indicates if the block has been replaced.

From the addressing point of view, the disk header is used by the controller when accessing data stored on the disk. Disk headers are read by the controller and compared for a match against the target block number when searching for a block of data. Both the header code and the block number must be valid in order to have a header match.

The controller checks the header code to ensure that the correct area of the disk is accessed. For example, if the controller is trying to access a sector in LBN space and the disk is actually positioned in XBN space, the code portion of the header would mismatch and an error would be reported.

The controller also checks the code field to see if the block has been revectored. If a block has been revectored, it is the controller's responsibility to determine where the data has been revectored to. There is more discussion on the revector process later.

Figure 12–2: Disk Header Format



12.2.3 How Header Codes are Used

Following is a list of the header codes and their purpose.

- Code 00—This code indicates the LBN data is usable and directs the controller to access the data following the header information.
- Code 03—This is the non–primary replacement code. This code indicates to the controller that the data following the header is invalid and directs the controller to retrieve the data from an RBN that is located on a different track than the track containing the LBN. The controller uses the RCT information to determine exactly which RBN was used. This operation is a non–primary revector.
- Code 05—This is the primary replacement code. This code indicates to the controller that the data following the header is invalid and directs the controller to read the data from the RBN at the end of the current track. For RA–based disks, each track contains one or more RBNs which can be used for primary replacement data. This operation is a primary revector.
- Code 06—This is the RBN header code. This code indicates to the controller that it is accessing an RBN.
- Code 11—This is the unusable block code.
- Code 12—This is the XBN header code. This code indicates to the controller that it is accessing FCT which resides in the external blocks (XBN) area of the disk.
- Code 14—This is the DBN header code. This code indicates to the controller that it is accessing a diagnostic block (DBN).

12.2.4 Special Uses of the Header Code Field

BBR does not apply to the RCT and FCT since they are multi–copy structures. The controller takes the following action based on RCT header codes:

- Code 00—This code indicates the LBN data is usable and directs the controller to access the data following the header information.
- Code 11— This is the unusable block code. This code indicates to the controller that the data following the header is invalid and directs the controller to retrieve the data from the next copy of the RCT or FCT. If all copies of the RCT or FCT are not readable, an uncorrectable error is reported.

12.2.5 EDC Protection

The DSA architecture provides an error detection code (EDC) mechanism to protect the controller data paths which are not protected by parity or ECC. The EDC is generated by the controller, at the bus interface, when a block of data is to be written to the disk. The EDC is then written to the disk along with the data and ECC. The EDC is checked, at the bus interface, when the controller transfers a block of data to the host.

If an EDC error is detected by the controller, it first checks to see if the error is a forced error (FE). An FE is detected by inverting the EDC and finding a result of zero. If an EDC error still exists after checking for the FE, the controller retries the read operation and logs an error. If the FE is detected, the controller reports the event to the host in the MSCP end packet and does not log the error to the error log.

To set the FE indicator, the host issues an MSCP WRITE command with the FE modifier set. When the controller detects the FE modifier, it calculates the normal EDC, then inverts it to set the indication of the FE.

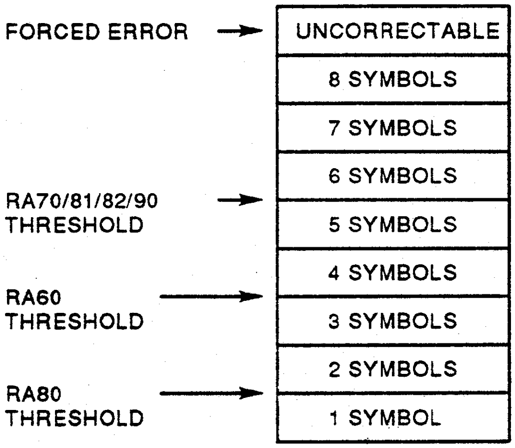
12.2.6 ECC Detection and Correction

The RA–series drives use a 170–bit ECC to correct up to 80 bits in error on a single disk sector. The controller generates the ECC and appends it to the data when writing a sector to the disk. The controller checks each sector of data read from the disk for the existence of an ECC error to determine if ECC correction is needed. If the controller determines that correction is needed, it enters the microcode algorithm to perform the correction. The ECC correction algorithm determines how many bits are in error by keeping a count of the number of symbols and bits per symbol in error on a single disk sector. A symbol is defined as 10 contiguous bits of correction information that the controller applies to the data in error to correct it. The maximum number of symbols that can be corrected before an an uncorrectable ECC error occurs is 8. Since each symbol contains information to correct a 10–bit burst, the maximum correction capability is 80 bits (8 symbols X 10 bits per symbol).

12.2.7 ECC Thresholding

The controller uses the ECC symbol count as a threshold to determine when a disk block is going bad and needs to be replaced. Each drive contains a threshold count parameter based on its media type, density, and head technology. The controller uses the parameter information to understand when to set the BBR flag and log ECC errors. The rules for thresholding are simple: If the number of symbols used to correct the ECC error is below the threshold, then perform correction only. If the number of symbols is greater than or equal to the threshold, then perform correction, set the BBR flag in the MSCP END packet, and log an ECC error. See Figure 12–3 for drive threshold settings.

Figure 12–3: ECC Symbols and Drive Threshold for BBR and Error Logging



12.3 BBR PROCESS OVERVIEW

Bad block replacement (BBR) is the mechanism that Digital Storage Architecture (DSA) uses to replace disk blocks (sectors) which are or may become unusable.

Normally, BBR is transparent to the user except for the error log events that are recorded by the BBR process. The only exception is when the block being replaced contains an uncorrectable ECC error. If an uncorrectable ECC is detected, the bad block is written with a forced error (FE) and the user is notified. Once a block is written with FE, the data is lost to the user. The possibility of recovering the data from a block after an uncorrectable ECC has occurred and FE has been applied is remote.

If RMS, or its equivalent, is the mechanism used to perform I/O operations when the uncorrectable ECC is detected, the job is terminated with an error message.

If QIO system service or its equivalent is the mechanism used to perform I/O operations when a uncorrectable ECC is detected, the user is notified with an MSCP END PACKET. In this case, the user has the option to continue or terminate the job.

It is important to note that the physical disk structure is permanently modified by BBR. The actual entities that BBR modifies are the bad block's header to reflect the replacement type used (primary or non–primary) and the replacement control table (RCT) descriptor to reflect the block used for replacement (RBN) is in use.

BBR is implemented in host software in UDA50, KDA50, and KDB50 controllers because there is not enough code space available in the controller ROM. BBR is implemented in HSC50 and HSC70 controllers after V250.

Three actions take place when invoking, performing, and reporting the replacement of a bad block:

- Notification that a block needs to be replaced — The controller sets the BBR flag to inform the replacement process that BBR is needed.
- Attempt to replace the bad block— This is the actual execution of the BBR algorithm.
- Report the results of the bad block replacement attempt to the error log.

12.3.1 Notification that a Block Needs to be Replaced

A request to perform BBR can only occur when the controller is attempting to transfer data to or from the disk. To understand when data transfers occur, consider the following: Data transfer requests are initiated by the host issuing a data transfer MSCP command, such as READ or WRITE, to the controller. The controller processes the command and performs the action requested. The controller checks for errors and sets the BBR flag, if needed. Reasons for setting the BBR flag include:

- The controller is not able to successfully read a header when trying to locate a sector of data to be transferred to or from the host. Before setting the BBR flag, the controller determines if the sector had been previously replaced.
- The controller cannot find the data sync on a sector when attempting to transfer the sector of data to the host.
- While the controller is attempting to read a sector of data from the disk, it detects an ECC error greater than or equal to the drive's error reporting threshold.

12.3.1.1 Host BBR

If the controller detects errors at the end of the data transfer, it places the error status in the MSCP END PACKET and possibly generates an error log packet. If BBR is needed, the controller also sets the BBR flag in the MSCP END PACKET. The controller then passes the end packet to the host. The host, upon receiving the END PACKET determines if errors occurred. If errors occurred, the host takes the appropriate action to handle the error. If the action needed is BBR, the host transfers control to the host BBR software.

12.3.1.2 Controller BBR

If the controller detects errors at the end of the data transfer, it places the error status in the MSCP END PACKET and possibly generates an error log packet. If BBR is needed, the controller sets the internal BBR flag. The internal BBR flag causes the controller firmware to enter its BBR routine.

12.3.2 Executing Bad Block Replacement

The host and controller perform the same functions to replace a block, so the following paragraphs represent both.

The BBR code retrieves and saves a copy of the bad block's data. This is because the suspected bad block must be read and written to confirm it is bad. This destroys the original data. If an uncorrectable ECC is detected while retrieving the bad block, the BBR algorithm sets an internal flag to remember that forced error must be set on the block.

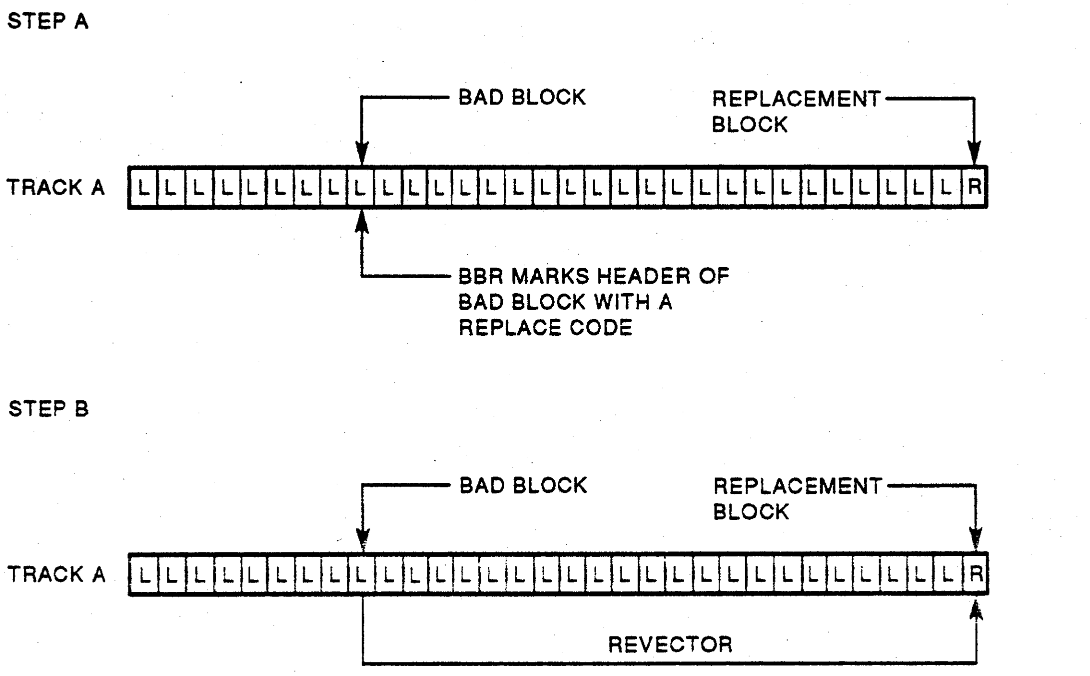
The BBR code determines if the LBN in question is really bad before attempting to replace it. This testing is done to protect from replacing too many blocks because of transient errors. The current BBR implementation uses the customer data instead of test patterns to test the bad block. Test patterns alone would not find the pattern–sensitive spots in the disk media.

If no errors are detected when the block is written and read with selected patterns and the customer data, the block is not replaced.

If the suspected bad LBN fails the tests, then the BBR code finds a substitute block called a replacement block (RBN) and moves the data from the bad LBN to the RBN. The actions used to move the bad block's data to the replacement block include:

- The BBR code marks the RCT descriptor which corresponds to the RBN used for the replacement "in use" so that it cannot be used again by subsequent replacements.
- The BBR code uses the MSCP REPLACE command to permanently mark the bad LBN header with a primary or non–primary replace code (Figure 12–4 and Figure 12–5, step A).
- The BBR code writes the data back to the bad block's LBN to force the controller to revector the data to the replacement block. This action is forced because the replacement code has been applied to the bad block's header (Figure 12–4 and Figure 12–5, step B).

Figure 12–4: Primary Replacement/Revector

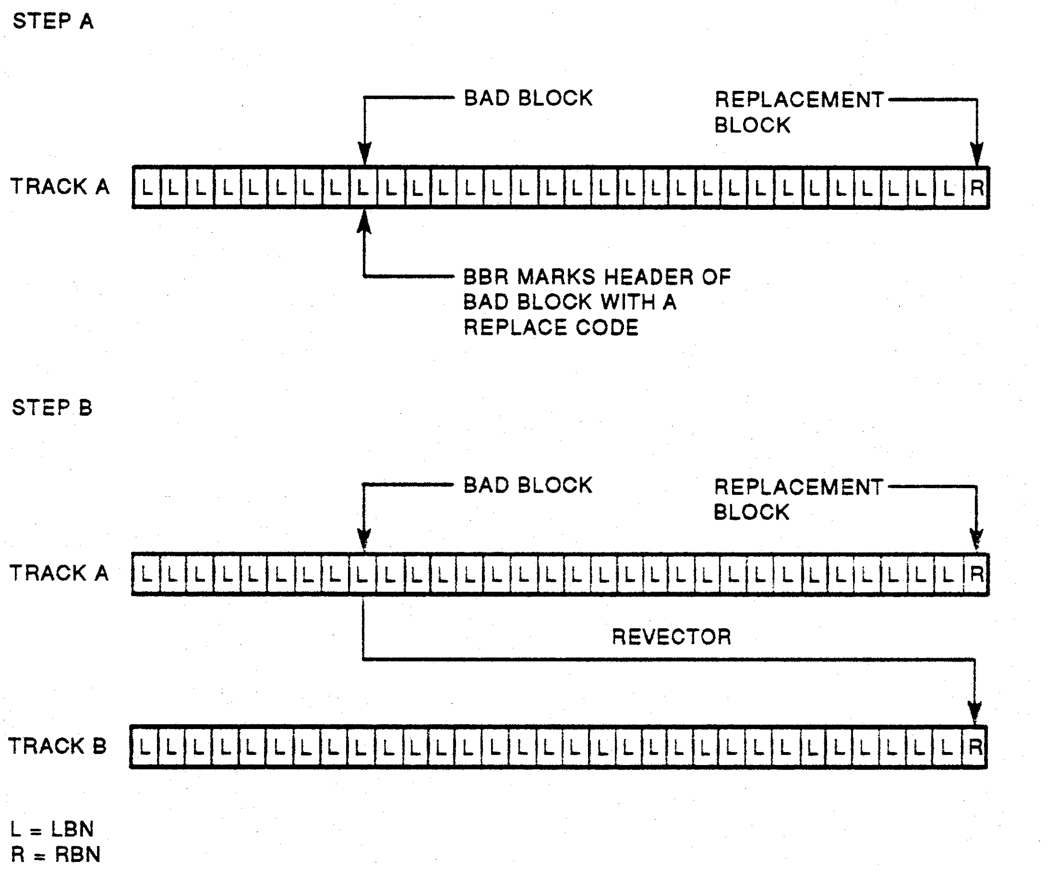


L = LBN
R = RBN

NOTE: BBR MOVES BAD BLOCK'S DATA TO REPLACEMENT BLOCK BY WRITING DATA BACK TO BAD BLOCK'S LBN. THIS CAUSES CONTROLLER TO REVECTOR DATA TO RBN.

CXO-2375A

Figure 12–5: Non-Primary Replacement/Revector



NOTE: BBR MOVES BAD BLOCK'S DATA TO REPLACEMENT BLOCK ON A DIFFERENT TRACK BY WRITING DATA BACK TO BAD BLOCK'S LBN. THIS CAUSES CONTROLLER TO USE POINTERS IN RCT TO REVECTOR DATA TO REPLACEMENT RBN.

CXO-2376A

12.3.3 Restarting BBR

The BBR code must keep track of its progress while performing BBR in case errors or power fail occur and BBR may need to be restarted. Block 0 and 1 of the RCT are used for this purpose.

RCT block 0 contains the state of the replacement while the BBR code is executing.

RCT block 1 contains the bad block's data. This data is good data if the ECC correction capability was not exceeded or is "best guess" data if the ECC correction capability is exceeded. Best guess data is passed to the host even though it contains errors.

The rest of the RCT contains the RBN descriptors. The descriptors are used to indicate whether an RBN is available for use, in use, or bad. There is one descriptor per RBN.

12.3.3.1 Host BBR

Host BBR generates many error log packets while handling BBR. This is because host BBR generates many MSCP commands during execution. Any of the commands can generate an error log event. As many as six error log events can be logged against one bad block replacement attempt.

12.3.3.2 Controller BBR

Controller BBR only generates one error log packet for the entire BBR operation since it generates no MSCP commands. Therefore, has better control of logging the event.

12.4 TROUBLESHOOTING BBR

BBR is a common, expected event in the DSA architecture. There will always be blocks which need to be replaced. However, when BBR becomes excessive, it is the symptom of a hardware problem. The following is a list of the most common occurrences and how to handle them.

- A disk experiencing BBRs in the range of one to two per month is reasonable. **There is nothing wrong.** BBRs in this range occur because the customer data pattern may affect a sector such that an ECC error above the threshold is generated.
- A disk is experiencing BBRs to the same LBN, but the LBN is not being replaced. This problem existed before VMS Version 4.4. The algorithm was using test patterns to test the block in question rather than using the customer's data. The test pattern didn't stress the bad block to the point where the algorithm thought the block was bad, so the block wasn't replaced. If this is really annoying the customer, eliminate the use of the block by creating a new copy of the file. You can also run the SCRUBBER utility (ZUDL or EVRLK) on the disk to attempt to replace the block. If the disk is attached to an HSC50/70, you can use DKUTIL to replace the block.
- BBR is causing excessive error log events to be placed in the error log, and it is hard to understand which events are meaningful. This problem applies to host BBR only, so systems with a KDA50, UDA50, or KDB50 are affected. This problem came about when the enhanced version of BBR was released to the field. The enhanced algorithm caused more error log events to be generated during testing of the bad block to determine if it was really bad. The change to the algorithm specifies that testing is to be done with correction and recovery disabled. By disabling correction and recovery, the controller generates an uncorrectable ECC error if the bad block has any ECC symbol errors.

There are only two meaningful error log events when BBR is attempted. They are the error log packet which indicates that an ECC error above the threshold occurred, and the BBR event error log packet. The error log packets after the packet which indicates ECC above the threshold and before the BBR error log packet have no meaning except to the BBR algorithm. To match the ECC over threshold event log packet with the BBR log packet do the following:

- Match the command reference number field of the error log packets.
- Match the unit number field of the error log packets.
- Match the LBN field of the error log packets.

When you match the fields of the error log packets and find the BBR packet, you have the necessary information about the BBR to understand what happened.

12.5 REVECTORING

The revector process is a controller function performed on disk blocks which have been replaced. The revector process is totally transparent unless an error occurs during revector. A disk block is considered revectored when its header has been written with a replace code. Once the block is replaced, the data portion of the sector is no longer accessible and the controller executes the revector procedure to determine where the data was revectored to.

12.6 QUESTIONS + ANSWERS

QUESTION: WHAT IS A FORCED ERROR AND HOW DOES IT OCCUR?

ANSWER: A forced error (FE) is a way of identifying a block of data that contained an uncorrectable ECC error. It is also a fast detection mechanism, since ECC correction takes about 17 ms of microcode execution time in the controller, and detection of an FE only takes the time to read one sector.

A forced error (FE) is manufactured by the BBR code when the sector being replaced contains an uncorrectable ECC error. The FE is applied to the data by an MSCP WRITE command with the FE modifier set. At this time, only the BBR code writes FEs.

QUESTION: CAN A FORCED ERROR BE UNDONE?

ANSWER: An FE can be cleared by issuing an MSCP WRITE command with the FE modifier CLEARED. Clearing the FE will not make the customer's data appear good again since the FE was placed on the sector to indicate the data is invalid (uncorrectable ECC). Running BACKUP will remove the FE, but it will also produce an error message indicating that a corrupt block was detected. One possible way to fix a block containing an FE is to:

- Read the LBN containing the FE. Today, DKUTIL is the only tool which allows you to read a block containing an FE. However, a program to read blocks which contain FE the QIO system service or its equivalent can be written and used.
- Examine the contents to see if something obvious is wrong and, if so, correct and rewrite the LBN with FE CLEARED.
- If the contents of the LBN cannot be fixed, get a backup copy of the file in question and restore the backup copy to the disk.
- If a backup copy of the file cannot be obtained and the LBN cannot be fixed, the customer must re-run the job used to create the file.

QUESTION: WHAT IS THE RCT?

ANSWER: The RCT is a multi–copy structure. There are four or more identical copies of the RCT for data integrity. Each copy of the RCT contains three areas of interest described below. The RCT is always physically located above the user LBN area of the disk. The RCT can be read by using the QIO – READRCT function.

RCT Block 0

The first area is RCT block 0 which contains state information while a replacement is in progress. If the BBR algorithm is aborted during replacement, then block 0 state is used to determine where to restart the algorithm. When a drive is brought on line, the host checks to see if a replacement was in progress. If it was, the information in block 0 is used to restart the algorithm. In the case of the RA81, RCT block 0 is at LBN 891072. (See Figure 12–6.)

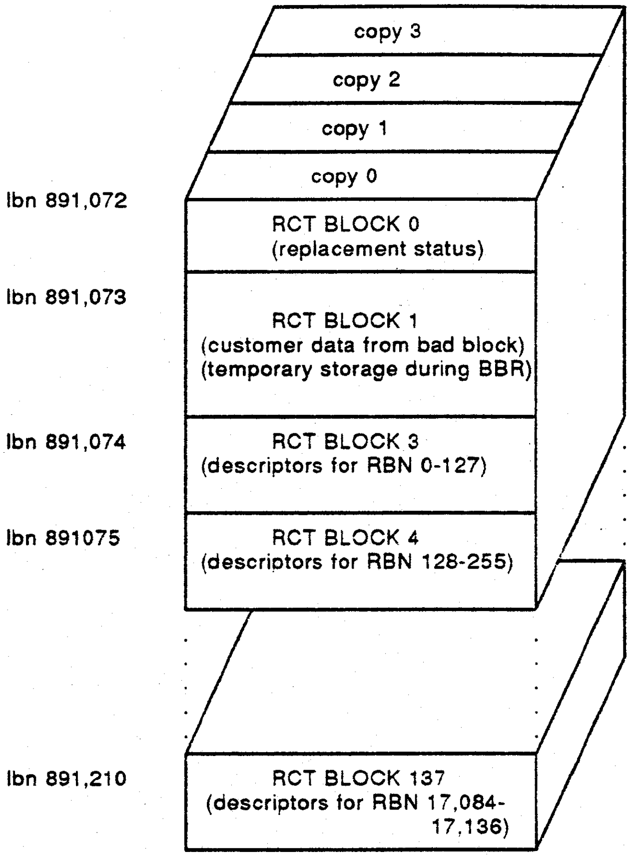
RCT Block 1

The second area of the RCT is used to save the data from the suspected bad block while the BBR algorithm is in progress. This data will be moved to the RBN after the bad block is marked replaced. In the case of the RA81, RCT block 1 is at LBN 891,073. (See Figure 12–6.)

The RCT Descriptor Blocks

The rest of the RCT is used to keep track of the disposition of the RBNs in the host LBN area of the disk. There is 1 descriptor per RBN. For example, the RA81 contains 17,472 descriptors in the RCT since there are 17,472 RBNs in host LBN space. There are no descriptors for the RCT since BBR does not protect the RCT area of the disk. The descriptors start at block 2 of the RCT and occupy as many blocks as necessary to account for the 17.472 descriptors. Each block contains 128 descriptors, so there are 136 blocks of descriptors on an RA81 (17,472 divided by 128). In the case of the RA81, the descriptor block area of the disk starts at LBN 891,074. See Figure 12–6.

Figure 12–6: RCT Layout for an RA81



QUESTION: WHAT IS THE FCT AND WHAT IS IT USED FOR?

ANSWER: The factory control table (FCT) contains a list of physical blocks which were found bad when the HDA was built and scanned for bad spots on the media. After formatting the HDA, the formatter uses the list of physical blocks contained in the FCT as the basis for blocks to replace. The formatter will construct the content of the RCT from the FCT contents. If the FCT contents are destroyed, the integrity of the HDA cannot be maintained after a format operation is performed. The FCT occupies four cylinders following the RCT and is a multi–copy structure for data integrity purposes. Only the controller can access the FCT. In the case of the HSC50 or HSC70, DKUTIL can be used to dump the contents of the FCT.

QUESTION: WHAT OPERATING SYSTEMS SUPPORT BBR AND ERROR LOGGING?

ANSWER: There are actually two versions of BBR. The original version did a poor job of replacing blocks, so enhancements were made and a new version was implemented. The table below lists which operating systems and versions contain the new version of BBR. Operating systems which supported BBR before the version listed in the table could have problems. A summary of the BBR enhancements include:

- Better testing of the potential bad block. This change came about because the old algorithm wasn't replacing blocks often enough. The fix was to use the customer's data for the test pattern. Since the customer's data almost always catches the ECC error, it is the best test pattern.
- Better stress testing of the bad block. The old algorithm wrote and read the block once, so the block almost always tested good. The new algorithm reads four times, then writes and rereads four more times before declaring the block good. In addition, testing is done with correction and recovery disabled.
- If a data path problem provoked BBR, there was a potential of recursively replacing blocks until the RCT filled up. The new algorithm allows two attempts to find a good replacement block before stopping the replacement attempt.

Table 12–1: Operating Systems Revisions

OPERATING SYSTEM	VERSION	ERROR LOG	BIT To TEXT	NEW BBR
DSM11	3.2	YES	YES	YES
IAS	ALL	NO	NO	Old
RT11	5.5	MIN	MIN	YES
RSTS/E	9.5	YES	YES	YES
MICRO RSTS	2.2	YES	YES	YES
RSX11M/S	4.3	YES	YES	YES
RSX11M+	3.2	YES	YES	YES
MICRO RSX11	3.2	YES	YES	YES
ULTRIX 11	3.0	YES	YES	NO
ULTRIX 32	2.0	YES	YES	YES
ULTRIX 32M micro	2.0	YES	YES	YES
UNIX AT&T SYS 5 Release 2 (32)	2.0	YES	YES	YES
BERKLEY UNIX	NO	NO	NO	NO
VAXELN	2.1	NO	NO	YES
VMS	4.4	YES	YES	YES
MICRO VMS	4.4	YES	YES	YES

QUESTION: CAN AN RBN BE REPLACED?

ANSWER: The BBR algorithm provides for RBN replacement. Replacement is accomplished by marking the corresponding descriptor in the RCT unusable, then finding another replacement block and revectoring the bad block's data to that block.

QUESTION: HOW CAN BBR BE DONE ON SYSTEMS WITHOUT THE BBR CAPABILITY?

ANSWER: The following tools can perform the BBR function even when the operating system doesn't support BBR:

- SCRUBBER (ZUDL or EVRLK)– This standalone tool searches the disk for LBNs to replace and replaces them.
- RABADS for ULTRIX– This is the same as SCRUBBER.
- System copy function – This function makes a new version of the file that contains the bad block. The new version occupies a different set of LBNs and, therefore, removes the bad block from use. This also removes LBNs from the system and causes the disk to shrink by the size of the file.

3 Modes
MANUAL
AUTO
VERIFY – RUN if Data Path in question

MDM> NAKDA

QUESTION: WHAT IS THE MEANING OF THE UDA50, KDA50–Q, KDB50 (xDA) CONTROLLER HANG?

ANSWER: When the (xDA) controller hangs, it is usually interpreted as a failure in the controller. Actually, when the CPU attached to the controller hangs, it forces the controller into a command timeout state. The command timeout state is entered when the (xDA) controller decrements its command timer to zero before receiving an MSCP command from the host CPU. Normally, if a command is received, the controller resets the command timer. If a command is not received within the timeout interval (approximately 2 minutes), the controller displays a timeout error code. Therefore, the controller timeout is usually a symptom of a failure elsewhere in the system.

if Scrubber it reads a block in error above threshold it tries 20 times
Then replaces it

≤ 20 reports as transient

> 20 Replaces it

Run 5 Passes min

Formatter won't always catch a bad block

Backup customers media
First Run 1 pass of Verify to be sure that the data path is ok
Then run Auto mode 5 Passes

QUESTION: WHAT IS THE NAME AND FUNCTION OF THE CURRENT DIAGNOSTIC SET?

ANSWER: Host based 'Diagnostics fr PDP11 & VAX

```
********************************************
*                                          *
*  PDP-11  DIAGNOSTIC SET for UDA50/KDA50  *
*                                          *
********************************************

       ZUDHA1 -- Tests 1-3
           >  Test 1 = UNIBUS Interrupt/Address Test
           >  Test 2 = Executes drive resident diagnostics
           >  Test 3 = Disk Function Test (Read/Write etc)
             -  Released -

       ZUDIA0 -- Test 4
           >  Test 4 = A disk exerciser
             -  Released -

       ZUDJA0 -- Test 5
           >  Test 5 = UDA50/KDA50 Subsystem  Exerciser -- This is
              an MSCP product that works like an operating system
              and reports MSCP error log packets, just  like  an
              operating system. A very good subsystem exerciser.
             - Released -
             - ZUDJB0 - Target release - Q2 87

       ZUDKB0 -- Formatter (DM Code Version 14)
           >  APPROXIMATE RUN TIME IS 2 (TWO) HOURS.
             - Released -
             - ZUDKC0 -  (DM Code Version 15) - Target release Q2 87

       ZUDLA0 -- Bad Block Replacement Utility
           >  "Scrubber" - MSCP product (Utility) for media maintenance.
             -  Released -

       ZUDMxx -- Disk Error Log Utility
           > Will display the 16 Error SILO in RA81, RA80
             - Target release Q2 87 -

       ZUDC -- OBSOLETE Tests 1-4

       ZUDE -- OBSOLETE FORMATTER
```

```
*********************************************
*                                           *
*  VAX BASED DIAGNOSTIC SET for UDA50/KDB50  *
*                                           *
*********************************************
```

```
EVRLB -- V 5.1, Formatter (See description above for PDP-11 ZUDK)
                (DM Code Version 14)
              - Released with VAX Diagnostic Release 24 -
      -- V 6.0, (DM Code Version 15)
              - VAX Diagnostic Release 26 - Jan 87

                     EVRLB prior to V5.1

EVRLF -- V 7.0, Tests 1-3 (See description above for PDP-11 ZUDH)
              - Released with VAX Diagnostic Release 24 -
      -- V 8.0 - VAX Diagnostic Release 25 - OCT 86

EVRLG -- V 7.0, Test 4 (See description above for PDP-11 ZUDI)
              - Released with VAX Diagnostic Release 24 -
      -- V 8.0 - VAX Diagnostic Release 25 - OCT 86

EVRLJ -- Test 5 (See description above for PDP-11 ZUDJ)
         (NOT RELEASED - target release - JAN 87 - for Release 26)

EVRLK -- V 2.0, Bad Block Replacement Utility (See PDP-11 ZUDL)
              - Released with VAX Diagnostic Release 24 -
      -- V 2.1 - VAX Diagnostic Release 25 - OCT 86

EVRLL -- Disk Error Log Utility
         (Will display the 16 Error SILO in RA81, RA80)
         (NOT RELEASED - target release - OCT 86 - for Release 25)

EVRLA -- OBSOLETE Tests 1-4 diagnostic
```

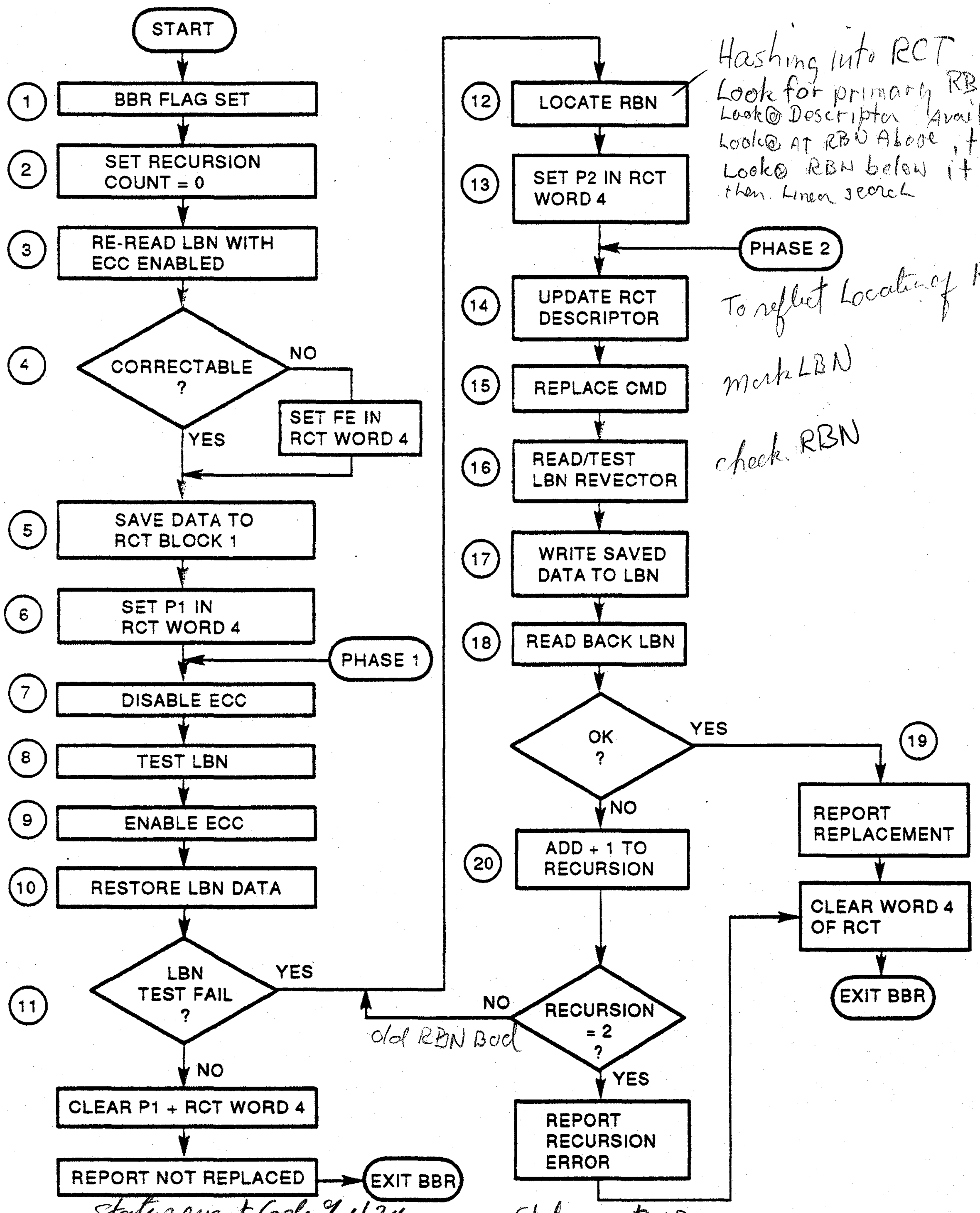
NOTE

The VAX diagnostic supervisors (VDS) that support the host level diagnostics require action by the user.

```
****************************************************
*                                                  *
*  VAX BASED DIAGNOSTIC SET for KDA50 (MicroVAX II) *
*                                                  *
****************************************************
```

```
  MicroVAX II uses the KDA50 controller and its  diagnostics
  are  written  for  the "Micro Diagnostic Monitor" (MDM).

NAKDAB -- Diagnostics for the uVAX-2 and R** drives -- All in one
       Tests 1-3 -
       Test  5 (Subsystem Exerciser)

      - Released with MicroVAX Diagnostic Release 112 -

NAKDAC --
      - Target Release Q2 87

NAKDAD -- Formatter upgrade to DM Code Version 15
      - Target Release Q3 87
```

Figure 12–7: BBR FLOW (simplified)(Sensative)



Hashing into RCT
Look for primary RBN first
Look@ Descriptor Available, go
Look@ At RBN Above it
Look@ RBN below it Ping
then Linear search

To reflect Location of RBN

mark LBN

check RBN

old RBN Bad

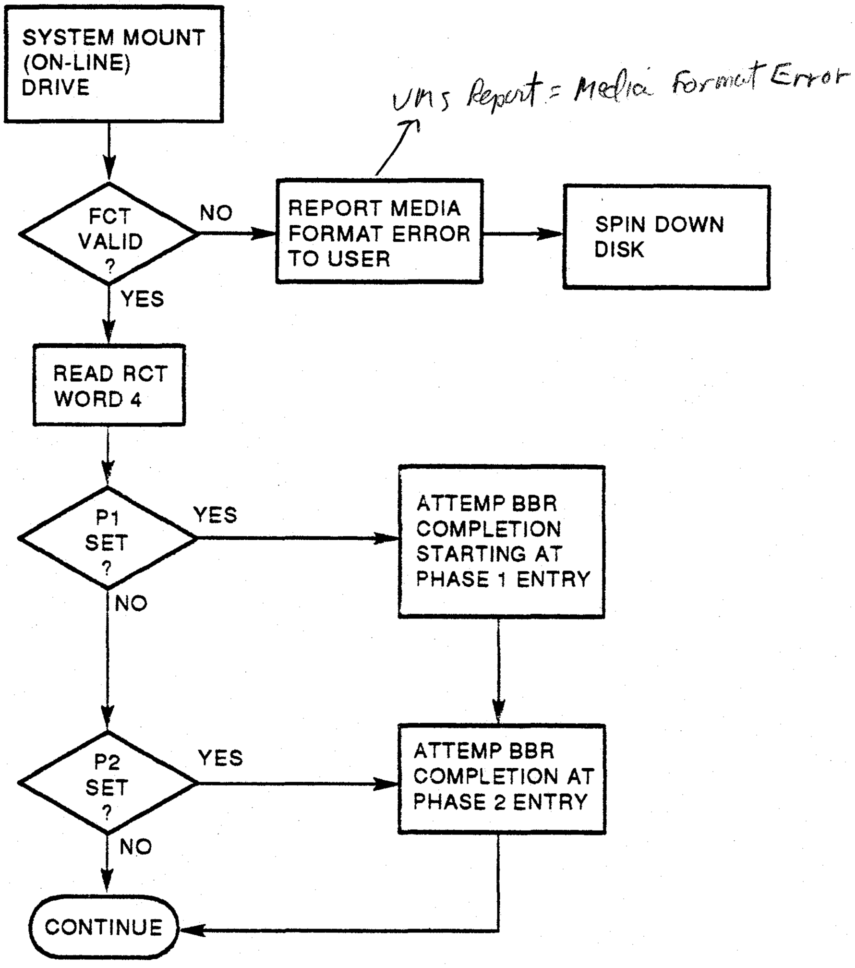
Status event Code %H34

Status event 4D
Probably R/W Data Path

CXO-2377A

old Terms	New/current
Primary	Primary
Secondary	No Primary
Tertiary	

Figure 12–8: Typical Mount Flow



VMS Report = Media Format Error

CXO-2378A

No method to write FE Block to tape if restored from this Tape now the data is corrupt
Make sure they see any forced errors at Backup time to note file & Restore from old backup

```
---+---+---+---+---+---+
   |   |   |   |   |   |
 i | g | i | t | a | l |
   |   |   |   |   |   |
---+---+---+---+---+---+
```

INTEROFFICE

MEMORANDUM

ike Mayfield
on Milano
ieth Brown
ave Varner
l Snyder

FROM: Glenn Scadden
DATE: 21-Feb-1986
DEPT: CX/CSSE
LOC.: CX01-1/P14
TEL.: 303-594-2345/522-2345
ENET: NERMAL::SCADDEN

Results of Feasibility Test -- 18 bit RA81 functionality on VAX

ROUND

After some reports of customers using 18 bit formatted RA81's and RA60's K processors, I volunteered to investigate this reported ability. n offices were also asking for a determination of the "supportability" bit converted media on a VAX. We have since "day 1" informed the field 18 bit media could not be used on 16 bit processors (via Tech Tip, Right etc).

ve offered KL customers several KL to VAX trade-in programs and tives. As a corporation we have been attempting to "migrate" these KL ners to VAX machines. Yet with all the financial incentives, when a KL mer that has RA disks looks at the financial impact of having to purchase ew HDA's/Packs, it significantly detracts from our trade-in programs. ny cases these KL customers have made a significant expenditure for RA and done so only recently. Being able to advertise 18 bit RA drive rsion to a VAX, will certainly improve customer satisfaction on the part ose customers who are converting. It may also enable a KL customer, who A drives, to purchase new VAX machines earlier than previously expected. customer may also feel incentive to purchase new 18 bit RA drives ng his investment will be completely compatible with a future VAX ase.

his test only related to the test of an RA81 18 to 16 bit conversion se on a VAX running VMS. For purposes of this report, when VAX is stated, an assume a VAX running VMS, unless otherwise stated.

My purpose was to conduct a test that would match that of a Field eer intending to make a conversion of an 18 bit RA81 for use on a VAX. a quick look at the different diagnostics and utilities that would be d to support that 18 bit media on a VAX. Both during the conversion and wards on the VAX system.

OAL

did not intend to conduct an exhaustive test of all the VMS operating stem functions, diagnostic functions and the large number of other ilities and programs that could be used on the 18 bit media, when on a X.

s initial feasibility test also only directed its efforts at the version of an 18 bit RA81. RA60 was not addressed in this feasibility t but should be if further investigation is necessary.

id not intend to test a "converted" RA81 on any other VAX, or PDP-11 rating system. VMS was the only operating system used.

TIONS

estricted my investigation to a customer that would have a KL10/HSC intending to move an RA81 to a VAX processor running VMS.

ike formatting an 18 bit HDA to 16 bit, a 16 bit (Burst written) /Pack CANNOT be converted for use as an 18 bit HDA/Pack.......

assumed their would be NO difference in the results of this test, if er VAX processors were used. As long as all other factors and resources nained the same.

e terms "18 bit" and "576 byte" are used interchangeably.

e terms "16 bit" and "512 byte" are used interchangeably.

RCES

I used the following resources during this investigation:

HSC50
RA81 with 18 bit Burst written and formatted HDA.
11780 with UDA50 and HSC50.

USION

decided to provide my conclusion at this point in the report. The next on will outline the test details. If not interested in that detail, this usion should provide the "bottom line".

THE CONVERSION OF AN 18 BIT BURST WRITTEN AND FORMATTED RA81 HDA TO A T FORMATTED HDA (STILL 18 BIT BURST WRITTEN) WAS ACCOMPLISHED WITHIN THE OF THIS INVESTIGATION. USE OF THIS 16 BIT FORMATTED AND 18 BIT BURST EN HDA ON A VMS OPERATING SYSTEM WORKED FOR THE LIMITED NUMBER OF TIES AND FUNCTIONS TESTED.

.. I recommend that CX/CSSE develop the best approach to "retracting" the long standing policy preventing 18 bit media from use on 16 bit machines.

  a. This "retraction" process should be coupled with a project for accomplishing a more extensive test. This was only a feasibility test and did NOT address all permutations of the HSC and VAX diagnostics, when used on a "converted" 18 bit RA81. Nor did it even attempt to try all VMS based utilities and functionality.

  b. A test of this conversion for an RA60-PE pack (18 bit pack) should be accomplished, before "retracting" our policy for RA60.

c. Complete confidence in the "supportability" aspects may require an SVT type process, with availability of all necessary resources. These resources would be systems, media, drives and manpower (to include VMS expertise) etc.

d. A decision is also required on the "scope" of any further testing into other VAX based (Like ULTRIX, SYSTEM V, VAX/ELN etc) and PDP-11 based operating systems. If other operating systems support is required, these resources and expertise would be needed.

. A "converted" 18 bit HDA will have 803,712 user LBN's available, when used under VMS. Normally, a 16 bit RA81 HDA has 891,072 user LBN's. This equates to only 9.8% less LBN space for a "converted" 18 bit HDA, as compared to a 16 bit HDA. Our current MLP for an RA81 16 bit HDA is in excess of $5,000.

Also, as bad as this may sound, a customer who makes the conversion would most likely obtain the full 891,072 LBN's upon the failure of the 18 bit HDA and replacement of it with a 16 bit HDA. It would be foolish to replace an 18 bit RA81 HDA on a VAX with another 18 bit HDA upon the failure of the 18 bit HDA, or during an FCO replacement. One more point. The last logistics price list I saw, showed the 18 bit HDA as being more expensive than the 16 bit HDA. So a thinking field service branch manager would naturally make the customer happy and save some money also.

ome important information was learned along the way, which will be
bed in the next section.

ETAILS
----

kept good notes of each step of this test. I will thus, only summarize
ignificant points.

ations:

RA81 contains a "jumper" on the front of the Preamp module of the HDA.
s "jumper" identifies to the drive logic whether the HDA is burst
tten in 16 or 18 bit. 18 Bit HDA's have this Jumper removed. The
version process maintains this jumper in the out state -- As the burst
te on the HDA never changes in the conversion process. The jumper also
nges the "characteristics" response from the drive on an SDI "Get
racteristics" response.

nging the jumper from the out to the in state results in drive faults
other unpredictable error conditions. AT ALL TIMES THE JUMPER SHOULD
IN THE OUT STATE.

HSC50 (All KL customers that have RA's have an HSC50) is "configured"
576 or 512 byte operation. Obviously, it is configured in 576 for
on the KL. However, when used on a VAX (as if a KL customer was to move
HSC and RA's to a VAX), the mode does not seem to matter. It works
the VAX or the KL when "configured" in 576 byte mode. The mode is set
the "SETSHO" utility command "Set Sector_size 576" or "Set Sector_size
" command.

e Media "mode" field in the FCT is changed automatically by the mode the rmatter is run in. i.e. When using either the HSC or VAX formatter to rmat an 18 bit HDA to 16 bit format, the FCT mode is automatically itched and appropriately displayed in the response to an MSCP "online" nmand from a host.

ease note that an 18 bit (576 byte) HDA once formatted to 16 bit, can be rmatted back to 18 bit without any trouble.

FCT contains two "subtables". One allocated for 576 byte and one for 2 byte formatted physical block numbers that are bad. For a 576 byte rst written HDA, both tables contain the same physical block numbers. r an 512 byte burst written HDA, only the 512 byte subtable contains ysical block numbers. For the case of the 576 byte burst written HDA, is was the correct thing to do. The SDI/DSDF spec is not at all clear this situation and could be easily misunderstood. Also, since the 512 byte rst written HDA CANNOT be field "converted" for 18 bit use, not having y contents in the 576 byte subtable is of no consequence.

e can "convert" a 576 byte burst written HDA with 18 bit format to bit format, without having to reformat. This "conversion" technique quires the use of the "write enabled HSC DKUTIL" program to modify the Mode eld of the FCT and to accomplish other conversion functions. However, rmatter "conversion" is by far the simplest and easiest method. nversion requires the ability to write all Host LBN's (Including the RCT ocks), otherwise uncorrectable ECC errors result from the "logic" the controller "picking-off" the ECC field from within the data field an 18 bit block.

nce the formatter diagnostic was never intended to be a "scanner" and lied upon to find bad blocks, the conversion may result in some marginal d blocks remaining on the media. A recommendation should be made, in any nversion instructions, to have the field engineer "keep an eye out" for d block reports in the VAX error log. The instructions should describe w to use the HSC DKUTIL to "revector" (replace) bad blocks. This assumes at since the 18 bit drive came from a KL, that an HSC50 will be available. is recommendation also, unfortunately, assumes that a possibility always ists that the drive may end up an a VAX running a UNIX based operating stem without dynamic BBR, regardless of any position statement relative the conversion.

test for any impact on VMS Dynamic BBR was also conducted. A large number block replacements were made during this test, without any problems pacting the BBR process noted.

ties and Diagnostics used

ormatter
erify
LEXER
KUTIL
LDISK

VDS) UDA50 based Diagnostics
  EVRLB Formatter
  EVRLF Tests 1-3
  EVRLG Test 4
  EVRLK "Scrubber"

N VMS FUNCTIONS

Concerns yet needing investigation

VMS Shadowing
Backup modes
  - VMS /Physical
  - HSC Backup

CHAPTER 13

TUTORIAL ON FORMATTING RA DRIVES

13.1 INTRODUCTION

This section reinforces and expands the information previously presented in this course relative to formatting RA drives. It covers when and when not to use the formatter. The formatter is:

- A beneficial tool when used properly for the right reasons.
- A destructive tool when used improperly or for the wrong reasons.

IT IS IMPORTANT FOR YOU TO UNDERSTAND THE FOLLOWING OBJECTIVES FROM THE DSA TROUBLESHOOTING COURSE:

- The RCT purpose and functionality.
- The FCT purpose and functionality.
- The purpose and functionality of drive thresholds.
- The purpose and functionality of the SCRUBBER diagnostic.
- The purpose and functionality of the HSC VERIFY and DKUTIL utilities.
- The purpose and functionality of the CX/CSSE RAUTIL, VMS error log tool, and other programs used during the DSA Troubleshooting Course.
- The purpose and functionality of dynamic bad block replacement (BBR).
- The purpose and functionality of revectoring.
- The basic DSA block (sector) header concepts.

13.2 BASIC FORMATTER FUNCTIONALITY REVIEW

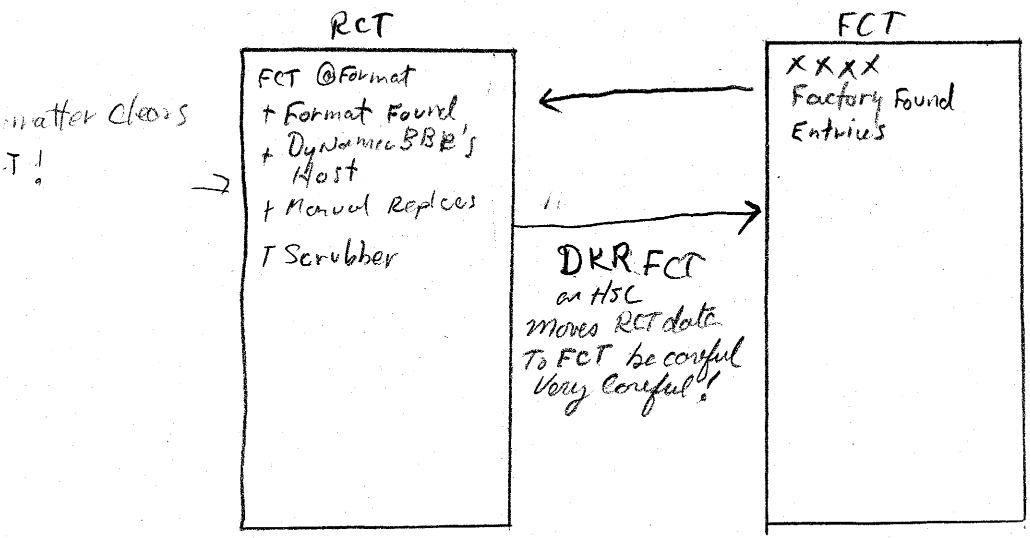
Several key concepts of formatter functionality are:

1. The formatter was designed and intended to format the media. The key functions of formatting are:
   - Write the headers (along with the associated data field, etc.) of the host LBN area, RCT area, and DBN area.

     NOTE

     DO NOT run the formatter in reconstruct mode. The XBN (FCT) area will be reformatted if you are running in reconstruct mode.

   - Replace the blocks identified as bad via the contents of the FCT (by default).
   - Replace any additional bad blocks (besides those identified in the FCT) that it finds.
   - Establish the contents and integrity of the RCT. After formatting, the RCT should contain replacement and revectoring information on all the blocks identified as bad. This will be the bad blocks as identified in the FCT plus any bad blocks that the formatter found during the formatting process.
2. The formatter was not intended to be a SCRUBBER, scanner, or diagnostic.
3. Only use the formatter on a drives in good working order.
   - Since the formatter is only intended to format media, the drive itself must be in good working order. If not, the formatter may:
     - Degrade the integrity of the logical structures on the media (such as the RCT) or make the HDA unusable.
     - Replace many good blocks. If a data path problem exists in the drive and the problem causes ECC errors (among others), then many good blocks on the media could be replaced by the formatter.
4. The formatter replaces bad blocks that are below the drive threshold.
   - The HSC formatter (not using special options) will replace all bad blocks that are two symbol in error and above.
   - The other formatters (EVRLB, ZUDK) will replace all bad blocks that are one symbol in error and above.
5. The SCRUBBER, since it is an MSCP product, only sees blocks that are bad above the drive threshold (threshold is 6 symbols in error for RA70/81/82/90, 4 symbols for the RA60, and 2 symbols for the RA80). Thus, the SCRUBBER will not replace bad blocks that are below the drive threshold.



13.3 SCRUBBER, FORMATTER, HSC50/70—WHAT REPLACES BLOCKS?

1. The HSC does controller–initiated BBR.
2. The RQDX controller (for RD disks) does controller–initiated BBR (even though the BBR algorithm used in the RQDX may not be the same as the one used by the HSC).
3. Operating systems do dynamic BBR if the operating system has the proper coding.
4. If an operating system has BBR capability and an HSC, the HSC does the BBR.
5. The SCRUBBER does BBR. The SCRUBBER was primarily intended for use when the operating system does not do dynamic BBR and there is no HSC controller.
    - UNIX (Berkeley)
    - RT–11, prior to Version 5.3
    - ELN, prior to Version 2.1
    - UNIX system V, prior to Version 2 Release 2
    - ULTRIX–11
    - DSM–11, prior to Version 3.1
    - IAS, prior to Version 3.2
    - MicroPower/PASCAL
    - ULTRIX–32, prior to Version 2

    However, the SCRUBBER is used quite successfully during the installation of a new drive or HDA. A quick pass of the SCRUBBER can clean up bad blocks up before the equipment is given to the customer.

    Also use the SCRUBBER to clean up bad blocks that the operating system is having a hard time doing. Make note of any bad blocks you wish to replace (from information in the error log, etc.), and invoke manual mode of the SCRUBBER.
6. The formatter does BBR.
    - The HSC formatter will replace blocks that are identified as bad in the format control table (FCT). It will also replace bad blocks that are above two symbols in error (or one symbol in error, using special options).
    - Since the formatter works at one (or two in HSC formatter without special options) symbols in error, it is unlikely to pass a bad block. However, it has happened.
7. HSC utilities do BBR.

    SCRUBBER–type functionality can be obtained by using the HSC utilities VERIFY and DKUTIL.

    1. VERIFY will give you a list of the bad blocks from one symbol in error to uncorrectable and header errors.
    2. Once you have obtained this list, you can decide at what level you want to make replacements.
    3. If you decide to replace all bad blocks that VERIFY shows are 4 symbols in error and above, you could use the REVECTOR command in DKUTIL. This allows you to manually replace all bad blocks that VERIFY shows are 4 symbols and above in error.

    When possible, use the HSC VERIFY/DKUTIL utilities rather than the formatter.

13.4 WHEN TO USE THE FORMATTER

Use the formatter under the following circumstances:

1. If you have a drive that has a data path problem or a worn spindle ground brush.

   Assuming you have an operating system that supports dynamic bad block replacement or an HSC controller that does controller initiated BBR, these a data path problem or a worn spindle ground brush will cause many ECC errors, header errors, and other data–related errors. BBR will most likely replace the blocks reporting the errors. However, because of the data path problem or worn spindle ground brush, most of the blocks reporting errors are not bad.

   The good blocks that have been unnecessarily replaced will begin to cause performance degradation due to revectoring.

   If you repair a drive that has a data path problem or worn spindle ground brush and the error log has accumulated lots of ECC/header errors, assume good blocks have been replaced. (VMS supports the BBR error log packet, and this indicates blocks being replaced for the data problem you have corrected. Many other operating systems do not support the BBR error log packet.) In severe cases, the good blocks that get replaced can be almost enough to fill the RCT (17,472 RBNs in the RA81).

   If you wish to see how many blocks are currently replaced, you can dump the contents of the RCT. This can be done in one of several ways:

   - HSC—Use the DISPLAY RCT command of DKUTIL.
   - HSC—Use the HSC VERIFY program.
   - UDA/KDA/KDB — Use the SCRUBBER program (VAX=EVRLK PDP–11=ZUDL).

   NOTE

   Remember there is no SCRUBBER for the MicroVAX–II environment (currently in development). For MicroVAX–II, the only way to see the RCT is with the CX/CSSE RAUTIL program (assuming the operating system is VMS).

   - RAUTIL—Use the SUM command to RAUTIL (assuming you have a copy available on site and VMS is the operating system).

   There is no official answer to how many replacements are too many. However, use the following as a guideline:

   - In an RA60, more than 600 replacements is suspect.
   - In an RA70, more than 500 replacements is suspect.
   - In an RA80, more than 400 replacements is suspect.
   - In an RA81, more than 1000 replacements is suspect.
   - In an RA82, more than 1000 replacements is suspect.
   - In an RA90, more than 1000 replacements is suspect.

   Suspect means that further analysis and determination is necessary. You must determine if scratches or other problems are causing the replacements. RAUTIL could be very helpful in this analysis.

   To compare the replacements the factory found to the contents of the RCT, use the HSC DKUTIL program. The DKUTIL command **DISPLAY FCT** or the VERIFY program shows the contents of the FCT. For other controllers, there is currently no way to see the contents of the FCT. However, the current version of the formatters will not begin formatting if a valid FCT does not exist.

CONCLUSION

If you determine that good blocks have been replaced due to a data path problem or worn spindle ground brush, use the formatter. **HOWEVER, ONLY USE THE FORMATTER AFTER YOU HAVE REPAIRED THE DISK DRIVE.**

The formatter replaces the bad blocks identified in the FCT and any other bad blocks it finds. Therefore, after correcting a data path problem (or a worn out spindle ground brush), the formatter returns the good blocks that BBR replaced into use and re–establishes the contents and integrity of the RCT to known values.

2. Use the formatter to correct a problem that relates to headers.

   Headers are never written unless you are formatting. Header errors (such as header not found, header compare error, or invalid header) also invoke BBR (assuming you have host or controller BBR available). Thus, the header errors may be corrected "dynamically." However, if you are troubleshooting intermittent header–related errors, then the formatter may help.

   Over time, the headers may diminish in amplitude. This can cause various problems, such as intermittent header–related errors. Also, under some circumstances, the automatic gain circuits may not be able to react properly, causing data–related errors (ECC type errors), when the cause is actually a low amplitude of the headers.

   The formatter is the only way to rewrite headers. Use the formatter to rewrite the headers and establish the amplitude intended for the drive "logic".

   CONCLUSION

   If the drive is working properly and the data path is not a source of errors, use the formatter. Make sure you understand the situation before deciding to use the formatter.

   If there are only a few blocks (1 to 10) reporting header errors, record the LBNs and manually replace those blocks using DKUTIL, RAUTIL, or the scrubber programs in manual mode. This will take considerably less time than formatting.

3. Although the formatter will replace bad blocks below the drive threshold, the formatter is not a "scanner" and may miss some bad blocks. In some cases, bad blocks can be at the marginal level of the drive threshold and cause intermittent errors, and never be replaced. This occurs most often in operating systems that do not have the most current BBR algorithm or that do not have BBR support (controller BBR or host BBR). In those cases, the fact that the formatters work below the drive threshold could be beneficial.

   CONCLUSION

   You may be able to eliminate many of the "marginal" bad blocks that are below the drive threshold but show up intermittently above the drive threshold (via error log entries).

4. You can also use the formatter for problems related to the media (HDA/pack) before replacing that HDA/pack. If a problem relates to the HDA/pack and you are considering replacing that HDA/pack, try the formatter first. Since the formatter rewrites everything (except the FCT), it may correct your problem and eliminate the need to replace the HDA/Pack.

13.5 WHEN NOT TO USE THE FORMATTER

This section discusses when you should use a method other than the formatter to correct a problem.

1. Avoid using the formatter as a troubleshooting tool. The formatter is not a diagnostic.
2. Avoid using the formatter on a broken drive. Using the formatter on a broken drive may cause damage to the HDA/pack.
3. Avoid using the formatter as a SCRUBBER or "scanner."
   - For scrubbing or scanning, use the utilities EVRLK-VAX (with UDA/KDB), and HSC VERIFY and DKUTIL REVECTOR.
   - Until an MDM scrubber is available for the MicroVAX II, formatting may be the only alternative. The formatter can be used as a scrubber to work below the drive threshold. However, if you have a bad block that keeps showing errors and BBR does not to replace it, manually replace it with the HSC DKUTIL REVECTOR command or use the manual mode of the scrubber. The RAUTIL VMS utility will also allow you to perform manual block replacements for RA-series drives.

If reconstruct has been run & FK Bit = 1 DKRFCT clears FK Bit

To Clear FK Bit FCT Blk 0 Word 21 <15> set to 0

if FCT is corrupted then do reconstruct if + only if FCT is corrupted Last chance to save HDA or if Bad RCT is written to FCT

Install time (do a format to clear transportation related errors)
also try scrubbing disk (multiple passes)

13.6 ITEMS OF FORMATTER INTEREST

This section outlines some items to consider about the formatter.

There could be situations where the scrubber is not available. Although the scrubber is preferred, you may have to rely on the formatter in two situations:

1. If you have a system that is on self maintenance (non-DIGITAL maintenance), the system will not have a scrubber. The scrubber is a licensed diagnostic. (However, you can bring a copy of the scrubber with you and use it.)
2. If the customer's operating system does not have dynamic BBR available.

In many cases, moving an HDA causes bad blocks to develop. If bad blocks are noted in the error log immediately after the installation of a new drive or replacement HDA, those bad blocks may be the result of the shipping process, rough handling, condensation, and so on. Those bad blocks are replaced via BBR (assuming the operating system has BBR available). You may want to check for and replace any bad blocks before allowing the customer to use the drive.

The scrubber is the best tool for detecting and replacing bad blocks. Since the replacement HDA or new drive is essentially scratch media, the scrubber could be used without backing up customer data. Three to five passes of the scrubber should eliminate any reportable bad blocks on the media.

Most often, the FCT and RCT contents match on a replacement HDA/pack or new drive. The manufacturing process scans for bad blocks and formats the media using results of the scanning process. This results in the contents of the RCT and FCT matching. If the contents of the FCT and RCT match and the drive is in good working order, formatting will not cause the loss of any critical information. After formatting, the RCT should contain the same bad block information as it contained before formatting, plus additional bad blocks that developed after the media was manufactured.

Occasionally, the FCT and the RCT do not match because the manufacturing process allows blocks to be replaced after the formatting step. During HDA burn–in, if a bad block is encountered, the technician can manually replace the suspect bad block(s).

On an HSC, determining if the FCT and RCT match is easy with DKUTIL commands (Display FCT/Display RCT). In non–HSC environments, it is not possible to make this determination.

Again, the scrubber is the best method for handling bad blocks in replacement HDAs or new drives. Use either the diagnostic scrubber or the scrubber methodology suggested for HSC environments.

Several good methods exist for obtaining the contents of the RCT, such as HSC DKUTIL or VERIFY, scrubber, and RAUTIL. If you can obtain the current contents of the RCT, you may have some information that will be necessary to "plug" back into the RCT after formatting. The formatter may replace many bad blocks that exist below the drive threshold that the scrubber is not able to see.

After using the formatter, you could dump the "new" contents of the RCT and compare it to the "old" contents. There may be some differences. If so, you could manually replace those blocks that were in the old RCT that do not show up in the new RCT. Manual replacement of blocks can be accomplished via the HSC DKUTIL REVECTOR command and manual mode of the scrubber (ZUDLXX, EVRLK). This procedure assumes that the contents of the old RCT did not contain any good blocks that had been replaced for a data path problem.

After formatting, test the drive before turning it over to the customer. The best tests for this are HSC ILEXER. For UDA, KDA, and KDB systems, use ZUDJ–PDP11 and EVRLJ–VAX.

Make sure the HDA/Pack has thermally stabilized before formatting. Stabilizing time varies, depending upon the state of the drive/HDA and the computer room environment.

13.7 FORMATTING SUMMARY

Proper use of the formatter and scrubber requires a good knowledge of DSA, the products, and the programs.

This knowledge and common sense will guide your decisions.

CAUTION

Above all, *DO NOT* use the formatter unless you have a good reason to do so.

Formatter formats DBN area by default
when formatter starts it sets Mode word to 0 in FCT

media format error if
if modeword = 0
+ Drive is mounted

To repair a disk that format was started on unintentionally
Clear out word 4
Put back SN's ! not necessary but nice
Set Mode word to 126736 (16 Bit mode)
2CT Run anal/disk/Read
Get data off
Reformat disk

Keep Good Records for your Disks

CHAPTER 14

DRIVE ERROR TOLERANCE

14.1 ILEXER SAMPLE 1

Logged to Host

Logs error to Host

corrected by ECC within Spec

```
                              SAMPLE 1

ILEXER>D>
ILEXER>D>Unit  R     Serial      Posi      Kbyte       Kbyte      Hard   Soft   Software
ILEXER>D> No   -     Number       tion     Read        Written    Error  Error  Corrected
ILEXER>D>D064     000000001F46  34874  0001371380  0000000000  00000  00000  00076
ILEXER>D>
ILEXER>D>Unit  R     Serial      Posi      Kbyte       Kbyte      Hard   Soft   Software
ILEXER>D> No   -     Number       tion     Read        Written    Error  Error  Corrected
ILEXER>D>D064     000000001F46  69698  0002740800  0000000000  00000  00000  00139
ILEXER>D>
ILEXER>D>Unit  R     Serial      Posi      Kbyte       Kbyte      Hard   Soft   Software
ILEXER>D> No   -     Number       tion     Read        Written    Error  Error  Corrected
ILEXER>D>D064     000000001F46  04420  0004110360  0000000000  00000  00000  00186
ILEXER>D>
ILEXER>D>Unit  R     Serial      Posi      Kbyte       Kbyte      Hard   Soft   Software
ILEXER>D> No   -     Number       tion     Read        Written    Error  Error  Corrected
ILEXER>D>D064     000000001F46  39109  0005479670  0000000000  00000  00000  00246
ILEXER>D>
ILEXER>D>Unit  R     Serial      Posi      Kbyte       Kbyte      Hard   Soft   Software
ILEXER>D> No   -     Number       tion     Read        Written    Error  Error  Corrected
ILEXER>D>D064     000000001F46  73822  0006848780  0000000000  00000  00000  00288
ILEXER>D>
```

14.2 ILEXER SAMPLE 2

SAMPLE 2

```
ILEXER>D>
ILEXER>D>Unit  R     Serial     Posi      Kbyte      Kbyte     Hard   Soft   Software
ILEXER>D> No   -     Number      tion     Read      Written    Error  Error  Corrected
ILEXER>D>D021     000000000802  67730  0008471300  0009078532  00000  00094  00191
ILEXER>D>
ILEXER>D>Unit  R     Serial     Posi      Kbyte      Kbyte     Hard   Soft   Software
ILEXER>D> No   -     Number      tion     Read      Written    Error  Error  Corrected
ILEXER>D>D021     000000000802  70877  0008631700  0009238632  00000  00095  00193
ILEXER>D>
ILEXER>D>Unit  R     Serial     Posi      Kbyte      Kbyte     Hard   Soft   Software
ILEXER>D> No   -     Number      tion     Read      Written    Error  Error  Corrected
ILEXER>D>D021     000000000802  74041  0008792300  0009399332  00000  00097  00199
ILEXER>D>
ILEXER>D>Unit  R     Serial     Posi      Kbyte      Kbyte     Hard   Soft   Software
ILEXER>D> No   -     Number      tion     Read      Written    Error  Error  Corrected
ILEXER>D>D021     000000000802  77205  0008952600  0009559932  00000  00098  00202
ILEXER>D>
ILEXER>D>Unit  R     Serial     Posi      Kbyte      Kbyte     Hard   Soft   Software
ILEXER>D> No   -     Number      tion     Read      Written    Error  Error  Corrected
ILEXER>D>D021     000000000802  80373  0009113400  0009720632  00000  00099  00205
ILEXER>D>
```

7 would be allowable
94 too much
need more info

- Try DBNova -
if same rate probably
not on Media
use tools to Isolate

14.3 ACCEPTABLE DRIVE ERROR RATES

RECOVERABLE R/W Errors

Sometimes referred to as SOFTWARE CORRECTED ERRORS when using HSC ILEXER for example. Errors that are correctable by ECC without Retry/Error-recovery sequences. The maximum number of recoverable R/W errors is:

For the RA60, RA70, RA80, RA81, RA82, RA90

1 error per: 10^7 bits read 1/MB

SOFT RECOVERABLE R/W Errors

Sometimes referred to as soft errors when using HSC ILEXER for example. Errors that are correctable with ECC and Retry/Error-recovery sequences. The maximum number of soft recoverable R/W errors is: Mostly 5

For the RA60, RA70, RA80, RA81, RA82, RA90

1 error per: 10^{10} bits read 1/1 GB

HARD UNRECOVERABLE R/W Errors

Sometimes referred to as hard errors when using HSC ILEXER for example. Errors that are not correctable by ECC nor by Retry/Error-recovery sequences. The maximum number of hard R/W errors is:

For the RA81, RA82, RA90

1 error per: 10^{13} bits read

For the RA60, RA70, RA80

1 error per: 10^{12} 1/100GB

RECOVERABLE SEEK Error

A seek error is any improper positioning of the heads which causes them to be at an undesired cylinder location, or a positioning operation timeout. The seek error is indicated by a seek timeout in the drive, a seek incomplete/error flag in the device, or by header verification mis-seek techniques. The maximum number of recoverable seek errors is:

For the RA60, RA70, RA80, RA81, RA82, RA90

1 error per: 10^6 seek or positioning operations

(1 million seeks)

CHAPTER 15

HSC50/70 DKUTIL USER GUIDE

15.1 INTRODUCTION

This section describes the use and operation of DKUTIL, the utility used to examine the structure or contents of a disk. It is also useful for modifying structures on the disk for repair. Operator input and common errors are described.

The DKUTIL utility was originally written to allow arbitrary blocks on a disk to be dumped along with information retained in the buffer about any error recovery done. It was intended for use in debugging utilities, diagnostics, error recovery, and bad block replacement. Several commands have been added to assist in this debugging. DKUTIL has become a general utility for displaying and modifying disk structures and data.

Unlike other utilities, DKUTIL is a command language interpreter. Initially, the user is prompted for a unit to which commands are to be applied. The program then goes into command mode prompting for a command, executing it, and then prompting for another. Execution is terminated by CTRL–C, CTRL–Y, CTRL–Z, or the EXIT command.

15.2 INITIATING DKUTIL

DKUTIL is initiated via the standard CRONIC command syntax. The user is asked for the unit number of the disk to examine:

```
DKUTIL-Q Enter unit number (U) [D0]?
```

Reply with the appropriate unit number. The first block of the FCT is read, if possible, and dumped in a format similar to what VERIFY prints. The unit is brought on line with the "ignore media format error" modifier so that drives which are improperly or incompletely formatted can be examined. If the FCT can't be read or the mode is invalid, DKUTIL prompts made for the sector size:

```
DKUTIL-Q Enter sector size (512/576) [512]?
```

The unit is placed in diagnostic mode so that the DBN area can be accessed. After the initial prompt(s), the program goes into command mode and prompts for a command:

```
DKUTIL>
```

Comment lines can be entered by prefixing them with an exclamation point (!). A null line is ignored. Entering a CTRL–Z terminates the program. Commands are executed immediately and usually take only the time necessary to print their results. Entering a CTRL–Y or CTRL–C at any time will abort the program and release the drive.

15.3 COMMAND SYNTAX

Commands, command options, and modifiers are recognized by initial substrings. For example, DUMP can entered as DUM, DU, or D. Where an initial substring can indicate one of several, the match depends on an order based on history and expected frequency of usage. Thus, D specifies DUMP, DI specifies DISPLAY, and DE specifies DEFAULT.

Some command options take optional parameters. If omitted, there are default parameters.

15.4 MODIFIERS

Some commands allow parameters. Parameters may appear anywhere after the command. Parameters are preceded by a slash (one slash for each). The following are equivalent:

```
DUMP/NOEDC RBN 0
DUMP /NOEDC RBN 0
DUMP RBN/NOEDC 0
DUMP RBN 0/NOEDC
DUMP RBN 0 /NOEDC
```

Modifiers are processed left to right and applied to the current default modifiers, if any. The default modifiers for DUMP can be changed via the DEFAULT command. The initial default modifiers for DUMP are /DATA, /EDC, and /IFERROR.

15.5 SAMPLE SESSION

The following is a sample session using DKUTIL. User input is in **boldface**.

```
 ^Y
 HSC50> RUN DKUTIL
DKUTIL-Q Enter unit number (U) [D0]?D133
    Serial Number:       0000000004
    Mode:                512
    First Formatted:     17-Nov-1858 00:35:47.48
    Date Formatted:      04-Apr-1984 00:05:09.20
    Format Instance:     6
    FCT:                 VALID
    DKUTIL> DIS/F FCT
            Factory Control Table for D133 (RA80)
    Serial Number:       0000000004
    Mode:                512
    First Formatted:     17-Nov-1858 00:35:47.48
    Date Formatted:      04-Apr-1984 00:05:09.20
    Format Instance:     6
    FCT:                 VALID
    Bad PBNs in FCT:     1 (512), 0 (576)
    Scratch Area Offset: 63
    Size (Not Last):     417
    Size (Last):         289
    Flags:               000000
    Format Version:      0
                              PBNs in 512 Byte Subtable
  (04) 244865 (LBN 237213),
  DKUTIL> REV 1000

  ERROR-W Bad Block Replacement (Success) at 04-Apr-1984 17:47:24.20
         Command Ref #   00000000
         RA80 Unit #     133.
         Err Seq #       6.
         Error Flags     80
         Event           0014
         Replace Flags   A400
         LBN             1000.
         Old RBN         32.
         New RBN         33.
         Cause Event     004A
 ERROR-I End of error.
 DKUTIL> DIS/F RCT
            Revector Control Table for D133 (RA80)

    Serial Number:       0000000004
    Flags:               000000
    LBN Being Replaced:  1000 (000000 001750)
    Replacement RBN:     33 (060000 000041)
    Bad RBN:             32 (060000 000040)
    Cache ID:            0000000000
    Cache Incarnation:   0
    Incarnation Date:    17-Nov-1858 00:00:00.00
  Bad RBN:      32,   1000 *->    33,  25512 -->   822,
  139512 -->  4500,
```

```
   RCT Statistics:        1 Bad RBNs,
                          3 Bad LBNs,
                          2 Primary Revectors,
                          1 Tertiary Revectors,
                          0 Probationary RBNs.
DKUTIL> DUMP LBN 1000
 ****** Buffer for LBN 1000 (000000 001750), MSCP Status: 000000

 Error Summary = header compare

 Original Error Bits   = 004000    BN = 1000 (000000 001750)
 Error Recovery Flags = 000       ECC Symbols Corrected = 0,0
 Error Retry Counts    = 0,1,0     Error Recovery Command = 000

     Header = 001750 030000 001750 030000 001750 030000 001750 030000

        EDC = 000105    Calculated EDC Difference = 000000

        ECC = 000000 000000 000000 000000 000000 000000
              000000 000000 000000 000003 000000 000000
DKUTIL> DIS CHAR LBN 1000

        Characteristics for LBN 1000 (000000 001750)

 Cylinder 1, Group 0, Track 4, Position 8

 PBN 1032 (000000 002010)

 Primary RBN 32 (060000 000040) in RCT Block 3 at Offset 128
DKUTIL> DIS CHAR DISK
        Drive Characteristics for D133

 Type:          RA80 (576 byte mode allowed)

 Media:         FIXED

 Cylinders:     275 LBN, 2 XBN, 2 DBN

 Geometry:      14 tracks/group, 2 groups/cylinder, 28 tracks/cylinder
                31 LBNs/track, 1 RBNs/track, 32 sectors/track, 32 XBNs
                896 XBNs/cylinder, 868 LBNs/cylinder, 28 RBNs/cylinder

 Group Offset:  16 (LBN), 16 (XBN)

 LBNs:          237212 (host), 238700 (total)

 RBNs:          7700

 XBNs:          1792

 DBNs:          1344 (read/write), 448 (read only)

 PBNs:          249984

 RCT:           465 (size), 63 (non-pad), 4 (copies)

 FCT:           480 (size), 63 (non-pad), 4 (copies)

 SDI Version:   3

 Transfer Rate: 97

 Timeouts:      3 (short), 7 (long)

 Retry Limit:   5

 Error Recover: 0 command levels

 ECC Threshold: 2 symbols

 Revision:      10 (microcode), 0 (hardware)

 Drive ID:      0A7A00000000
```

```
 Drive Type ID: 1

 DBN RO Groups: 1

 Preamble Size: 11 (data), 4 (header)
DKUTIL> DUMP RCT BLOCK 3

        ****** RCT Block 3, Copy 1 ******

 ****** Buffer for LBN 237214 (000003 117236), MSCP Status: 000000

        Data = 000000 000000 000000 000000 000000 000000 000000 000000
          +16  000000 000000 000000 000000 000000 000000 000000 000000
          +32  000000 000000 000000 000000 000000 000000 000000 000000
          +48  000000 000000 000000 000000 000000 000000 000000 000000
          +64  000000 000000 000000 000000 000000 000000 000000 000000
          +80  000000 000000 000000 000000 000000 000000 000000 000000
          +96  000000 000000 000000 000000 000000 000000 000000 000000
         +112  000000 000000 000000 000000 000000 000000 000000 000000
         +128  000000 040000 001750 030000 000000 000000 000000 000000
         +144  000000 000000 000000 000000 000000 000000 000000 000000
         +160  000000 000000 000000 000000 000000 000000 000000 000000
         +176  000000 000000 000000 000000 000000 000000 000000 000000
         +192  000000 000000 000000 000000 000000 000000 000000 000000
         +208  000000 000000 000000 000000 000000 000000 000000 000000
         +224  000000 000000 000000 000000 000000 000000 000000 000000
         +240  000000 000000 000000 000000 000000 000000 000000 000000
         +256  000000 000000 000000 000000 000000 000000 000000 000000
         +272  000000 000000 000000 000000 000000 000000 000000 000000
         +288  000000 000000 000000 000000 000000 000000 000000 000000
         +304  000000 000000 000000 000000 000000 000000 000000 000000
         +320  000000 000000 000000 000000 000000 000000 000000 000000
         +336  000000 000000 000000 000000 000000 000000 000000 000000
         +352  000000 000000 000000 000000 000000 000000 000000 000000
         +368  000000 000000 000000 000000 000000 000000 000000 000000
         +384  000000 000000 000000 000000 000000 000000 000000 000000
         +400  000000 000000 000000 000000 000000 000000 000000 000000
         +416  000000 000000 000000 000000 000000 000000 000000 000000
         +432  000000 000000 000000 000000 000000 000000 000000 000000
         +448  000000 000000 000000 000000 000000 000000 000000 000000
         +464  000000 000000 000000 000000 000000 000000 000000 000000
         +480  000000 000000 000000 000000 000000 000000 000000 000000
         +496  000000 000000 000000 000000 000000 000000 000000 000000

         EDC = 023277    Calculated EDC Difference = 000000

         DKUTIL> EXIT
```

15.6 DETAILED COMMAND DESCRIPTIONS

Following are descriptions for the DKUTIL commands. Command options are shown by separate lines in the syntax specification. Parameters are indicated in the syntax by braces ({ }) and lower case. Options which may be omitted are indicated by brackets ([ ]).

15.6.1 CHECK Command

Purpose:	To fill the host LBN area with data unique to each LBN, or to check the area for containing that data.
Syntax:	CHECK [READ] CHECK WRITE set CSSE_write_ON
Parameters:	none
Modifiers:	/BBR Normally, when a block is accessed, bad block replacement is inhibited. If this modifier is specified, bad block replacement is allowed. It will only occur, however, if the block being accessed is detected as bad by the error recovery code and is an LBN in the host area.
Usage:	If the WRITE command option is given, all LBNs in the host area will be written with a pattern which consists of the LBN number repeated enough times to fill the sector. Thus, each LBN will have unique data. If the READ command option is selected (the default if no option is given), every LBN in the host area is read and checked for the expected unique data pattern. For both options, hard errors are reported with a status line showing the MSCP status returned along with the LBN number. For the READ option, an MSCP status of ST.CMP (000007) indicates that the data did not compare to what was expected. This status will override other errors such as forced error. Therefore, for CHECK READ, a status other than ST.CMP indicates that the data in the LBN was correct.
Examples:	CHECK/BBR WRITE set CSSE_write_ON ZCH READ C

Use customer data by using ILEXER

Sequential
Read/only
multiple Process
On Customer Data

15.6.2 DEFAULT Command

Purpose:	To change the default modifiers for the DUMP command.
Syntax:	DEFAULT
Parameters:	none
Modifiers:	/IFERROR (NOIFERROR)

If this modifier is specified (default on), the error, header, and ECC fields in the buffer will be dumped if an error occurs when reading the block. When used in conjunction with the /RAW modifier, the error must occur on the reread of the block with the header code extracted from the first read.

/ERRORS (NOERRORS)

If this modifier is specified (default off), the error fields in the buffer will be dumped.

/EDC (NOEDC)

If this modifier is specified (default on), the EDC and calculated EDC fields in the buffer will be dumped.

/ECC (NOECC)

If this modifier is specified (default off), the ECC fields in the buffer will be dumped.

/DATA (NODATA)

If this modifier is specified (default on), the data in the buffer will be displayed unless the /NZ modifier has also been specified. See below.

/HEADERS (NOHEADERS)

If this modifier is specified (default off), the header fields in the buffer will be displayed.

/ALL (NONE)

This is the same as /ERRORS/EDC/ECC/DATA/HEADERS. It requests that all fields be displayed. Its opposite, /NONE, requests that no fields be displayed. In this case, only the MSCP status line will print.

/RAW (NORAW)

This modifier requests that data from the specified LBN be dumped instead of the data from the RBN, if the data had been previously replaced. The /IFERROR modifier, if in effect, applies only to the reread. This modifier only has an effect on dumping an LBN which is revectored.

/NZ (NONZ)

This modifier (default off), does not display data that is all zero. Instead, a single line indicating that the data is zero will be printed. It has no effect if the /DATA modifier is not specified (or is defaulted off).

/BBR (NOBBR)

Normally, when a block is accessed, bad block replacement is inhibited. If this modifier (default off), is specified, bad block replacement will occur. It will only occur, however, if the block being accessed is detected as bad by the error recovery code and is an LBN in the host area.

/ORIGINAL (NOORIGINAL)

When a block is accessed for dumping, the data is seen by the program twice if an error occurs. It is seen first just after the K detects the error and sends it to error recovery. It is then seen again after error recovery takes place and the data has been corrected or reread. Normally, the data is saved for displaying when it is last seen. If this modifier (default off), is specified, the data saved for display will be the data first seen.

Usage: The modifiers specified are applied to the current default modifiers for the DUMP command. The result becomes the new default.

Examples:

```
DEFAULT/NONE
DEF/RAW/NODATA
DE/A/OR/NZ
```

15.6.3 DISPLAY Command

Purpose:

To display the disk characteristics, the characteristics of a given block, the error history in the drive, the FCT, or the RCT.

Syntax:

DISPLAY ALL
DISPLAY CHARACTERISTICS DBN {block}
DISPLAY CHARACTERISTICS DISK
DISPLAY CHARACTERISTICS LBN {block}
DISPLAY CHARACTERISTICS PBN {block}
DISPLAY CHARACTERISTICS RBN {block}
DISPLAY CHARACTERISTICS XBN {block}
DISPLAY ERRORS
DISPLAY FCT
DISPLAY RCT

Parameters:

block is a number specifying the DBN, LBN, PBN, RBN, or XBN whose characteristics are to be displayed. The default radix is decimal. It can be changed to octal by prefixing the number with the letter O.

Modifiers:

/FULL

If this modifier is specified, all fields in xCT block 0 which are defined will be displayed. It applies only to the RCT and FCT command options. Normally, for the RCT option the bad block replacement and write back caching fields in RCT block 0 are only displayed if the appropriate flags in the flags field are set indicating they are currently in use (BBR or caching in progress). This modifier forces all fields to be displayed regardless of the flags settings. For the FCT option, the number of bad PBNs field is normally displayed only if the FCT is VALID. Also, the scratch area parameters, format version, and format flags are normally not displayed. This modifier forces all fields in FCT block 0 to be displayed.

/NOITEMS

If this modifier is specified, the individual items in the FCT or RCT are not displayed. It applies only to the FCT and RCT command options. If given, only the block 0 information is displayed.

Usage:

DISPLAY ALL

The disk characteristics, FCT, RCT, and error history are displayed. Because the error history in the drive is dumped by this option, do not use it for RA60 drives; the SDI command to read the error history is illegal and causes the drive to go inoperative.

DISPLAY CHARACTERISTICS DISK

The drive type, media, cylinders, geometry, group offsets, number of LBNs, number of RBNs, number of XBNs, number of DBNs, number of PBNs, RCT parameters, FCT parameters, SDI version, transfer rate, SDI time outs, SDI retry limit, error recovery command levels, ECC threshold, revision levels, drive ID, drive type ID, DBN Read/Only groups, and preamble sizes are displayed.

DISPLAY CHARACTERISTICS xBN {block}

The characteristics of the given block are displayed. For DBNs and XBNs, these are the block number in decimal and octal, cylinder, group, track, position, and PBN in decimal and octal. For RBNs, the RCT block number and offset are also displayed. For LBNs, the primary RBN number and its RCT block number and offset are also displayed. For PBNs, what is displayed depends on the type of the block: DBN, LBN, RBN, or XBN.

DISPLAY ERRORS

The error history in the drive is read from region 2, offset 0, and dumped in hexadecimal. Do not use this option used for RA60 drives; it will cause them to go inoperative.

DISPLAY FCT

The information in FCT block 0 is displayed. Certain fields will not be displayed unless the /FULL modifier is specified. The list of bad PBNs is displayed unless the /NOITEMS modifier is given. For each item in the list, the header bits, PBN number, type (DBN, LBN, RBN, or XBN), and xBN number are displayed.

DISPLAY RCT

The information in RCT block 0 is displayed. Certain fields will not be displayed unless the /FULL modifier is specified. The list of revectors, bad RBNs, and probationary RBNs are displayed unless the /NOITEMS modifier is given. For bad and probationary RBNs, just the RBN number is displayed in decimal. For revectors, the LBN number and RBN number to which it is revectored are displayed in decimal. A primary revector is distinguished by the character sequence "–>". A secondary (tertiary) revector is distinguished by the character sequence "*–>.

Examples:

```
DISPLAY/FULL ALL
DI/F A
DI C D
DIS CHAR LBN 1000
DI/NOI RCT
```

15.6.4 DUMP Command

Purpose: To dump the given block or table of blocks.

Syntax:

```
DUMP [BUFFER]
DUMP DBN [{block}]
DUMP FCT [BLOCK {number}] [COPY {copy}]
DUMP LBN [{block}]
DUMP RBN [{block}]
DUMP RCT [BLOCK {number}] [COPY {copy}]
DUMP XBN [{block}]
```

Parameters:

block is a number specifying the DBN, LBN, RBN, or XBN to be dumped. The default radix is decimal. It can be changed to octal by prefixing the number with the letter O.

number is the relative block number in the FCT or RCT to be dumped. The default radix is decimal. It can be changed to octal by prefixing the number with the letter O. The value must be in the range 1 through non-pad FCT or RCT size. That is, the first block is number 1 (not 0) and the block must be in the non-pad area.

copy specifies which copy of the given block in the FCT or RCT that is to be dumped. The first copy is number 1. The value must not exceed the number of copies.

Modifiers:

/IFERROR (NOIFERROR)

If this modifier is specified (default on), the error, header, and ECC fields in the buffer will be dumped if an error occurs when reading the block. When used in conjunction with the /RAW modifier, the error must occur on the reread of the block with the header code extracted from the first read.

/ERRORS (NOERRORS)

If this modifier is specified (default off), the error fields in the buffer will be dumped.

/EDC (NOEDC)

If this modifier is specified (default on), the EDC and calculated EDC fields in the buffer will be dumped.

/ECC (NOECC)

If this modifier is specified (default off), the ECC fields in the buffer will be dumped.

/DATA (NODATA)

If this modifier is specified (default on), the data in the buffer will be displayed unless the /NZ modifier has also been specified. See below.

/HEADERS (NOHEADERS)

If this modifier is specified (default off), the header fields in the buffer will be displayed.

/ALL (NONE)

This is the same as /ERRORS/EDC/ECC/DATA/HEADERS. It requests that all fields be displayed. Its opposite, /NONE, requests that no fields be displayed. In this case, only the MSCP status line will print.

/RAW (NORAW)

This modifier requests that data from the specified LBN be dumped instead of the data from the RBN, if the data had been previously replaced. The /IFERROR modifier, if in effect, applies only to the reread. This modifier only has an effect on dumping an LBN which is revectored.

/NZ (NONZ)

This modifier (default off), does not display data if it is all zero. Instead a single line indicating that the data is zero will be printed. It has no effect if the /DATA modifier is not specified (or is defaulted off).

/BBR (NOBBR)

Normally, when a block is accessed, bad block replacement is inhibited. If this modifier (default off), is specified bad block replacement will be allowed to occur. It will only occur, however, if the block being accessed is detected as bad by the error recovery code and is an LBN in the host area.

/ORIGINAL (NOORIGINAL)

When a block is accessed for dumping, the data is seen by the program twice if an error occurs. It is seen first just after the K detects the error and sends it to error recovery. It is then seen again after error recovery takes place and the data has been corrected or reread. Normally, the data is saved for displaying when it is last seen. If this modifier, which defaults off, is specified, the data saved for display will be the data first seen.

Usage: DUMP [BUFFER]

The current buffer is dumped subject to the given modifiers. If there is no current buffer, an error message will be printed.

DUMP xBN [{block}]

The specified DBN, LBN, RBN, or XBN is read in and dumped subject to the given modifiers. If the block number is not specified, it defaults to zero (0).

DUMP xCT [BLOCK {number}] [COPY {copy}]

If a BLOCK number is given, that block in the FCT or RCT is read in and dumped. If none is specified, every block in the non-pad area of the FCT or RCT is read in and dumped. If COPY is not specified, it defaults to copy 1.

Examples:

```
DUMP RCT BLOCK 3 COPY 4
DU/NZ RCT C 2
DU LBN 1000
D F B 2
D X
D/DATA
```

15.6.5 EXIT Command

Purpose:	To terminate execution of the program.
Syntax:	EXIT
Parameters:	none
Modifiers:	none
Usage:	The current drive is released, all resources are returned, and the program exits.
Examples:	EXIT E

15.6.6 GET Command

Purpose:	To change the current drive.
Syntax:	GET [{drive}]
Parameters:	drive is a valid drive unit specification of the form "Dnnn". If this parameter is left out, it defaults to "D000" (unit 0).
Modifiers:	/NOACQUIRE Normally, when a new drive is selected, it is first acquired for diagnostic use by the program before being brought online. In fact, the online operation will fail if it has not been acquired. However, if the drive was previously acquired by the program and not released by the /NORELEASE modifier in a previous GET command, the drive does not need to be acquired again. Under these circumstances, the GET will fail unless /NOACQUIRE is used. /NOIMF By default, a new drive is brought online with the IMF (MD.IMF) MSCP modifier which inhibits reading the FCT block 0 to determine the mode, and reading and writing of RCT block 0 to verify that the RCT is sane. If this modifier is specified, these actions will take place. /NORELEASE This command normally makes the current drive available and releases it from diagnostic use before selecting a new drive. If this modifier is specified, the current drive will be left online. It should be used with great caution because the drive left online will be in limbo until the HSC50 reboots or the drive is re-selected with a GET/NOACQUIRE and then released. /SHADOW If this modifier is specified, the drive will be brought online with the MSCP SHADOW (MD.SHD) modifier. The shadow unit (virtual unit) will be 0 and the unit will be made a part of a shadow set. This modifier must be used in conjunction with the /NOIMF modifier. /WP If this modifier is specified, the drive will be brought online with the MSCP SET WRITE PROTECT modifier (MD.SWP) and WRITE PROTECT unit flag (UF.WPS). The drive will be software or volume write protected. /NOWP If this modifier is specified, the drive will be brought online with the MSCP SET WRITE PROTECT modifier. The drive will not be software write protected.
Usage:	The current drive is released unless the /NORELEASE modifier is specified. The new drive is acquired unless the /NOACQUIRE modifier is specified (the unit was previously acquired). It is then brought online with the requested modifiers and unit flags. If the drive is nonexistent, in use, or inoperative, the user is prompted for another unit. The modifiers cannot be changed for this other unit. If the mode word in FCT block 0 is invalid or all copies of FCT block 0 are bad, the user is prompted for the sector size to use.
Examples:	GET D133 G/WP D64 G

15.6.7 MODIFY Command

Purpose: To modify a location or set of consecutive locations in the current buffer.

Syntax: MODIFY {offset} [{value} ...]

Parameters: offset is a number specifying the initial offset in the current buffer where modification is to start. The default radix is decimal. It can be changed to octal by prefixing the number with the letter O. The value is forced even.

value is a number used to modify the next consecutive word in the current buffer. The default radix is decimal. It can be changed to octal by prefixing the number with the letter O.

Modifiers: none

Usage: The specified word in the current buffer is changed to the given value. The following consecutive words are changed to the subsequent values, if any. Modification stops when the offset exceeds 574. The modified buffer can be written to an arbitrary block with the WRITE command which will recompute the checksum on the buffer.

Examples:

```
MODIFY 130 O040000
M 0 0 0 0
```

15.6.8 POP Command

Purpose: To restore the data in the current buffer from the save buffer.

Syntax: POP

Parameters: none

Modifiers: none

Usage: The data in the save buffer is restored to the current buffer. The data in the current buffer is lost.

Examples:

```
POP
P
```

15.6.9 PUSH Command

Purpose: To save the data in the current buffer in the save buffer.

Syntax: PUSH

Parameters: none

Modifiers: none

Usage: The data in the current buffer is saved in the save buffer. The data in the save buffer is lost.

Examples:

```
PUSH
PU
```

15.6.10 REVECTOR Command – (Manual LBN Replacement)

Purpose:	To force bad block replacement to occur for a given LBN.
Syntax:	REVECTOR {block}
Parameters:	block is a number specifying the LBN to be replaced. The default radix is decimal. It can be changed to octal by prefixing the number with the letter O.
Modifiers:	none
Usage:	The specified LBN is sent to the bad block replacement module to be revectored. If it is not a valid LBN or is not in the RCT, the revector will fail and an error message will be printed. Otherwise, the result of the replace attempt will be shown in the error log produced, if the appropriate level message level is enabled (INFO). The data in the replacement RBN is read from the specified LBN.
Examples:	REVECTOR 1000 R 100

15.6.11 SET Command

SET

Purpose:	To change various program parameters.
Syntax:	**SET [SIZE {size}]**
Parameters:	size specifies the new sector size to be used for the current drive. It must be either 512 or 576.
Modifiers:	none
Usage:	SET SIZE {size} The sector size is changed to the given value and the disk parameters are recomputed. This new sector size is used when doing I/O to the LBN area and is also reflected in the parameters printed by the DISPLAY CHARACTERISTICS DISK command.
Examples:	SET SIZE 576 S S 512

15.6.11.1 SET CSSE_WRITE_ON

Syntax: **SET CSSE_WRITE_ON**

Usage: This command is only applicable to the DKUTIL utility supplied with HSC software Version 390 or higher. When executed, this command will enable proper operation of the following "disk-write" commands:

CHECK WRITE
MODIFY
WRITE

This feature is intended for field service use only. The special disk-write commands will only be enabled until the user terminates execution of DKUTIL. Upon restarting DKUTIL, these commands will again be disabled until the next execution of the SET CSSE_WRITE_ON command. This command eliminates the need for a special DKUTIL ODT patch to enable the disk-write features.

NOTE

The REVECTOR command is not affected. The REVECTOR command is always enabled.

This SET command may not be abbreviated.

Example: SET CSSE_WRITE_ON

15.6.12 WRITE Command

Purpose: To write a given block or all copies of a given RCT or FCT block.

Syntax:

```
WRITE [BUFFER]
WRITE DBN [{block}]
WRITE FCT [BLOCK {number}] [COPY {copy}]
WRITE LBN [{block}]
WRITE RBN [{block}]
WRITE RCT [BLOCK {number}] [COPY {copy}]
WRITE XBN [{block}]
```

Parameters:

block   is a number specifying the DBN, LBN, RBN, or XBN to be written. The default radix is decimal. It can be changed to octal by prefixing the number with the letter O.

number   is the relative block number in the FCT or RCT to be written. The default radix is decimal. It can be changed to octal by prefixing the number with the letter O. The value must be in the range 1 through non–pad FCT or RCT size. That is, the first block is number 1 (not 0) and the block must be in the non–pad area.

copy   specifies which copy of the given block in the FCT or RCT to be written. The first copy is number 1. The value must not exceed the number of copies.

Modifiers:

/BADEDC

If this modifier is specified, the EDC written is forced to be bad (illegal). The actual EDC used is the correct EDC plus 1. Using this modifier will cause a number of error log messages to be generated. The block(s) written will cause controller errors (ST.CNT) when read.

/FE

If this modifier is specified, the EDC written will cause the block(s) to have forced errors. The actual EDC used is the complement of the correct EDC. The block(s) written will cause forced errors (ST.DAT) when read.

/BBR

Normally, when a block is accessed, bad block replacement is inhibited. If this modifier is specified bad block replacement will be allowed to occur. It will only occur, however, if the block being accessed is detected as bad by the error recovery code and is an LBN in the host area.

Usage: A new EDC is computed on the data in the current buffer and modified according to the given modifiers. The current buffer is written with this new EDC to the specified block or blocks. If the FCT or RCT option is used, the first block (actual block 0) will be written if the BLOCK parameter was not given. If the COPY parameter is not given, all copies of the block will be written.

Examples:

```
WRITE RCT BLOCK 3
WR LBN 1000
WR/BBR L 100
W/FE
W/BAD
```

15.7 COMMAND SUMMARY

CHECK	Fill or check for unique data in host LBN area. CHECK [READ] CHECK WRITE
DEFAULT	Change default modifiers for DUMP command.
DISPLAY	Display characteristics, error history, RCT, or FCT. DISPLAY ALL DISPLAY CHARACTERISTICS DBN {block} DISPLAY CHARACTERISTICS DISK DISPLAY CHARACTERISTICS LBN {block} DISPLAY CHARACTERISTICS PBN {block} DISPLAY CHARACTERISTICS RBN {block} DISPLAY CHARACTERISTICS XBN {block} DISPLAY ERRORS DISPLAY FCT DISPLAY RCT
DUMP	Dump given block or table of blocks. DUMP [BUFFER] DUMP DBN [{block}] DUMP FCT [BLOCK {number}] [COPY {copy}] DUMP LBN [{block}] DUMP RBN [{block}] DUMP RCT [BLOCK {number}] [COPY {copy}] DUMP XBN [{block}]
EXIT	Terminate execution of the program.
GET	Change the current drive. GET [{drive}]
MODIFY	Modify location(s) in the current buffer. MODIFY {offset} [{value} ...]
POP	Restore save buffer to current buffer.
PUSH	Save current buffer in save buffer.
REVECTOR	Force bad block replacement for the given LBN. REVECTOR {block}
SET	Change various program parameters. SET [SIZE {size}] SET CSSE_WRITE_ON (enables disk-write commands)
WRITE	Write a given block or all copies of a FCT or RCT block. WRITE [BUFFER] WRITE DBN [{block}] WRITE FCT [BLOCK {number}] [COPY {copy}] WRITE LBN [{block}] WRITE RBN [{block}] WRITE RCT [BLOCK {number}] [COPY {copy}] WRITE XBN [{block}]

15.8 ERRORS and INFORMATION MESSAGES

Following is a list of error and information messages which may be printed out by DKUTIL. Variable output is as follows:

n	a decimal number
par	BLOCK or COPY
parm	the part of the command in error (modifier, etc.)
status	MSCP status (an octal number)
text	the actual text in error
xBN	DBN, LBN, etc.
xCT	FCT or RCT

15.8.1 DKUTIL-S CTRL/Y or CTRL/C Abort!

This termination message is printed if the user aborts DKUTIL by typing CTRL–C or CTRL–Y.

15.8.2 DKUTIL-F Insufficient resources to RUN!

This message is printed if DKUTIL cannot acquire the necessary resources or if the disk functional code is not loaded. The program terminates after this message is printed.

15.8.3 DKUTIL-F Drive went OFFLINE!

This message is printed if the unit selected goes off line while DKUTIL is running. The program terminates after this message is printed.

15.8.4 DKUTIL-F I/O request was rejected!

This message is printed if the diagnostic interface (DDUSUB) rejects a request to start an I/O operation. It indicates a bug in DKUTIL and should be reported to field service. The program terminates after this message is printed.

15.8.5 DKUTIL-E Illegal response to start-up question.

This message is printed if an invalid response is entered for a start–up question or a prompt for the GET command. The user is prompted again with the same question.

15.8.6 DKUTIL-E Nonexistent unit number.

This error message is printed if the unit number entered does not correspond to any known unit. The user is prompted again for a unit number.

15.8.7 DKUTIL-E Unit is not available.

This message is printed if the unit requested is unavailable. It may be in use by a host or another diagnostic. It may be inoperative. The user is prompted again for another unit.

15.8.8 DKUTIL-E Cannot ONLINE unit.

This message is printed if the requested unit is available but the ONLINE command failed. The unit is released, and the user is prompted again for another unit.

15.8.9 DKUTIL-E Invalid decimal number.

This message is printed if the user entered an invalid decimal number in a command line.

15.8.10 DKUTIL-E Invalid octal number.

This message is printed if the user entered an invalid octal number in a command line.

15.8.11 DKUTIL-E Missing parameter.

This message is printed if a command line is entered and a required parameter is missing.

15.8.12 DKUTIL-E There is no buffer to dump.

This message is printed if the DUMP BUFFER command is entered and there is no current buffer. This can only happen if a drive has just been selected.

15.8.13 DKUTIL-E Missing modifier (only a slash (/) was specified).

This message is printed if a command line is entered with a slash (/) not followed by a modifier.

15.8.14 DKUTIL-E SDI command was unsuccessful.

This message is printed when an SDI command is rejected by the drive. A DISPLAY ERRORS command for a RA60 drive will generate this message.

15.8.15 DKUTIL-E n is an invalid par number; maximum is n.

This message is printed if an out–of–range number is entered for a BLOCK or COPY value for either the DUMP or the WRITE command.

15.8.16 DKUTIL-E "text" is an invalid parm.

This generic error message is printed when an invalid command, command option, modifier, block type, or SET option is specified in a command line.

15.8.17 DKUTIL-E Invalid block number for xBN space.

This message is printed if the block number specified for a DISPLAY CHARACTERISTICS xBN command is out–of–range for the given space.

15.8.18 DKUTIL-E Copy n of xCT Block n (xBN n) is bad.

This message is printed for FCT or RCT blocks which cannot be read correctly with error recovery. It will occur when the FCT or RCT is being read just after a drive has been selected or for the DISPLAY FCT or DISPLAY RCT command.

15.8.19 DKUTIL-E All copies of of xCT Block n are bad.

This message is printed for FCT or RCT blocks where all copies are bad. It will occur when the FCT or RCT is being read just after a drive has been selected or for the DISPLAY FCT or DISPLAY RCT command.

15.8.20 DKUTIL-E Could not write xBN n, MSCP Status: status

This message is printed if a write (for the WRITE command) fails.

15.8.21 DKUTIL-E Invalid sector size; only 512 and 576 are legal.

This message is printed if the sector size entered for the SET SIZE command is not 512 or 576.

15.8.22 DKUTIL-E Revector for LBN n failed, MSCP Status:

This message is printed if a revector (for the REVECTOR command) fails.

15.8.23 DKUTIL-E CHECK READ for LBN n failed, MSCP Status:

This message is printed for any LBN read, by CHECK READ, that has a non-zero MSCP status returned.

15.8.24 DKUTIL-E CHECK WRITE for LBN n failed, MSCP Status:

This message is printed for any LBN written, by CHECK WRITE, that has a non-zero MSCP status returned.

15.9 DKUTIL Lab Samples

```
HSC50> RUN DD1:DKUTIL

DKUTIL functionality has changed in this release.
It no longer prompts the user for the drive number.
Instead, use the GET Dxxx command.

DKUTIL> GET D230

   Serial Number:        0000160992
   Mode:                 512    (126736) :: Mode byte   0 means modeword wrong "Corrupt
                                                          a incomplete format
   First Formatted:      17-Mar-1988 18:00:58.00
   Date Formatted:       18-Mar-1988 00:00:00.00
   Format Instance:      1
   FCT:                  VALID - NULL   FK Bit Set

DKUTIL>
```

```
DKUTIL> DISP CHARACTERISTICS DISK
        Drive Characteristics for D0230

 Type:          RA81
 Media:         FIXED
 Cylinders:     1252 LBN, 4 XBN, 2 DBN
 Geometry:      1 track/group, 14 groups/cylinder, 14 tracks/cylinder
                51 LBNs/track, 1 RBN/track, 52 sectors/track, 52 XBNs/track
                728 XBNs/cylinder, 714 LBNs/cylinder, 14 RBNs/cylinder
 Group Offset:  14 (LBN), 14 (XBN)
 LBNs:          891072 (host), 893928 (total)
 RBNs:          17528
 XBNs:          2912
 DBNs:          728 (read/write), 728 (read only)
 PBNs:          915824
 RCT:           765 (size), 139 (non-pad), 4 (copies)
 FCT:           780 (size), 139 (non-pad), 4 (copies)
 SDI Version:   3
 Transfer Rate: 174
 Timeouts:      3 (short), 7 (long)
 Retry Limit:   5
 Error Recover: 0 command levels
 ECC Threshold: 6 symbols
 Revision:      8 (microcode), 8 (hardware)
 Drive ID:      010C00030000
 Drive Type ID: 5
 DBN RO Groups: 14
 Preamble Size: 19 (data), 12 (header)
DKUTIL>
```

V3.7 + 3.9

Running on RA60 will corrupt DKUTIL must use purge & reload DKUTIL

```
DKUTIL> DISPLAY ERRORS

  *----------------------------------------------------------------------*
  |           Disk Drive Internal Error Log Display                      |
  *----------------------------------------------------------------------*
  | This command will display the internal error log of disk drives      |
  | that support internal error logging.                                 |
  |                                                                      |
  | For the RA80, RA81 and RA82 only 16 bytes of error log data will be  |
  | displayed.  For the RA60, no error log is implemented.  For later    |
  | drives, the internal error log data will be displayed.               |
  *----------------------------------------------------------------------*

  *----------------------------------------------------------------------*
   Region 2 Data (byte 0 on the right is the oldest Drive Error)
  *----------------------------------------------------------------------*

   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

DKUTIL>
```

Errors on silo may be more useful than error logs use these first

```
DKUTIL> DISPLAY RCT

   Serial Number:        0000160992           }
   Flags:                000000               }
                                              } comes from control block
   LBN Being Replaced:   0 (000000 000000)    }
   Replacement RBN:      0 (060000 000000)    }

  11403 -->   223,  13822 -->   271,  14536 -->   285,
  15293 -->   299,  22714 -->   445,  51330 -->  1006,
 Bad RBN:    1475,  83559 -->  1638,  93562 -->  1834,
 100244 -->  1965, 114133 -->  2237, 114847 -->  2251,
 115561 -->  2265, 116275 -->  2279, 116989 -->  2293,
 117394 -->  2301, 117703 -->  2307, 120021 -->  2353,
 121987 -->  2391, 150682 -->  2954, 164702 -->  3229,
 171620 -->  3365, 181380 -->  3556, 192415 -->  3772,
 193520 -->  3794, 210660 -->  4130, 212809 -->  4172,
 225652 -->  4424, 231370 -->  4536, 242077 -->  4746,
 253500 -->  4970, 254214 -->  4984, 262070 -->  5138,
 266355 -->  5222, 272069 -->  5334, 272623 -->  5345,
 274359 -->  5379, 275905 -->  5409, 282789 -->  5544,
 305893 -->  5997, 308464 -->  6048, 317579 -->  6227,
 340423 *->  6674, 365116 -->  7159, 365829 -->  7173,
 374947 -->  7351, 379896 -->  7448, 389162 -->  7630,
 389586 -->  7638, 390300 -->  7652, 391014 -->  7666,
 391728 -->  7680, 420792 -->  8250, 433639 -->  8502,
 442706 -->  8680, 446995 -->  8764, 493817 -->  9682,
 497662 -->  9758, 507686 -->  9954, 516964 --> 10136,
 567338 --> 11124, 567925 --> 11135, 577411 --> 11321,
 578125 --> 11335, 614076 --> 12040, 640975 --> 12568,
 659759 --> 12936, 671472 --> 13166, 671958 --> 13175,
 672672 --> 13189, 680737 --> 13347, 681451 --> 13361,
 682165 --> 13375, 682879 --> 13389, 684307 *-> 13417,
 685021 --> 13431, 686449 --> 13459, 687163 --> 13473,
 687877 --> 13487, 688591 --> 13501, 689305 --> 13515,
 699525 --> 13716, 721806 --> 14153, 729217 --> 14298,
 744479 --> 14597, 746413 --> 14635, 768332 --> 15065,
 773565 --> 15167, 790535 --> 15500, 800493 --> 15695,
 814216 --> 15965, 814930 --> 15979, 815644 --> 15993,
 859503 --> 16853, 869680 --> 17052, 885017 --> 17353,
 889896 --> 17448,

   RCT Statistics:           1 Bad RBN
                            96 Bad LBNs
                            94 Primary Revectors
                        * =  2 Non-Primary Revectors
                             0 Probationary RBNs

DKUTIL>
```

```
DKUTIL> DISP RCT/FULL
           Revector Control Table for D0230 (RA81)

   Serial Number:       0000160992
   Flags:               000000

   LBN Being Replaced:  0 (000000 000000)
   Replacement RBN:     0 (060000 000000)
   Bad RBN:             0 (060000 000000)

   Cache ID:            0000000000
   Cache Incarnation:   0
   Incarnation Date:    17-Nov-1858 00:00:00.00

  11403 -->   223,  13822 -->   271,  14536 -->   285,
  15293 -->   299,  22714 -->   445,  51330 -->  1006,
 Bad RBN:    1475,  83559 -->  1638,  93562 -->  1834,
 100244 -->  1965, 114133 -->  2237, 114847 -->  2251,
 115561 -->  2265, 116275 -->  2279, 116989 -->  2293,
 117394 -->  2301, 117703 -->  2307, 120021 -->  2353,
 121987 -->  2391, 150682 -->  2954, 164702 -->  3229,
 171620 -->  3365, 181380 -->  3556, 192415 -->  3772,
 193520 -->  3794, 210660 -->  4130, 212809 -->  4172,
 225652 -->  4424, 231370 -->  4536, 242077 -->  4746,
 253500 -->  4970, 254214 -->  4984, 262070 -->  5138,
 266355 -->  5222, 272069 -->  5334, 272623 -->  5345,
 274359 -->  5379, 275905 -->  5409, 282789 -->  5544,
 305893 -->  5997, 308464 -->  6048, 317579 -->  6227,
 340423 *->  6674, 365116 -->  7159, 365829 -->  7173,
 374947 -->  7351, 379896 -->  7448, 389162 -->  7630,
 389586 -->  7638, 390300 -->  7652, 391014 -->  7666,
 391728 -->  7680, 420792 -->  8250, 433639 -->  8502,
 442706 -->  8680, 446995 -->  8764, 493817 -->  9682,
 497662 -->  9758, 507686 -->  9954, 516964 --> 10136,
 567338 --> 11124, 567925 --> 11135, 577411 --> 11321,
 578125 --> 11335, 614076 --> 12040, 640975 --> 12568,
 659759 --> 12936, 671472 --> 13166, 671958 --> 13175,
 672672 --> 13189, 680737 --> 13347, 681451 --> 13361,
 682165 --> 13375, 682879 --> 13389, 684307 *-> 13417,
 685021 --> 13431, 686449 --> 13459, 687163 --> 13473,
 687877 --> 13487, 688591 --> 13501, 689305 --> 13515,
 699525 --> 13716, 721806 --> 14153, 729217 --> 14298,
 744479 --> 14597, 746413 --> 14635, 768332 --> 15065,
 773565 --> 15167, 790535 --> 15500, 800493 --> 15695,
 814216 --> 15965, 814930 --> 15979, 815644 --> 15993,
 859503 --> 16853, 869680 --> 17052, 885017 --> 17353,
 889896 --> 17448, 891071 --> 17471,

   RCT Statistics:          1 Bad RBN
                           96 Bad LBNs
                           94 Primary Revectors
                            2 Non-Primary Revectors
                            0 Probationary RBNs

DKUTIL>
```

```
DKUTIL> DISPLAY FCT

         Factory Control Table for D0230 (RA81)

   Serial Number:         0000160992
   Mode:                  512
   First Formatted:       17-Mar-1988 18:00:58.00
   Date Formatted:        18-Mar-1988 00:00:00.00
   Format Instance:       1
   FCT:                   VALID

   Bad PBNs in FCT:       98 (512), 0 (576)
                                PBNs in 512 Byte Subtable
 (04) 910999 (LBN 893514), (14) 907296 (LBN 889896), (14) 902364 (LBN 885017),
 (14) 886732 (LBN 869680), (14) 876406 (LBN 859503), (14) 831655 (LBN 815644),
 (14) 830927 (LBN 814930), (14) 830199 (LBN 814216), (14) 816150 (LBN 800493),
 (14) 806011 (LBN 790535), (14) 788698 (LBN 773565), (14) 783411 (LBN 768332),
 (14) 761066 (LBN 746413), (14) 759046 (LBN 744479), (14) 743519 (LBN 729217),
 (14) 735985 (LBN 721806), (14) 713277 (LBN 699525), (14) 702786 (LBN 689305),
 (14) 702058 (LBN 688591), (14) 701330 (LBN 687877), (14) 700602 (LBN 687163),
 (14) 699874 (LBN 686449), (14) 698418 (LBN 685021), (14) 697690 (LBN 684307),
 (14) 696234 (LBN 682879), (14) 695506 (LBN 682165), (14) 694778 (LBN 681451),
 (14) 694050 (LBN 680737), (14) 685875 (LBN 672672), (14) 685147 (LBN 671958),
 (14) 684670 (LBN 671472), (14) 672695 (LBN 659759), (14) 653579 (LBN 640975),
 (14) 626116 (LBN 614076), (11) 589430 (LBN 578125), (11) 588702 (LBN 577411),
 (14) 579026 (LBN 567925), (14) 578470 (LBN 567338), (14) 527100 (LBN 516964),
 (14) 517640 (LBN 507686), (14) 507420 (LBN 497662), (14) 503507 (LBN 493817),
 (14) 455759 (LBN 446995), (14) 451386 (LBN 442706), (14) 442145 (LBN 433639),
 (14) 429046 (LBN 420792), (14) 399364 (LBN 391728), (14) 398636 (LBN 391014),
 (14) 397908 (LBN 390300), (14) 397180 (LBN 389586), (14) 396792 (LBN 389162),
 (11) 387344 (LBN 379896), (14) 382260 (LBN 374947), (14) 373020 (LBN 365829),
 (14) 372293 (LBN 365116), (14) 347081 (LBN 340423), (14) 323804 (LBN 317579),
 (11) 314512 (LBN 308464), (11) 311856 (LBN 305893), (14) 288333 (LBN 282789),
 (11) 281280 (LBN 275905), (14) 279728 (LBN 274359), (14) 277966 (LBN 272623),
 (14) 277403 (LBN 272069), (14) 271577 (LBN 266355), (14) 267208 (LBN 262070),
 (14) 259198 (LBN 254214), (14) 258470 (LBN 253500), (14) 246823 (LBN 242077),
 (14) 235906 (LBN 231370), (14) 230076 (LBN 225652), (14) 216981 (LBN 212809),
 (14) 214790 (LBN 210660), (14) 197314 (LBN 193520), (14) 196167 (LBN 192415),
 (14) 184936 (LBN 181380), (14) 175003 (LBN 171620), (14) 167953 (LBN 164702),
 (11) 153636 (LBN 150682), (14) 124376 (LBN 121987), (14) 122388 (LBN 120021),
 (14) 120008 (LBN 117703), (14) 119661 (LBN 117394), (14) 119280 (LBN 116989),
 (14) 118552 (LBN 116275), (14) 117824 (LBN 115561), (14) 117096 (LBN 114847),
 (14) 116368 (LBN 114133), (14) 102227 (LBN 100244), (14)  95396 (LBN  93562),
 (14)  85197 (LBN  83559), (04)  76717 (RBN   1475), (14)  52348 (LBN  51330),
 (14)  23157 (LBN  22714), (14)  15558 (LBN  15293), (14)  14839 (LBN  14536),
 (14)  14111 (LBN  13822), (14)  11600 (LBN  11403),

DKUTIL>
```

16 bit will have just 512
18 bit will have both

can be DBN's
RBN's
XBN's
LBN's

```
DKUTIL> DISPLAY CHARA LBN 100
                                                Header Code
        Characteristics for LBN 100 (000000 000144)
 Cylinder 0, Group 1, Track 0, Position 11
 PBN 63 (000000 000077)                                          in bytes
 Primary RBN 1 (060000 000001) in RCT Block 3 at Offset 4
DKUTIL>
```

```
DKUTIL> DISPLAY CHARA DBN 2

        Characteristics for DBN 2 (140000 000002)
 Cylinder 1256, Group 0, Track 0, Position 2
 PBN 914370 (000015 171702)

DKUTIL>
```

DKUTIL Lab Sample – 9

```
DKUTIL> DISPLAY CHARA RBN 24

        Characteristics for RBN 24 (060000 000030)
 Cylinder 1, Group 10, Track 0, Position 35
 PBN 1283 (000000 002403)
 Located in RCT Block 3 at Offset 96

DKUTIL>
```

```
DKUTIL> DISPLAY CHARA XBN 400

        Characteristics for XBN 400 (120000 000620)
 Cylinder 1252, Group 7, Track 0, Position 30
 PBN 911850 (000015 164752)
DKUTIL>
```

```
DKUTIL> DUMP LBN 100/ALL

 ****** Buffer for LBN 100 (000000 000144), MSCP Status: 000000
                          Not always Valid Look@ HSC error log
 Error Summary =

 Original Error Bits   = 000000      BN = 0 (000000 000000)
 Error Recovery Flags = 000          ECC Symbols Corrected = 0,0
 Error Retry Counts    = 0,0,0       Error Recovery Command = 000
               Lower   Upper
     Header = 000144 000000 000144 000000 000144 000000 000144 000000
Octal=100 decimal ↑  ↑ HDR Code
       Data = 063146 177776 177774 177770 177760 177740 177700 177600
         +16 177400 177000 176000 174000 170000 160000 140000 100000
         +32 000000 177776 177774 177770 177760 177740 177700 177600
         +48 177400 177000 176000 174000 170000 160000 140000 100000
         +64 000000 177776 177774 177770 177760 177740 177700 177600
         +80 177400 177000 176000 174000 170000 160000 140000 100000
         +96 000000 177776 177774 177770 177760 177740 177700 177600
        +112 177400 177000 176000 174000 170000 160000 140000 100000
        +128 000000 177776 177774 177770 177760 177740 177700 177600
        +144 177400 177000 176000 174000 170000 160000 140000 100000
        +160 000000 177776 177774 177770 177760 177740 177700 177600
        +176 177400 177000 176000 174000 170000 160000 140000 100000
        +192 000000 177776 177774 177770 177760 177740 177700 177600
        +208 177400 177000 176000 174000 170000 160000 140000 100000
        +224 000000 177776 177774 177770 177760 177740 177700 177600
        +240 177400 177000 176000 174000 170000 160000 140000 100000
        +256 000000 177776 177774 177770 177760 177740 177700 177600
        +272 177400 177000 176000 174000 170000 160000 140000 100000
        +288 000000 177776 177774 177770 177760 177740 177700 177600
        +304 177400 177000 176000 174000 170000 160000 140000 100000
        +320 000000 177776 177774 177770 177760 177740 177700 177600
        +336 177400 177000 176000 174000 170000 160000 140000 100000
        +352 000000 177776 177774 177770 177760 177740 177700 177600
        +368 177400 177000 176000 174000 170000 160000 140000 100000
        +384 000000 177776 177774 177770 177760 177740 177700 177600
        +400 177400 177000 176000 174000 170000 160000 140000 100000
        +416 000000 177776 177774 177770 177760 177740 177700 177600
        +432 177400 177000 176000 174000 170000 160000 140000 100000
        +448 000000 177776 177774 177770 177760 177740 177700 177600
        +464 177400 177000 176000 174000 170000 160000 140000 100000
        +480 000000 177776 177774 177770 177760 177740 177700 177600
        +496 177400 177000 176000 174000 170000 160000 140000 100000
                                                                 0 = ok
        EDC = 063043    Calculated EDC Difference = 000000 — Most important, 17777 = F

        ECC = 000000 000000 000000 000000 000000 000000 — not always read the same
              000000 000000 010400 000000 000000 000000   in DKUTIL

DKUTIL>
```

```
DKUTIL> DUMP/ALL DBN 123

 ****** Buffer for DBN 123 (140000 000173), MSCP Status: 000000

 Error Summary =

 Original Error Bits   = 000000     BN = 0 (000000 000000)
 Error Recovery Flags = 000        ECC Symbols Corrected = 0,0
 Error Retry Counts    = 0,0,0      Error Recovery Command = 000
                       ⌐USABLE DBN
     Header = 000173 140000 000173 140000 000173 140000 000173 140000

       Data = 177777 155555 133333 066666 155555 133333 066666 155555
         +16  133333 066666 155555 133333 066666 155555 133333 066666
         +32  155555 133333 066666 155555 133333 066666 155555 133333
         +48  066666 155555 133333 066666 155555 133333 066666 155555
         +64  133333 066666 155555 133333 066666 155555 133333 066666
         +80  155555 133333 066666 155555 133333 066666 155555 133333
         +96  066666 155555 133333 066666 155555 133333 066666 155555
        +112  133333 066666 155555 133333 066666 155555 133333 066666
        +128  155555 133333 066666 155555 133333 066666 155555 133333
        +144  066666 155555 133333 066666 155555 133333 066666 155555
        +160  133333 066666 155555 133333 066666 155555 133333 066666
        +176  155555 133333 066666 155555 133333 066666 155555 133333
        +192  066666 155555 133333 066666 155555 133333 066666 155555
        +208  133333 066666 155555 133333 066666 155555 133333 066666
        +224  155555 133333 066666 155555 133333 066666 155555 133333
        +240  066666 155555 133333 066666 155555 133333 066666 155555
        +256  133333 066666 155555 133333 066666 155555 133333 066666
        +272  155555 133333 066666 155555 133333 066666 155555 133333
        +288  066666 155555 133333 066666 155555 133333 066666 155555
        +304  133333 066666 155555 133333 066666 155555 133333 066666
        +320  155555 133333 066666 155555 133333 066666 155555 133333
        +336  066666 155555 133333 066666 155555 133333 066666 155555
        +352  133333 066666 155555 133333 066666 155555 133333 066666
        +368  155555 133333 066666 155555 133333 066666 155555 133333
        +384  066666 155555 133333 066666 155555 133333 066666 155555
        +400  133333 066666 155555 133333 066666 155555 133333 066666
        +416  155555 133333 066666 155555 133333 066666 155555 133333
        +432  066666 155555 133333 066666 155555 133333 066666 155555
        +448  133333 066666 155555 133333 066666 155555 133333 066666
        +464  155555 133333 066666 155555 133333 066666 155555 133333
        +480  066666 155555 133333 066666 155555 133333 066666 155555
        +496  133333 066666 155555 133333 066666 155555 133333 066666

        EDC = 030206    Calculated EDC Difference = 000000

        ECC = 000000 000000 000000 000000 000000 000000
              000000 000000 007400 000002 000000 000000

DKUTIL>
```

```
DKUTIL> DUMP/ALL RBN 2000

 ****** Buffer for RBN 2000 (060000 003720), MSCP Status: 000010 Status event code

 Error Summary = EDC  Forced error is a type of EDC error

 Original Error Bits   = 000020    BN = 2000 (060000 003720)
 Error Recovery Flags = 000       ECC Symbols Corrected = 0,0
 Error Retry Counts    = 0,0,0     Error Recovery Command = 000
                        ←USABLE RBN
     Header = 003720 060000 003720 060000 003720 060000 003720 060000

       Data = 177777 155555 133333 066666 155555 133333 066666 155555
         +16 133333 066666 155555 133333 066666 155555 133333 066666
         +32 155555 133333 066666 155555 133333 066666 155555 133333
         +48 066666 155555 133333 066666 155555 133333 066666 155555
         +64 133333 066666 155555 133333 066666 155555 133333 066666
         +80 155555 133333 066666 155555 133333 066666 155555 133333
         +96 066666 155555 133333 066666 155555 133333 066666 155555
        +112 133333 066666 155555 133333 066666 155555 133333 066666
        +128 155555 133333 066666 155555 133333 066666 155555 133333
        +144 066666 155555 133333 066666 155555 133333 066666 155555
        +160 133333 066666 155555 133333 066666 155555 133333 066666
        +176 155555 133333 066666 155555 133333 066666 155555 133333
        +192 066666 155555 133333 066666 155555 133333 066666 155555
        +208 133333 066666 155555 133333 066666 155555 133333 066666
        +224 155555 133333 066666 155555 133333 066666 155555 133333  Dec STD format F
        +240 066666 155555 133333 066666 155555 133333 066666 155555
        +256 133333 066666 155555 133333 066666 155555 133333 066666
        +272 155555 133333 066666 155555 133333 066666 155555 133333
        +288 066666 155555 133333 066666 155555 133333 066666 155555
        +304 133333 066666 155555 133333 066666 155555 133333 066666
        +320 155555 133333 066666 155555 133333 066666 155555 133333
        +336 066666 155555 133333 066666 155555 133333 066666 155555
        +352 133333 066666 155555 133333 066666 155555 133333 066666
        +368 155555 133333 066666 155555 133333 066666 155555 133333
        +384 066666 155555 133333 066666 155555 133333 066666 155555
        +400 133333 066666 155555 133333 066666 155555 133333 066666
        +416 155555 133333 066666 155555 133333 066666 155555 133333
        +432 066666 155555 133333 066666 155555 133333 066666 155555
        +448 133333 066666 155555 133333 066666 155555 133333 066666
        +464 155555 133333 066666 155555 133333 066666 155555 133333
        +480 066666 155555 133333 066666 155555 133333 066666 155555
        +496 133333 066666 155555 133333 066666 155555 133333 066666

        EDC = 147571    Calculated EDC Difference = 177777 = Forced error

        ECC = 000000 000000 000000 000000 000000 000000
              000000 000000 006400 000001 000000 000000

DKUTIL>
```

```
DKUTIL> DUMP/ALL RBN 223

 ****** Buffer for RBN 223 (060000 000337), MSCP Status: 000000

 Error Summary =

 Original Error Bits   = 000000    BN = 0 (000000 000000)
 Error Recovery Flags = 000       ECC Symbols Corrected = 0,0
 Error Retry Counts   = 0,0,0     Error Recovery Command = 000

     Header = 000337 060000 000337 060000 000337 060000 000337 060000

       Data = 073567 133331 133331 133331 133331 133331 133331 133331
         +16  133331 133331 133331 133331 133331 133331 133331 133331
         +32  133331 133331 133331 133331 133331 133331 133331 133331
         +48  133331 133331 133331 133331 133331 133331 133331 133331
         +64  133331 133331 133331 133331 133331 133331 133331 133331
         +80  133331 133331 133331 133331 133331 133331 133331 133331
         +96  133331 133331 133331 133331 133331 133331 133331 133331
        +112  133331 133331 133331 133331 133331 133331 133331 133331
        +128  133331 133331 133331 133331 133331 133331 133331 133331
        +144  133331 133331 133331 133331 133331 133331 133331 133331
        +160  133331 133331 133331 133331 133331 133331 133331 133331
        +176  133331 133331 133331 133331 133331 133331 133331 133331
        +192  133331 133331 133331 133331 133331 133331 133331 133331
        +208  133331 133331 133331 133331 133331 133331 133331 133331
        +224  133331 133331 133331 133331 133331 133331 133331 133331
        +240  133331 133331 133331 133331 133331 133331 133331 133331  Diag Pattern
        +256  133331 133331 133331 133331 133331 133331 133331 133331
        +272  133331 133331 133331 133331 133331 133331 133331 133331
        +288  133331 133331 133331 133331 133331 133331 133331 133331
        +304  133331 133331 133331 133331 133331 133331 133331 133331
        +320  133331 133331 133331 133331 133331 133331 133331 133331
        +336  133331 133331 133331 133331 133331 133331 133331 133331
        +352  133331 133331 133331 133331 133331 133331 133331 133331
        +368  133331 133331 133331 133331 133331 133331 133331 133331
        +384  133331 133331 133331 133331 133331 133331 133331 133331
        +400  133331 133331 133331 133331 133331 133331 133331 133331
        +416  133331 133331 133331 133331 133331 133331 133331 133331
        +432  133331 133331 133331 133331 133331 133331 133331 133331
        +448  133331 133331 133331 133331 133331 133331 133331 133331
        +464  133331 133331 133331 133331 133331 133331 133331 133331
        +480  133331 133331 133331 133331 133331 133331 133331 133331
        +496  133331 133331 133331 133331 133331 133331 133331 133331

        EDC = 140753    Calculated EDC Difference = 000000  OK

        ECC = 000000 000000 000000 000000 000000 000000
              000000 000000 030000 000002 000000 000000

DKUTIL>
```

```
DKUTIL> DUMP/ALL XBN 0

 ****** Buffer for XBN 0 (120000 000000), MSCP Status: 000000

 Error Summary =

 Original Error Bits  = 000000    BN = 0 (000000 000000)
 Error Recovery Flags = 000       ECC Symbols Corrected = 0,0
 Error Retry Counts   = 0,0,0     Error Recovery Command = 000
                      ┌USABLE XBN
     Header = 000000 120000 000000 120000 000000 120000 000000 120000
                 ┌──→Mode word = 512 byte Mode
       Data = 126736 000001 072340 000002 000000 000000 024400 006522
         +16  177626 000220 100000 032532 177710 000220 000142 000000
         +32  000000 000000 000213 001201 000661 000000 000000 000000
         +48  000000 000000 000000 000000 000000 000000 000000 000000
         +64  000000 000000 000000 000000 000000 000000 000000 000000
         +80  000000 000000 000000 000000 000000 000000 000000 000000
         +96  000000 000000 000000 000000 000000 000000 000000 000000
        +112  000000 000000 000000 000000 000000 000000 000000 000000
        +128  000000 000000 000000 000000 000000 000000 000000 000000
        +144  000000 000000 000000 000000 000000 000000 000000 000000
        +160  000000 000000 000000 000000 000000 000000 000000 000000
        +176  000000 000000 000000 000000 000000 000000 000000 000000
        +192  000000 000000 000000 000000 000000 000000 000000 000000
        +208  000000 000000 000000 000000 000000 000000 000000 000000
        +224  000000 000000 000000 000000 000000 000000 000000 000000
        +240  000000 000000 000000 000000 000000 000000 000000 000000
        +256  000000 000000 000000 000000 000000 000000 000000 000000
        +272  000000 000000 000000 000000 000000 000000 000000 000000
        +288  000000 000000 000000 000000 000000 000000 000000 000000
        +304  000000 000000 000000 000000 000000 000000 000000 000000
        +320  000000 000000 000000 000000 000000 000000 000000 000000
        +336  000000 000000 000000 000000 000000 000000 000000 000000
        +352  000000 000000 000000 000000 000000 000000 000000 000000
        +368  000000 000000 000000 000000 000000 000000 000000 000000
        +384  000000 000000 000000 000000 000000 000000 000000 000000
        +400  000000 000000 000000 000000 000000 000000 000000 000000
        +416  000000 000000 000000 000000 000000 000000 000000 000000
        +432  000000 000000 000000 000000 000000 000000 000000 000000
        +448  000000 000000 000000 000000 000000 000000 000000 000000
        +464  000000 000000 000000 000000 000000 000000 000000 000000
        +480  000000 000000 000000 000000 000000 000000 000000 000000
        +496  000000 000000 000000 000000 000000 000000 000000 000000

        EDC = 124152    Calculated EDC Difference = 000000

        ECC = 000000 000000 000000 000000 000000 000000
              000000 000000 007400 000001 000000 000000

DKUTIL>
```

MSB = FK bit

FCT Valid bit 15

FCT invalid bit 15

Generally from a

Block 0

Logical Command

```
DKUTIL> DUMP/ALL FCT BLOCK 1 COPY 1

        ****** FCT Block 1, Copy 1 ******

 ****** Buffer for XBN 0 (120000 000000), MSCP Status: 000000

 Error Summary =

 Original Error Bits   = 000000    BN = 0 (000000 000000)
 Error Recovery Flags = 000       ECC Symbols Corrected = 0,0
 Error Retry Counts   = 0,0,0     Error Recovery Command = 000

     Header = 000000 120000 000000 120000 000000 120000 000000 120000

       Data = 126736 000001 072340 000002 000000 000000 024400 006522
         +16  177626 000220 100000 032532 177710 000220 000142 000000
         +32  000000 000000 000213 001201 000661 000000 000000 000000
         +48  000000 000000 000000 000000 000000 000000 000000 000000
         +64  000000 000000 000000 000000 000000 000000 000000 000000
         +80  000000 000000 000000 000000 000000 000000 000000 000000
         +96  000000 000000 000000 000000 000000 000000 000000 000000
        +112  000000 000000 000000 000000 000000 000000 000000 000000
        +128  000000 000000 000000 000000 000000 000000 000000 000000
        +144  000000 000000 000000 000000 000000 000000 000000 000000
        +160  000000 000000 000000 000000 000000 000000 000000 000000
        +176  000000 000000 000000 000000 000000 000000 000000 000000
        +192  000000 000000 000000 000000 000000 000000 000000 000000
        +208  000000 000000 000000 000000 000000 000000 000000 000000
        +224  000000 000000 000000 000000 000000 000000 000000 000000
        +240  000000 000000 000000 000000 000000 000000 000000 000000
        +256  000000 000000 000000 000000 000000 000000 000000 000000
        +272  000000 000000 000000 000000 000000 000000 000000 000000
        +288  000000 000000 000000 000000 000000 000000 000000 000000
        +304  000000 000000 000000 000000 000000 000000 000000 000000
        +320  000000 000000 000000 000000 000000 000000 000000 000000
        +336  000000 000000 000000 000000 000000 000000 000000 000000
        +352  000000 000000 000000 000000 000000 000000 000000 000000
        +368  000000 000000 000000 000000 000000 000000 000000 000000
        +384  000000 000000 000000 000000 000000 000000 000000 000000
        +400  000000 000000 000000 000000 000000 000000 000000 000000
        +416  000000 000000 000000 000000 000000 000000 000000 000000
        +432  000000 000000 000000 000000 000000 000000 000000 000000
        +448  000000 000000 000000 000000 000000 000000 000000 000000
        +464  000000 000000 000000 000000 000000 000000 000000 000000
        +480  000000 000000 000000 000000 000000 000000 000000 000000
        +496  000000 000000 000000 000000 000000 000000 000000 000000

        EDC = 124152    Calculated EDC Difference = 000000

        ECC = 000000 000000 000000 000000 000000 000000
              000000 000000 007400 000001 000000 000000

DKUTIL>
```

```
DKUTIL> DUMP/ALL XBN 1

 ****** Buffer for XBN 1 (120000 000001), MSCP Status: 000000

 Error Summary =

 Original Error Bits   = 000000    BN = 0 (000000 000000)
 Error Recovery Flags = 000       ECC Symbols Corrected = 0,0
 Error Retry Counts   = 0,0,0     Error Recovery Command = 000

     Header = 000001 120000 000001 120000 000001 120000 000001 120000
               Lower  High
       Data = 163227 040015 154040 140015 142334 140015 103714 140015
         +16  057566 140015 130247 140014 126717 140014 125367 140014
         +32  072026 140014 046173 140014 004332 140014 172063 140013
         +48  116352 140013 112406 140013 054137 140013 035361 140013
         +64  161075 140012 134502 140012 133152 140012 131622 140012
         +80  130272 140012 126742 140012 124062 140012 122532 140012
         +96  117652 140012 116322 140012 114772 140012 113442 140012
        +112  073463 140012 072133 140012 071176 140012 041667 140012
        +128  174413 140011 106704 140011 177166 110010 175636 110010
        +144  152722 140010 151646 140010 005374 140010 163010 140007
        +160  137034 140007 127323 140007 172117 140006 161472 140006
        +176  137441 140006 105766 140006 014004 140006 012454 140006
        +192  011124 140006 007574 140006 006770 140006 164420 110005
        +208  152464 140005 130434 140005 127105 140005 045711 140005
        +224  170334 140004 146220 110004 141060 110004 063115 140004
        +240  045300 110004 042260 140004 036716 140004 035633 140004
        +256  022331 140004 011710 140004 172176 140003 170646 140003
        +272  142047 140003 114602 140003 101274 140003 047625 140003
        +288  043406 140003 001302 140003 177107 140002 151150 140002
        +304  125633 140002 110021 140002 054044 110002 162730 140001
        +320  157024 140001 152310 140001 151555 140001 150760 140001
        +336  147430 140001 146100 140001 144550 140001 143220 140001
        +352  107523 140001 072244 140001 046315 140001 025655 040001
        +368  146174 140000 055165 140000 036306 140000 034767 140000
        +384  033437 140000 026520 140000 000000 000000 000000 000000
        +400  000000 000000 000000 000000 000000 000000 000000 000000
        +416  000000 000000 000000 000000 000000 000000 000000 000000
        +432  000000 000000 000000 000000 000000 000000 000000 000000
        +448  000000 000000 000000 000000 000000 000000 000000 000000
        +464  000000 000000 000000 000000 000000 000000 000000 000000
        +480  000000 000000 000000 000000 000000 000000 000000 000000
        +496  000000 000000 000000 000000 000000 000000 000000 000000

        EDC = 142337    Calculated EDC Difference = 000000

        ECC = 000000 000000 000000 000000 000000 000000
              000000 000000 006000 000001 000000 000000

DKUTIL>
```

```
DKUTIL> DUMP/ALL FCT BLOCK 2 COPY 1

        ****** FCT Block 2, Copy 1 ******
 ****** Buffer for XBN 1 (120000 000001), MSCP Status: 000000

 Error Summary =

 Original Error Bits  = 000000    BN = 0 (000000 000000)
 Error Recovery Flags = 000       ECC Symbols Corrected = 0,0
 Error Retry Counts   = 0,0,0     Error Recovery Command = 000

     Header = 000001 120000 000001 120000 000001 120000 000001 120000

       Data = 163227 040015 154040 140015 142334 140015 103714 140015
         +16  057566 140015 130247 140014 126717 140014 125367 140014
         +32  072026 140014 046173 140014 004332 140014 172063 140013
         +48  116352 140013 112406 140013 054137 140013 035361 140013
         +64  161075 140012 134502 140012 133152 140012 131622 140012
         +80  130272 140012 126742 140012 124062 140012 122532 140012
         +96  117652 140012 116322 140012 114772 140012 113442 140012
        +112  073463 140012 072133 140012 071176 140012 041667 140012
        +128  174413 140011 106704 140011 177166 110010 175636 110010
        +144  152722 140010 151646 140010 005374 140010 163010 140007
        +160  137034 140007 127323 140007 172117 140006 161472 140006
        +176  137441 140006 105766 140006 014004 140006 012454 140006
        +192  011124 140006 007574 140006 006770 140006 164420 110005
        +208  152464 140005 130434 140005 127105 140005 045711 140005
        +224  170334 140004 146220 110004 141060 110004 063115 140004
        +240  045300 110004 042260 140004 036716 140004 035633 140004
        +256  022331 140004 011710 140004 172176 140003 170646 140003
        +272  142047 140003 114602 140003 101274 140003 047625 140003
        +288  043406 140003 001302 140003 177107 140002 151150 140002
        +304  125633 140002 110021 140002 054044 110002 162730 140001
        +320  157024 140001 152310 140001 151555 140001 150760 140001
        +336  147430 140001 146100 140001 144550 140001 143220 140001
        +352  107523 140001 072244 140001 046315 140001 025655 040001
        +368  146174 140000 055165 140000 036306 140000 034767 140000
        +384  033437 140000 026520 140000 000000 000000 000000 000000
        +400  000000 000000 000000 000000 000000 000000 000000 000000
        +416  000000 000000 000000 000000 000000 000000 000000 000000
        +432  000000 000000 000000 000000 000000 000000 000000 000000
        +448  000000 000000 000000 000000 000000 000000 000000 000000
        +464  000000 000000 000000 000000 000000 000000 000000 000000
        +480  000000 000000 000000 000000 000000 000000 000000 000000
        +496  000000 000000 000000 000000 000000 000000 000000 000000

        EDC = 142337    Calculated EDC Difference = 000000

        ECC = 000000 000000 000000 000000 000000 000000
              000000 000000 006000 000001 000000 000000

DKUTIL>
```

```
DKUTIL> DUMP/ALL RCT BLOCK 1 COPY 1   Control Block

        ****** RCT Block 1, Copy 1 ******

 ****** Buffer for LBN 891072 (000015 114300), MSCP Status: 000000

 Error Summary =

 Original Error Bits  = 000000    BN = 0 (000000 000000)
 Error Recovery Flags = 000       ECC Symbols Corrected = 0,0
 Error Retry Counts   = 0,0,0     Error Recovery Command = 000
                     USABLE LBN
     Header = 114300 000015 114300 000015 114300 000015 114300 000015 - All the SAME "ok"
               Serial #                     — control word
       Data = 072340 000002 000000 000000 000000 000000 000000 000000
         +16  000000 000000 000000 000000 000000 000000 000000 000000
         +32  000000 000000 000000 000000 000000 000000 000000 000000
         +48  000000 000000 000000 000000 000000 000000 000000 000000
         +64  000000 000000 000000 000000 000000 000000 000000 000000
         +80  000000 000000 000000 000000 000000 000000 000000 000000
         +96  000000 000000 000000 000000 000000 000000 000000 000000
        +112  000000 000000 000000 000000 000000 000000 000000 000000
        +128  000000 000000 000000 000000 000000 000000 000000 000000
        +144  000000 000000 000000 000000 000000 000000 000000 000000
        +160  000000 000000 000000 000000 000000 000000 000000 000000
        +176  000000 000000 000000 000000 000000 000000 000000 000000
        +192  000000 000000 000000 000000 000000 000000 000000 000000
        +208  000000 000000 000000 000000 000000 000000 000000 000000
        +224  000000 000000 000000 000000 000000 000000 000000 000000
        +240  000000 000000 000000 000000 000000 000000 000000 000000
        +256  000000 000000 000000 000000 000000 000000 000000 000000
        +272  000000 000000 000000 000000 000000 000000 000000 000000
        +288  000000 000000 000000 000000 000000 000000 000000 000000
        +304  000000 000000 000000 000000 000000 000000 000000 000000
        +320  000000 000000 000000 000000 000000 000000 000000 000000
        +336  000000 000000 000000 000000 000000 000000 000000 000000
        +352  000000 000000 000000 000000 000000 000000 000000 000000
        +368  000000 000000 000000 000000 000000 000000 000000 000000
        +384  000000 000000 000000 000000 000000 000000 000000 000000
        +400  000000 000000 000000 000000 000000 000000 000000 000000
        +416  000000 000000 000000 000000 000000 000000 000000 000000
        +432  000000 000000 000000 000000 000000 000000 000000 000000
        +448  000000 000000 000000 000000 000000 000000 000000 000000
        +464  000000 000000 000000 000000 000000 000000 000000 000000
        +480  000000 000000 000000 000000 000000 000000 000000 000000
        +496  000000 000000 000000 000000 000000 000000 000000 000000

        EDC = 070244    Calculated EDC Difference = 000000

        ECC = 000000 000000 000000 000000 000000 000000
              000000 000000 174400 000002 000000 000000

DKUTIL>
```

```
DKUTIL> DUMP/ALL LBN 891072

 ****** Buffer for LBN 891072 (000015 114300), MSCP Status: 000000

 Error Summary =

 Original Error Bits   = 000000     BN = 0 (000000 000000)
 Error Recovery Flags = 000        ECC Symbols Corrected = 0,0
 Error Retry Counts    = 0,0,0      Error Recovery Command = 000

     Header = 114300 000015 114300 000015 114300 000015 114300 000015

       Data = 072340 000002 000000 000000 000000 000000 000000 000000
         +16  000000 000000 000000 000000 000000 000000 000000 000000
         +32  000000 000000 000000 000000 000000 000000 000000 000000
         +48  000000 000000 000000 000000 000000 000000 000000 000000
         +64  000000 000000 000000 000000 000000 000000 000000 000000
         +80  000000 000000 000000 000000 000000 000000 000000 000000
         +96  000000 000000 000000 000000 000000 000000 000000 000000
        +112  000000 000000 000000 000000 000000 000000 000000 000000
        +128  000000 000000 000000 000000 000000 000000 000000 000000
        +144  000000 000000 000000 000000 000000 000000 000000 000000
        +160  000000 000000 000000 000000 000000 000000 000000 000000
        +176  000000 000000 000000 000000 000000 000000 000000 000000
        +192  000000 000000 000000 000000 000000 000000 000000 000000
        +208  000000 000000 000000 000000 000000 000000 000000 000000
        +224  000000 000000 000000 000000 000000 000000 000000 000000
        +240  000000 000000 000000 000000 000000 000000 000000 000000
        +256  000000 000000 000000 000000 000000 000000 000000 000000
        +272  000000 000000 000000 000000 000000 000000 000000 000000
        +288  000000 000000 000000 000000 000000 000000 000000 000000
        +304  000000 000000 000000 000000 000000 000000 000000 000000
        +320  000000 000000 000000 000000 000000 000000 000000 000000
        +336  000000 000000 000000 000000 000000 000000 000000 000000
        +352  000000 000000 000000 000000 000000 000000 000000 000000
        +368  000000 000000 000000 000000 000000 000000 000000 000000
        +384  000000 000000 000000 000000 000000 000000 000000 000000
        +400  000000 000000 000000 000000 000000 000000 000000 000000
        +416  000000 000000 000000 000000 000000 000000 000000 000000
        +432  000000 000000 000000 000000 000000 000000 000000 000000
        +448  000000 000000 000000 000000 000000 000000 000000 000000
        +464  000000 000000 000000 000000 000000 000000 000000 000000
        +480  000000 000000 000000 000000 000000 000000 000000 000000
        +496  000000 000000 000000 000000 000000 000000 000000 000000

        EDC = 070244    Calculated EDC Difference = 000000

        ECC = 000000 000000 000000 000000 000000 000000
              000000 000000 174400 000002 000000 000000

DKUTIL>
```

DKUTIL Lab Sample – 21

```
DKUTIL> DUMP/ALL LBN 100

 ****** Buffer for LBN 100 (000000 000144), MSCP Status: 000000

 Error Summary =

 Original Error Bits  = 000000    BN = 0 (000000 000000)
 Error Recovery Flags = 000       ECC Symbols Corrected = 0,0
 Error Retry Counts   = 0,0,0     Error Recovery Command = 000

     Header = 000144 000000 000144 000000 000144 000000 000144 000000

       Data = 063146 177776 177774 177770 177760 177740 177700 177600
         +16  177400 177000 176000 174000 170000 160000 140000 100000
         +32  000000 177776 177774 177770 177760 177740 177700 177600
         +48  177400 177000 176000 174000 170000 160000 140000 100000
         +64  000000 177776 177774 177770 177760 177740 177700 177600
         +80  177400 177000 176000 174000 170000 160000 140000 100000
         +96  000000 177776 177774 177770 177760 177740 177700 177600
        +112  177400 177000 176000 174000 170000 160000 140000 100000
        +128  000000 177776 177774 177770 177760 177740 177700 177600
        +144  177400 177000 176000 174000 170000 160000 140000 100000
        +160  000000 177776 177774 177770 177760 177740 177700 177600
        +176  177400 177000 176000 174000 170000 160000 140000 100000
        +192  000000 177776 177774 177770 177760 177740 177700 177600
        +208  177400 177000 176000 174000 170000 160000 140000 100000
        +224  000000 177776 177774 177770 177760 177740 177700 177600
        +240  177400 177000 176000 174000 170000 160000 140000 100000
        +256  000000 177776 177774 177770 177760 177740 177700 177600
        +272  177400 177000 176000 174000 170000 160000 140000 100000
        +288  000000 177776 177774 177770 177760 177740 177700 177600
        +304  177400 177000 176000 174000 170000 160000 140000 100000
        +320  000000 177776 177774 177770 177760 177740 177700 177600
        +336  177400 177000 176000 174000 170000 160000 140000 100000
        +352  000000 177776 177774 177770 177760 177740 177700 177600
        +368  177400 177000 176000 174000 170000 160000 140000 100000
        +384  000000 177776 177774 177770 177760 177740 177700 177600
        +400  177400 177000 176000 174000 170000 160000 140000 100000
        +416  000000 177776 177774 177770 177760 177740 177700 177600
        +432  177400 177000 176000 174000 170000 160000 140000 100000
        +448  000000 177776 177774 177770 177760 177740 177700 177600
        +464  177400 177000 176000 174000 170000 160000 140000 100000
        +480  000000 177776 177774 177770 177760 177740 177700 177600
        +496  177400 177000 176000 174000 170000 160000 140000 100000

        EDC = 063043    Calculated EDC Difference = 000000

        ECC = 000000 000000 000000 000000 000000 000000
              000000 000000 010400 000000 000000 000000

DKUTIL>
```

decimal entry octal printout
↓

```
DKUTIL> MODIFY 32 1111 2222 3333 4444 5555 6666

DKUTIL> DUMP/ALL BUFFER

 ****** Buffer for LBN 100 (000000 000144), MSCP Status: 000000

 Error Summary =

 Original Error Bits   = 000000     BN = 0 (000000 000000)
 Error Recovery Flags = 000        ECC Symbols Corrected = 0,0
 Error Retry Counts    = 0,0,0      Error Recovery Command = 000

     Header = 000144 000000 000144 000000 000144 000000 000144 000000

       Data = 063146 177776 177774 177770 177760 177740 177700 177600
         +16  177400 177000 176000 174000 170000 160000 140000 100000
----->   +32  002127 004256 006405 010534 012663 015012 177700 177600  <------
         +48  177400 177000 176000 174000 170000 160000 140000 100000
         +64  000000 177776 177774 177770 177760 177740 177700 177600
         +80  177400 177000 176000 174000 170000 160000 140000 100000
         +96  000000 177776 177774 177770 177760 177740 177700 177600
        +112  177400 177000 176000 174000 170000 160000 140000 100000
        +128  000000 177776 177774 177770 177760 177740 177700 177600
        +144  177400 177000 176000 174000 170000 160000 140000 100000
        +160  000000 177776 177774 177770 177760 177740 177700 177600
        +176  177400 177000 176000 174000 170000 160000 140000 100000
        +192  000000 177776 177774 177770 177760 177740 177700 177600
        +208  177400 177000 176000 174000 170000 160000 140000 100000
        +224  000000 177776 177774 177770 177760 177740 177700 177600
        +240  177400 177000 176000 174000 170000 160000 140000 100000
        +256  000000 177776 177774 177770 177760 177740 177700 177600
        +272  177400 177000 176000 174000 170000 160000 140000 100000
        +288  000000 177776 177774 177770 177760 177740 177700 177600
        +304  177400 177000 176000 174000 170000 160000 140000 100000
        +320  000000 177776 177774 177770 177760 177740 177700 177600
        +336  177400 177000 176000 174000 170000 160000 140000 100000
        +352  000000 177776 177774 177770 177760 177740 177700 177600
        +368  177400 177000 176000 174000 170000 160000 140000 100000
        +384  000000 177776 177774 177770 177760 177740 177700 177600
        +400  177400 177000 176000 174000 170000 160000 140000 100000
        +416  000000 177776 177774 177770 177760 177740 177700 177600
        +432  177400 177000 176000 174000 170000 160000 140000 100000
        +448  000000 177776 177774 177770 177760 177740 177700 177600
        +464  177400 177000 176000 174000 170000 160000 140000 100000
        +480  000000 177776 177774 177770 177760 177740 177700 177600
        +496  177400 177000 176000 174000 170000 160000 140000 100000

        EDC = 063043    Calculated EDC Difference = 000000

        ECC = 000000 000000 000000 000000 000000 000000
              000000 000000 010400 000000 000000 000000

DKUTIL>
```

DKUTIL Lab Sample - 23

word offset modify's whole word

0 (Octal) Not 0

```
DKUTIL> MODIFY 32 01111 02222 03333 04444 05555 06666 07777

DKUTIL> DUMP/ALL BUFFER

 ****** Buffer for LBN 100 (000000 000144), MSCP Status: 000000

 Error Summary =

 Original Error Bits  = 000000     BN = 0 (000000 000000)
 Error Recovery Flags = 000        ECC Symbols Corrected = 0,0
 Error Retry Counts   = 0,0,0      Error Recovery Command = 000

     Header = 000144 000000 000144 000000 000144 000000 000144 000000

       Data = 063146 177776 177774 177770 177760 177740 177700 177600
         +16  177400 177000 176000 174000 170000 160000 140000 100000
----->   +32  001111 002222 003333 004444 005555 006666 007777 177600  <------
         +48  177400 177000 176000 174000 170000 160000 140000 100000
         +64  000000 177776 177774 177770 177760 177740 177700 177600
         +80  177400 177000 176000 174000 170000 160000 140000 100000
         +96  000000 177776 177774 177770 177760 177740 177700 177600
        +112  177400 177000 176000 174000 170000 160000 140000 100000
        +128  000000 177776 177774 177770 177760 177740 177700 177600
        +144  177400 177000 176000 174000 170000 160000 140000 100000
        +160  000000 177776 177774 177770 177760 177740 177700 177600
        +176  177400 177000 176000 174000 170000 160000 140000 100000
        +192  000000 177776 177774 177770 177760 177740 177700 177600
        +208  177400 177000 176000 174000 170000 160000 140000 100000
        +224  000000 177776 177774 177770 177760 177740 177700 177600
        +240  177400 177000 176000 174000 170000 160000 140000 100000
        +256  000000 177776 177774 177770 177760 177740 177700 177600
        +272  177400 177000 176000 174000 170000 160000 140000 100000
        +288  000000 177776 177774 177770 177760 177740 177700 177600
        +304  177400 177000 176000 174000 170000 160000 140000 100000
        +320  000000 177776 177774 177770 177760 177740 177700 177600
        +336  177400 177000 176000 174000 170000 160000 140000 100000
        +352  000000 177776 177774 177770 177760 177740 177700 177600
        +368  177400 177000 176000 174000 170000 160000 140000 100000
        +384  000000 177776 177774 177770 177760 177740 177700 177600
        +400  177400 177000 176000 174000 170000 160000 140000 100000
        +416  000000 177776 177774 177770 177760 177740 177700 177600
        +432  177400 177000 176000 174000 170000 160000 140000 100000
        +448  000000 177776 177774 177770 177760 177740 177700 177600
        +464  177400 177000 176000 174000 170000 160000 140000 100000
        +480  000000 177776 177774 177770 177760 177740 177700 177600
        +496  177400 177000 176000 174000 170000 160000 140000 100000

        EDC = 063043     Calculated EDC Difference = 000000

        ECC = 000000 000000 000000 000000 000000 000000
              000000 000000 010400 000000 000000 000000
```

```
DKUTIL> WRITE LBN 891071

DKUTIL> DUMP LBN 891071

 ****** Buffer for LBN 891071 (000015 114277), MSCP Status: 000000

        Data = 063146 177776 177774 177770 177760 177740 177700 177600
          +16  177400 177000 176000 174000 170000 160000 140000 100000
----->    +32  001111 002222 003333 004444 005555 006666 007777 177600  <------
          +48  177400 177000 176000 174000 170000 160000 140000 100000
          +64  000000 177776 177774 177770 177760 177740 177700 177600
          +80  177400 177000 176000 174000 170000 160000 140000 100000
          +96  000000 177776 177774 177770 177760 177740 177700 177600
         +112  177400 177000 176000 174000 170000 160000 140000 100000
         +128  000000 177776 177774 177770 177760 177740 177700 177600
         +144  177400 177000 176000 174000 170000 160000 140000 100000
         +160  000000 177776 177774 177770 177760 177740 177700 177600
         +176  177400 177000 176000 174000 170000 160000 140000 100000
         +192  000000 177776 177774 177770 177760 177740 177700 177600
         +208  177400 177000 176000 174000 170000 160000 140000 100000
         +224  000000 177776 177774 177770 177760 177740 177700 177600
         +240  177400 177000 176000 174000 170000 160000 140000 100000
         +256  000000 177776 177774 177770 177760 177740 177700 177600
         +272  177400 177000 176000 174000 170000 160000 140000 100000
         +288  000000 177776 177774 177770 177760 177740 177700 177600
         +304  177400 177000 176000 174000 170000 160000 140000 100000
         +320  000000 177776 177774 177770 177760 177740 177700 177600
         +336  177400 177000 176000 174000 170000 160000 140000 100000
         +352  000000 177776 177774 177770 177760 177740 177700 177600
         +368  177400 177000 176000 174000 170000 160000 140000 100000
         +384  000000 177776 177774 177770 177760 177740 177700 177600
         +400  177400 177000 176000 174000 170000 160000 140000 100000
         +416  000000 177776 177774 177770 177760 177740 177700 177600
         +432  177400 177000 176000 174000 170000 160000 140000 100000
         +448  000000 177776 177774 177770 177760 177740 177700 177600
         +464  177400 177000 176000 174000 170000 160000 140000 100000
         +480  000000 177776 177774 177770 177760 177740 177700 177600
         +496  177400 177000 176000 174000 170000 160000 140000 100000

         EDC = 167125    Calculated EDC Difference = 000000

DKUTIL>
```

REPLACE

```
DKUTIL>

DKUTIL> REVECTOR 891071

DKUTIL>

DKUTIL> DISPLAY/FULL RCT

DKUTIL> ERROR-W Bad Block Replacement (Success) at  9-May-1988 13:50:09.93
        Command Ref #    00000000
        RA81 unit #      230.
        Err Seq #        1.
        Format Type      09
        Error Flags      80
        Event            0014
        Replace Flags    8000
        LBN              891071.
        Old RBN          0.
        New RBN          17471.
        Cause Event      004A
ERROR-I End of error.

DKUTIL>
```

Successful BBR

```
DKUTIL> DISPLAY/FULL RCT

           Revector Control Table for D0230 (RA81)

   Serial Number:       0000160992
   Flags:               000000

   LBN Being Replaced:  891071 (000015 114277)
   Replacement RBN:     17471 (060000 042077)
   Bad RBN:             0 (060000 000000)

   Cache ID:            0000000000
   Cache Incarnation:   0
   Incarnation Date:    17-Nov-1858 00:00:00.00

  11403 -->   223,  13822 -->   271,  14536 -->   285,
  15293 -->   299,  22714 -->   445,  51330 -->  1006,
 Bad RBN:     1475,  83559 -->  1638,  93562 -->  1834,
 100244 -->  1965, 114133 -->  2237, 114847 -->  2251,
 115561 -->  2265, 116275 -->  2279, 116989 -->  2293,
 117394 -->  2301, 117703 -->  2307, 120021 -->  2353,
 121987 -->  2391, 150682 -->  2954, 164702 -->  3229,
 171620 -->  3365, 181380 -->  3556, 192415 -->  3772,
 193520 -->  3794, 210660 -->  4130, 212809 -->  4172,
 225652 -->  4424, 231370 -->  4536, 242077 -->  4746,
 253500 -->  4970, 254214 -->  4984, 262070 -->  5138,
 266355 -->  5222, 272069 -->  5334, 272623 -->  5345,
 274359 -->  5379, 275905 -->  5409, 282789 -->  5544,
 305893 -->  5997, 308464 -->  6048, 317579 -->  6227,
 340423 *->  6674, 365116 -->  7159, 365829 -->  7173,
 374947 -->  7351, 379896 -->  7448, 389162 -->  7630,
 389586 -->  7638, 390300 -->  7652, 391014 -->  7666,
 391728 -->  7680, 420792 -->  8250, 433639 -->  8502,
 442706 -->  8680, 446995 -->  8764, 493817 -->  9682,
 497662 -->  9758, 507686 -->  9954, 516964 --> 10136,
 567338 --> 11124, 567925 --> 11135, 577411 --> 11321,
 578125 --> 11335, 614076 --> 12040, 640975 --> 12568,
 659759 --> 12936, 671472 --> 13166, 671958 --> 13175,
 672672 --> 13189, 680737 --> 13347, 681451 --> 13361,
 682165 --> 13375, 682879 --> 13389, 684307 *-> 13417,
 685021 --> 13431, 686449 --> 13459, 687163 --> 13473,
 687877 --> 13487, 688591 --> 13501, 689305 --> 13515,
 699525 --> 13716, 721806 --> 14153, 729217 --> 14298,
 744479 --> 14597, 746413 --> 14635, 768332 --> 15065,
 773565 --> 15167, 790535 --> 15500, 800493 --> 15695,
 814216 --> 15965, 814930 --> 15979, 815644 --> 15993,
 859503 --> 16853, 869680 --> 17052, 885017 --> 17353,
 889896 --> 17448, 891071 --> 17471,                    <--- new replacement
                                                             added for 891071

   RCT Statistics:          1 Bad RBN
                           97 Bad LBNs
                           95 Primary Revectors
                            2 Non-Primary Revectors
                            0 Probationary RBNs

DKUTIL>
```

```
DKUTIL> DUMP/ALL LBN 891071

 ****** Buffer for LBN 891071 (000015 114277), MSCP Status: 000000

 Error Summary = header compare

 Original Error Bits   = 004000     BN = 891071 (000015 114277)
 Error Recovery Flags = 200         ECC Symbols Corrected = 0,0
 Error Retry Counts   = 0,0,0       Error Recovery Command = 000

     Header = 114277 050015 114277 050015 114277 050015 114277 050015

       Data = 063146 177776 177774 177770 177760 177740 177700 177600
         +16  177400 177000 176000 174000 170000 160000 140000 100000
         +32  001111 002222 003333 004444 005555 006666 007777 177600
         +48  177400 177000 176000 174000 170000 160000 140000 100000
         +64  000000 177776 177774 177770 177760 177740 177700 177600
         +80  177400 177000 176000 174000 170000 160000 140000 100000
         +96  000000 177776 177774 177770 177760 177740 177700 177600
        +112  177400 177000 176000 174000 170000 160000 140000 100000
        +128  000000 177776 177774 177770 177760 177740 177700 177600
        +144  177400 177000 176000 174000 170000 160000 140000 100000
        +160  000000 177776 177774 177770 177760 177740 177700 177600
        +176  177400 177000 176000 174000 170000 160000 140000 100000
        +192  000000 177776 177774 177770 177760 177740 177700 177600
        +208  177400 177000 176000 174000 170000 160000 140000 100000
        +224  000000 177776 177774 177770 177760 177740 177700 177600
        +240  177400 177000 176000 174000 170000 160000 140000 100000
        +256  000000 177776 177774 177770 177760 177740 177700 177600
        +272  177400 177000 176000 174000 170000 160000 140000 100000
        +288  000000 177776 177774 177770 177760 177740 177700 177600
        +304  177400 177000 176000 174000 170000 160000 140000 100000
        +320  000000 177776 177774 177770 177760 177740 177700 177600
        +336  177400 177000 176000 174000 170000 160000 140000 100000
        +352  000000 177776 177774 177770 177760 177740 177700 177600
        +368  177400 177000 176000 174000 170000 160000 140000 100000
        +384  000000 177776 177774 177770 177760 177740 177700 177600
        +400  177400 177000 176000 174000 170000 160000 140000 100000
        +416  000000 177776 177774 177770 177760 177740 177700 177600
        +432  177400 177000 176000 174000 170000 160000 140000 100000
        +448  000000 177776 177774 177770 177760 177740 177700 177600
        +464  177400 177000 176000 174000 170000 160000 140000 100000
        +480  000000 177776 177774 177770 177760 177740 177700 177600
        +496  177400 177000 176000 174000 170000 160000 140000 100000

         EDC = 071732     Calculated EDC Difference = 000000

         ECC = 000000 000000 000000 000000 000000 000000
               000000 000000 066400 000000 000000 000000

DKUTIL>
```

Compules header Code for LBN if it finds something Else ie: 05 Primary Replacement it expected 06

This is New R DATA

```
DKUTIL> DUMP/ALL RBN 17471

 ****** Buffer for RBN 17471 (060000 042077), MSCP Status: 000000

 Error Summary =

 Original Error Bits   = 000000    BN = 0 (000000 000000)
 Error Recovery Flags = 000       ECC Symbols Corrected = 0,0
 Error Retry Counts   = 0,0,0     Error Recovery Command = 000

     Header = 042077 060000 042077 060000 042077 060000 042077 060000

       Data = 063146 177776 177774 177770 177760 177740 177700 177600
         +16  177400 177000 176000 174000 170000 160000 140000 100000
         +32  001111 002222 003333 004444 005555 006666 007777 177600
         +48  177400 177000 176000 174000 170000 160000 140000 100000
         +64  000000 177776 177774 177770 177760 177740 177700 177600
         +80  177400 177000 176000 174000 170000 160000 140000 100000
         +96  000000 177776 177774 177770 177760 177740 177700 177600
        +112  177400 177000 176000 174000 170000 160000 140000 100000
        +128  000000 177776 177774 177770 177760 177740 177700 177600
        +144  177400 177000 176000 174000 170000 160000 140000 100000
        +160  000000 177776 177774 177770 177760 177740 177700 177600
        +176  177400 177000 176000 174000 170000 160000 140000 100000
        +192  000000 177776 177774 177770 177760 177740 177700 177600
        +208  177400 177000 176000 174000 170000 160000 140000 100000
        +224  000000 177776 177774 177770 177760 177740 177700 177600
        +240  177400 177000 176000 174000 170000 160000 140000 100000
        +256  000000 177776 177774 177770 177760 177740 177700 177600
        +272  177400 177000 176000 174000 170000 160000 140000 100000
        +288  000000 177776 177774 177770 177760 177740 177700 177600
        +304  177400 177000 176000 174000 170000 160000 140000 100000
        +320  000000 177776 177774 177770 177760 177740 177700 177600
        +336  177400 177000 176000 174000 170000 160000 140000 100000
        +352  000000 177776 177774 177770 177760 177740 177700 177600
        +368  177400 177000 176000 174000 170000 160000 140000 100000
        +384  000000 177776 177774 177770 177760 177740 177700 177600
        +400  177400 177000 176000 174000 170000 160000 140000 100000
        +416  000000 177776 177774 177770 177760 177740 177700 177600
        +432  177400 177000 176000 174000 170000 160000 140000 100000
        +448  000000 177776 177774 177770 177760 177740 177700 177600
        +464  177400 177000 176000 174000 170000 160000 140000 100000
        +480  000000 177776 177774 177770 177760 177740 177700 177600
        +496  177400 177000 176000 174000 170000 160000 140000 100000

        EDC = 167125    Calculated EDC Difference = 000000

        ECC = 000000 000000 000000 000000 000000 000000
              000000 000000 052000 000001 000000 000000

DKUTIL>
```

```
DKUTIL> DUMP/ALL/RAW LBN 891071

 ****** Buffer for LBN 891071 (000015 114277), MSCP Status: 000000

 Error Summary =

 Original Error Bits   = 000000     BN = 0 (000000 000000)
 Error Recovery Flags = 000        ECC Symbols Corrected = 0,0
 Error Retry Counts    = 0,0,0      Error Recovery Command = 000

     Header = 114277 050015 114277 050015 114277 050015 114277 050015

       Data = 000000 000000 000000 000000 000000 000000 000000 000000
         +16  000000 000000 000000 000000 000000 000000 000000 000000
         +32  000000 000000 000000 000000 000000 000000 000000 000000
         +48  000000 000000 000000 000000 000000 000000 000000 000000
         +64  000000 000000 000000 000000 000000 000000 000000 000000
         +80  000000 000000 000000 000000 000000 000000 000000 000000
         +96  000000 000000 000000 000000 000000 000000 000000 000000
        +112  000000 000000 000000 000000 000000 000000 000000 000000
        +128  000000 000000 000000 000000 000000 000000 000000 000000
        +144  000000 000000 000000 000000 000000 000000 000000 000000
        +160  000000 000000 000000 000000 000000 000000 000000 000000
        +176  000000 000000 000000 000000 000000 000000 000000 000000
        +192  000000 000000 000000 000000 000000 000000 000000 000000
        +208  000000 000000 000000 000000 000000 000000 000000 000000
        +224  000000 000000 000000 000000 000000 000000 000000 000000
        +240  000000 000000 000000 000000 000000 000000 000000 000000
        +256  000000 000000 000000 000000 000000 000000 000000 000000
        +272  000000 000000 000000 000000 000000 000000 000000 000000
        +288  000000 000000 000000 000000 000000 000000 000000 000000
        +304  000000 000000 000000 000000 000000 000000 000000 000000
        +320  000000 000000 000000 000000 000000 000000 000000 000000
        +336  000000 000000 000000 000000 000000 000000 000000 000000
        +352  000000 000000 000000 000000 000000 000000 000000 000000
        +368  000000 000000 000000 000000 000000 000000 000000 000000
        +384  000000 000000 000000 000000 000000 000000 000000 000000
        +400  000000 000000 000000 000000 000000 000000 000000 000000
        +416  000000 000000 000000 000000 000000 000000 000000 000000
        +432  000000 000000 000000 000000 000000 000000 000000 000000
        +448  000000 000000 000000 000000 000000 000000 000000 000000
        +464  000000 000000 000000 000000 000000 000000 000000 000000
        +480  000000 000000 000000 000000 000000 000000 000000 000000
        +496  000000 000000 000000 000000 000000 000000 000000 000000

        EDC = 000105    Calculated EDC Difference = 000000

        ECC = 000000 000000 000000 000000 000000 000000
              000000 000000 146400 000003 000000 000000

DKUTIL>
```

Origional Blo
from #27

```
DKUTIL> DUMP RCT BLOCK 2
```

Temp Storage used during BBR Caching Block in RCT

```
        ****** RCT Block 2, Copy 1 ******

 ****** Buffer for LBN 891073 (000015 114301), MSCP Status: 000000

       Data = 063146 177776 177774 177770 177760 177740 177700 177600
         +16  177400 177000 176000 174000 170000 160000 140000 100000
         +32  001111 002222 003333 004444 005555 006666 007777 177600
         +48  177400 177000 176000 174000 170000 160000 140000 100000
         +64  000000 177776 177774 177770 177760 177740 177700 177600
         +80  177400 177000 176000 174000 170000 160000 140000 100000
         +96  000000 177776 177774 177770 177760 177740 177700 177600
        +112  177400 177000 176000 174000 170000 160000 140000 100000
        +128  000000 177776 177774 177770 177760 177740 177700 177600
        +144  177400 177000 176000 174000 170000 160000 140000 100000
        +160  000000 177776 177774 177770 177760 177740 177700 177600
        +176  177400 177000 176000 174000 170000 160000 140000 100000
        +192  000000 177776 177774 177770 177760 177740 177700 177600
        +208  177400 177000 176000 174000 170000 160000 140000 100000
        +224  000000 177776 177774 177770 177760 177740 177700 177600
        +240  177400 177000 176000 174000 170000 160000 140000 100000
        +256  000000 177776 177774 177770 177760 177740 177700 177600
        +272  177400 177000 176000 174000 170000 160000 140000 100000
        +288  000000 177776 177774 177770 177760 177740 177700 177600
        +304  177400 177000 176000 174000 170000 160000 140000 100000
        +320  000000 177776 177774 177770 177760 177740 177700 177600
        +336  177400 177000 176000 174000 170000 160000 140000 100000
        +352  000000 177776 177774 177770 177760 177740 177700 177600
        +368  177400 177000 176000 174000 170000 160000 140000 100000
        +384  000000 177776 177774 177770 177760 177740 177700 177600
        +400  177400 177000 176000 174000 170000 160000 140000 100000
        +416  000000 177776 177774 177770 177760 177740 177700 177600
        +432  177400 177000 176000 174000 170000 160000 140000 100000
        +448  000000 177776 177774 177770 177760 177740 177700 177600
        +464  177400 177000 176000 174000 170000 160000 140000 100000
        +480  000000 177776 177774 177770 177760 177740 177700 177600
        +496  177400 177000 176000 174000 170000 160000 140000 100000

        EDC = 167125    Calculated EDC Difference = 000000

DKUTIL>
```

```
DKUTIL> DUMP RCT BLOCK 3

        ****** RCT Block 3, Copy 1 ******

 ****** Buffer for LBN 891074 (000015 114302), MSCP Status: 000000

       Data = 000000 000000 000000 000000 000000 000000 000000 000000
         +16  000000 000000 000000 000000 000000 000000 000000 000000
         +32  000000 000000 000000 000000 000000 000000 000000 000000
         +48  000000 000000 000000 000000 000000 000000 000000 000000
         +64  000000 000000 000000 000000 000000 000000 000000 000000
         +80  000000 000000 000000 000000 000000 000000 000000 000000
         +96  000000 000000 000000 000000 000000 000000 000000 000000
        +112  000000 000000 000000 000000 000000 000000 000000 000000
        +128  000000 000000 000000 000000 000000 000000 000000 000000
        +144  000000 000000 000000 000000 000000 000000 000000 000000
        +160  000000 000000 000000 000000 000000 000000 000000 000000
        +176  000000 000000 000000 000000 000000 000000 000000 000000
        +192  000000 000000 000000 000000 000000 000000 000000 000000
        +208  000000 000000 000000 000000 000000 000000 000000 000000
        +224  000000 000000 000000 000000 000000 000000 000000 000000
        +240  000000 000000 000000 000000 000000 000000 000000 000000
        +256  000000 000000 000000 000000 000000 000000 000000 000000
        +272  000000 000000 000000 000000 000000 000000 000000 000000
        +288  000000 000000 000000 000000 000000 000000 000000 000000
        +304  000000 000000 000000 000000 000000 000000 000000 000000
        +320  000000 000000 000000 000000 000000 000000 000000 000000
        +336  000000 000000 000000 000000 000000 000000 000000 000000
        +352  000000 000000 000000 000000 000000 000000 000000 000000
        +368  000000 000000 000000 000000 000000 000000 000000 000000
        +384  000000 000000 000000 000000 000000 000000 000000 000000
        +400  000000 000000 000000 000000 000000 000000 000000 000000
        +416  000000 000000 000000 000000 000000 000000 000000 000000
        +432  000000 000000 000000 000000 000000 000000 000000 000000
        +448  000000 000000 000000 000000 000000 000000 000000 000000
        +464  000000 000000 000000 000000 000000 000000 000000 000000
        +480  000000 000000 000000 000000 000000 000000 000000 000000
        +496  000000 000000 000000 000000 000000 000000 000000 000000

        EDC = 000105    Calculated EDC Difference = 000000

DKUTIL>
```

```
DKUTIL> DUMP RCT BLOCK 5

        ****** RCT Block 5, Copy 1 ******

 ****** Buffer for LBN 891076 (000015 114304), MSCP Status: 000000

       Data = 000000 000000 000000 000000 000000 000000 000000 000000
         +16  000000 000000 000000 000000 000000 000000 000000 000000
         +32  000000 000000 000000 000000 000000 000000 000000 000000
         +48  000000 000000 000000 000000 000000 000000 032776 020000 <---
         +64  000000 000000 000000 000000 000000 000000 000000 000000
         +80  000000 000000 000000 000000 000000 000000 000000 000000
         +96  000000 000000 000000 000000 000000 000000 000000 000000
        +112  000000 000000 034310 020000 000000 000000 000000 000000
        +128  000000 000000 000000 000000 000000 000000 000000 000000
        +144  000000 000000 000000 000000 000000 000000 000000 000000
        +160  000000 000000 000000 000000 000000 000000 035675 020000
        +176  000000 000000 000000 000000 000000 000000 000000 000000
        +192  000000 000000 000000 000000 000000 000000 000000 000000
        +208  000000 000000 000000 000000 000000 000000 000000 000000
        +224  000000 000000 000000 000000 000000 000000 000000 000000
        +240  000000 000000 000000 000000 000000 000000 000000 000000
        +256  000000 000000 000000 000000 000000 000000 000000 000000
        +272  000000 000000 000000 000000 000000 000000 000000 000000
        +288  000000 000000 000000 000000 000000 000000 000000 000000
        +304  000000 000000 000000 000000 000000 000000 000000 000000
        +320  000000 000000 000000 000000 000000 000000 000000 000000
        +336  000000 000000 000000 000000 000000 000000 000000 000000
        +352  000000 000000 000000 000000 000000 000000 000000 000000
        +368  000000 000000 000000 000000 000000 000000 000000 000000
        +384  000000 000000 000000 000000 000000 000000 000000 000000
        +400  000000 000000 000000 000000 000000 000000 000000 000000
        +416  000000 000000 000000 000000 000000 000000 000000 000000
        +432  000000 000000 000000 000000 000000 000000 000000 000000
        +448  000000 000000 000000 000000 000000 000000 000000 000000
        +464  000000 000000 000000 000000 000000 000000 000000 000000
        +480  000000 000000 000000 000000 000000 000000 000000 000000
        +496  000000 000000 000000 000000 000000 000000 000000 000000

        EDC = 050431    Calculated EDC Difference = 000000

DKUTIL>
```

Descriptor Code=02 =
Allocated RBN (Primary)

DKUTIL Lab Sample – 33

```
DKUTIL> DUMP RCT BLOCK 9999

DKUTIL-E 9999 is an invalid BLOCK number; maximum is 139.


DKUTIL>
```

= use to find Last RCT
see next SAMPle

```
DKUTIL> DUMP RCT BLOCK 139

        ****** RCT Block 139, Copy 1 ******

 ****** Buffer for LBN 891210 (000015 114512), MSCP Status: 000000

       Data = 000000 000000 000000 000000 000000 000000 000000 000000
         +16  000000 000000 000000 000000 000000 000000 000000 000000
         +32  000000 000000 000000 000000 000000 000000 000000 000000
         +48  000000 000000 000000 000000 000000 000000 000000 000000
         +64  000000 000000 000000 000000 000000 000000 000000 000000
         +80  000000 000000 000000 000000 000000 000000 000000 000000
         +96  000000 000000 000000 000000 000000 000000 000000 000000
        +112  000000 000000 000000 000000 000000 000000 000000 000000
        +128  000000 000000 000000 000000 000000 000000 000000 000000
        +144  000000 000000 000000 000000 000000 000000 000000 000000
        +160  112050 020015 000000 000000 000000 000000 000000 000000
        +176  000000 000000 000000 000000 000000 000000 000000 000000
        +192  000000 000000 000000 000000 000000 000000 000000 000000
        +208  000000 000000 000000 000000 000000 000000 000000 000000
        +224  000000 000000 000000 000000 000000 000000 000000 000000
----->  +240  000000 000000 000000 000000 000000 000000 114277 020015 <-----
        +256  000000 000000 000000 000000 000000 000000 000000 000000
        +272  000000 000000 000000 000000 000000 000000 000000 000000
        +288  000000 000000 000000 000000 000000 000000 000000 000000
        +304  000000 000000 000000 000000 000000 000000 000000 000000
        +320  000000 000000 000000 000000 000000 000000 000000 000000
        +336  000000 000000 000000 000000 000000 000000 000000 000000
        +352  000000 000000 000000 000000 000000 000000 000000 000000
        +368  000000 000000 000000 000000 000000 000000 000000 000000
        +384  000000 000000 000000 000000 000000 000000 000000 000000
        +400  000000 000000 000000 000000 000000 000000 000000 000000
        +416  000000 000000 000000 000000 000000 000000 000000 000000
        +432  000000 000000 000000 000000 000000 000000 000000 000000
        +448  000000 000000 000000 000000 000000 000000 000000 000000
        +464  000000 000000 000000 000000 000000 000000 000000 000000
        +480  000000 100000 000000 100000 000000 100000 000000 100000
        +496  000000 100000 000000 100000 000000 100000 000000 100000

        EDC = 071732    Calculated EDC Difference = 000000

DKUTIL>
```

End of Table (of this copy)

DKUTIL Lab Sample – 35

```
DKUTIL>  MODIFY 0 0177777

DKUTIL-E "MODIFY" is an invalid command.

DKUTIL>
```

```
DKUTIL> DISPLAY RCT

       Revector Control Table for D0232 (RA81)

   Serial Number:       0000160978
   Flags:               000000

  19784 -->   387,  20143 -->   394,  27548 -->   540,
  47784 -->   936,  48344 -->   947, Bad RBN:    1179,
  81181 -->  1591,  93337 -->  1830,  93620 -->  1835,
  94334 -->  1849,  95762 -->  1877,  97853 -->  1918,
  99436 -->  1949, 111798 -->  2192, 113602 -->  2227,
 114316 -->  2241, 115744 -->  2269, 125505 -->  2460,
 130442 -->  2557, 153071 -->  3001, 153204 -->  3004,
 153918 -->  3018, 155437 -->  3047, 156158 -->  3061,
 156330 -->  3065, 176124 -->  3453, 183653 -->  3601,
 184367 -->  3615, 192131 -->  3767, 195256 -->  3828,
 203282 -->  3985, 203947 -->  3998, 213933 -->  4194,
 218759 -->  4289, 227509 -->  4460, 254669 -->  4993,
 291924 -->  5724, 293960 -->  5763, 348920 -->  6841,
 349045 -->  6844, 358890 -->  7037, 364645 -->  7149,
 438151 -->  8591, 438994 -->  8607, 457022 -->  8961,
 458862 -->  8997, 468130 -->  9179, 486718 -->  9543,
 488026 -->  9569, 494505 -->  9696, 498756 -->  9779,
 509499 -->  9990, 512513 --> 10049, 517416 --> 10145,

    "        "       "          "       "          "
    "        "       "          "       "          "
    "        "       "          "       "          "
    "        "       "          "       "          "

 809040 --> 15863, 809756 --> 15877, 809754 *-> 15878,
 810470 --> 15891, 810468 *-> 15892, 811184 --> 15905,
 811182 *-> 15906, 811898 --> 15919, 812612 --> 15933,
 813326 --> 15947, 813322 *-> 15948, 814036 --> 15961,
 838216 --> 16435, 848022 --> 16627, 848736 --> 16641,
 853831 --> 16741, 886659 --> 17385,

   RCT Statistics:            1 Bad RBN
                            157 Bad LBNs
                            153 Primary Revectors
                              4 Non-Primary Revectors
                              0 Probationary RBNs

DKUTIL>
```

```
DKUTIL> DISPLAY RCT

           Revector Control Table for D0117 (RA81)

   Serial Number:       3067705207
   Flags:               133331

   LBN Being Replaced:  3067721433 (133331 133331)
   Bad RBN:             3067721433 (173331 133331)

   Cache ID:            3067721433
   Cache Incarnation:   3067721433
   Incarnation Date:    5824 06:31:06.85

******* *->     0,******* *->      1,******* *->    2,
******* *->     3,******* *->      4,******* *->    5,
******* *->     6,******* *->      7,******* *->    8,
******* *->     9,******* *->     10,******* *->   11,
******* *->    12,******* *->     13,******* *->   14,
******* *->    15,******* *->     16,******* *->   17,
******* *->    18,******* *->     19,******* *->   20,
******* *->    21,******* *->     22,******* *->   23,
******* *->    24,******* *->     25,******* *->   26,
******* *->    27,******* *->     28,******* *->   29,
******* *->    30,******* *->     31,******* *->   32,
******* *->    33,******* *->     34,******* *->   35,
******* *->    36,******* *->     36,******* *->   38,
******* *->    39,******* *->     40,******* *->   41,
******* *->    42,******* *->     43,******* *->   44,
******* *->    45,******* *->     46,******* *->   47,
******* *->    48,******* *->     49,******* *->   50,
******* *->    51,******* *->     52,** ^C

DKUTIL-I CTRL/Y or CTRL/C Abort !
HSC70>
```

Result of Broke K.SDI

```
DKUTIL> DUMP RCT BLOCK 1

        ****** RCT Block 1, Copy 1 ******

 ****** Buffer for LBN 891072 (000015 114300), MSCP Status: 000000

       Data = 073567 133331 133331 133331 133331 133331 133331 133331
         +16  133331 133331 133331 133331 133331 133331 133331 133331
         +32  133331 133331 133331 133331 133331 133331 133331 133331
         +48  133331 133331 133331 133331 133331 133331 133331 133331
         +64  133331 133331 133331 133331 133331 133331 133331 133331
         +80  133331 133331 133331 133331 133331 133331 133331 133331
         +96  133331 133331 133331 133331 133331 133331 133331 133331
        +112  133331 133331 133331 133331 133331 133331 133331 133331
        +128  133331 133331 133331 133331 133331 133331 133331 133331
        +144  133331 133331 133331 133331 133331 133331 133331 133331
        +160  133331 133331 133331 133331 133331 133331 133331 133331
        +176  133331 133331 133331 133331 133331 133331 133331 133331
        +192  133331 133331 133331 133331 133331 133331 133331 133331
        +208  133331 133331 133331 133331 133331 133331 133331 133331
        +224  133331 133331 133331 133331 133331 133331 133331 133331
        +240  133331 133331 133331 133331 133331 133331 133331 133331
        +256  133331 133331 133331 133331 133331 133331 133331 133331
        +272  133331 133331 133331 133331 133331 133331 133331 133331
        +288  133331 133331 133331 133331 133331 133331 133331 133331
        +304  133331 133331 133331 133331 133331 133331 133331 133331
        +320  133331 133331 133331 133331 133331 133331 133331 133331
        +336  133331 133331 133331 133331 133331 133331 133331 133331
        +352  133331 133331 133331 133331 133331 133331 133331 133331
        +368  133331 133331 133331 133331 133331 133331 133331 133331
        +384  133331 133331 133331 133331 133331 133331 133331 133331
        +400  133331 133331 133331 133331 133331 133331 133331 133331
        +416  133331 133331 133331 133331 133331 133331 133331 133331
        +432  133331 133331 133331 133331 133331 133331 133331 133331
        +448  133331 133331 133331 133331 133331 133331 133331 133331
        +464  133331 133331 133331 133331 133331 133331 133331 133331
        +480  133331 133331 133331 133331 133331 133331 133331 133331
        +496  133331 133331 133331 133331 133331 133331 133331 133331

        EDC = 140753    Calculated EDC Difference = 000000

DKUTIL>
```

DSA TROUBLESHOOTING COURSE

Lab Exercise #3

RAUTIL

RAUTIL Lab
Lab Exercise 3

Refer to the RAUTIL section of the Student Guide during this exercise.

1. Log into your student account.

2. **$ SET DEF [STUDENTx.RAUTIL]**

3. **$ SET PROCESS/PRIV=ALL**

4. Specify the desk assigned to you by the instructor.

5. **$ SHOW DEV D**

   If your disk is not mounted, mount it using:

   ```
   $ MOUNT/FOR Device:
   ```

   Also make a note of some other mounted disks (disk names) for use later during the exercise.

6. **$ RUN RAUTIL**
   - Select the disk assigned to you for this exercise.
   - When prompted, select an output log file so that you may obtain a hardcopy of the results after the exercise.

7. Execute the **HELP** command.

8. Execute the **ANALYZE** command. Based on Contents of RCT

9. Execute the **SUMMARY** command.
   - Write down the head numbers that have replacements logged.

10. Execute the **HEAD** command for each of the heads on the disk that have replacements logged in the **SUMMARY** command.

    Use CONTROL–C to get out of the head selection prompt mode and back to the normal RAUTIL mode.

11. Execute the **DD** command.

12. Execute the **TL** command.

13. Execute the **TR** command.

14. Execute the **DUMP** command. Read the instructions in of the RAUTIL section of the Student Manual for proper command formats.
    - Dump some random LBNs in the host area.
    - Dump blocks 0, 1, and 2 from the RCT.

15. Using the **DUMP** command, locate an unused LBN that contains either all zeros or the DEC Standard Format Pattern. On an RA81, look in the vicinity of LBN 891050.

    Try the **WRITE** command using different write patterns of your choice, and use the **DUMP** command to verify the results.

16. Using the available LBN found during the previous step, execute the **MODIFY** command. Pick your own patterns and bytes to modify. Use the **DUMP** command to verify your modifications.

17. Execute the **BBR** command.

    NOTE

    If you are using a disk that is connected to an HSC controller, you will get a message that indicates you should use DKUTIL. This is normal as the HSC controls ALL BBR activity and prevents RAUTIL from performing this function. Have your instructor clarify this point if necessary.

18. Use the **NEXT** command and select another disk drive. Select one of the drives that is already mounted. (Refer to your notes from step 5.)

19. Use the **ANALYZE**, **SUMMARY**, and **HEAD** commands to review these drives.

20. If time permits, re-select the original drive assigned to you by the instructor, and try the **SCRUB** command.

21. Read the TROUBLESHOOTING section of the RAUTIL user guide in your Student Manual.

CHAPTER 16

RAUTIL USER GUIDE

16.1 OVERVIEW

RAUTIL is an executable image which runs under VMS and is capable of examining RA–based drives which have been previously mounted using the VMS MOUNT command. RAUTIL allows the user to examine disks that are already mounted and being used interactively by the system without any interference to another user or user's data. If the user intends to only perform read operations to examine the disk structures and contents, the drive does not have to be dismounted. If the drive is mounted FOREIGN, the user is allowed to write, modify, or replace an LBN. Some of the capabilities that may be utilized by RAUTIL are:

- Display the number of replacements that exist on a drive.
- Show the relative position of the bad block on a given track to identify where scratches and other defects may exist.
- Determine if any replaced blocks contain a forced error.
- Display the contents of an LBN within the host area or the RCT area.
- Scrub the disk to cause automatic replacements to occur.
- Manually force replacement of a block (xDA controllers only).
- Modify the contents of an LBN in the host area or RCT area.

16.1.1 Restrictions

Some features of RAUTIL cannot be executed for drives connected to an HSC–type controller. This is due to the way the program is constructed and the limitations that the HSC presents to RAUTIL. These will be noted throughout this document. All functionality of RAUTIL is available to xDA–type controllers.

16.2 GETTING STARTED

16.2.1 Compile RAUTIL.MAR

RAUTIL makes use of several system references that are different from system to system. For this reason, RAUTIL must be compiled on the target VMS system or any one of the system nodes in a VAX/VMS cluster. This may be accomplished using the following steps:

1. Obtain a copy of the source input file RAUTIL.MAR.
2. At the VMS prompt, enter the following commands:

```
$ MACRO RAUTIL+SYS$LIBRARY:LIB/LIB
$ LINK RAUTIL+SYS$SYSTEM:SYS.STB+SCSDEF
```

To run RAUTIL, you must have PHY_IO (physical I/O) privilege and CMKRNL (change kernel mode) privilege.

16.2.2 Invoke RAUTIL.EXE

1. Select a target drive to interrogate or test. To determine which drives are available, type SHOW DEVICE D at your terminal. If the target drive is not mounted, mount it foreign. If it is mounted and you wish to perform write functions to the drive, you will have to dismount and remount it foreign.

   The program performs physical QIO functions, so be sure you have enabled PHY_IO privilege. To enable the privilege, type SET PROC/PRIV=PHY_IO at your terminal. If you intend to write to the disk, you must also set CMKRNL privilege.

2. Start the program by typing RUN RAUTIL at your terminal. A sample of the dialog follows.

   - At the first prompt, "what device ? ", enter the disk you wish to test. This must be a device name that appeared when you issued the SHOW DEVICE D command. For shadowed disks, be sure to select the device name for a member of the shadow set and not the shadow set name.

     Alternately, you may enter one of the following generic device names rather than a specific device:

     RA60, RA70, RA80, RA81, RA82, RA90

     This allows the program to be used for information–only purposes to perform translations for the selected device type. The following commands may then be used:

DD	–	Display device parameters
EXIT	–	Exit the program
HELP	–	Display command summary
NEXT	–	Select another "real" device or device
TL	–	Translate LBN
TR	–	Translate RBN

     Commands that require selection of a specific device will be non–operational when you select a generic device type.

   - The next prompt will ask if you wish to create a log file. If you respond with "Y", a log file containing a copy of the commands and responses displayed on your terminal during the RAUTIL session will be generated.

   - The next prompt will ask if you want to verify the RCT consistency. If the controller is an xDA-type and you wish to check RCT consistency, type "Y". This will verify the multiple copies of the RCT.

     If you are connected to an HSC controller, the consistency cannot be checked since the HSC will only allow the host to read RCT copy 1.

   - The next prompt will be the main RAUTIL prompt. From this point, any of the functions listed in Section 16.3 or Section 16.4 can be executed, provided you have the proper privileges.

Sample dialog of RAUTIL startup

```
$ RUN RAUTIL

RA drive analysis utility version 9.3, type "HELP" for help

what device ? $1$DUA40

create a log file ? Y

creating log file $1$DUA40.dat

device is an RA81, serial# 137579, attached to a HSC70,
on node (SLEAZY), error count is 0

do you want to verify rct copy consistency ? (y/n) N

RAUTIL>
```

16.3 COMMAND SUMMARY

The following list summarizes the commands that may be used with RAUTIL. A more detailed explanation of these commands and some examples are found in Section 16.4.

ANALYZE	Analyze and list all replacements and verify RCT/replaced LBNs.
BBR	Manually replace a bad block (xDA controllers only).
DD	Display device parameters.
DUMP	Display the contents of a block in the user LBN area or the RCT.
EXIT	Exit the program or exit to the main RAUTIL prompt.
HEAD	Display replacements for a single head or all heads listed individually.
HELP	Display a summary list of RAUTIL commands.
MODIFY	Modify the contents of an LBN in the user LBN area or in the RCT.
NEXT	Release the current drive and select a new drive to examine.
SCRUB	Scrub the disk.
SUMMARY	Summarize replacements only.
TL	Translate LBN.
TR	Translate RBN.
WRITE	Write an LBN with a pattern.

16.4 COMMAND DETAILS and EXAMPLES

This section explains each individual RAUTIL command in more detail, including examples of command entries and responses.

In these examples, text enclosed in parentheses ( ) is informational only.

Unless otherwise noted, the following commands should work for all disk controllers in the DSA I environment (HSC, UDA, KDA, KDB, etc.).

16.4.1 ANALYZE

The RCT is searched for replacements, and all replacements are verified. All replacements are also checked for the occurrence of a forced error. If a forced error is detected, it is reported on the user terminal. Only LBNs that have been replaced are actually read by this function. Blocks that contain forced errors are generally the result of BBR (bad block replacement) where the original data was uncorrectable. In some instances, blocks could be flagged with a forced error and not be replaced. These blocks will not be seen or displayed by the ANALYZE command. To locate non–replaced blocks with forced errors, EDC errors, or other problems, use the SCRUB command.

All replacements are listed on the terminal in order of ascending LBNs, then rolled into a summary on the terminal. The summary shows the replacements allocated by physical head and categorized by primary replaced LBNs, non–primary replaced LBNs, and bad RBNs.

Refer to the following example for an illustration of using the ANALYZE command. Following the example is a legend explaining most of the terminology used in the display.

Example 16–1: ANALYZE Command

```
RAUTIL> ANALYZE

  DB       LBN       RBN     CYL HEAD POS    DESC      TYP
-----------------------------------------------------------

(  2.)      56. =>     2.    0.   1.  19.  3000 0038   NON-PRI
(  2.)     343. =>     6.    0.   6.  17.  2000 0157   PRI
(  3.)    8514. =>   166.   11.  12.   8.  2000 2142   PRI
(  6.)   26341. =>   516.   36.  12.  37.  2000 66E5   PRI
(  6.)   27260. =>   534.   38.   2.   2.  2000 6A7C   PRI
(  7.)   36333. =>   712.   50.  12.  33.  2000 8DED   PRI
(  7.)   38838. =>   761.   54.   5.  45.  2000 97B6   PRI
( 12.)   68754. =>  1348.   96.   4.  10.  2001 0C92   PRI
( 13.)   72438. =>  1420.  101.   6.  50.  2001 1AF6   PRI
( 13.) *******. =>  1533.  109.   7.  45.  4000 0000   UNUSABLE
( 14.)   82039. =>  1608.  114.  12.  43.  2001 4077   PRI
( 19.)  114914. =>  2253.  160.  13.  37.  2001 C0E2   PRI
( 20.) *******. =>  2333.  166.   9.  21.  4000 0000   UNUSABLE

              "          "          "       "          "
              "          "          "       "          "

(137.)  882628. => 17306. 1236.   2.  50.  200D 77C4   PRI
(137.)  882695. => 17307. 1236.   3.  28.  200D 7807   PRI
(137.)  884817. => 17349. 1239.   3.   8.  200D 8051   PRI
(137.)  887751. => 17406. 1243.   4.  49.  200D 8BC7   PRI
(138.)  888465. => 17420. 1244.   4.  49.  200D 8E91   PRI
(138.)  888520. => 17421. 1244.   5.  15.  200D 8EC8   PRI

replacement by head for the RA81, unit# $5$DUA230, serial # 137579

HEAD        0   1   2   3   4   5   6   7   8   9  10  11  12  13  TOTAL
           ------------------------------------------------------

PRI         0   2  46  25  17   6   8  13   2   6   5   5  16  24    175

NON-PRI     0   1   1   0   0   0   0   1   0   0   0   0   0   0      3

BAD-RBN     0   1   1   0   0   0   0   1   0   1   0   0   1   0      5

TOTAL       0   4  48  25  17   6   8  15   2   7   5   5  17  24    183

RAUTIL>
```

Table 16–1: Legend for ANALYZE Command

RBN	Replacement block. This is the block that currently contains the data for the corresponding logical block listed in the LBN column.	
DB	Descriptor block number. This is the relative block number within the RCT that contains the descriptor for this entry.	
LBN	Logical block. This is a block from the usable host area that has been replaced as described in the RCT.	
CYL	Cylinder. This is the physical cylinder containing the corresponding LBN.	
HEAD	Disk R/W head. This is the physical R/W head that would be used to read the corresponding LBN. It also describes the media surface or a portion of the media surface containing the LBN.	
POS	(Position from index.) This is the physical sector from index that corresponds to the LBN.	
DESC	(Descriptor contents.) This is the hexadecimal contents of the specific RCT descriptor that corresponds to the LBN/RBN entry listed in the output.	
TYP	(Type of descriptor.) This is a translation of the descriptor code field within the RCT descriptor.	
	PRI =	Primary replaced LBN
	NON–PRI =	Non–primary replaced LBN
	UNUSABLE =	BAD RB

NOTE

Asterisks (*******) in the LBN column indicate that an RBN descriptor was found in the RCT table referencing that particular RBN as bad and unusable.

16.4.2 Manual Bad Block Replacement (BBR)

When executed to a drive connected to an xDA–type controller, this command allows the user to manually force the replacement of a specified LBN in the host/user LBN area.
This command cannot be executed on HSC controllers. (Use the utility DKUTIL for this purpose when your disk is connected to an HSC controller.) A sample of the results is shown below.

```
RAUTIL> BBR

what lbn  ? 891020

      flags....  P_2
      lbn......  891020
      new rbn..  17470
      old rbn..  0

is replacement information correct ? Y

replacing LBN 891020

RAUTIL>
```

```
The information in
this area is for
information only. You
will normally answer
"Y" (yes) unless you
wish to abort the
replacement operation.
```

16.4.3 DD - Display Drive

This command displays the device parameters for the currently selected drive. A sample of the results is shown below.

```
RAUTIL> DD

            drive is an RA81
     -----------------------------------

         lbns per trk .........  51
         trks per group........  1
         groups per cyl........  14
         group offset..........  14
         rbns per track........  1
         number of heads.......  14
         number of host lbns...  891072
         rct copy size.........  765
         number of rct copies..  4

RAUTIL>
```

16.4.4 DUMP

This command allows you to display the contents of data in both the host/user LBN area and blocks in the RCT. It is particularly useful for reviewing suspected bad data blocks or structures in the RCT. Refer to the following example for an illustration of using the DUMP command. To perform this operation, you must first enter the command DUMP at the RAUTIL> prompt. This will put the program into the dump mode of operation.

When selecting LBNs in the host area, use the following format:

```
dmp> L xxxx              where xxxx is the LBN number
```

When selecting blocks in the RCT area, use the following format:

```
dmp> R x,y               where x is the relative block number
                         within a particular copy of the RCT; and
                         y is the copy number.
```

The dump display is organized into 32–bit longwords. Each longword is subdivided into two 16–bit words (hexadecimal). There are four longwords per line numbered from right to left starting with zero. When using the L xxxx format, the program will also attempt to translate the contents of the block into ASCII and display it in the far right columns.

Use CONTROL–C to exit from the DUMP mode of operation. This will cause a return to the RAUTIL> prompt.

When using this command with HSC controllers, only the first copy of any RCT block will be displayed, regardless of which copy you specify during the "R x,y" format.

```
RAUTIL> DUMP

dmp> L 55               (This selects LBN # 55 in the host/user area)

4227 2720   7374 7265   766E 6F63   2079 6C65   0    "ely converts ''B"
3120 646E   6120 3938   2073 2727   4550 5954   4    "TYPE''s 89 and 1"
5357 2024   0046 2220   5357 2024   0006 3132   8    "21..$ WS "F.$ WS"
2020 3E52   433C 2020   3035 3A35   3420 2220   12   " " 45:50  <CR>  "
2074 7265   766E 6F63   206C 6C69   5720 2020   16   "   Will convert "
6220 7327   2745 5059   5442 2727   206C 6C61   20   "all ''BTYPE''s b"
3035 2064   6E61 2035   3420 6E65   6577 7465   24   "etween 45 and 50"
2020 2020   2020 2220   5357 2024   0017 222E   28   ".."..$ WS "      "

 "    "      "    "      "    "      "    "     "
 "    "      "    "      "    "      "    "     "

0028 2264   6574 6365   6C65 7320   6570 7954   112  "Type selected"(."
7275 4320   6F20 2020   2020 2220   5357 2024   116  "$ WS "      o Cur"
7320 6570   7954 2065   646F 4D20   746E 6572   120  "rent Mode Type s"
2220 5357   2024 0029   2264 6574   6365 6C65   124 "elected").$ WS ""


dmp> R 2,1              (This selects block 2, copy 1, in the RCT area)

0000 0000   3000 0038   4000 0000   0000 0000   +  0
0000 0000   2000 0157   0000 0000   0000 0000   +  4
0000 0000   0000 0000   0000 0000   0000 0000   +  8
0000 0000   0000 0000   0000 0000   0000 0000   + 12
0000 0000   0000 0000   0000 0000   0000 0000   + 16
0000 0000   0000 0000   0000 0000   0000 0000   + 20

 "    "      "    "      "    "      "    "
 "    "      "    "      "    "      "    "

0000 0000   0000 0000   0000 0000   0000 0000   + 112
0000 0000   0000 0000   0000 0000   0000 0000   + 116
0000 0000   0000 0000   0000 0000   0000 0000   + 120
0000 0000   0000 0000   0000 0000   0000 0000   + 124

dmp> <CONTROL-C>

RAUTIL>
```

16.4.5 EXIT

This command causes an exit from RAUTIL when entered from the RAUTIL> prompt. When using functions within RAUTIL (such as the dump or the head commands) use CONTROL–C to exit and return to the RAUTIL> prompt.

16.4.6 HEAD

This command allows you to examine the failures on a head–by–head basis to see where scratches exist on the media.

At the RAUTIL> prompt, enter the command HEAD. When prompted "What head ?", you must respond with the particular head number (decimal). Entering ALL will cause a replacement listing on a head–by–head basis for all heads individually. Use CONTROL–C to exit from the HEAD mode of operation. The RAUTIL> prompt will return. An example follows. Following the example is a legend explaining most of the terminology used in the display.

```
RAUTIL> HEAD

What head ? 12

  DB       LBN       RBN     PCYL  GRP  HEAD POS   TYP
------------------------------------------------------

(  3.)    8514. =>   166.   11.   12.   12.  8.  PRI
(  6.)   26341. =>   516.   36.   12.   12. 37.  PRI
(  7.)   36333. =>   712.   50.   12.   12. 33.  PRI
( 14.)   82039. =>  1608.  114.   12.   12. 43.  PRI
( 37.)  229123. =>  4492.  320.   12.   12. 43.  PRI
( 48.) *******. =>  5892.  420.   12.   12. 11.  UNUSABLE
( 56.)  352638. =>  6914.  493.   12.   12. 36.  PRI
( 63.)  402638. =>  7894.  563.   12.   12.  4.  PRI
( 69.)  440459. =>  8636.  616.   12.   12. 35.  PRI
( 79.)  503313. =>  9868.  704.   12.   12.  5.  PRI
( 83.)  531854. => 10428.  744.   12.   12. 38.  PRI
(100.)  643937. => 12626.  901.   12.   12. 23.  PRI
(101.)  649639. => 12738.  909.   12.   12. 13.  PRI
(102.)  656793. => 12878.  919.   12.   12. 27.  PRI
(107.)  688219. => 13494.  963.   12.   12. 37.  PRI
(120.)  774614. => 15188. 1084.   12.   12. 38.  PRI
(136.)  877433. => 17204. 1228.   12.   12. 41.  PRI

replacement by head for the RA81, unit# $5$DUA230, serial # 137579

HEAD     0  1  2  3  4  5  6  7  8  9 10 11 12 13  TOTAL

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

PRI      0  0  0  0  0  0  0  0  0  0  0  0 16  0    16

NON-PRI  0  0  0  0  0  0  0  0  0  0  0  0  0  0     0

BAD-RBN  0  0  0  0  0  0  0  0  0  0  0  0  1  0     1

TOTAL    0  0  0  0  0  0  0  0  0  0  0  0 17  0    17

What head ? <CONTROL-C>

RAUTIL>
```

Table 16–2: Legend for HEAD Command

DB	Descriptor block number. This is the relative block number within the RCT that contains the descriptor for this entry.	
LBN	Logical block. This is a block from the usable host area that has been replaced as described in the RCT.	
RBN	Replacement block. This is the block that currently contains the data for the corresponding logical block listed in the LBN column.	
PCYL	Physical cylinder. This is the physical cylinder containing the corresponding LBN.	
GRP	This is the logical group number within the cylinder that contains the LBN.	
HEAD	Disk R/W head. This is the physical R/W head that would be used to read the corresponding LBN. It also describes the media surface or a portion of the media surface containing the LBN.	
POS	(Position from index.) This is the physical sector from index that corresponds to the LBN.	
DESC	(Descriptor contents.) This is the hexadecimal contents of the specific RCT descriptor that corresponds to the LBN/RBN entry listed in the output.	
TYP	(Type of descriptor.) This is a translation of the descriptor code field within the RCT descriptor.	
	PRI =	Primary replaced LBN
	NON–PRI =	Non–primary replaced LBN
	UNUSABLE =	BAD RBN

NOTE

Asterisks (*******) in the LBN column indicate that an RBN descriptor was found in the RCT table referencing that particular RBN as bad and unusable.

16.4.7 HELP

The HELP command displays a summary of all commands on your terminal. The summary displayed is similar to that listed in Section 16.3 section of this document.

16.4.8 MODIFY

The MODIFY command allows you to modify an LBN on a longword boundary. The drive must be mounted FOREIGN and you must have CMKRNL (change mode kernel) privileges to execute this command.

An example follows. In this example the user entered the MODIFY command at the RAUTIL> prompt. When prompted for the LBN to modify, the user entered "891050." The program then dumps the current contents of the specified LBN.

Next, you are prompted for the number of a longword to modify. In this example, the user selected longword number 4. The program then requests the pattern that is to be entered into the specified longword. This value must be 1 to 8 characters long specified in hexadecimal format. Values less than 8 characters are zero–filled for the entire longword.

The program continues to request longword numbers and patterns. When you are satisfied that all modifications have been entered, enter WRITE command at the longword number prompt to terminate the changes. At this point, the program will display the dump buffer showing all the modifications performed.

You will then be prompted to write the record. A "Y" will write the LBN (including modifications) back to the disk. Any other response will abort the MODIFY operation.

This command may not be used to modify blocks in the RCT on an HSC controller. Use the HSC utility DKUTIL for this purpose.

```
RAUTIL> MODIFY

what lbn ? 891050

0000 0000   0000 0000   0000 0000   0000 0000   0    "................"
0000 0000   0000 0000   0000 0000   0000 0000   4    "................"
0000 0000   0000 0000   0000 0000   0000 0000   8    "................"
0000 0000   0000 0000   0000 0000   0000 0000   12   "................"
0000 0000   0000 0000   0000 0000   0000 0000   16   "................"

 "    "      "    "      "    "      "    "    "
 "    "      "    "      "    "      "    "    "

0000 0000   0000 0000   0000 0000   0000 0000   120  "................"
0000 0000   0000 0000   0000 0000   0000 0000   124  "................"


longword to modify, or "WR" to write the block  4
what pattern (hex) ? 12345678
longword to modify, or "WR" to write the block  15
what pattern (hex) ? FFFFF
longword to modify, or "WR" to write the block  WRITE

the modified record contents is:

0000 0000   0000 0000   0000 0000   0000 0000   0    "................"
0000 0000   0000 0000   0000 0000   1234 5678   4    "xV4............."
0000 0000   0000 0000   0000 0000   0000 0000   8    "................"
000F FFFF   0000 0000   0000 0000   0000 0000   12   "................"
0000 0000   0000 0000   0000 0000   0000 0000   16   "................"

 "    "      "    "      "    "      "    "    "
 "    "      "    "      "    "      "    "    "

0000 0000   0000 0000   0000 0000   0000 0000   120  "................"
0000 0000   0000 0000   0000 0000   0000 0000   124  "................"

write the record  (y/n) ? Y

RAUTIL>
```

16.4.9 NEXT

The NEXT command deselects the current drive and allows you to request the next drive or select another generic device type. (See Section 16.2.2.)

NOTE

The NEXT command only releases the channel associated with the previous drive, and no VMS dismounts are attempted.

If a log file was generated for the previous drive, it will be closed. After executing the NEXT function, you will be prompted for another log file (optional) for the new device selected. An example follows:

```
RAUTIL> NEXT

what device ? NERMAL$DUA33

create a log file ? N

device is an RA81, serial# 137427, attached to a UDA50,
error count is 0

do you want to verify rct copy consistency ? (y/n) Y

RAUTIL>
```

16.4.10 SCRUB

The SCRUB command causes the entire user LBN area of the disk to be scanned in either full–track or one–sector mode. You may select the starting LBN and the mode.

The term SCRUB means scanning the entire disk, reading all host/user LBNs, and replacing any blocks that fail the BBR (bad block replacement) algorithm. With this command, all host/user LBNs are scanned and read sequentially. Any blocks read by a DSA controller that cause the BBR flag to set will ultimately cause the BBR software to be invoked. Blocks that fail the BBR test software will, in fact, be replaced.

On VAX/VMS systems using HSC controllers, the HSC performs the actual test and replacement process. On VAX/VMS systems using xDA–type controller, VMS performs the test and replacement process. The SCRUB function of RAUTIL merely provides the means to force the entire host/user area of a DSA disk to be scanned and read using the existing BBR functionality within the system. If you wish to force a block replacement, use the BBR command described in Section 16.4.2.

There is another benefit in using the SCRUB command. Since this command forces a read of all host/user LBNs, any blocks that contain forced errors, EDC errors, or exhibit other problems, will also be displayed. This is somewhat different from the ANALYZE command that only reads replaced LBNs to find forced errors.

Since RAUTIL may be interrogating (or scrubbing) a disk that is being interactively accessed by other users, the time to complete the scrub function varies depending upon system load, disk activity, or controller usage. This operation typically requires 10 minutes to over an hour to complete. If single sector mode is selected, it takes considerably longer.

Example 16–2 shows the results of a SCRUB operation.

Example 16–2: SCRUB Operation

```
RAUTIL> SCRUB

Starting LBN ? 0
Single sector scrub (y/n) ? N
```

Don't do this it will Take Forever

```
starting scrub at 23-SEP-1987 13:26:36.79
starting error count is 0

reading lbn      51. at Lcyl 0.
reading lbn    5763. at Lcyl 8.
reading lbn   11475. at Lcyl 16.
reading lbn   17187. at Lcyl 24.
reading lbn   22899. at Lcyl 32.
reading lbn   28611. at Lcyl 40.

 "        "      "     "    "    "
 "        "      "     "    "    "

reading lbn  856851. at Lcyl 1200.
reading lbn  862563. at Lcyl 1208.
reading lbn  868275. at Lcyl 1216.
reading lbn  873987. at Lcyl 1224.
reading lbn  879699. at Lcyl 1232.
reading lbn  885411. at Lcyl 1240.

scrub completed at 23-SEP-1987 13:46:40.29
ending error count is 0

RAUTIL>
```

16.4.11 SUMMARY

The SUMMARY command is similar to the ANALYZE command except that only a summary report of the replacements is displayed on the user terminal. An example of the SUMMARY command follows.

Example 16–3: SUMMARY Command

```
RAUTIL> SUMMARY

replacement by head for the RA81, unit# $5$DUA230, serial # 137579

HEAD       0  1  2  3  4  5  6  7  8  9 10 11 12 13   TOTAL

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

PRI        0  2 46 25 17  6  8 13  2  6  5  5 16 24     175

NON-PRI    0  1  1  0  0  0  0  1  0  0  0  0  0  0       3

BAD-RBN    0  1  1  0  0  0  0  1  0  1  0  0  1  0       5

TOTAL      0  4 48 25 17  6  8 15  2  7  5  5 17 24     183

RAUTIL>
```

16.4.12 TL - TRANSLATE LBN

This command translates an LBN into a cylinder, group, track, or sector. An example of this command follows.

Example 16–4: TL Command

```
RAUTIL> TL

what lbn ? 8943

       pri rbn....................  175
       lbns protected by pri rbn..  8925-8976
       cyl........................  12
       grp........................  7
       trk........................  0
       head.......................  7
       position from index........  12
       phys block number..........  9112
       rct blk....................  3
       offset into rct blk........  47
       lbn addr of rct block......  891075
       rbn range..................  128-255

what lbn ? <CONTROL-C>

RAUTIL>
```

16.4.13 TR—TRANSLATE RBN

This command translates an RBN into a cylinder, group, track, or sector. An example of this command follows.

Example 16–5: TR Command

```
RAUTIL> TR

what rbn ? 983

        rbn..........  983
        cyl..........  70
        grp..........  3
        trk..........  0
        head.........  3
        pos..........  41
        phy blk num..  51157

what rbn ? <CONTROL-C>

RAUTIL>
```

16.4.14 WRITE

The WRITE command provides the capability to write a longword pattern to an LBN and fill the entire block. The longword pattern will be repeated 128 times in the block. The drive must be mounted FOREIGN and the user must have CMKRNL (change mode kernel) privileges to execute this command. This command may not be used to write blocks into the RCT on an HSC controller. Use the HSC utility DKUTIL for this purpose. In the example below, the user is prompted for the LBN to write and the pattern to use. Only one LBN may be specified at a time. The pattern entered must be a hexadecimal value from 1 to 8 characters long. If less than 8 characters are specified, the longword is zero–filled. The example also shows the use of the DUMP command to verify the contents of the specified LBN that was written.

Example 16–6: WRITE Command

```
RAUTIL> WRITE

what lbn ? 891050

what pattern (hex) ? 1111FFFF

RAUTIL> DUMP

dmp> L 891050

1111 FFFF   1111 FFFF   1111 FFFF   1111 FFFF  0    "................"
1111 FFFF   1111 FFFF   1111 FFFF   1111 FFFF  4    "................"
1111 FFFF   1111 FFFF   1111 FFFF   1111 FFFF  8    "................"

 "    "      "    "      "    "      "    "    "
 "    "      "    "      "    "      "    "    "

1111 FFFF   1111 FFFF   1111 FFFF   1111 FFFF  116  "................"
1111 FFFF   1111 FFFF   1111 FFFF   1111 FFFF  120  "................"
1111 FFFF   1111 FFFF   1111 FFFF   1111 FFFF  124  "................"

dmp> <CONTROL-C>

RAUTIL>
```

16.5 TROUBLESHOOTING and USING RAUTIL

The following section describes some uses for RAUTIL and what information can be derived from its operation. This section does not include all the capabilities of RAUTIL but provides some basic samples and interpretations.

16.5.1 Radial Scratches

Refer to the following example in which the user has used the HEAD command to analyze the LBN replacements associated with head 4 on an RA80. Notice that there are 18 line entries that all have the same POS value of 21. This represents 18 logical blocks that have been replaced. The common factor is that these blocks are all positioned on the same physical sector (sector 21) from index. Also note that most of the values displayed in the PCYL (physical cylinder) column indicate that these LBNs are on adjacent cylinders. Since this display represents replaced blocks associated with the same head, we know that they are all on the same surface (or a portion of the same surface for the RA80) and, therefore, adjacent tracks.

Given these facts, we can conclude that this display probably represents some radial scratches on the media. A radial scratch (or defect in the media) is one that generally has the property of being aligned perpendicular to the rotation of the media.

In this example, there are three radial scratches.

1. Scratch #1 is positioned at physical sector 21 and crosses 6 adjacent tracks (noted by cylinders 58 through 63).
2. Scratch #2 is positioned at physical sector 21 and crosses 12 adjacent tracks (noted by cylinders 107 through 118).
3. Scratch #3 is positioned at physical sector 30 and crosses 3 adjacent tracks (noted by cylinders (174 through 176).

It is also possible that scratches 1 and 2 are the same scratch and that the tracks between cylinder 63 and 107 are not so severely affected by the defect as to cause block replacement. It is also possible that the defect was created by a "skipping" action, in the same way a rock can be made to skip across a pond. Radial defects are generally not caused while the media is spinning but rather when it is stationary. There are probably a hundred reasons why scratches occur. Some of them include excessive mishandling of an HDA, misuse of the HDA lock lever, drive failures causing head movement while the media is not spinning, manufacturing defects, and so on.

Note that since there are 478 tracks/inch in an RA80 HDA, the largest of these scratches is only about 0.025 inches long. Also remember that you are observing the analysis of LBNs whose data has been relocated to RBNs elsewhere on the disk. This is a result of either the manufacturing scanner/format process or error recovery and BBR techniques employed by the Digital Storage Architecture (DSA). You are observing a theory and probability of why the LBNs have been replaced.

If the defects noted here are severe enough, the scratches may increase in size over time. Monitor this by periodically using RAUTIL and its log file feature to take snapshots of this disk and compare the results. If further errors are encountered and/or the number of replacements associated with these scratches increase, you may have to replace the media.

If no further replacements or errors are associated with these LBNs, there is no cause for alarm. The data is safely stored elsewhere on the disk. Most disks contain scratches. With RAUTIL, you now have more visibility into the disk.

```
RAUTIL> HEAD

What head ? 4

  DB        LBN        RBN      PCYL    GRP  HEAD POS TYP
---------------------------------------------------------
(  8.)   25317. =>    816.    58.      0.     4. 21.  PRI
(  8.)   25735. =>    830.    59.      1.     4. 21.  PRI
(  8.)   26185. =>    844.    60.      0.     4. 21.  PRI
(  8.)   26603. =>    858.    61.      1.     4. 21.  PRI
(  8.)   27053. =>    872.    62.      0.     4. 21.  PRI
(  8.)   27471. =>    886.    63.      1.     4. 21.  PRI
( 13.)   46567. =>   1502.   107.      1.     4. 21.  PRI
( 13.)   47017. =>   1516.   108.      0.     4. 21.  PRI
( 13.)   47435. =>   1530.   109.      1.     4. 21.  PRI
( 14.)   47885. =>   1544.   110.      0.     4. 21.  PRI
( 14.)   48303. =>   1558.   111.      1.     4. 21.  PRI
( 14.)   48753. =>   1572.   112.      0.     4. 21.  PRI
( 14.)   49171. =>   1586.   113.      1.     4. 21.  PRI
( 14.)   49621. =>   1600.   114.      0.     4. 21.  PRI
( 14.)   50039. =>   1614.   115.      1.     4. 21.  PRI
( 14.)   50489. =>   1628.   116.      0.     4. 21.  PRI
( 14.)   50907. =>   1642.   117.      1.     4. 21.  PRI
( 14.)   51357. =>   1656.   118.      0.     4. 21.  PRI
( 21.)   75670. =>   2440.   174.      0.     4. 30.  PRI
( 21.)   76088. =>   2454.   175.      1.     4. 30.  PRI
( 21.)   76538. =>   2468.   176.      0.     4. 30.  PRI

replacement by head for the RA80, unit# $1$DUA50, serial # 18245

HEAD        0   1   2   3   4   5   6   7   8   9  10  11  12  13   TOTAL
-------------------------------------------------------------------

PRI         0   0   0   0  21   0   0   0   0   0   0   0   0   0      21

NON-PRI     0   0   0   0   0   0   0   0   0   0   0   0   0   0       0

BAD-RBN     0   0   0   0   0   0   0   0   0   0   0   0   0   0       0

TOTAL       0   0   0   0  21   0   0   0   0   0   0   0   0   0      21

RAUTIL>
```

16.5.2 Forced Errors

Refer to the following example. The user has performed an ANALYZE command for an RA81 HDA. The result shows some forced errors at LBNs 36620, 36623, 36624, 36627, and 36623. Remember, the ANALYZE command only scans LBNs whose data has already been stored into replacement blocks called RBNs. These forced errors represent data that is correctly stored into replacement blocks and was at one time unrecoverable.

Several points should be taken into consideration here. These may or may not be the only forced errors on the disk. The ANALYZE function only scans LBNs whose data has been replaced by RBNs and have replacement descriptor entries in the RCT. In order to determine if other forced errors exist, you would have to use the SCRUB command which causes RAUTIL to read every user LBN on the disk.

When dealing with forced errors, the only guaranteed method of recovery is to replace the file(s) containing the forced errors with known good copies, or again perform the operation that creates these files. RAUTIL does not deal with file structures. It deals with absolute blocks referenced as LBNs and makes no attempt at associating LBNs with file structures.

In the VAX/VMS environment, it is probable that several of these forced errors are not currently part of an existing file. Some or all may, in fact, be forced errors that exist with blocks that are not currently part of an existing file but instead part of a pool of currently unused, available blocks. When these blocks become allocated and rewritten (with new data), the forced error will no longer appear or be associated with these particular blocks. Blocks with forced errors that are unused by VMS will generally have no affect on the operating system.

```
RAUTIL> ANALYZE

  DB       LBN       RBN       CYL HEAD POS    DESC   TYP
-----------------------------------------------------------

(  2.)       15.  =>     0.     0.  0.  15.  2000 000F  PRI
(  4.)    14440.  =>   283.    20.  3.  49.  2000 3868  PRI
(  7.)    36630.  =>   714.    51.  4.  16.  3000 8F16  NON-PRI
lbn 36624, PC= 000051D5 2144 %SYSTEM-F-FORCEDERROR, forced error
flagged in last sector read
(  7.)    36624.  =>   715.    51.  4.  10.  3000 8F10  NON-PRI
lbn 36620, PC= 000051D5 2144 %SYSTEM-F-FORCEDERROR, forced error
flagged in last sector read
(  7.)    36620.  =>   716.    51.  4.   6.  3000 8F0C  NON-PRI
(  7.)    36616.  =>   717.    51.  3.  39.  2000 8F08  PRI
(  7.)    36625.  =>   718.    51.  4.  11.  2000 8F11  PRI
lbn 36623, PC= 000051D5 2144 %SYSTEM-F-FORCEDERROR, forced error
flagged in last sector read
(  7.)    36623.  =>   719.    51.  4.   9.  3000 8F0F  NON-PRI
lbn 36632, PC= 000051D5 2144 %SYSTEM-F-FORCEDERROR, forced error
flagged in last sector read
(  7.)    36632.  =>   720.    51.  4.  18.  3000 8F18  NON-PRI
lbn 36627, PC= 000051D5 2144 %SYSTEM-F-FORCEDERROR, forced error
flagged in last sector read
(  7.)    36627.  =>   721.    51.  4.  13.  3000 8F13  NON-PRI
(  7.)    36634.  =>   722.    51.  4.  20.  3000 8F1A  NON-PRI
(  7.)    36619.  =>   723.    51.  4.   5.  3000 8F0B  NON-PRI
               "          "         "      "            "
               "          "         "      "            "
(137.)   884889.  => 17350. 1239.  4.  43.  200D 8099  PRI
(138.)   891023.  => 17471. 1247. 13.  28.  200D 988F  PRI

replacement by head for the RA81, unit# $5$DUA1, serial # 1271811603

HEAD      0  1  2   3   4   5  6  7  8   9  10 11  12  13  TOTAL
----------------------------------------------------------

PRI      10  4  8  23   4   9  2  0  5  12  3   1   1   3     85

NON-PRI   0  0  0   0  12   1  0  0  0  0   0   0   0   0     13

BAD-RBN   0  2  0   0   0   0  0  0  0  0   0   0   0   0      2

TOTAL    10  6  8  23  16  10  2  0  5  12  3   1   1   3    100

RAUTIL>
```

If, however, some of these forced errors are part of one or more files, how do we know which files need to be replaced? One method is to use the ANALYZE/DISK command which is a VMS DCL command and not a RAUTIL command. To do this, make sure the disk is mounted (not foreign) and perform the following command at the VMS prompt.

```
$ ANALYZE/DISK/READ_CHECK  Device-name:
```

This will cause VMS to read and check the structures of all existing files on the selected disk and report any discrepancies on the user terminal. This will include reporting any files that contain forced errors. Now it is merely a matter of replacing these files with known good copies. Also note that the VMS ANALYZE/DISK command will also report other structure errors that may exist. A discussion of these is beyond the scope of this document.

You can also use a variation of this command to create a soft copy of the results of the read_check operation. Use the following:

```
$ SPAWN/OUTPUT=File_spec  ANALYZE/DISK/READ_CHECK Device-name:
```

Here you specify the name of a file to be created to contain the results of the ANALYZE/DISK command. This file can later be displayed on the terminal or printed in hardcopy format.

Another observation should be noted about the RAUTIL ANALYZE results. Notice that all of the forced errors are associated with the same head (head 4). Also note that there are several other entries associated with head 4, and many of them indicate non–primary replacement. You may be inclined to perform the HEAD command to this disk and select head 4 for further analysis. Section 16.5.3 discusses a further analysis of this sample.

16.5.3 Circular Defects

This section is a continuation of a sample analysis started in the previous section.

Refer to the following example. The user has elected to analyze the replacement associated with head 4 on a RA81. Notice that there are 13 entries that indicate LBNs replaced from the same physical cylinder (PCYL 51). Since the analysis here is limited to show replacements for head 4, the conclusion is that all 13 LBNs replaced were from the same physical track. Since only one replacement can be primary on any given track, the remaining replacements had to be revectored as non–primary. Since most of the LBNs replaced occurred from the same track, the geometry of the symptom is considered circular.

We know from the previous section that several of the replacements resulted in forced errors. We can conclude that a failure probably occurred while the heads were positioned over cylinder 51, and something (contact erasure, contaminate, head/disk interference, defective spot in the media, vibration, etc.) caused the degradation and subsequent replacement of several blocks under head 4.

This may or may not be serious. If more LBNs continue to be replaced from this track or adjacent tracks, the problem may be growing as indicated by additional forced errors or errors in the VMS error log. Use of the SCRUB command may provide this additional knowledge. In this case, the media or a R/W head (for removable media) may need to be replaced. On the other hand, if the defect does not appear to be growing by evidence of continued errors and/or replacements, there may be no further cause for alarm. In this case, it will have merely been an exercise for the user to understand the symptoms that may have resulted in unexpected forced errors or errors in the error log.

The "logging" feature of RAUTIL could be used to record these results and be used for comparisons to another analysis at a later time. This would allow the user to monitor this particular disk and determine if problems are growing or just instantaneous in time. The DSA architecture was designed to provide adequate recovery during instantaneous failures. Deal with growing problems before they become catastrophic.

```
RAUTIL> HEAD

What head ? 4

  DB       LBN       RBN     PCYL   GRP  HEAD POS TYP
-----------------------------------------------------
(  7.)    36622. =>   712.    51.      4.     4.  8.  NON-PRI
(  7.)    36629. =>   713.    51.      4.     4. 15.  NON-PRI
(  7.)    36630. =>   714.    51.      4.     4. 16.  NON-PRI
(  7.)    36624. =>   715.    51.      4.     4. 10.  NON-PRI
(  7.)    36620. =>   716.    51.      4.     4.  6.  NON-PRI
(  7.)    36625. =>   718.    51.      4.     4. 11.  PRI
(  7.)    36623. =>   719.    51.      4.     4.  9.  NON-PRI
(  7.)    36632. =>   720.    51.      4.     4. 18.  NON-PRI
(  7.)    36627. =>   721.    51.      4.     4. 13.  NON-PRI
(  7.)    36634. =>   722.    51.      4.     4. 20.  NON-PRI
(  7.)    36619. =>   723.    51.      4.     4.  5.  NON-PRI
(  7.)    36655. =>   724.    51.      4.     4. 41.  NON-PRI
(  7.)    36645. =>   725.    51.      4.     4. 31.  NON-PRI
( 79.)   504332. =>  9888.   706.      4.     4. 48.  PRI
(137.)   884175. => 17336. 1238.       4.     4. 43.  PRI
(137.)   884889. => 17350. 1239.       4.     4. 43.  PRI

replacement by head for the RA81, unit# S5$DUA1, serial # 1271811603

HEAD        0   1   2   3   4   5   6   7   8   9  10  11  12  13   TOTAL
    ------------------------------------------------------

PRI         0   0   0   0   4   0   0   0   0   0   0   0   0   0       4

NON-PRI     0   0   0   0  12   0   0   0   0   0   0   0   0   0      12

BAD-RBN     0   0   0   0   0   0   0   0   0   0   0   0   0   0       0

TOTAL       0   0   0   0  16   0   0   0   0   0   0   0   0   0      16

What head ? <CONTROL-C>

RAUTIL>
```

16.5.4 Summary Analysis

Refer to the following example. The user elected to perform a SUMMARY command for an RA82 HDA. A couple of observations may be noted. The replacements associated with the even numbered heads appears to be higher than those of the odd numbered heads. This is probably normal since the even numbered heads are associated with the innermost cylinders on an RA82 where the bit density is much higher.

The replacements associated with head 10 are more than twice that of the other heads. You would probably be inclined to perform the HEAD command and select head number 10 for further analysis. If you compared this analysis to an RAUTIL log copy you obtained earlier for this disk and head 10 showed a relatively large number of replacements over a short period of time (within hours or a day), you may have to consider replacing the HDA.

If the comparison showed considerable replacements associated with all heads in a short time period, you should consider the possibility of a disk R/W electronics data path problem (R/W module, hybrid module, SDI problem, cable problem, etc.). Since the SUMMARY command allows you to compare replacements associated with all the heads, knowledge of the head select logic for the specified disk may prove very valuable. A single head select line from a microprocessor controlled circuit may affect a certain combination of read/write heads. If this combination of heads accounted for a large number of replacements, you could isolate the problem to a specific field replaceable unit (FRU) using the SUMMARY command in RAUTIL.

```
RAUTIL> SUMMARY

replacement by head for the RA82, unit# $1$DUA0, serial # 0

HEAD      0  1  2  3  4  5  6  7  8  9 10 11 12 13 14   TOTAL

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

PRI      29 14 30  6 14  4 12 15 38 33 89 10 18 15 33    360

SEC       1  0  0  0  0  0  0  0  0  1  4  0  0  0  2      8

BAD-RBN   0  0  0  0  0  0  0  1  0  0  0  0  1  0  2      4

TOTAL    30 14 30  6 14  4 12 16 38 34 93 10 19 15 37    372

RAUTIL>
```

16.5.5 EDC Errors

The following example shows how an EDC error would appear. Here, there is an EDC error detected while verifying LBN 56 during the ANALYZE command. Remember, since the ANALYZE command verifies replaced LBNs, the EDC error actually occurred while reading RBN 2 that contains the data (and EDC character) for LBN 56.

Most error detection code (EDC) errors are the result of controller data path problems. If, however, the EDC errors are accompanied with ECC errors, the problem is likely in the disk R/W data path, media, or SDI R/W data path.

```
RAUTIL> ANALYZE

  DB      LBN       RBN        CYL HEAD POS    DESC   TYP
----------------------------------------------------------

(  2.) *******. =>     1.    0.  1.  13.  4000 0000   UNUSABLE

lbn 56, PC= 000051D5 0054 %SYSTEM-F-CTRLERR, fatal controller error
do you want to continue ? (y/n) Y

(  2.)      56. =>     2.    0.  1.  19.  3000 0038   NON-PRI
(  2.)     343. =>     6.    0.  6.  17.  2000 0157   PRI
(  3.)    8514. =>   166.   11. 12.   8.  2000 2142   PRI
( 13.)   72438. =>  1420.  101.  6.  50.  2001 1AF6   PRI
( 14.)   82039. =>  1608.  114. 12.  43.  2001 4077   PRI
( 20.) *******. =>  2333.  166.  9.  21.  4000 0000   UNUSABLE

            "           "         "       "           "
            "           "         "       "           "
(137.)  887751. => 17406. 1243.  4.  49.  200D 8BC7   PRI
(138.)  888465. => 17420. 1244.  4.  49.  200D 8E91   PRI
(138.)  888520. => 17421. 1244.  5.  15.  200D 8EC8   PRI

replacement by head for the RA81, unit# $5$DUA230, serial # 137579

HEAD       0   1   2   3   4   5   6   7   8   9  10  11  12  13  TOTAL
           ------------------------------------------------------

PRI        0   2  46  25  17   6   8  13   2   6   5   5  16  24    175

NON-PRI    0   1   1   0   0   0   0   1   0   0   0   0   0   0      3

BAD-RBN    0   1   1   0   0   0   0   1   0   1   0   0   1   0      5

TOTAL      0   4  48  25  17   6   8  15   2   7   5   5  17  24    183

RAUTIL>
```

- Anal/Disk/Read-check Device:
- Must be mounted
- Forced errors
- EDC errors
- This will show you the bad files.

```
$ Spawn/out=Filename.Typ - Anal/Disk/Read Dev:
    /Nowait ; gets $ back
```

DSA TROUBLESHOOTING COURSE

Lab Exercise #4

DSAERR/DSA301 Error Log Tool

Review the DSAERR/DSA301 user guide section of the Student Guide before proceeding with these exercises.

1. Log into your student account

2. **$ SET DEF [STUDENTx.ERRORLOG_TOOL]**

3. **$ RUN DSA301**

NOTE

If your system is using VMS version 5.0 or higher, then run DSA303 instead of DSA301 for the duration of this lab exercise.

4. Enter the following parameters for the program:

```
Input file          [SYS$ERRORLOG:ERRLOG.SYS] ?         MASTER.DAT
Output file (file-name.ext)               [TERMINAL] ?    MASTER_1.OUT
Device(s),Type(s)  (Daan,Rann) 'DJ%1,RA60'  [ALL] ?     <cr>
HEX Event Code(s)   (nnnn) '1AB,E8,%%6B,*4'  [ALL] ?     <cr>
Starting date (dd-mmm-yyyy hh:mm:ss.cc)   [FIRST] ?     <cr>
Ending date (dd-mmm-yyyy hh:mm:ss)         [LAST] ?     <cr>
Report Type (Physical, Geographic, Summary, Verbose) [P] ?  <cr>
```

MASTER.OUT will be the master list of all error log entries contained in the binary error log file, MASTER.DAT. Print MASTER.OUT for reference during this lab exercise. This physical report type contains entries in the order in which they originated from the binary file. Review this report to get familiar with the various types of entries it contains.

5. Run DSA301 using the following parameters:

```
Input file          [SYS$ERRORLOG:ERRLOG.SYS] ?          MASTER.DAT
Output file (file-name.ext)               [TERMINAL] ?     MASTER_2.OUT
Device(s),Type(s)  (Daan,Rann) 'DJ%1,RA60'  [ALL] ?      <cr>
HEX Event Code(s)   (nnnn) '1AB,E8,%%6B,*4'  [ALL] ?      <cr>
Starting date (dd-mmm-yyyy hh:mm:ss.cc)   [FIRST] ?      <cr>
Ending date (dd-mmm-yyyy hh:mm:ss)         [LAST] ?      <cr>
Report Type (Physical, Geographic, Summary, Verbose) [P] ?  G <cr>
```

These selections are similar to those in step 4 except for the report type "G" (geographic). Type or print the results and notice how the entries have been sorted in numeric order according to device name, block number, cylinder number, track/head number, etc.

6. Run DSA301 and select the following parameters:

```
Input file          [SYS$ERRORLOG:ERRLOG.SYS] ?          MASTER.DAT
Output file (file-name.ext)              [TERMINAL] ?     SAMPLE_1.OUT
Device(s),Type(s)   (Daan,Rann) 'DJ%1,RA60'   [ALL] ?     DUA2
HEX Event Code(s)   (nnnn) '1AB,E8,%%6B,*4'   [ALL] ?     *8
Starting date (dd-mmm-yyyy hh:mm:ss.cc)    [FIRST] ?     <cr>
Ending date (dd-mmm-yyyy hh:mm:ss)          [LAST] ?     <cr>
Report Type (Physical, Geographic, Summary, Verbose) [P] ?  <cr>
```

DUA2 is specified here to limit the selections to entries with device name DUA2. We have entered event code *8 to further limit the selection process to only those entries with MSCP status/event codes that end in the number 8 (hexadecimal). These codes will cause selections that are mostly R/W data or transfer related, such as ECC errors, header sync errors, etc.

This is one way to customize the report to fit your specific needs. In this case, review the data errors associated with DUA2. Type or print the output file for further review. Review the DSA301 user guide section of your Student Guide for further details on the many ways to select device names, device types, and event codes to customize the reporting results.

7. Run DSA301 and use the same parameters as in the previous step, except this time specify report type "S" for a summary report.

```
Input file          [SYS$ERRORLOG:ERRLOG.SYS] ?          MASTER.DAT
Output file (file-name.ext)              [TERMINAL] ?     SAMPLE_2.OUT
Device(s),Type(s)   (Daan,Rann) 'DJ%1,RA60'   [ALL] ?     DUA2
HEX Event Code(s)   (nnnn) '1AB,E8,%%6B,*4'   [ALL] ?     *8
Starting date (dd-mmm-yyyy hh:mm:ss.cc)    [FIRST] ?     <cr>
Ending date (dd-mmm-yyyy hh:mm:ss)          [LAST] ?     <cr>
Report Type (Physical, Geographic, Summary, Verbose) [P] ?  S <cr>
```

The summary report provides you with a map of how the data–related entries in the error log file are distributed with respect to the physical translations (head and cylinder) that are automatically provided by the program. Type or print the output file and review the results.

Review the DSA301 user guide section of your Student Guide for further information on how to use a summary report.

8. Run DSA301 and use the following selection parameters:

```
Input file                [SYS$ERRORLOG:ERRLOG.SYS] ?        MASTER.DAT
Output file (file-name.ext)              [TERMINAL] ?        SAMPLE_3.OUT
Device(s),Type(s)   (Daan,Rann) 'DJ%1,RA60'   [ALL] ?        ,RA82
HEX Event Code(s)   (nnnn) '1AB,E8,%%6B,*4'   [ALL] ?        EB
Starting date (dd-mmm-yyyy hh:mm:ss.cc)    [FIRST] ?        <cr>
Ending date (dd-mmm-yyyy hh:mm:ss)          [LAST] ?        <cr>
Report Type (Physical, Geographic, Summary, Verbose) [P] ?  <cr>
```

In this example, we have selected a device type (RA82) instead of device name. By placing a comma before the "RA82", we have instructed the program to accept any device name, as long as the device type was an RA82. We also specified event code EB to select all entries that contained drive detected errors (status/event code EB). Look at the physical report (SAMPLE_3.OUT) and notice that the entries now contain values for LED codes. The LED codes represent the drive LED error codes (drive detected errors).

9. Run DSA301 and select the following parameters:

```
Input file                [SYS$ERRORLOG:ERRLOG.SYS] ?       MASTER.DAT
Output file (file-name.ext)              [TERMINAL] ?       SAMPLE_4.OUT
Device(s),Type(s)  (Daan,Rann) 'DJ%1,RA60'  [ALL] ?       <cr>
HEX Event Code(s)  (nnnn) `1AB,E8,%%6B,*4'  [ALL] ?       34
Starting date (dd-mmm-yyyy hh:mm:ss.cc)   [FIRST] ?       <cr>
Ending date (dd-mmm-yyyy hh:mm:ss)         [LAST] ?       <cr>
Report Type (Physical, Geographic, Summary, Verbose) [P] ?  P <cr>
```

In this example, the parameters selected caused the program to select all entries containing an MSCP status/event code of 34. This event code indicates host LBNs that were flagged for BBR (bad block replacement) during a read operation but did not fail the BBR test and were not replaced. Using this approach, you can easily spot LBNs that were considered marginal by noting the number of times they appear in the error log.

CAUTION ! The transfer of an LBN during a single MSCP command could result in multiple entries in the error log. Use the verbose report to determine if all the entries for an identical block are logged for the same command reference number. If the same LBN appears with event code 34 and it is logged against several different commands (different command reference numbers), that block may be a candidate for manual replacement.

Using the same parameters but selecting the summary report would provide a geographic map of the LBNs that were flagged for BBR and **NOT REPLACED**. This technique may provide some useful information for troubleshooting.

10. Run DSA301 and select the following parameters:

```
Input file                [SYS$ERRORLOG:ERRLOG.SYS] ?       MASTER.DAT
Output file (file-name.ext)              [TERMINAL] ?       SAMPLE_5.OUT
Device(s),Type(s)  (Daan,Rann) 'DJ%1,RA60'  [ALL] ?       <cr>
HEX Event Code(s)  (nnnn) `1AB,E8,%%6B,*4'  [ALL] ?       14
Starting date (dd-mmm-yyyy hh:mm:ss.cc)   [FIRST] ?       <cr>
Ending date (dd-mmm-yyyy hh:mm:ss)         [LAST] ?       <cr>
Report Type (Physical, Geographic, Summary, Verbose) [P] ?  P <cr>
```

This demonstrates a slight variation to the previous example. Here we have generated a physical report (SAMPLE_5.OUT) to show all the blocks that **HAVE BEEN REPLACED** by BBR during the period of time specified. This report (physical) and a summary report may also prove useful in troubleshooting DSA disk/controllers.

11. Run DSA301 and select the following parameters:

```
Input file              [SYS$ERRORLOG:ERRLOG.SYS] ?        MASTER.DAT
Output file (file-name.ext)              [TERMINAL] ?        SAMPLE_6.OUT
Device(s),Type(s)  (Daan,Rann) ’DJ%1,RA60’   [ALL] ?        <cr>
HEX Event Code(s)  (nnnn) ‘1AB,E8,%%6B,*4’   [ALL] ?        14,34,48,6B
Starting date (dd-mmm-yyyy hh:mm:ss.cc)   [FIRST] ?        <cr>
Ending date (dd-mmm-yyyy hh:mm:ss)         [LAST] ?        <cr>
Report Type (Physical, Geographic, Summary, Verbose) [P] ?  G <cr>
```

The resulting report contains all error log entries with event codes of 48 and 6B (header–related errors) as well as 14 and 34 (BBR testing–related events). Notice that LBN 35896 had an event code of 48 (indicating a corrupt header), followed by an event code of 14 (indicating the block was subsequently replaced).

DUA124, another RA81, shows several entries with status event code 6B, positioner error (header not found). There are several LBNs logged with the code 6B but none of these blocks are duplicated with a code of 34 or 14 to indicate any BBR activity. Also note that most of these LBNs are on the same cylinder and associated with track/heads 1 and 3. There may be a positioning problem with cylinder 627, especially if errors continued to occur on other track/heads within this cylinder. There may also be integrity problems with the media in the area of these blocks causing header sync timeout (status/event=6B on HSC). If these are the only blocks flagged, then manual block replacement would be a simple solution.

This example shows how to customize the program selection process and use the results to look for trends in the DSA/disk subsystem. When using the program to look for trends related to specific block numbers, use the geographic report (report type "G") to assure that all of the entries for a given block number are grouped together for easier reference.

12. Run DSA301 using the following selection parameters:

```
Input file              [SYS$ERRORLOG:ERRLOG.SYS] ? MASTER.DAT
Output file (file-name.ext)              [TERMINAL] ? SAMPLE_7.OUT
Device(s),Type(s)  (Daan,Rann) ’DJ%1,RA60’   [ALL] ? DUA2,RA81
HEX Event Code(s)  (nnnn) ‘1AB,E8,%%6B,*4’   [ALL] ? <cr>
Starting date (dd-mmm-yyyy hh:mm:ss.cc)   [FIRST] ? 22-AUG-1986 14:06:23.00
Ending date (dd-mmm-yyyy hh:mm:ss)         [LAST] ? 22-AUG-1986 14:06:25.37
Report Type (Physical, Geographic, Summary, Verbose) [P] ? P <cr>
```

This example shows how to use the START DATE and END DATE prompts. The parameters were selected to illustrate an important concept. All the entries on this report have the same command reference number (you could confirm this with the verbose report). Carefully review the results. Consult your instructor if you are unclear.

13. Run DSA301 and select some parameters of your own choosing. Try the wild card features for device names and event codes. Experiment with the different report types and device types. Refer to the DSA301 user guide section of your Student Guide for further information about selection features and wild card options.

14. Locate the current binary system error log resident on your system at SYS$ERRORLOG. Copy the error log to your student account. The file name may either be ERRLOG.SYS or ERRLOG.OLD. You may need some assistance from the system manager, depending upon the privileges that have been assigned to your account.

Run DSA301, use this file as your input, and try some of the selections from the previous steps in this lab exercise.

DSA TROUBLESHOOTING COURSE

Lab Exercise #6

Forced Errors/EDC Errors

Forced Errors/EDC Errors
Lab Exercise 6

In this exercise, you will create some bad blocks containing forced errors and EDC errors on your scratch disk. Then you will use some tools and techniques to isolate these blocks and the files that contain these bad blocks. Your scratch disk contains a VMS directory similar to the student account on the system. The scratch disk also contains files similar to the ones in your system account. You will select blocks in files on the scratch disk. Be sure to follow the steps carefully. At no time should you be changing blocks in your student account on the system.

1. Log into your system account

2. **$ SET DEF DISK:[STUDENTM.MISC]**

   NOTE

   DISK: is the name of the scratch disk selected by the instructor. Use the VMS command SHOW DEVICE D to determine the exact name to use in place of the DISK:, including any allocation class information.

   Be sure to use the specific name of your scratch disk in place of the term DISK: throughout this exercise.

3. **$ COPY BLOCK.COM OLD1.COM**

4. **$ COPY BLOCK.COM OLD2.COM**

5. **$ DUMP/HEADER/BLOCK=(START:0,END:0) OLD1.COM**

   Make a note of the LBNs that are allocated to OLD1.COM. The information displayed by this command is extensive. Ask your instructor help you determine which LBN numbers are assigned to the file OLD1.COM.

6. **$ DUMP/HEADER/BLOCK=(START:0,END:0) OLD2.COM**

   Make a note of the LBNs that are allocated to OLD2.COM.

7. **$ DISM/NOUNL DISK:** our Disk ȯ

8. Go to the HSC for your disk, RUN DKUTIL, and install the write patch.

9. Display the RCT and save the hardcopy.

10. Refer to the LBNs allocated to these files in your notes from steps 5 and 6. Select three arbitrary LBNs from each of the files (OLD1.COM and OLD2.COM) and install them into the blank spaces in the work table below. As illustrated, two of the three blocks in each file should not be in the RCT list, and one of the three blocks in each file should be in the RCT list.

```
OLD1.COM     LBN  A = ________________   is NOT in the RCT list
             LBN  B = ________________   is NOT in the RCT list
             LBN  C = ________________   IS in the RCT list

OLD2.COM     LBN  D = ________________   is NOT in the RCT list
             LBN  E = ________________   is NOT in the RCT list
             LBN  F = ________________   IS in the RCT list
```

11. Using DKUTIL, issue the following commands. Ignore any errors displayed as a result of these commands.

NOTE

Be sure to substitute the actual LBN numbers from your work table in place of the letters A, B, C, D, E, and F in these commands.

```
DKUTIL> DUMP LBN A
DKUTIL> WRITE/FE LBN A

DKUTIL> DUMP LBN C
DKUTIL> WRITE/FE LBN C

DKUTIL> DUMP LBN D
DKUTIL> WRITE/FE LBN D

DKUTIL> DUMP LBN B
DKUTIL> WRITE/BADEDC LBN B

DKUTIL> DUMP LBN E
DKUTIL> WRITE/BADEDC LBN E

DKUTIL> DUMP LBN F
DKUTIL> WRITE/BADEDC LBN F
```

12. Exit from DKUTIL and return to your account on the VMS terminal.

13. Remount your scratch disk (DISK:) and set your default directory:

    $ SET DEF DISK:[STUDENTM.MISC]

14. Try to execute OLD1.COM using the command:

    $ @OLD1

    Enter arbitrary selections until you encounter errors from VMS. Observe how these errors are displayed for a typical user.

15. Try the command:

    $ COPY OLD1.COM TEMP.COM

    Note any errors that VMS may produce.

16. Try the command:

    $ TYPE OLD1.COM

    Note any errors that VMS may produce.

17. Enter the following command:

    $ ANALYZE/DISK/READ_CHECK DISK:

    Note any errors associated with OLD1.COM and OLD2.COM.

18. **$ SET DEF DISK:[STUDENTM.RAUTIL]**

19. RUN RAUTIL and select your scratch disk (DISK:).

20. Perform the RAUTIL ANALYZE command.

    Note that only LBNs C and F show up with errors. This is because the RAUTIL ANALYZE command only causes replaced LBNs to be scanned and verified.

21. Perform the RAUTIL SCRUB command.

    Now notice that all of the LBNs you modified show up with errors. The SCRUB command causes RAUTIL to read every physical LBN in the host area of the disk.

    Abort the scrub operation using Control–C after all of your selected LBNs have been displayed.

22. Enter the following VMS command:

    $ DELETE OLD1.COM

23. Re–execute the VMS command:

    $ ANALYZE/DISK/READ DISK:

    Note that only OLD2.COM shows errors for your arbitrary LBNs.

24. Re–execute the RAUTIL command SCRUB.

Notice that all of your arbitrary LBNs are still flagged as errors. This is because the VMS DELETE command only deletes the file pointer/header but does not actually rewrite or erase the allocated LBNs. These LBNs still exist with their errors and will not normally affect VMS. They will be rewritten (and corrected) the next time VMS allocates them to store some other file.

25. Execute the following VMS command:

$ **DELETE/ERASE OLD2.COM**

26. Perform the VMS command:

$ **ANALYZE/DISK/READ DISK:**

Note that all of your arbitrary LBNs are excluded.

27. Re–execute the RAUTIL SCRUB command.

Note that the arbitrary LBNs from OLD2.COM are no longer flagged. That's because the VMS DELETE/ERASE actually rewrites all LBNs associated with the OLD2.COM file as well as de–allocating them.

The arbitrary LBNs associated with OLD1.COM are still flagged as explained in step 24.

28. $ **DISMOUNT/NOUNLOAD DISK:**

29. Use DKUTIL on the HSC and dump each of the two arbitrary LBNs associated with OLD1.COM and rewrite them to the disk using the standard DKUTIL WRITE command with no special modifiers. This should correct errors associated with them.

Use DKUTIL on the HSC, select your scratch disk, and enter the following commands:

NOTE

Be sure to substitute the actual LBN numbers from your work table in place of the letters A, B, and C in these commands.

```
DKUTIL> DUMP LBN A
DKUTIL> WRITE LBN A

DKUTIL> DUMP LBN B
DKUTIL> WRITE LBN B

DKUTIL> DUMP LBN C
DKUTIL> WRITE LBN C
```

30. Exit from DKUTIL and return to your VMS account.

31. Re-mount the DISK:

32. Use the RAUTIL SCRUB command to verify that all of the arbitrary LBNs are no longer flagged with errors.

In this exercise, you used DUMP and WRITE commands to correct the modified LBNs. Writing an LBN using a known good controller will clear the forced error indicator and write good EDC. In our case, we knew that the contents of the LBNs were good (since we never changed them).

Normally, when you encounter blocks with forced errors or EDC errors, you will not know if the data is corrupt or not. Therefore, assume the LBNs are corrupt and replace the files with **KNOWN GOOD COPIES OF THE FILES**.

1 Summary

A. Replace files with forced errors using known good backup copies or recreated from a known good source.

B. ANAL/DISK/READ_CHECK (VMS command) identifies existing files that contain forced errors and/or EDC errors. Replace these files with copies from known good backup(s).

C. ANAL/DISK/READ_CHECK (VMS command) does not report errors for unused or unallocated LBNs. These will still exist, but they will be rewritten and corrected the next time VMS allocates them for storage.

D. The RAUTIL ANALYZE command only reports errors for LBNs that have been replaced (according to the RCT table).

E. The RAUTIL SCRUB command reports errors encountered for any host LBN whether it has been replaced or not.

LBNs that are not allocated or not currently used by operating system software but are written with bad EDC or forced error can be corrected by rewriting them with DKUTIL if they become a nuisance. This is not normally necessary as most system configurations ignore errors associated with unused LBNs

CHAPTER 17

DSAERR V3.01 USER DOCUMENT

17.1 OVERVIEW

DSAERR is an executable image which runs under VMS. DSAERR is capable of extracting selected DSA disk information from a VMS binary error log file and displaying it into a variety of formats. These formats are more condensed than the conventional styles presented by ERRFMT as used with the ANALYZE/ERROR command within VMS. The selected information provides only the elements necessary to understand the root nature of most DSA disk errors.

DSAERR is an extremely powerful service tool for analyzing DSA disk–related errors and is sometimes referred to as an error log tool. Each error log entry presented to VMS often results in one to two pages of VMS error log reporting. DSAERR reduces this entry to a single line. Some of the advantages of DSAERR include:

- Reducing error log entries from one or two pages to a single line.
- Performing automatic translation of logical block numbers (LBNs) to physical cylinder, track, sector, and head information associated with R/W transfer and bad block replacement error log entries.
- Providing manual translation of block numbers.
- Allowing the user to sort through the error log for specific status/event codes, disk drive types, device names, etc.
- Providing the ability to sort or summarize error log entries by geographic characteristics (cylinder, track, sector).
- Providing a soft copy output which may be printed as hardcopy at the user's discretion.
- Providing the use of wild card characters (* and %) to make the selection and sorting process more versatile.

DSAERR currently supports RA60, RA70, RA80, RA81, RA82, RA90 and associated DSA controllers (UDA, KDA, KDB, HSC).

17.1.1 Restrictions

Two forms of the DSAERR program are available: DSAERR.EXE which is linked and ready for execution using the VMS RUN command, and DSAERR.OBJ which is not linked to the libraries in VMS. For most systems, the executable (DSAERR.EXE) version is all that is required by the user. Occasionally, the particular version and configuration of the VMS system may require the program to be linked on the actual target system. To assure compatibility, it is recommended that DSAERR.OBJ be obtained and linked on the system for which it is intended. The VMS command to perform this is:

```
$ LINK DSAERR.OBJ

$ RUN DSAERR
```

NOTE

DSAERR.OBJ (Version 3.03) or DSA303.OBJ is required to execute with VMS Version 5.0 or higher.

17.2 SELECTION PARAMETERS

When run, DSAERR prompts the user for a variety of selection characteristics. A summary of all the prompts is shown below. The following sections describe each prompt and provide examples.

```
Input file:
Output file:
Device(s):
Event(s):
After:
Before:
Report:
```

Enter a question mark (?) at any prompt to obtain a brief summary of the information that may be supplied to that prompt. Enter HELP at any prompt to obtain more detailed information about what may be entered.

17.2.1 Input File

```
Input file:
```

Enter a carriage return <cr> to select the default VMS binary ERRLOG.SYS file. This will be extracted from the default SYS$ERRORLOG:ERRLOG.SYS. Users sometimes rename this file ERRLOG.OLD. Some users may prefer to use the ANALYZE/ERROR/BINARY command to extract a portion of the full ERRLOG.SYS file and produce a limited binary output file which can be used as the input file to DSAERR. Any file name specification may be used as long as it is a binary, formatted VMS error log file. Only one file specification may be entered. Following is an example.

```
Input file:  SYS$SYSMAINTENANCE:DISK_ERRORS.BINARY <cr>
```

If you enter TT <cr> in response to the input file prompt, the program will enter manual translation mode. Details for program operation in the manual translation mode are discussed in Section 17.3

17.2.2 Output File

```
Output file:
```

The user may respond in one of two ways to the output file prompt.

A carriage return <cr> to the output file prompt will cause the program to display all results to the user terminal (SYS$OUTPUT as the default).

1. Enter a file name specification to cause the program to generate a soft copy of all error log output generated by the program.

```
Output file:  TEST.OUT <cr>
```

17.2.3 Device(s) and Type(s)

```
Device(s):
```

The device name and device type may be specified a number of ways. A carriage return <cr> will default to all error log entries containing supported DSA disks and device names found in the specified input file. The general format for responding to this prompt is:

```
Daan,Rann        (Device name and/or device type separated by a comma)
```

"Daan" specifies the device name (DUA123, DUB2, DJA6, etc.). Wild card characters (* and %) may be used in the device name specification. Multiple device entries may be used if they are separated by a comma.

Examples:

DUA*	Selects all DUA devices ending with any number.
DJA10%	Selects all DJA devices with three–digit numbers beginning with 10.
DJ,DUA*1	Selects all DJ devices and DUA devices ending with a 1.

The device name may also include the controller name. This option allows you to capture error log events associated with a single controller channel. For example:

```
HSC007$DUA5          BRAVAX$DJA*
```

"Rann" specifies the device type(s) for selection. These must be specified exactly as one or more of the following: RA60, RA70, RA80, RA81, RA82, RA90. Wild cards may not be used when selecting a device type. However, one or more device types may be specified, and they must be separated by commas. For example:

```
DUA*,RA80,RA82          DJ*,RA60          D*,RA70,RA80,RA81
```

17.2.4 Event Codes

Event(s):

This prompt allows you to sort and tailor the selection of error log events displayed according to a specific MSCP status/event code or a list of codes. Wild card characters (* or %) may be specified in each MSCP code entered. A maximum of 100 MSCP codes may be specified, separated by commas. Entries are specified in hexadecimal, and leading zeros may be omitted. A default carriage return <cr> causes all entries with any MSCP event to be selected. For example:

1AB,*8	Limits entries selected to those containing MSCP event code 1AB or any event code ending in 8.
%%6B	Selects event codes four digits long with the last two digits of 6B.
*6B	Selects event codes of any length with the last two digits of 6B.

17.2.5 After

Starting date:

A default carriage return <cr> causes the program to start selection with the first entry in the binary input file. Entering a date/time will cause the program to select entries with a date/time equal to or greater than the value entered. The format for the response is:

```
dd-mmm-yyyy hh:mm:ss:cc
```

A date or time may be selected independently. If you specify both the date and time, the intervening space is required. You can omit any of the trailing fields in the date or time parameter.

17.2.6 Before

Ending date:

A default carriage return <cr> causes the program to make selections up to and including the last entry in the binary input file. Entering a date/time will cause the program to select entries with date/time less than or equal to the value entered. The format for the response is:

```
dd-mmm-yyyy hh:mm:ss:cc
```

The date or time may be selected independently. If you specify both the date and time, the intervening space is required. You can omit any of the trailing fields in the date or time parameter.

17.2.7 Report

Report type:

This prompt allows you to specify the format and style of the output report generated. The report type selected will also dictate the amount of information provided in conjunction with the parameters previously provided. There are five different report formats that may be selected.

```
          Physical     Geographic     Summary     Verbose     Time
```

You may enter the first character or the entire word to select the desired report. A carriage return <cr> defaults to P (physical report). Details and example of each are described in the following sections.

17.2.7.1 Physical Report (P)

This is the default report type. An example follows. The error log entries selected and reported are displayed in the same order in which they appeared in the binary input file. The information provided includes:

Device name (including controller path if applicable)
Drive type
Drive LED code (if applicable)
MSCP status/event code
Block number (LBN or RBN)
Translation of block number into:

Cyl – Physical cylinder
Hd – Head
S – Physical sector from index

Volume serial number
Date/Time of entry

PHYSICAL report example:

```
                               -<* DSAERR V3.01 *>-

               Drv  Drv  MSCP  Block
Device Name    Type Led  Event Number     Cyl Hd  S  Vol-sn  yy-mm-dd hh:mm:ss:cc
------------   ---- ---  ----  -------   ---- -- --  ------  --------------------

HSC007$DUA66   RA82      01AB        0      0  0  0  634003  86/04/08 13:37:56.68
HSC007$DUA66   RA82      01AB        0      0  0  0  634003  86/04/08 13:37:56.79
HSC007$DUA66   RA82      01AB        0      0  0  0  634003  86/04/08 13:37:56.91
HSC007$DUA66   RA82      006B   608485    711 10 34  634003  86/04/08 13:37:56.91
DUA3           RA80      00EB        0      0  0  0       0  86/04/08 14:00:24.89
DUA3           RA80 07   00EB        0      0  0  0       0  86/04/08 14:00:24.89
DUA3           RA80 07   002B        0      0  0  0       0  86/04/08 14:00:34.11
HSC007$DUA66   RA82 4F   00EB        0      0  0  0  634003  86/04/09 16:48:17.97
HSC007$DUA66   RA82 4F   00EB        0      0  0  0  634003  86/04/09 16:48:23.41
HSC007$DUA66   RA82      006B    87036    101 11 34  634003  86/04/09 16:48:28.89
HSC007$DUA66   RA82      0045  1217578   1424  1 15  634003  86/04/08 14:48:44.25
HSC007$DUA66   RA82 C0   00EB        0      0  0  0  634003  86/04/08 16:17:54.27
HSC007$DUA66   RA82      002B        0      0  0  0  634003  86/04/09 09:32:30.25
HSC007$DUA66   RA82 4D   00EB        0      0  0  0  634003  86/04/09 09:32:51.77
HSC007$DUA66   RA82      002B        0      0  0  0  634003  86/04/09 09:33:13.05
HSC007$DUA66   RA82 4D   00EB        0      0  0  0  634003  86/04/09 09:33:13.17
HSC007$DUA66   RA82      006B   608483    711 10 32  634003  86/04/09 10:14:19.34
HSC007$DUA66   RA82      006B   608483    711 10 32  634003  86/04/09 10:14:19.50
HSC007$DUA66   RA82      0045  1216666   1423  0  1  634003  86/04/09 10:14:36.89
HSC007$DUA66   RA82      0045  1218490   1425  2 29  634003  86/04/09 10:14:37.17
HSC007$DUA66   RA82      0045  1216665   1423  0  0  634003  86/04/09 10:14:37.43
HSC007$DUA66   RA82      0045  1216667   1423  0  2  634003  86/04/09 10:15:12.53
HSC007$DUA66   RA82      0094      342      0  6 26  634003  86/04/09 10:15:12.65
HSC007$DUA66   RA82      002B        0      0  0  0  634003  86/04/09 15:28:00.33
HSC007$DUA66   RA82 26   00EB        0      0  0  0  634003  86/04/09 15:28:00.47
HSC007$DUA66   RA82 26   00EB        0      0  0  0  634003  86/04/09 15:28:00.50
DUA2           RA81      0048    35896     50  3 33   21198  86/08/22 13:24:03.38
DUA2           RA81      0048    35896     50  3 33   21198  86/08/22 13:24:04.18
DUA2           RA81      0048    35896     50  3 33   21198  86/08/22 13:24:04.92
DUA2           RA81      0014    35896     50  3 33   21198  86/08/22 13:24:05.94
DUA2           RA81      00E8    28047     39  3 38   21198  86/08/22 13:24:06.90
DUA2           RA81      00E8    28047     39  3 38   21198  86/08/22 13:24:07.33
DUA2           RA81      00E8    28047     39  3 38   21198  86/08/22 13:24:07.79
```

17.2.7.2 Geographic Report (G)

This report provides the same information as the physical (P) report but sorts the entries according to geographic cylinder, head, and sector. The sorting priorities are:

1. Device name
2. Cylinder
3. Head
4. Sector

An example follows. The information provided includes:

Device name (including controller path if applicable)
Drive type
Drive LED code (if applicable)
MSCP status/event code
Block number (LBN or RBN)
Translation of block number into:

Cyl – Physical cylinder
Hd – Head
S – Physical sector from index

Volume serial number
Date/Time of entry

GEOGRAPHIC report sample:

```
                                 Geography V3.0

              Drv  Drv  MSCP  Block
Device Name   Type Led  Event Number  Cyl  Hd  S  Vol-sn  yy-mm-dd hh:mm:ss:cc
-----------   ---- ---  ----- ------- ---- -- --  ------  --------------------

BRIVAX$DUA0   RA81      010B    25237   35  4 47      10  86/10/03 11:41:37.20
BRIVAX$DUA0   RA81      010B   187530  262  9 25      10  86/09/29 16:30:49.15
BRIVAX$DUA0   RA81      010B   189322  265  2 38      10  86/09/25 19:28:31.42
BRIVAX$DUA0   RA81      010B   240521  336 12 17      10  86/09/25 19:29:45.03
BRIVAX$DUA3   RA80      0168   186137  428 12 13      25  86/10/06 13:50:12.04
BRIVAX$DUA3   RA80      0034   186137  428 12 13      25  86/10/06 13:50:12.74
BRIVAX$DUA3   RA80      0128   186137  428 12 13      25  86/10/06 13:50:12.08
BRIVAX$DUA3   RA80      0148   186579  429 12 21      25  86/10/06 13:50:16.57
BRIVAX$DUA3   RA80      0034   186579  429 12 21      25  86/10/06 13:50:17.21
BRIVAX$DUA3   RA80      0128   186579  429 12 21      25  86/10/06 13:50:16.53
BRIVAX$DUA3   RA80      0128   195246  449 12  8      25  86/10/06 14:47:24.40
BRIVAX$DUA3   RA80      0034   195246  449 12  8      25  86/10/06 14:47:25.13
BRIVAX$DUA3   RA80      0128   214791  494 12 23      25  86/10/06 13:49:04.47
BRIVAX$DUA3   RA80      00E8   214791  494 12 23      25  86/10/06 13:49:04.91
BRIVAX$DUA3   RA80      0014   214791  494 12 23      25  86/10/06 13:49:05.53
BRIVAX$DUA3   RA80      0128     6928  494 12 31      25  86/10/06 13:49:05.36
BRIVAX$DUA3   RA80      0014   216954  499 12 16      25  86/10/06 13:49:24.70
BRIVAX$DUA3   RA80      00E8   216954  499 12 16      25  86/10/06 13:49:24.08
BRIVAX$DUA3   RA80      0128   216954  499 12 16      25  86/10/06 13:49:23.63
BRIVAX$DUA3   RA80      0148   216955  499 12 17      25  86/10/06 13:49:23.65
BRIVAX$DUA3   RA80      00E8   218704  503 12 30      25  86/10/06 13:49:25.24
BRIVAX$DUA3   RA80      0128   218704  503 12 30      25  86/10/06 13:49:24.82
BRIVAX$DUA3   RA80      0128   218704  503 12 30      25  86/10/06 13:49:24.77
HSC007$DUA66  RA82 4D   00EB        0    0  0  0   63400  86/04/09 09:33:13.17
HSC007$DUA66  RA82 F1   00EB        0    0  0  0   63400  86/04/08 17:19:01.01
HSC007$DUA66  RA82 26   00EB        0    0  0  0   63400  86/04/09 15:28:00.45
HSC007$DUA66  RA82 F1   00EB        0    0  0  0   63400  86/04/08 17:19:57.67
HSC007$DUA66  RA82      006B       96    0  1 53   63400  86/04/08 14:48:27.02
HSC007$DUA66  RA82      0094       96    0  1 53   63400  86/04/08 14:48:44.35
HSC007$DUA66  RA82      006B      342    0  6 26   63400  86/04/09 10:14:36.72
HSC007$DUA66  RA82      0094      342    0  6 26   63400  86/04/09 10:15:12.65
HSC007$DUA66  RA82      0043    16586   19  5 10   63400  86/04/09 15:28:00.53
HSC007$DUA66  RA82      006B    87036  101 11 34   63400  86/04/09 16:48:28.89
HSC007$DUA66  RA82      006B   608483  711 10 32   63400  86/04/09 10:14:19.50
HSC007$DUA66  RA82      006B   608483  711 10 32   63400  86/04/09 10:14:19.34
HSC007$DUA66  RA82      006B   608485  711 10 34   63400  86/04/08 13:37:56.91
HSC007$DUA66  RA82      0045  1216665 1423  0  0   63400  86/04/09 10:14:54.61
HSC007$DUA66  RA82      0045  1216665 1423  0  0   63400  86/04/09 10:14:37.43
HSC007$DUA66  RA82      0045  1216666 1423  0  1   63400  86/04/09 10:14:36.89
HSC007$DUA66  RA82      0045  1216667 1423  0  2   63400  86/04/09 10:14:36.68
HSC007$DUA66  RA82      0045  1216667 1423  0  2   63400  86/04/09 10:15:12.53
```

17.2.7.3 Summary Report (S)

This report provides a summary map of all selected entries in the error log. The map organizes the entries according to the disk R/W heads associated with the translation of each of the block numbers if applicable for a R/W transfer error log entry. This gives a geographic view of the displacement of R/W–related entries across the disk media.

Entries associated with non–transfer errors will have zeros for block numbers. Due to the nature of the program, these entries are tallied and entered in the coordinate associated with cylinder 0 and head 0. Do not to let these lead you to believe there is a head 0 or cylinder 0 problem. One solution would be to select MSCP status/event codes associated with R/W data transfer errors, such as:

8	Forced error
48	Invalid (corrupted) header
68	Data sync timeout
88	Correctable error in ECC field
E8	Uncorrectable ECC error
128	Two–symbol ECC error
148	Three–symbol ECC error
168	Four–symbol ECC error
188	Five–symbol ECC error
1A8	Six–symbol ECC error
1C8	Seven–symbol ECC error
1E8	Eight–symbol ECC error

A simpler method would be to use the wild card feature when responding to the status/event code prompt of the program and select any disk entry with status/event codes ending in 8.

```
HEX Event Code(s)   (nnnn) '1AB,E8,%%6B,*4'   [ALL] ?  *8 <cr>
```

Three (3) examples follow. The summary associated with disk DUA3 shows all of the logged entries as being associated with head number 12 (after automatic translation). The head may be defective or the media area associated with head 12 may be defective.

The summary associated with DUA2 shows a distribution of errors across many different R/W heads. If the errors occurred in a relatively short amount of time, a read/write data path problem may exist with disk electronics, the SDI path, or the SDI electronics within the controller. A number of other possibilities also exist here. Examination of the specific event codes in a PHYSICAL report may provide more information to help isolate the problem.

The summary for disk HSC007$DUA66 is an example where all the errors are logged at the coordinate for head 0, cylinder 0. These are likely not R/W related but instead SDI related with no LBNs associated with the errors. You could confirm this by obtaining a PHYSICAL report for this drive.

SUMMARY report sample:

```
-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
Volume Ser#:     25205
                         ( Device Name: DUA3           )

        0    1    2    3    4    5    6    7    8    9   10   11   12   13   14
CYL#   ---  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---

 428  -    -    -    -    -    -    -    -    -    -    -    -    -  3 -    -
 429  -    -    -    -    -    -    -    -    -    -    -    -    -  3 -    -
 449  -    -    -    -    -    -    -    -    -    -    -    -    -  2 -    -
 494  -    -    -    -    -    -    -    -    -    -    -    -    -  4 -    -
 499  -    -    -    -    -    -    -    -    -    -    -    -    -  4 -    -
 503  -    -    -    -    -    -    -    -    -    -    -    -    -  5 -    -
 532  -    -    -    -    -    -    -    -    -    -    -    -    -  4 -    -
 541  -    -    -    -    -    -    -    -    -    -    -    -    -  6 -    -



-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
Volume Ser#:     21198
                         ( Device Name: DUA2           )

        0    1    2    3    4    5    6    7    8    9   10   11   12   13   14
CYL#   ---  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---
   0  -  3 -    -    -    -    -    -    -    -    -    -    -    -    -    -
  25  -    -    -    -    -    -    -    -    -    -    -    -    -    -  4 -
  28  -    -    -    -    -    -    -  4 -    -    -    -    -    -    -    -
  30  -    -    -    -    -    -    -  5 -    -    -    -    -    -    -    -
  31  -    -    -    -    -    -    -  5 -    -    -    -    -    -  3 -    -
  37  -    -  5 -    -  5 -    -    -    -    -    -    -    -    -    -    -
  38  -    -    -  4 -    -    -    -    -    -    -    -    -    -    -  5 -
  39  -    -  4 -    -  4 -    -    -    -    -    -    -  7 -    -    -  4 -
  40  -    -    -  4 -    -    -    -  8 -    -    -    -  3 -    -  4 -    -
  41  -    -    -  4 -    -    -    -    -    -    -    -    -    -    -    -
  42  -    -    -  4 -    -    -    -    -    -  4 -  9 -    -    -    -    -
  43  -    -    -  4 -    -    -    -    -    -  4 -    -    -    -    -    -
  44  -    -    -  9 -    -  4 -    -    -    -  4 -    -    -    -    -    -
  45  -    -    -    -    -  4 -    -    -    -    -    -    -    -    -    -
  47  -    -    -  8 -    -    -    -    -    -    -    -    -    -    -    -
  48  -    -    -    -    -    -    -    -    -    -    -    -    -  5 -    -
  49  -    -    -  5 -    -  4 -    -    -    -    -    -    -    -  5 -    -
  50  -    -    -    -  4 -  8 -    -    -    -    -  2 -    -    -    -    -
  51  -    -    -    -  4 -  4 -    -    -    -    -    -    -    -    -    -
  52  -    -    -    -    -    -    -    -    -    -    -    -    -    -  5 -
  54  -    -    -    -    -    -    -    -    -  5 -    -    -    -    -    -



-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
Volume Ser#:  63400354
                         ( Device Name: HSC007$DUA66   )

        0    1    2    3    4    5    6    7    8    9   10   11   12   13   14
CYL#   ---  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---

   0 - 24 -    -    -    -    -    -    -    -    -    -    -    -    -    -
```

17.2.7.4 Verbose Report (V)

The verbose report displays all the MSCP information for each of the selected error log events. Unlike the standard VMS error log report that may display one to two pages for each entry, DSAERR condenses the information into about a half a page for each entry. This format is intended for experienced users who are more familiar with DSA/MSCP and need these details.

Examples for a disk transfer error log entry, an SDI error log entry, and a bad block replacement entry follow.

VERBOSE report sample:

```
-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-==
     DISK TRANSFER ERROR logged at  8-APR-1986 13:37:56.91 on SID 01380A4F

B.ENTRY_CLASS     : 100.                  B.RECOVERY_LEVEL  : 7.
B.ENTRY_TYPE      : 0.                    B.RECOVERY_COUNT  : 0.
W.ERR_SEQ         : 30.                   L.DRV_SER         : 264.
W.SEQ_NUM         : 433.                  B.UNIT_SVR        : 01
B.DD_CLASS        : 1.                    B.UNIT_HVR        : 0F
B.DD_TYPE         : 30.  RA82             B.UNIT_TYPE       : 11.
B.DD_NUM          : 66.                   B.UNIT_CLASS      : 2.
B.DD_NAME         : HSC007$DUA            L.VOL_SER         : 63400354.
W.MESSAGE_TYPE    : 0001                  L.BLOCK_NUM       : 608485.  LBN
L.CMD_REF         : 130E0006              W.ORIG_ERR_FLAGS  : 014000
W.UNIT            : 66.                   W.RECOVERY_FLAGS  : 000002
W.SEQ_NUM         : 01B1                  B.LVL_A_RETRY_CNT : 3.
B.FORMAT          : 02                    B.LVL_B_RETRY_CNT : 0.
B.FLAGS           : 00                    W.BUFFER_ADDR     : 141706
W.EVENT           : 006B                  B.SOURCE_REQ      : 5
Q.CNT_ID          : 010100000000F807      B.DETECT_REQ      : 5
B.CNT_SVR         : 02
B.CNT_HVR         : 00
W.MULTI_UNIT      : 0050
```

```
-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
        SDI MESSAGE logged at 25-SEP-1987 14:31:28.60 on SID 01380A4F

B.ENTRY_CLASS     : 100.                  B.RECOVERY_LEVEL  : 0
B.ENTRY_TYPE      : 0.                    B.RECOVERY_COUNT  : 0.
W.ERR_SEQ         : 13.                   L.DRV_SER         : 173816.
W.SEQ_NUM         : 12.                   B.UNIT_SVR        : 08
B.DD_CLASS        : 1.                    B.UNIT_HVR        : 08
B.DD_TYPE         : 21.  RA81             B.UNIT_TYPE       : 5.
B.DD_NUM          : 116.                  B.UNIT_CLASS      : 2.
B.DD_NAME         : HSC015$DUA            L.VOL_SER         : 140582.
W.MESSAGE_TYPE    : 0001                  L.BLOCK_NUM       : 0.  LBN
L.CMD_REF         : 000000                L.SDI_INFO        : 0080001B
W.UNIT            : 116.                  B.SDI_RETRY_CNT   : 0.
W.SEQ_NUM         : 000C                  B.PRV_CMD         : 8E
B.FORMAT          : 03                    B.SDI_STATUS      : 00
B.FLAGS           : 40                    W.CURRENT_CYL     : 627.
W.EVENT           : 00EB                  B.CURRENT_GROUP   : 3.
Q.CNT_ID          : 010100000000F807      B.DRIVE_LED_CODE  : F1
B.CNT_SVR         : 02                    B.DRV_FAULT_CODE  : 1A
B.CNT_HVR         : 00                    B.SDI_S_REQ       : 3.
W.MULTI_UNIT      : 0033                  B.SDI_D_REQ       : 3.
```

```
-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
    REPLACEMENT MESSAGE logged at 22-AUG-1986 14:30:32.90 on SID 03003A00

B.ENTRY_CLASS     : 100.                  B.RECOVERY_LEVEL  : 0
B.ENTRY_TYPE      : 0.                    B.RECOVERY_COUNT  : 192.
W.ERR_SEQ         : 3098.                 L.DRV_SER         : 76478.
W.SEQ_NUM         : 65535.                B.UNIT_SVR        : 07
B.DD_CLASS        : 1.                    B.UNIT_HVR        : 06
B.DD_TYPE         : 21.  RA81             B.UNIT_TYPE       : 5.
B.DD_NUM          : 2.                    B.UNIT_CLASS      : 2.
B.DD_NAME         : DUA                   L.VOL_SER         : 21198.
W.MESSAGE_TYPE    : 0001                  L.BLOCK_NUM       : 30837.  LBN
L.CMD_REF         : E2510006              B.BBR_FLAGS       : C000
W.UNIT            : 2.                    L.BAD_LBN         : 30837.  LBN
W.SEQ_NUM         : FFFF                  L.OLD_RBN         : 0.
B.FORMAT          : 09                    W.BBR_CAUSE       : 00E8
B.FLAGS           : A0                    L.NEW_RBN         : 604.
W.EVENT           : 0014
Q.CNT_ID          : 010600815ACA44
B.CNT_SVR         : 04
B.CNT_HVR         : 00
W.MULTI_UNIT      : 0002
```

17.2.7.5 Time Report (T)

The time report is similar to the physical report, but the entries are sorted by the date/time in which they occurred rather than the order in which they appear in the binary input file. The information provided includes:

Device name (including controller path if applicable)
Drive type
Drive LED code (if applicable)
MSCP status/event code
Block number (LBN or RBN)
Translation of block number into:

Cyl – Physical cylinder
Hd – Head
S – Physical sector from index

Volume serial number
Date/Time of entry

TIME report sample:

```
                                 -<* DSAERR V3.01 *>-

               Drv  Drv  MSCP  Block
Device Name    Type Led  Event Number    Cyl Hd  S  Vol-sn  yy-mm-dd hh:mm:ss:cc
------------   ---- ---  ----  -------  ---- --  --  ------  -------------------

HSC007$DUA66   RA82      01AB        0     0  0  0  634003  86/04/08 13:37:56.68
HSC007$DUA66   RA82      01AB        0     0  0  0  634003  86/04/08 13:37:56.79
HSC007$DUA66   RA82      01AB        0     0  0  0  634003  86/04/08 13:37:56.91
HSC007$DUA66   RA82      006B   608485   711 10 34  634003  86/04/08 13:37:56.91
DUA3           RA80      00EB        0     0  0  0       0  86/04/08 14:00:24.89
DUA3           RA80 07   00EB        0     0  0  0       0  86/04/08 14:00:24.89
DUA3           RA80 07   002B        0     0  0  0       0  86/04/08 14:00:34.11
HSC007$DUA66   RA82      006B       96     0  1 53  634003  86/04/08 14:48:27.02
HSC007$DUA66   RA82      0045  1217578  1424  1 15  634003  86/04/08 14:48:44.25
HSC007$DUA66   RA82      0094       96     0  1 53  634003  86/04/08 14:48:44.35
HSC007$DUA66   RA82 C0   00EB        0     0  0  0  634003  86/04/08 16:17:54.27
HSC007$DUA66   RA82      002B        0     0  0  0  634003  86/04/09 09:32:30.25
HSC007$DUA66   RA82 4D   00EB        0     0  0  0  634003  86/04/09 09:32:30.37
HSC007$DUA66   RA82      002B        0     0  0  0  634003  86/04/09 09:32:51.65
HSC007$DUA66   RA82 4D   00EB        0     0  0  0  634003  86/04/09 09:32:51.77
HSC007$DUA66   RA82      002B        0     0  0  0  634003  86/04/09 09:33:13.05
HSC007$DUA66   RA82 4D   00EB        0     0  0  0  634003  86/04/09 09:33:13.17
HSC007$DUA66   RA82      006B   608483   711 10 32  634003  86/04/09 10:14:19.34
HSC007$DUA66   RA82      006B   608483   711 10 32  634003  86/04/09 10:14:19.50
HSC007$DUA66   RA82      0045  1216667  1423  0  2  634003  86/04/09 10:14:36.68
HSC007$DUA66   RA82      006B      342     0  6 26  634003  86/04/09 10:14:36.72
HSC007$DUA66   RA82      0045  1216666  1423  0  1  634003  86/04/09 10:14:36.89
HSC007$DUA66   RA82      0045  1218490  1425  2 29  634003  86/04/09 10:14:37.17
HSC007$DUA66   RA82      0045  1216665  1423  0  0  634003  86/04/09 10:14:37.43
HSC007$DUA66   RA82      0045  1216665  1423  0  0  634003  86/04/09 10:14:54.61
HSC007$DUA66   RA82      0045  1218489  1425  2 28  634003  86/04/09 10:15:11.95
HSC007$DUA66   RA82      0045  1216667  1423  0  2  634003  86/04/09 10:15:12.53
HSC007$DUA66   RA82      0094      342     0  6 26  634003  86/04/09 10:15:12.65
HSC007$DUA66   RA82      002B        0     0  0  0  634003  86/04/09 15:28:00.33
HSC007$DUA66   RA82 26   00EB        0     0  0  0  634003  86/04/09 15:28:00.47
HSC007$DUA66   RA82 26   00EB        0     0  0  0  634003  86/04/09 15:28:00.50
HSC007$DUA66   RA82 4F   00EB        0     0  0  0  634003  86/04/09 16:48:17.97
HSC007$DUA66   RA82 4F   00EB        0     0  0  0  634003  86/04/09 16:48:23.41
HSC007$DUA66   RA82      006B    87036   101 11 34  634003  86/04/09 16:48:28.89
DUA2           RA81      0048    35896    50  3 33   21198  86/08/22 13:24:03.38
DUA2           RA81      0048    35896    50  3 33   21198  86/08/22 13:24:04.18
DUA2           RA81      0048    35896    50  3 33   21198  86/08/22 13:24:04.92
DUA2           RA81      0014    35896    50  3 33   21198  86/08/22 13:24:05.94
DUA2           RA81      00E8    28047    39  3 38   21198  86/08/22 13:24:06.90
DUA2           RA81      00E8    28047    39  3 38   21198  86/08/22 13:24:07.33
DUA2           RA81      00E8    28047    39  3 38   21198  86/08/22 13:24:07.79
DUA2           RA81      0014    28047    39  3 38   21198  86/08/22 13:24:08.48
DUA2           RA81      00E8    31848    44  8 32   21198  86/08/22 13:24:37.66
DUA2           RA81      00E8    31848    44  8 32   21198  86/08/22 13:24:37.94
DUA2           RA81      00E8    31848    44  8 32   21198  86/08/22 13:24:39.14
DUA2           RA81      0014    31848    44  8 32   21198  86/08/22 13:24:39.82
```

17.2.8 Using the Selection Process

The following pages provide some examples of the selection parameters previously described and the resulting displays.

```
Input file:      errorlog.eng <cr>
Output file:     <cr>
Device(s):       DU*,RA80,RA81 <cr>
Event(s):        *8,14 <cr>
After:           <cr>
Before:          <cr>
Report:          P <cr>
```

```
                              -<* DSAERR V3.01 *>-

                 Drv  Drv  MSCP  Block
Device Name      Type Led  Event Number    Cyl Hd  S  Vol-sn  yy-mm-dd hh:mm:ss:cc
------------     ---- ---  ----  ------- ---- -- --  ------  --------------------
DUA2             RA81      0048    35896   50  3 33   21198  86/08/22 13:24:03.38
DUA2             RA81      0048    35896   50  3 33   21198  86/08/22 13:24:04.18
DUA2             RA81      0048    35896   50  3 33   21198  86/08/22 13:24:04.92
DUA2             RA81      0014    35896   50  3 33   21198  86/08/22 13:24:05.94
DUA2             RA81      00E8    28047   39  3 38   21198  86/08/22 13:24:06.90
DUA2             RA81      00E8    28047   39  3 38   21198  86/08/22 13:24:07.33
DUA2             RA81      00E8    28047   39  3 38   21198  86/08/22 13:24:07.79
DUA2             RA81      0014    28047   39  3 38   21198  86/08/22 13:24:08.48
DUA2             RA81      00E8    31848   44  8 32   21198  86/08/22 13:24:37.66
DUA2             RA81      00E8    31848   44  8 32   21198  86/08/22 13:24:37.94
DUA2             RA81      00E8    31848   44  8 32   21198  86/08/22 13:24:39.14
DUA2             RA81      0014    31848   44  8 32   21198  86/08/22 13:24:39.82
DUA2             RA81      01E8    22440   31  6 32   21198  86/08/22 13:24:41.22
BRIVAX$DUA3      RA80      0128   231274  532 12 14   25205  86/10/06 13:48:31.04
BRIVAX$DUA3      RA80      00E8   231274  532 12 14   25205  86/10/06 13:48:31.47
BRIVAX$DUA3      RA80      0128     7460  532 12 31   25205  86/10/06 13:48:31.91
BRIVAX$DUA3      RA80      0014   231274  532 12 14   25205  86/10/06 13:48:32.08
BRIVAX$DUA3      RA80      0128   214791  494 12 23   25205  86/10/06 13:49:04.47
BRIVAX$DUA3      RA80      00E8   214791  494 12 23   25205  86/10/06 13:49:04.91
```

```
Input file:     binary.dat <cr>
Output file:    my.out <cr>
Device(s):      DUA* <cr>
Event(s):       4% <cr>
After:          8-APR-1986 14:00:00 <cr>
Before:         11-APR-1986 <cr>
Report:         <cr>
```

```
                                      -<* DSAERR V3.01 *>-

              Drv  Drv  MSCP  Block
Device Name   Type Led  Event Number   Cyl Hd  S  Vol-sn  yy-mm-dd hh:mm:ss:cc
------------  ---- ---  ----  ------- ---- -- --  ------  --------------------
HSC007$DUA66  RA82      0045  1217578 1424  1 15  634003  86/04/08 14:48:44.25
HSC007$DUA66  RA82      0045  1216667 1423  0  2  634003  86/04/09 10:14:36.68
HSC007$DUA66  RA82      0045  1216666 1423  0  1  634003  86/04/09 10:14:36.89
HSC007$DUA66  RA82      0045  1218490 1425  2 29  634003  86/04/09 10:14:37.17
HSC007$DUA66  RA82      0045  1216665 1423  0  0  634003  86/04/09 10:14:37.43
HSC007$DUA66  RA82      0045  1216665 1423  0  0  634003  86/04/09 10:14:54.61
HSC007$DUA66  RA82      0045  1218489 1425  2 28  634003  86/04/09 10:15:11.95
HSC007$DUA66  RA82      0045  1216667 1423  0  2  634003  86/04/09 10:15:12.53
HSC007$DUA66  RA82      0043    16586   19  5 10  634003  86/04/09 15:28:00.53
MUFFIN$DUA0   RA70      004B        0    0  0  0       0  86/04/10 08:10:03.65
MUFFIN$DUA0   RA70      004B        0    0  0  0       0  86/04/10 08:10:03.66
MUFFIN$DUA0   RA70      004B        0    0  0  0       0  86/04/10 08:10:03.66
```

```
       Input file:     <cr>
       Output file:    <cr>
       Device(s):      DUA*,RA80,RA81,RA82 <cr>
       Event(s):       <cr>
       After:          <cr>
       Before:         <cr>
       Report:         G <cr>

                             -<* DSAERR V3.01 *>-

              Drv  Drv  MSCP  Block
Device Name   Type Led  Event Number   Cyl Hd  S  Vol-sn  yy-mm-dd hh:mm:ss:cc
-----------   ---- ---  ----- -------- ---- -- --  ------  --------------------

BRIVAX$DUA0   RA81        010B    25237   35  4 47      10  86/10/03 11:41:37.20
BRIVAX$DUA0   RA81        010B   187530  262  9 25      10  86/09/29 16:30:49.15
BRIVAX$DUA0   RA81        010B   189322  265  2 38      10  86/09/25 19:28:31.42
BRIVAX$DUA0   RA81        010B   240521  336 12 17      10  86/09/25 19:29:45.03
BRIVAX$DUA3   RA80        0168   186137  428 12 13      25  86/10/06 13:50:12.04
BRIVAX$DUA3   RA80        0034   186137  428 12 13      25  86/10/06 13:50:12.74
BRIVAX$DUA3   RA80        0128   186137  428 12 13      25  86/10/06 13:50:12.08
BRIVAX$DUA3   RA80        0148   186579  429 12 21      25  86/10/06 13:50:16.57
BRIVAX$DUA3   RA80        0034   186579  429 12 21      25  86/10/06 13:50:17.21
BRIVAX$DUA3   RA80        0128   186579  429 12 21      25  86/10/06 13:50:16.53
BRIVAX$DUA3   RA80        0128   195246  449 12  8      25  86/10/06 14:47:24.40
BRIVAX$DUA3   RA80        0034   195246  449 12  8      25  86/10/06 14:47:25.13
BRIVAX$DUA3   RA80        0128   214791  494 12 23      25  86/10/06 13:49:04.47
BRIVAX$DUA3   RA80        00E8   214791  494 12 23      25  86/10/06 13:49:04.91
BRIVAX$DUA3   RA80        0014   214791  494 12 23      25  86/10/06 13:49:05.53
BRIVAX$DUA3   RA80        0128     6928  494 12 31      25  86/10/06 13:49:05.36
BRIVAX$DUA3   RA80        0014   216954  499 12 16      25  86/10/06 13:49:24.70
BRIVAX$DUA3   RA80        00E8   216954  499 12 16      25  86/10/06 13:49:24.08
BRIVAX$DUA3   RA80        0128   216954  499 12 16      25  86/10/06 13:49:23.63
BRIVAX$DUA3   RA80        0148   216955  499 12 17      25  86/10/06 13:49:23.65
BRIVAX$DUA3   RA80        00E8   218704  503 12 30      25  86/10/06 13:49:25.24
BRIVAX$DUA3   RA80        0128   218704  503 12 30      25  86/10/06 13:49:24.82
BRIVAX$DUA3   RA80        0128   218704  503 12 30      25  86/10/06 13:49:24.77
HSC007$DUA66  RA82  4D  00EB        0    0  0  0   63400  86/04/09 09:33:13.17
HSC007$DUA66  RA82  F1  00EB        0    0  0  0   63400  86/04/08 17:19:01.01
HSC007$DUA66  RA82  26  00EB        0    0  0  0   63400  86/04/09 15:28:00.45
HSC007$DUA66  RA82  F1  00EB        0    0  0  0   63400  86/04/08 17:19:57.67
HSC007$DUA66  RA82      006B       96    0  1 53   63400  86/04/08 14:48:27.02
HSC007$DUA66  RA82      0094       96    0  1 53   63400  86/04/08 14:48:44.35
HSC007$DUA66  RA82      006B      342    0  6 26   63400  86/04/09 10:14:36.72
HSC007$DUA66  RA82      0094      342    0  6 26   63400  86/04/09 10:15:12.65
HSC007$DUA66  RA82      0043    16586   19  5 10   63400  86/04/09 15:28:00.53
HSC007$DUA66  RA82      006B    87036  101 11 34   63400  86/04/09 16:48:28.89
HSC007$DUA66  RA82      006B   608483  711 10 32   63400  86/04/09 10:14:19.50
HSC007$DUA66  RA82      006B   608483  711 10 32   63400  86/04/09 10:14:19.34
HSC007$DUA66  RA82      006B   608485  711 10 34   63400  86/04/08 13:37:56.91
HSC007$DUA66  RA82      0045  1216665 1423  0  0   63400  86/04/09 10:14:54.61
HSC007$DUA66  RA82      0045  1216665 1423  0  0   63400  86/04/09 10:14:37.43
HSC007$DUA66  RA82      0045  1216666 1423  0  1   63400  86/04/09 10:14:36.89
```

```
       Input file:     <cr>
       Output file:    <cr>
       Device(s):      DUA9,RA81
       Event(s):       *8 <cr>
       After:          <cr>
       Before:         <cr>
       Report:         S <cr>

-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
Volume Ser#:     23448

                        ( Device Name: DUA9               )

        0    1    2    3    4    5    6    7    8    9   10   11   12   13   14
CYL#   ---  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---
  25  -    -    -    -    -    -    -    -    -    -    -    -    -    -  4 -
  28  -    -    -    -    -    -    -  4 -    -    -    -    -    -    -    -
  30  -    -    -    -    -    -    -  5 -    -    -    -    -    -    -    -
  31  -    -    -    -    -    -    -  5 -    -    -    -    -    -  3 -    -
  37  -    -  5 -    -  5 -    -    -    -    -    -    -    -    -    -    -
  38  -    -    -  4 -    -    -    -    -    -    -    -    -    -    -  5 -
 139  -    -  4 -    -  4 -    -    -    -    -    -    -  7 -    -    -  4 -
 140  -  2 -    -  4 -    -    -    -  8 -    -    -    -  3 -    -  4 -    -
 141  -    -  3 -  4 -    -    -    -    -    -    -    -    -    -    -    -
 142  -    -    -  4 -    -  3 -  4 -    -    -  4 -  9 -    -    -    -    -
 143  -    -    -  4 -    -    -    -    -    -  4 -    -    -    -    -    -
 144  -    -    -  9 -    -  4 -    -    -    -  4 -    -    -    -    -    -
 145  -    -    -    -    -  4 -    -    -    -    -    -    -    -    -    -
 236  -    -    -  8 -    -    -    -    -    -    -    -    -    -    -    -
 248  -    -    -    -    -    -    -    -    -    -    -    -    -  5 -    -
 249  -    -    -  5 -    -  4 -    -    -    -    -    -    -    -  5 -    -
 254  -    -    -    -    -    -    -    -    -  5 -    -    -    -    -    -
```

```
Input file:      binary.dat <cr>
Output file:     LED_CODES.OUT <cr>
Device(s):       DUA*
Event(s):        EB <cr>
After:           <cr>
Before:          <cr>
Report:          <cr>
```

```
                              -<* DSAERR V3.01 *>-

               Drv  Drv  MSCP  Block
Device Name    Type Led  Event Number    Cyl Hd  S  Vol-sn  yy-mm-dd hh:mm:ss:cc
------------   ---- ---  ----  ------- ---- -- --  ------  --------------------
DUA3           RA80  07  00EB        0    0  0  0       0  86/04/08 14:00:24.89
DUA3           RA80  07  002B        0    0  0  0       0  86/04/08 14:00:34.11
HSC007$DUA66   RA82  4F  00EB        0    0  0  0  634003  86/04/09 16:48:17.97
HSC007$DUA66   RA82  4F  00EB        0    0  0  0  634003  86/04/09 16:48:23.41
HSC007$DUA66   RA82  C0  00EB        0    0  0  0  634003  86/04/08 16:17:54.27
HSC007$DUA66   RA82  4D  00EB        0    0  0  0  634003  86/04/09 09:32:30.37
HSC007$DUA66   RA82  4D  00EB        0    0  0  0  634003  86/04/09 09:32:51.77
HSC007$DUA66   RA82  4D  00EB        0    0  0  0  634003  86/04/09 09:33:13.17
HSC007$DUA66   RA82  26  00EB        0    0  0  0  634003  86/04/09 15:28:00.45
HSC007$DUA66   RA82  26  00EB        0    0  0  0  634003  86/04/09 15:28:00.46
HSC007$DUA66   RA82  26  00EB        0    0  0  0  634003  86/04/09 15:28:00.47
HSC007$DUA66   RA82  26  00EB        0    0  0  0  634003  86/04/09 15:28:00.50
GRANPA$DUA124  RA81  F1  00EB        0    0  0  0       5  87/09/28 16:23:19.63
GRANPA$DUA124  RA81  F1  00EB        0    0  0  0       5  87/09/28 16:23:32.06
GRANPA$DUA124  RA81  4B  00EB        0    0  0  0       5  87/09/28 16:23:53.26
GRANPA$DUA124  RA81  4B  00EB        0    0  0  0       5  87/09/28 16:23:43.06
GRANPA$DUA124  RA81  F1  00EB        0    0  0  0       5  87/09/28 16:24:18.95
GRANPA$DUA124  RA81  4B  00EB        0    0  0  0       5  87/09/28 16:24:30.60
```

17.3 MANUAL TRANSLATION of DSA BLOCK NUMBERS

When prompted for the input file, you may enter TT <cr> to cause the program to enter a mode where manual block translation may be performed. Following is a summary of the prompts that will be displayed for manual translation to occur.

```
Drive Type:
```

Entering a question mark (?) will cause the program to display the drive types supported by this program. Otherwise, enter the desired type of drive associated with the blocks you wish to translate. Drives supported at the time this course was developed include RA60, RA70, RA80, RA81, RA82, and RA90. Enter Control_Z if you wish to exit the program.

```
Display drive parameter table (Y/N) [N] ?
```

A carriage return <cr> will cause the program to exit to the previous prompt. Following is a sample of the results provided if the user selects "Y" and RA81 was previously entered for the drive type:

```
              RA81 TOPOLOGY information  V3.0

   1258  :  Physical CYLINDERS.
     14  :  Physical HEADS/Cylinder.
     52  :  Physical BLOCKS/Track.
   1258  :  Logical CYLINDERS.
     14  :  Logical GROUPS/Cylinder.
      1  :  Logical TRACKS/Group.
     51  :  Logical BLOCKS/Track (LBNs).
      1  :  Replacement BLOCKS/Track (RBNs).
     14  :  Sectors of GROUP OFFSET.
      0  :  Starting cylinder in Host LBN Area.
   1252  :  Starting cylinder in FCT Area.
   1256  :  Starting cylinder in DBN Area.
 891072  :  LBNs in Host Area.
 893928  :  Total LBNs (Host + RCT).
   2912  :  Extended Blocks in FCT area (XBNs).
   2912  :  Diagnostic Blocks in DBN area (DBNs).
 915824  :  Total Physical Blocks (PBNs).
```

```
BLOCK Type:
```

A carriage return <cr> will cause the program to return to the "Drive Type" prompt. Otherwise, select the type of block (LBN, RBN, PBN, XBN or DBN) that you wish to translate. Enter a question mark (?) to obtain help.

```
Block(s):
```

A carriage return <cr> will cause the program to return to the previous prompt. Otherwise, enter the block number(s) you wish to have translated. You may enter any of the following:

1	A single block number	
2	Several block numbers separated by a comma	10,11
3	A range of numbers separated by a colon	111:120
4	A mix of 2 and 3 above	10,11,111:120,999

Selecting drive type RA81, block type LBN, and block numbers 10,11,111:120,999 would result in the display shown on the next page.

```
                 -<* DSAERR V3.01 *>-
                   TRK                  SEC
         LOG  LOG  IN   LOG   PHY  PHY FROM
   LBN#  CYL  GRP  GRP  SEC   CYL  HD   IDX  . . . . . Usage . . . .
-----------------------------------------------------------------------
     10    0    0    0   10     0    0   10  Host ......... PBN #     10
     11    0    0    0   11     0    0   11  Host ......... PBN #     11
    111    0    2    0    9     0    2   37  Host ......... PBN #    113
    112    0    2    0   10     0    2   38  Host ......... PBN #    114
    113    0    2    0   11     0    2   39  Host ......... PBN #    115
    114    0    2    0   12     0    2   40  Host ......... PBN #    116
    115    0    2    0   13     0    2   41  Host ......... PBN #    117
    116    0    2    0   14     0    2   42  Host ......... PBN #    118
    117    0    2    0   15     0    2   43  Host ......... PBN #    119
    118    0    2    0   16     0    2   44  Host ......... PBN #    120
    119    0    2    0   17     0    2   45  Host ......... PBN #    121
    120    0    2    0   18     0    2   46  Host ......... PBN #    122
    999    1    5    0   30     1    5   48  Host ......... PBN #   1018
```

CHAPTER 18

FAKDSK (ON HSC)

18.1 FAKDSK (on HSC V300/V350)

ON THE HSC CONSOLE:

HSC50>**RUN DD1:FAKDSK**	The cassette light will indicate the program is loading. No other response will occur on the HSC terminal. Two new devices will be available from VMS using device numbers 256 and 257.
Use Control Y or Control C	To abort and exit from FAKDSK operation.

ON THE VMS TERMINAL:

$ **EXCHANGE <cr>**	Program to exchange file information between VMS and RT–11 (in the HSC).
$ EXCHANGE> **MOUNT 1DUA256:**	This mounts cassette drive 0 in the HSC50 or floppy 0 in the HSC70.
$ EXCHANGE> **MOUNT 1DUA257:**	This mounts cassette drive 1 in the HSC50 or floppy 1 in the HSC70.
$ EXCHANGE> **DIR 1DUA257:**	To get a directory of files in drive 1
$ EXCHANGE> **COPY <cr>** from: **1DUA257:*.*** to: ***.*/LOG/TRANS=BLOCK**	To copy all files from drive 1 to your account. The /Log option lets you monitor the activity. The /Trans=block option assures image copy by block for compatibility when copying back to another HSC cassette of floppy.
$ EXCHANGE> **COPY <cr>** from: *.* to: **1DUA257:*.*/LOG/TRANS=BLOCK**	To copy all files from your account to drive 1. The /Log option lets you monitor the activity. The /Trans=block option assures image copy by block for compatibility between HSC (RT–11) and the VMS environment.
$ EXCHANGE> **INIT1DUA257:/VOLUME_FORMAT=RT11**	Initialize a new cassette (in the HSC, drive 1) for use. You must then mount it to transfer files.

NOTE:

You can specify /VERIFY when transferring files to the HSC media and verify for read back and compare checking, but this will add considerable time to the COPY operation.

When transferring lots of files to the HSC (write to HSC tape/floppy), it will have to rewind the tape after each transfer to update the directory. Thus, this particular operation takes longer.

CAUTION:

Aborting FAKDSK while any transfer operations (e.g., COPY) are in progress will crash the HSC.

Do Not Mix utilities between HSC50's + 70's or between versions of Cronic

18.2 FAKDSK (on HSC V370 and up)

HSC V370 has modified the disk path to support VMS access to the load device. As a result, FAKDSK is no longer needed, and FAKDSK support in the disk path no longer exists. Now, access to the load device is initiated through some SETSHO commands.

To enable creation of a fake unit from the load device, use the following SETSHO commands:

```
HSC70> R SETSHO

SETSHO> ENABLE REBOOT
SETSHO-S The HSC will reboot on exit.

SETSHO> SET SERVER DISK/LOAD_ACCESS

SETSHO> EXIT

SETSHO-S Rebooting HSC, type Y to continue, CTRL/Y to abort:  Y

INIPIO-I Booting...
```

Two fake units will be created and assigned unique unit numbers in the range of 4096–32767. The ENABLE REBOOT command is necessary to cause generation of the necessary structures for the virtual units. These units will be retained across subsequent reboots.

The assigned unit numbers can then be found by issuing a SHOW DISK command. Also, a SHOW SERVER command will indicate whether or not this server option is enabled.

Conversely, the following command disables virtual unit creation upon reboot:

```
SETSHO> SET SERVER DISK/NOLOAD_ACCESS
```

On each reboot, the virtual units default to "no host access." In order to enable access to a load device, use the SETSHO command:

```
SETSHO> SET Dn HOST_ACCESS
```

Once a virtual unit is set to host access, it can be accessed in the same manner as if it were running FAKDSK, such as using EXCHANGE.

Always issue the following command after you are done using the fake unit.

```
SETSHO> SET Dn NOHOST_ACCESS
```

18.3 SUMMARY (HSC Version 370 and up)

Use the following sequence of events to enable a load device for host access:

1. See if the virtual units have already been created:

   ```
   HSC70> SHOW DISK
   ```

   If they have, skip to step 4.

2. Create virtual units:

   ```
   HSC70> R SETSHO
   SETSHO> ENABLE REBOOT
   SETSHO-S The HSC will reboot on exit.
   SETSHO> SET SERVER DISK/LOAD_ACCESS
   SETSHO> EXIT

   SETSHO-S Rebooting HSC, type Y to continue, CTRL/Y to abort:  Y

   INIPIO-I Booting...
   ```

3. Obtain the unit number(s), (Dn):

   ```
   HSC70> SHOW DISK
   ```

4. Enable host access:

   ```
   HSC70> SET Dn HOST_ACCESS
   ```

5. Perform desired activity on load device.

Use the following to disable host access:

```
HSC70> SET Dn NOHOST_ACCESS
```

DSA TROUBLESHOOTING COURSE

Lab Exercise #5

SET HOST HSC and DKRFCT LAB

SET HOST HSC and DKRFCT Lab
Lab Exercise 5

1. Log into your student account.

2. $ **SET PROCESS/PRIV=DIAGNOSE** (You need this privilege to perform SET HOST/HSC functions.)

3. $ **SET HOST/HSC/LOG Node-name** (Use HSC node name assigned to you.)

4. Run DKUTIL and select your target disk using the GET command.

5. Using DKUTIL, set the FK bit = 1 in the mode word of the FCT volume control block. (This is bit 15 of word 21 in the first block of the FCT.)

6. Using the DKUTIL GET command, select the target disk again. Note that the "FCT:" should now indicate NULL since you set the FK bit.

7. DISPLAY the FCT and the RCT.

8. Locate an LBN that has not been replaced. 533333

9. Use DKUTIL and manually replace the LBN.

10. Run DKRFCT and select the target disk. Note the LBN(s) that were added to the FCT.

11. Run DKUTIL again and select your target disk. Note that the "FCT:" should now indicate VALID. This is because the FK bit is now cleared.

12. Dump the FCT volume control block. Note that the FK bit=0. This is because DKRFCT clears the FK bit when it adds entries into the FCT.

13. Display the FCT and verify that your manually replaced LBN is now permanently part of the FCT. Note any other LBNs that were added during step 10.

14. Exit from HSC console using CONTROL \ (back_slash).

15. Print a hardcopy of the file HSCPAD.LOG and review it. This is a log of your activities to/from the HSC.

DSA TROUBLESHOOTING FLOW CHART

DSA TROUBLESHOOTING FLOW CHART

Order Number: EK–DSATF–TM–PRE

This guide is the property of DIGITAL EQUIPMENT CORPORATION and is considered for DIGITAL INTERNAL USE ONLY.

Digital Equipment Corporation makes no representation that use of its products with those of other manufacturers will not infringe existing or future patent rights. The descriptions contained herein do not imply the granting of a license to make, use, or sell equipment or software as described in this manual.

Digital Equipment Corporation assumes no responsibility or liability for the proper performance of other manufacturers' products used with its products.

Digital Equipment Corporation believes that information in this publication is accurate as of its publication date. Such information is subject to change without notice. Digital Equipment Corporation is not responsible for any inadvertent errors.

Class A Computing Devices:

NOTICE: This equipment generates, uses, and may emit radio radio frequency energy. It has been tested and found to comply with the limits for a Class A computing device pursuant to Subpart J of Part 15 of FCC rules for operation in a commercial environment. This equipment, when operated in a residential area, may cause interference to radio/TV communications. In such event the user (owner), at his/her own expense, may be required to take corrective measures.

Revision/Update Information: PRELIMINARY, April 1989

This is the first document release from CX/CSSE and supersedes all previous versions. All revisions and known error corrections up to March/89 have been incorporated into this manual

CX/CSSE PRELIMINARY April, 1989

The postpaid READER'S COMMENTS form on the last page of this document requests the user's critical evaluation to assist in preparing future documentation.

The following are trademarks of Digital Equipment Corporation:

DEC
DEC/CMS
DEC/MMS
DECnet
DECsystem-10
DECSYSTEM-20
DECUS
DECwriter
DIBOL
EduSystem
IAS
MASSBUS
PDP
PDT
RSTS
RSX
UNIBUS
VAX
VAXcluster
VMS
VT
digital™

The following are also trademarks of Digital Equipment Corporation:

CI
DDCMP
DDIF
DEBET
DSA
DECconnect
DECdirect
DECdisk
DECmail
DECmat
DECmate
DECnet/E
DECnet-RT
DECnet-ULTRIX
DECserver
DECservice
DECtape
DELNI
DELUA
DEMPR
DEQNA
DESTA
DEUNA
DMS
DRB32
DSRVB-AA
HSC
IVIS
KA10
KD11
KDA50-Q
KDB50-A
KDB50-B
KI
KL10
KS10
LA50
LN01
LN03
MicroPDP-11
MicroVAX
MicroVMS
MSCP
PDP-11
Q-bus
RA60
RA70
RA80
RA81
RA90
RC25
RQDX3
RMS-11
RSX-11
RSX-11M
RSX-11S
RX33
SA482
SA550
SA600
SA650
TA78-81
TMS-11
TK50
TOPS-10
TOPS-20
TU78-81
UDA50
UETP
ULTRIX
ULTRIX-11
ULTRIX-32
VAXELN
VAX/VMS
VAXsimPLUS
VMS
RA82

This document was prepared using VAX DOCUMENT, Version 1.0

TABLE OF CONTENTS

FIGURES

TABLES

Preface

DSA Troubleshooting Overview

Manual Objectives

WHAT IT "IS"

The *DSA Troubleshooting Flowchart* provides the following objectives:

- The *DSA Troubleshooting Flowchart* presents a logical approach to DSA troubleshooting based on the experience of thousands of DSA service calls.
- Written by support engineers with many years of DSA troubleshooting experience, this manual helps to logically analyze the extensive error information and data that usually surround errors in the DSA environment. The main emphasis will be on the use of ERROR LOG information. Furthermore, this manual supplements existing service documentation and training.
- This document will be most beneficial to the Field Engineer who has some DSA training and/or experience. The experienced DSA troubleshooter can benefit from exposure to a troubleshooting approach that has been used successfully by other engineers. DSA trained engineers will find this document easier to use and understand than non–DSA trained engineers.

WHAT IT IS "NOT"

The following items are not within the scope of the *DSA Troubleshooting Flowchart:*

- This document is NOT a do–it–all "cookbook."
- This document is NOT meant to be self–sufficient.
- This document is NOT intended to replace existing documentation or training.
- This document is NOT intended to replace individual troubleshooting techniques or styles developed through experience.
- This document does not address DSA tape subsystems.

TROUBLESHOOTING OVERVIEW

Keep in mind the following points when troubleshooting:

- The same features that make DSA a "high availability/ fault tolerant" architecture also make a "traditional" approach to troubleshooting often inappropriate. (For example, trying to force the problem to recur, using diagnostics.)
- Accurate diagnosis often must be (and usually can be) made based on careful examination and analysis of information that already exists without resorting to diagnostic simulation and the resulting disruption to customer operations (symptom–directed vs. test–directed diagnosis).
- This document is a guide for troubleshooting DSA disk subsystems, reflecting RAxx series disk drives and associated controllers (HSC, xDA, and so on). The troubleshooting flows and/or techniques may, however, apply to other DSA products (such as RDxx).

Use the following steps for more effective troubleshooting:

A. Gather as much information as you can about the problem before using the flowchart, including:

- Error logs
- Messages
- Customer input
- Device error indications

B. Stay organized, since there may be an overwhelming amount of information and data to consider, especially surrounding intermittent problems:

- Make notes, keep records
- Get hard copies, if possible

C. Take time to analyze the data

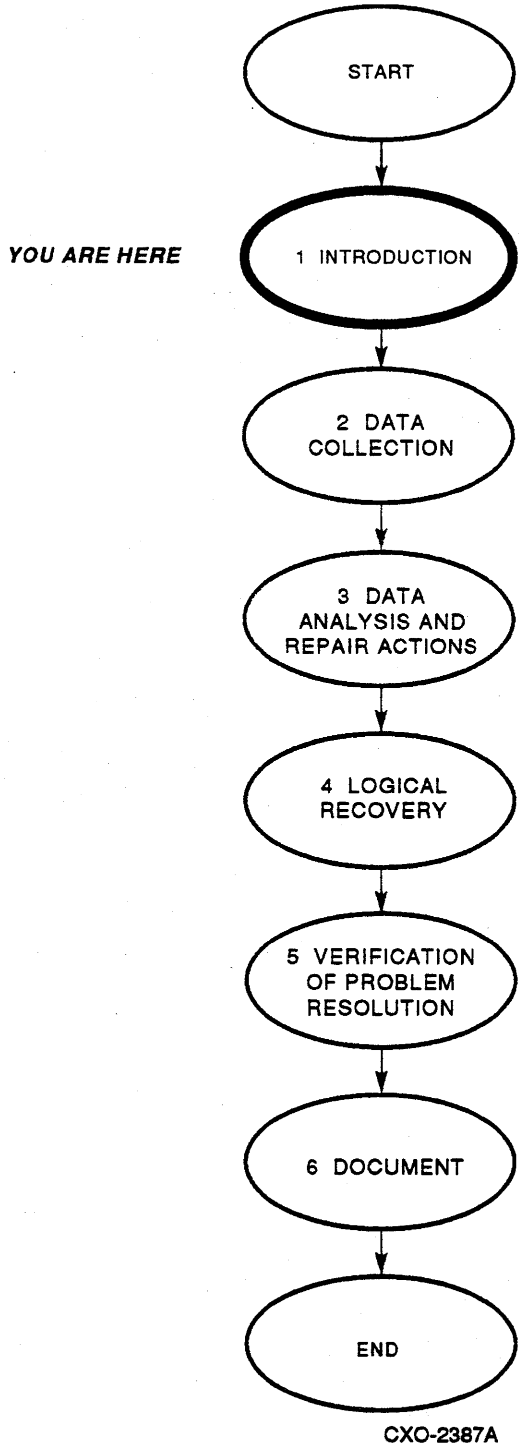
- Use 20 minutes to analyze the data and understand the problem. That can save possibly hours of downtime caused by a "shotgun" approach.
- In most cases, the customer's system is probably not DOWN at the moment, so take the time to understand the problem *before* taking down the machine or system for a service action. Many service actions, such as FRU replacements, can and should be "scheduled" with the customer, to minimize disruption to any operations. (**Remember** DSA is a "FAULT–TOLERANT" architecture.)

D. When in doubt, use the following services:

- Remote support
- Local district/area support
- Other field engineers

THERE IS LOTS OF HELP AVAILABLE — USE IT WHEN YOU NEED IT.

Figure 1: Flow Map – Introduction



CHAPTER 1

INTRODUCTION

1.1 Structure of This Document

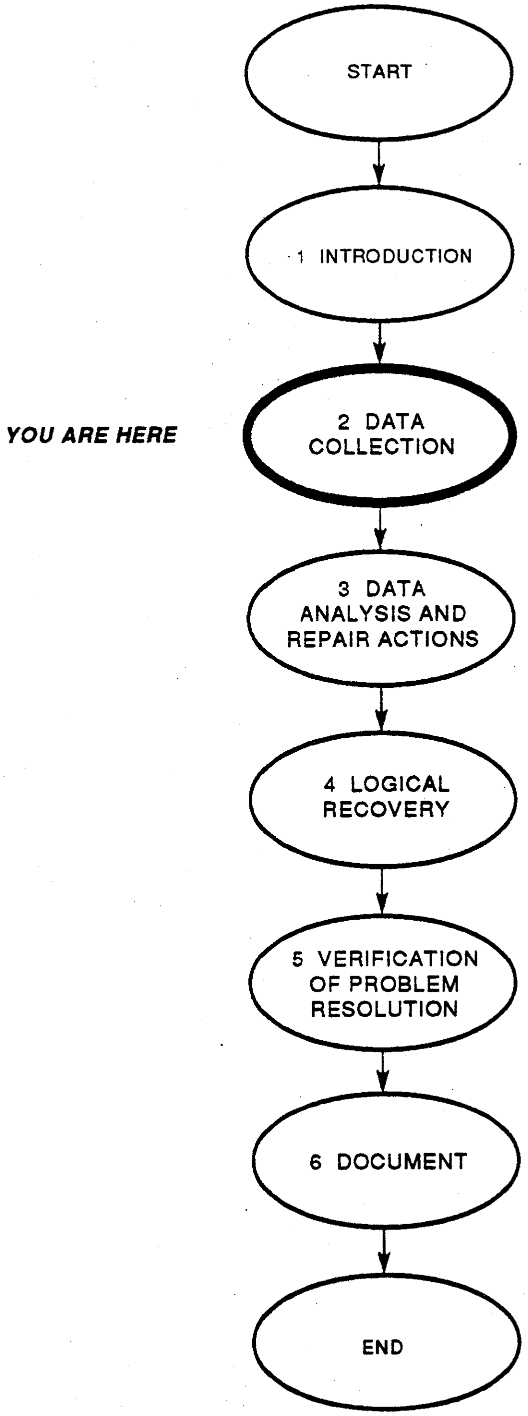
This flow process incorporates a collection of troubleshooting techniques from a variety of resources and represents the most up–to–date approach. This includes the basic 6–step troubleshooting strategy incorporated by all of the recently announced disk products.

This document is divided into six chapters that map the DSA troubleshooting flow process. Refer to the diagram on the opposite page. Flow diagrams appear on a *left–facing* pages and all pertinent notes appear on a *right–facing* pages.

Initial review of this document may appear overwhelming, which is expected. As you gain experience in using the methods outlined here, this document will become a resource for most typical DSA problems, as well as a guide for occasional "difficult" DSA problems.

Review the entire process to understand the approach and become familiar with the location of most of the material in this document. Later, with experience, you will find that you can skip much of the material you already understand or material that does not apply to the problem you are troubleshooting.

Figure 2: Flow Map – Data Collection



CXO-2388A

CHAPTER 2

DATA COLLECTION

2.1 Overview

The Mass Storage Control Protocol (MSCP) error logging and reporting system is perhaps the single most important maintainability feature of the DSA architecture. All higher level error analysis tools (VAXsimPLUS, SPEAR, and so on) use the error log information as their source data.

If error logging is available in any form (such as HSC console, system error log), it is your primary tool for obtaining error information.

NOTE

> **Most errors in the DSA environment are soft (recoverable) but they will be reported to the error log in most cases.**

Section 2.2 focuses on the use of the error log as the source of error information.

The amount of data information in error log packets can be overwhelming and possibly confusing at times. However, only a few key fields of the packet are necessary to make a diagnosis in most cases. The key starting point in error log packet interpretation is the **STATUS/EVENT** code.

2.2 Host Error Log

The use of host error logs is a data collection process. Begin this data collection process by accessing the host error logs, obtaining the STATUS/EVENT codes, and noting the Logical Block Numbers (LBNs) for any read/write (R/W) disk transfer errors.

While performing this data collection step, if the following errors are detected by the controller, note the LBN(s) being reported:

- Data errors
- ECC errors
- Uncorrectable ECC errors
- Header not found errors
- Invalid header errors
- Header compare errors
- Format errors
- Data sync timeout errors

The primary elements of an error log entry for troubleshooting are:

- MSCP STATUS/EVENT code
- Master drive error code or LED code
- LBN
- Date/time of error
- Cylinder
- R/W head
- Sector

The cylinder, R/W head, and sector represent physical disk elements that are obtained by translating LBNs from an error log entry. Those physical characteristics are NOT usually indicated in most error logs. Several techniques and utilities are available to provide the necessary translation. Some of those utilities include BLOCK.COM, VAXSIM$LBN.COM, DSAERR, RAUTIL, and disk internal error logs. Appendix C provides manual conversion algorithms in the event that a conversion utility is not readily available.

The data collection step is critical to effectively troubleshoot DSA subsystems.

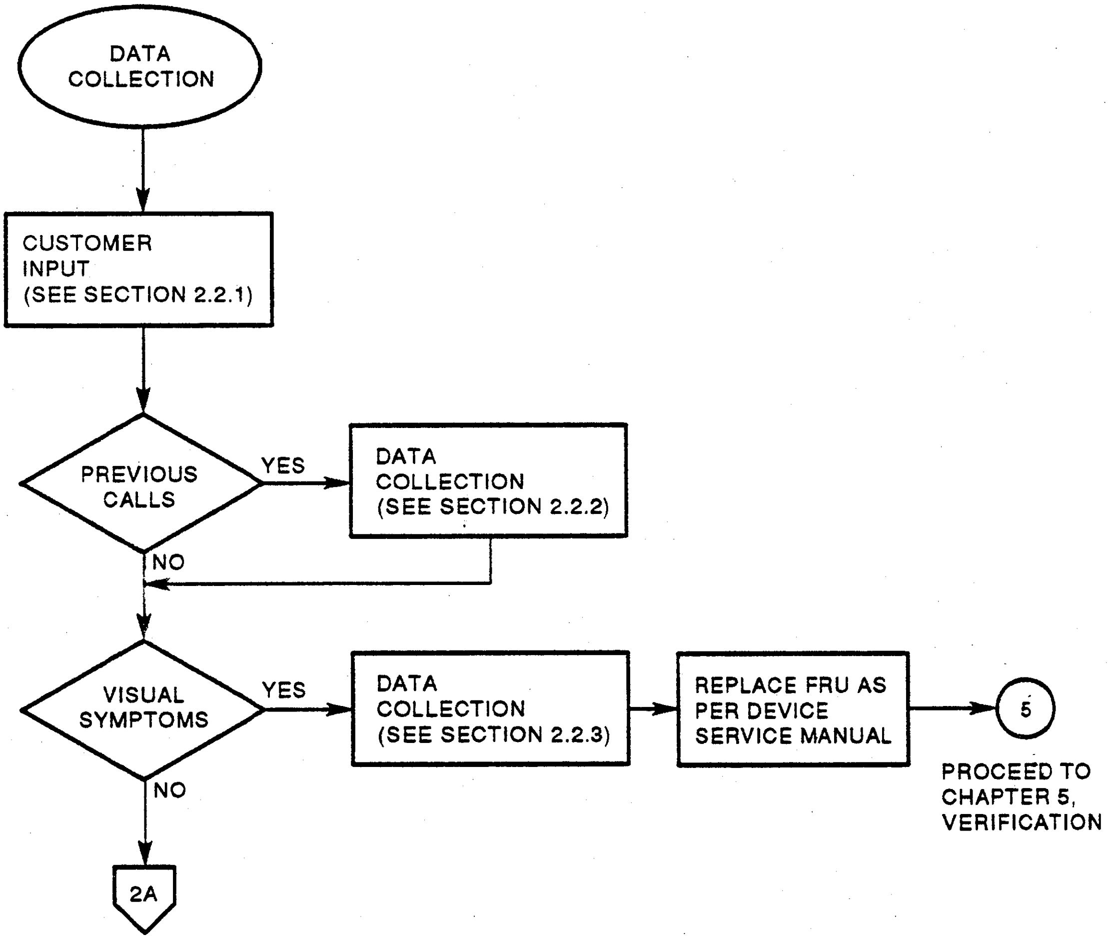
1. Gather as much information as you can about the problem before using the flowchart, including:
    - Error logs
    - Messages
    - Customer input
2. Stay organized, since there may be an overwhelming amount of information and data to consider, especially surrounding intermittent problems:
    - Make notes, keep records
    - Get hard copies, if possible

This chapter outlines most of the available resources for collecting data while troubleshooting a problem on site. It is important to get the *full scope* of the problem. With all of the available information collected, you will usually be able to obtain the essential information needed by this flow technique. Namely:

- MSCP STATUS/EVENT code
- Master drive error code (if applicable)
- LBNs (if applicable)
- Date/time of errors

This chapter outlines the data collection resources for a variety of DSA system configurations. For a specific site configuration, not all of these resources will apply. Use those that are applicable. The resources noted here are in "preferred priority" to maximize system availability to the customer. A "symptom–directed" data collection process prescribes the "best resources" first and other resources later.

Figure 3: Flow – Data Collection (Start)



CXO-2389A

Service personnel should use one or more of the data collection mechanisms before continuing the analysis of the error types discussed in Chapter 3.

2.2.1 Customer Input

Discuss the problem the operator experienced and how often the problem appeared during system operation. Operators/users may provide valuable information concerning system activity at the time of the errors (such as applications that were running, affected users, impact on other applications, and so on).

NOTE

Obtain any error messages from the user terminal.

2.2.2 Previous Call History

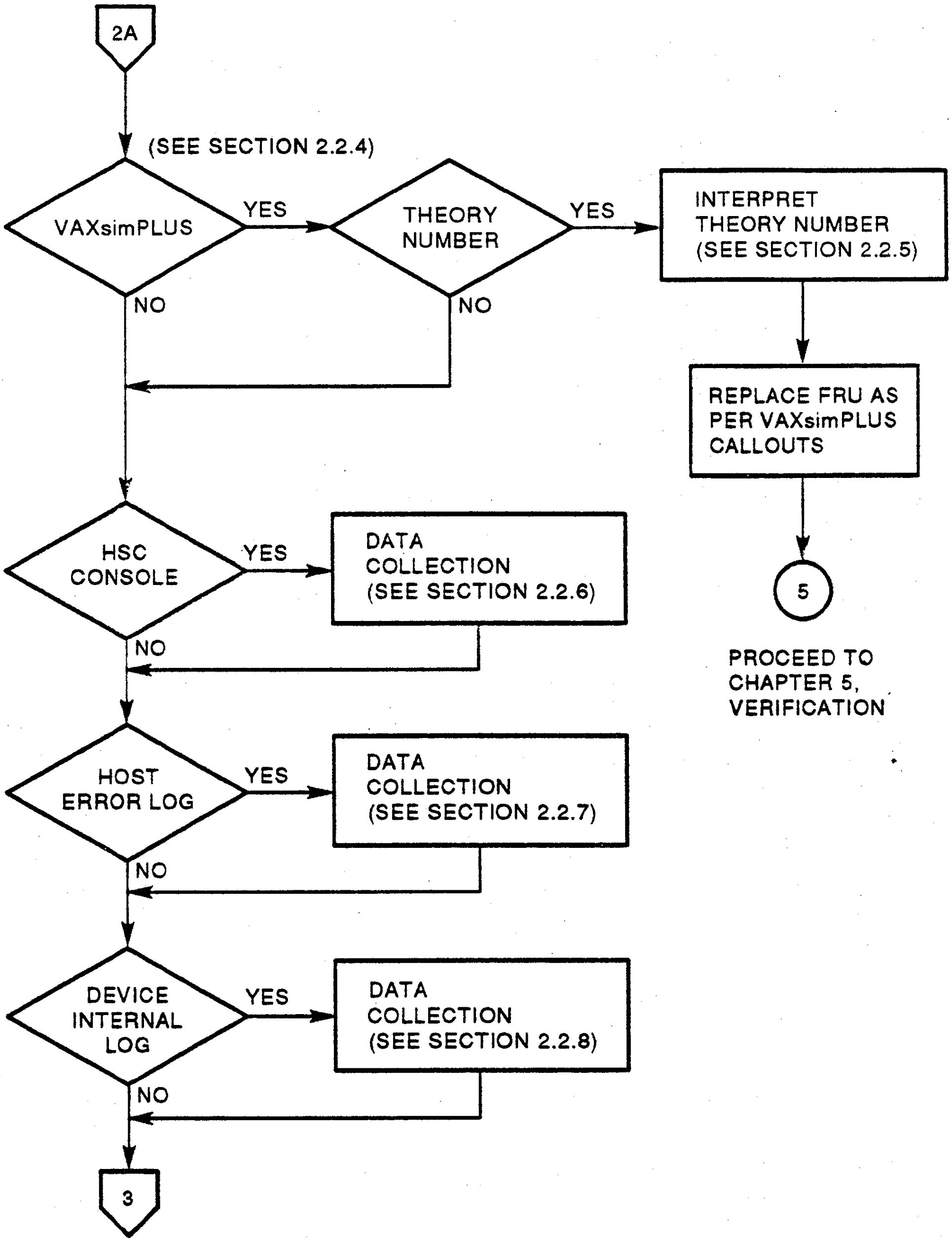
The site guide *may* have valuable information that should be considered before you proceed with the service call. Look over what problems the site has been experiencing recently. There may be indications of a repeat call or an intermittent problem. It is possible that you have previously analyzed an intermittent problem and selected one of the multiple recommended Field Replaceable Units (FRUs) that could contribute to the current symptom (as previously documented). If this problem is a repeat of the same symptom, select the next suggested FRU for replacement.

2.2.3 Visual Symptoms

Hard/repeat faults displayed in the drive Operator Control Panel (OCP) may directly correlate to an FRU. Hard/repeat faults may lead to directly analyzing the drive internal error log, if available. Hard/repeat faults include fault lights, error Light Emitting Diodes (LEDs), noise, smoke, mechanical failures, power problems, drive spin up/down problems, and internal diagnostic failures.

Refer to the appropriate device service manual for details on the error code and, if applicable, for the FRU.

Figure 4: Flow – Data Collection (2A)



CXO-2390A

2.2.4 VAXsimPLUS

If such tools as VAXsimPLUS or SPEAR are available, a quick look at the summary report containing evidence information may lead to direct identification of the failing/faulty device.

2.2.5 Theory Number

Using analysis of the errors being recorded by the VMS system error logger, VAXsimPLUS can determine which FRU(s) may need replacement based on the analysis and frequency of errors. VAXsimPLUS identifies the failure with a theory number, which can be cross–referenced to a particular FRU. Call the CSC to cross–reference a theory number to the FRU. Use the following numbers for assistance:

Installation and Usage –	PL01 = CSC/AT (Atlanta)	1–800–241–2546
	PL31 = CSC/CX (Colorado)	1–800–525–6570
Europe and GIA areas –	The telephone numbers are area dependent. Check with your support center for the correct telephone number.	

2.2.6 HSC Console

If the subsystem is HSC based, check the HSC console log. The HSC console log may indicate which drive has a problem. Correlating user–provided time information of a disk subsystem error occurring during a user operation eases searching through hardcopy HSC console trails to identify a failure. This assumes that the HSC time is the same as that of the user setable system clock.

NOTE

Refer to Appendix A for specific "drive–detected error" examples.

2.2.7 Host Error Log

Study available host error logs. Error summaries provided by some error log utilities enable you to identify the suspect drives. It is useful to correlate user–provided information around the time of the errors to the time stamps in the error log. For a complete description of the host error log messages, refer to the *DSA Error Log Reference Manual (EK–DSAEL–MN–002).*

2.2.8 Device Internal Error Log

Device internal error logs and/or error silos contain useful information occasionally not found in sources previously discussed. They may also contain misleading information. For example:

- Error information from a previous or unrelated problem
- Status information not related to the immediate problem

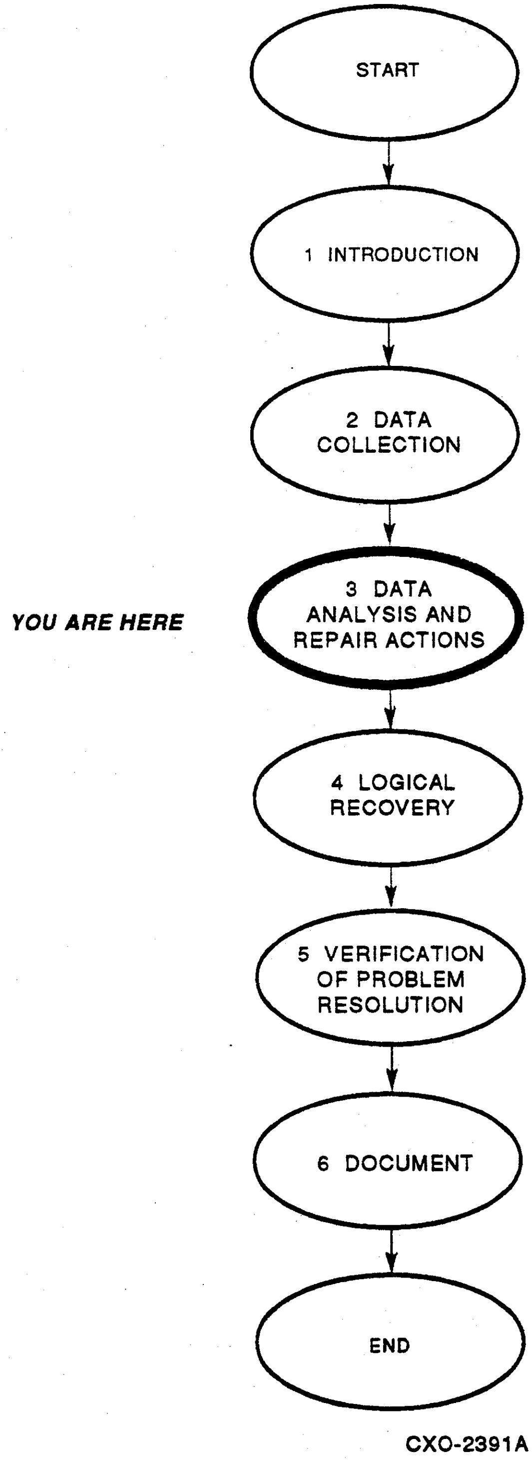
Internal error logs accumulate data over a period of time. Therefore, select the information that appears appropriate for the current problem. The device internal error logs can be dumped by using DKUTIL, NAKDAx, or EVRLL/ZUDMxx.

NOTE:

Using NAKDAx or EVRLL/ZUDM standalone diagnostics will remove system availability from the users.

After you have collected all available data, proceed to Chapter 3, Data Analysis and Repair Actions, page 13.

Figure 5: Flow Map – Data Analysis and Repair Actions



CHAPTER 3

DATA ANALYSIS AND REPAIR ACTIONS

SECTION 3.1

3.1 Overview

This chapter integrates two primary areas of troubleshooting into one.

1. Analyzing the data you collected in Chapter 2 (Data Collection) to derive a specific symptom.
2. Suggesting what remedial action could be performed to resolve the problem.

The symptom analysis is based on two key areas. The first area (most preferred) presumes you have collected status/event codes.

1. Scan through the flow until you find a status/event code decision diamond that matches one or more of those codes you isolated using data collection.
2. Then follow the flows and notes for the specific status/event code to further isolate the problem and select appropriate repair actions as described in this section.

Logical repair (non–physical) is covered in Chapter 4 of this document. The most commonly encountered status/event codes are provided. If you encounter status/event codes not discussed in the document, consult your nearest support engineer or remote support organization.
The second area of analysis provides further information to troubleshooting symptoms that often occur with no corresponding status/event codes logged in the host/controller error logs. These decision diamonds "follow" the strings of status/event code flows. Only the most frequently encountered symptoms are listed.

In short, if data collection provides one or more status/event codes, then use the status/event flow for troubleshooting. If symptoms occur without status/event codes in the error log, then skip the status/event code decision flows and proceed directly to Section 3.1.12 on page 27.

The status/event code flow is a "prioritized" flow. If you have gathered "multiple" status/event codes from the data collection process, select the status/event code decision diamonds that match your codes and appear as near the beginning of the flow as possible. This will allow you to troubleshoot and usually resolve the "root" problems first, which are often responsible for the occurrence of "other" symptoms and status/event codes reported.

Finally, a few status/event code decisions involve more complex flows for subsequent analysis. These areas include drive–detected errors, communication errors, data errors, and so on. Flows for these topics are further described in other areas of this document. These flows are kept separate so as not to clutter or cause confusion with the main status/event code analysis flow.

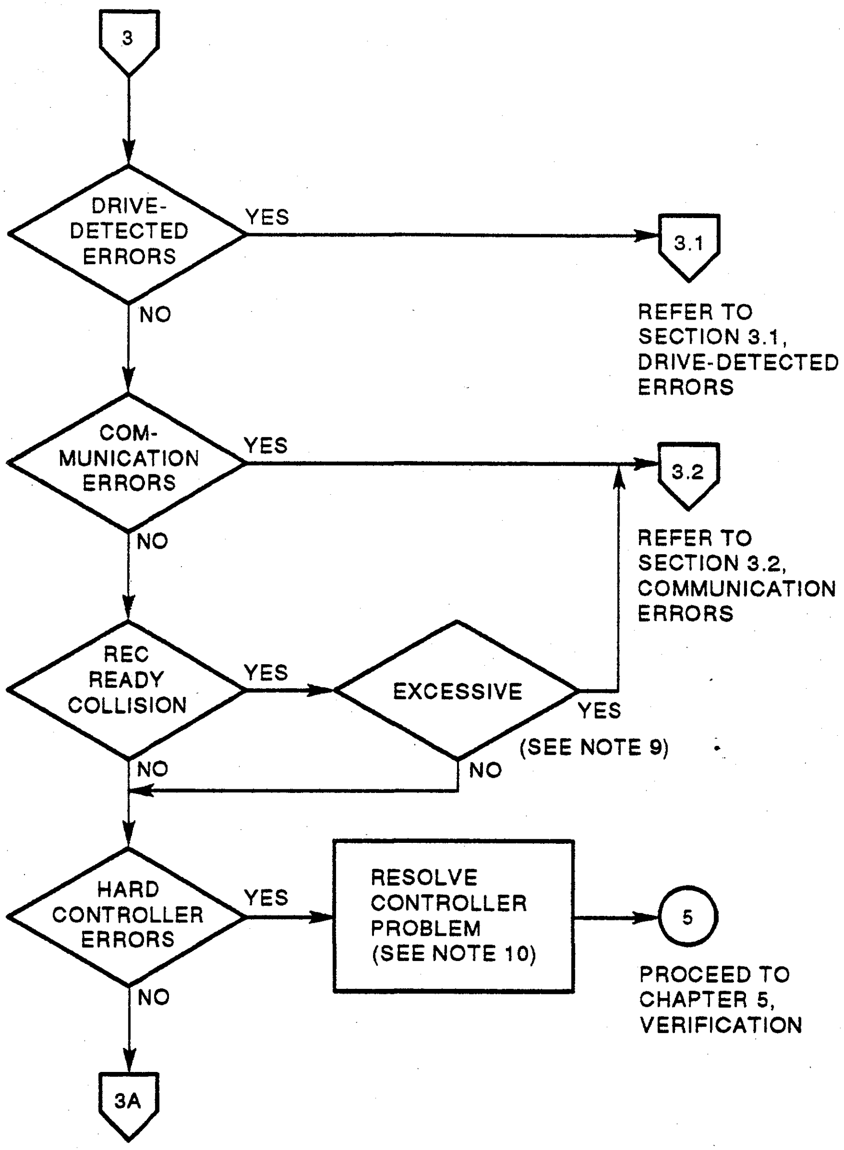
After this chapter, you can continue with the logical recovery step (Chapter 4) or proceed to the verification step (Chapter 5). This will depend on the decision flows you followed during the data analysis and repair actions steps.

Figure 6: Flow – Data Analysis and Repair (3)

BEGIN DATA ANALYSIS AND REPAIR PROCESS ON STATUS/EVENT CODES. FOR YOUR CONVENIENCE, THESE EVENT CODES ARE DOCUMENTED IN BOTH HEX AND OCTAL.

STATUS/EVENT CODE

HEX	OCTAL
EB	353
4B	113
10B	413
14B	513
1AB	653
2A	052
6A	152
8A	212
10A	412
12A	452
14A	512
16A	552



CXO-2392A

3.1.1 Excessive Receiver–Ready Collisions (1AB)

These are recoverable events that do not result in data loss unless other SDI bus errors occur at the same time. This event usually occurs when the drive attempts to raise RECEIVER READY (RTDS status bit), indicating the drive is ready to receive a command from the controller while the controller had previously raised RECEIVER READY (RTCS status bit), indicating the controller was ready to receive a drive response.

- This is not an error, but an event within the subsystem.
- **All DSA drives and controllers** will occasionally result in this event being logged.
- There is no performance impact associated with occasional occurrence of this event since the normal recovery time from this event is very fast.
- Data integrity is assured and no data corruption is associated with the occurrence of this event provided there are no other SDI bus errors at the same time.
- Testing of HSC software Version 370 has indicated a noticeable reduction of the normal occurrence of receiver–ready collision events.
- Assuring that all drives and controllers are up to the latest hardware and software revision levels will contribute to the reduction of receiver–ready collisions.

A receiver–ready collision rate of one or two events a week per drive to one or two events a day per drive is usually acceptable for most sites. This presumes that:

- Physical SDI interconnects are not being broken (plugging and unplugging SDI cables, worn connectors, and so on).
- Controller initialization is not occurring.
- Controller failover operations are not occurring.

If you encounter excessive receiver–ready collisions, proceed to Section 3.3, Communication Errors, page 39.

NOTE

> **There are many possible causes for this event. The occurrence of receiver–ready collisions happens primarily when both A and B ports are enabled at the drive.**

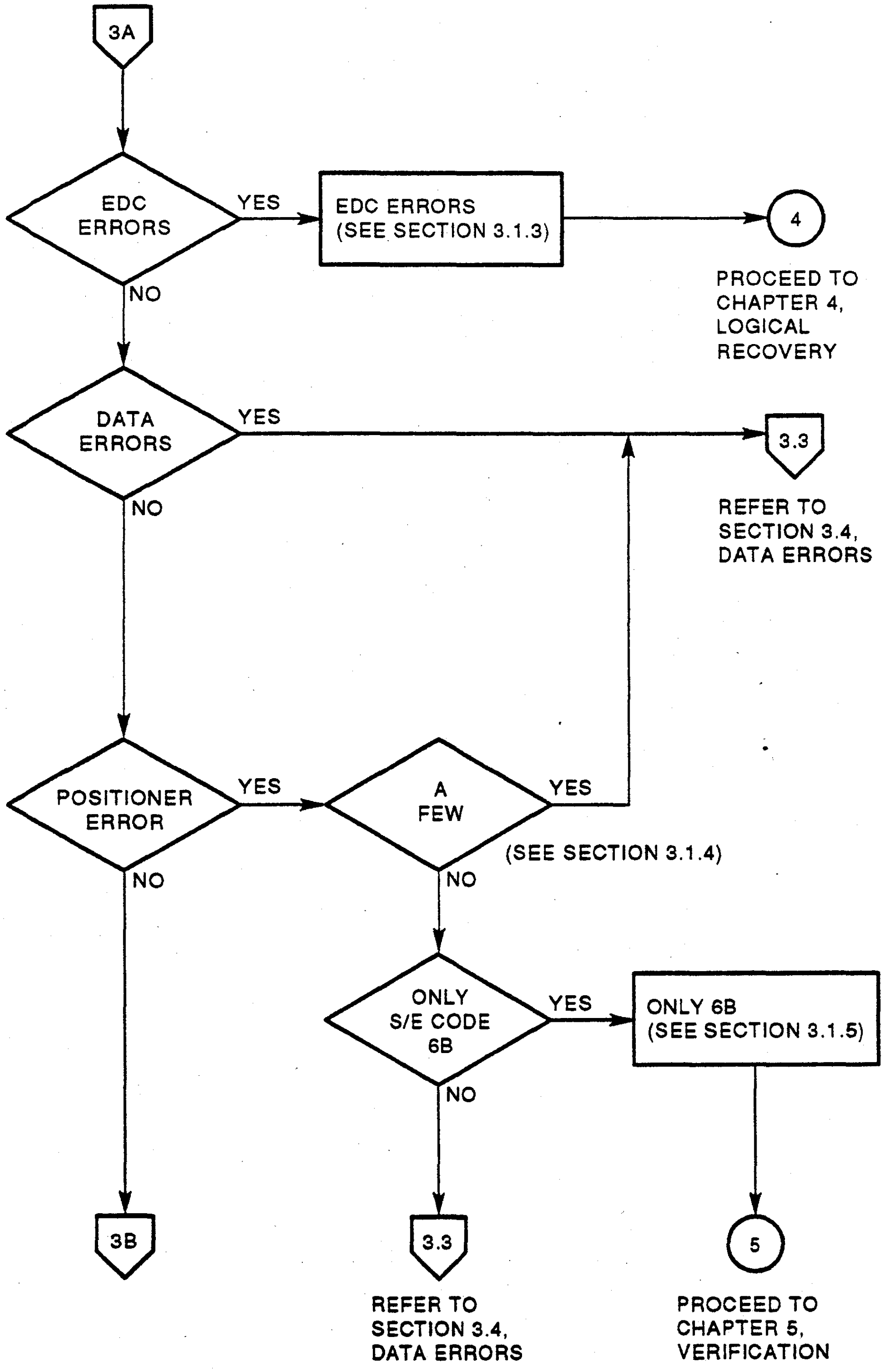
3.1.2 Hard Controller Errors

1. Refer to the *DSA Error Log Reference Manual (EK–DSAEL–MN–002)* for a complete description of the status/event code.
2. Refer to the specific controller service manual to identify the appropriate list of FRUs for replacement, based on the specific description of the status/event code symptoms.

Figure 7: Flow – Data Analysis and Repair (3A)

STATUS/EVENT CODE

HEX	OCTAL
25	045
4A	112
45	105
48	110
65	145
68	150
88	210
E5	345
38	350
128	450
148	510
168	550
188	610
1A8	650
1C8	710
1E8	750
6B	153



CXO-2393A

3.1.3 EDC Errors

Error Detection Code (EDC) is a data–protection mechanism to ensure data integrity of the controller internal data path. In contrast, the ECC mechanism ensures data integrity from the controller through the drive, to the media, and back again. ECC provides error detection and correction for both the data field and the EDC field within a block of data read from the disk.

It is important to note the differences in how controllers implement the EDC mechanism. For the KDA/KDB/UDA family of controllers, EDC is generated on a sector of data at the CPU bus interface as the data is initially read from host memory. It is verified on a sector basis as the data is written to host memory from the internal controller memory. Therefore, with the xDA/xDB controllers, it is generated and checked at this CPU bus interface within the controller by the microcode engine of the controller.

For HSC controllers, the EDC is generated on a sector of data at the K.pli port processor module as the data streams in from host memory over the CI bus. The EDC then becomes an integral part of the user data as the data is transferred to the HSC data memory. As this data is read out of the HSC data memory by the K.sdi modules and transmitted to the drive, the EDC for the user data is regenerated in the K.sdi and compared to the EDC characters appended to the data by the K.pli module.

The EDC characters must match or the write transfer to the disk will be aborted. The HSC re–requests the data from host memory and requeues the write transfer to the disk when data is again available in the HSC data memory. If the EDC verifies correctly at the K.sdi on a write to disk, the EDC and ECC codes are appended to the data stream and written to the disk, with the ECC mechanism ensuring data integrity of the customer data and the EDC code.

Note also that EDC errors can be caused by the host if an incomplete data transfer results. As an example, the controller may prematurely terminate a write transfer due to a host initialization, CPU crash or reboot during the transfer.

For a read from the disk, the data as it is read by the K.sdi (over the SDI read/response line) is checked for good ECC, then the data plus EDC characters are stored in HSC data memory. As the data is sent to host memory, the K.pli, while transferring the data to host memory, verifies that good EDC exists for the customer data block but does not transfer EDC characters to host memory. If the EDC is bad, the K.pli informs the HSC functional code to re–request the same data from the disk.

If EDC errors are detected without ECC errors, the problem is in the controller. This is because the ECC is protecting the data to and from the disk and checking the integrity of the data at the SDI port module logic.

NOTE

A properly functioning controller always reports bad EDC written to disks. If bad EDC is written to a disk (improperly functioning controller), each time the block containing bad EDC is read, EDC errors are logged against the drive. Only after the data is restored or rewritten to the disk with good EDC by a good controller will the errors be resolved.

First resolve the problem in the controller that caused the EDC errors. Then proceed to Chapter 4, Logical Recovery, page 53 to resolve the EDC errors that were written to the disk.

3.1.4 Few 6B Errors

If either of the following symptoms exist, then treat these errors as data errors:

1. Only a few (for example, less than 10 per day) error log entries contain status/event codes of 6B

   or

2. Any number of 6Bs are logged with other data transfer–related errors (ECC, header, and so on)

Then treat the errors as data errors and proceed to Section 3.4, Data Errors, page 43.

3.1.5 Many 6B Errors

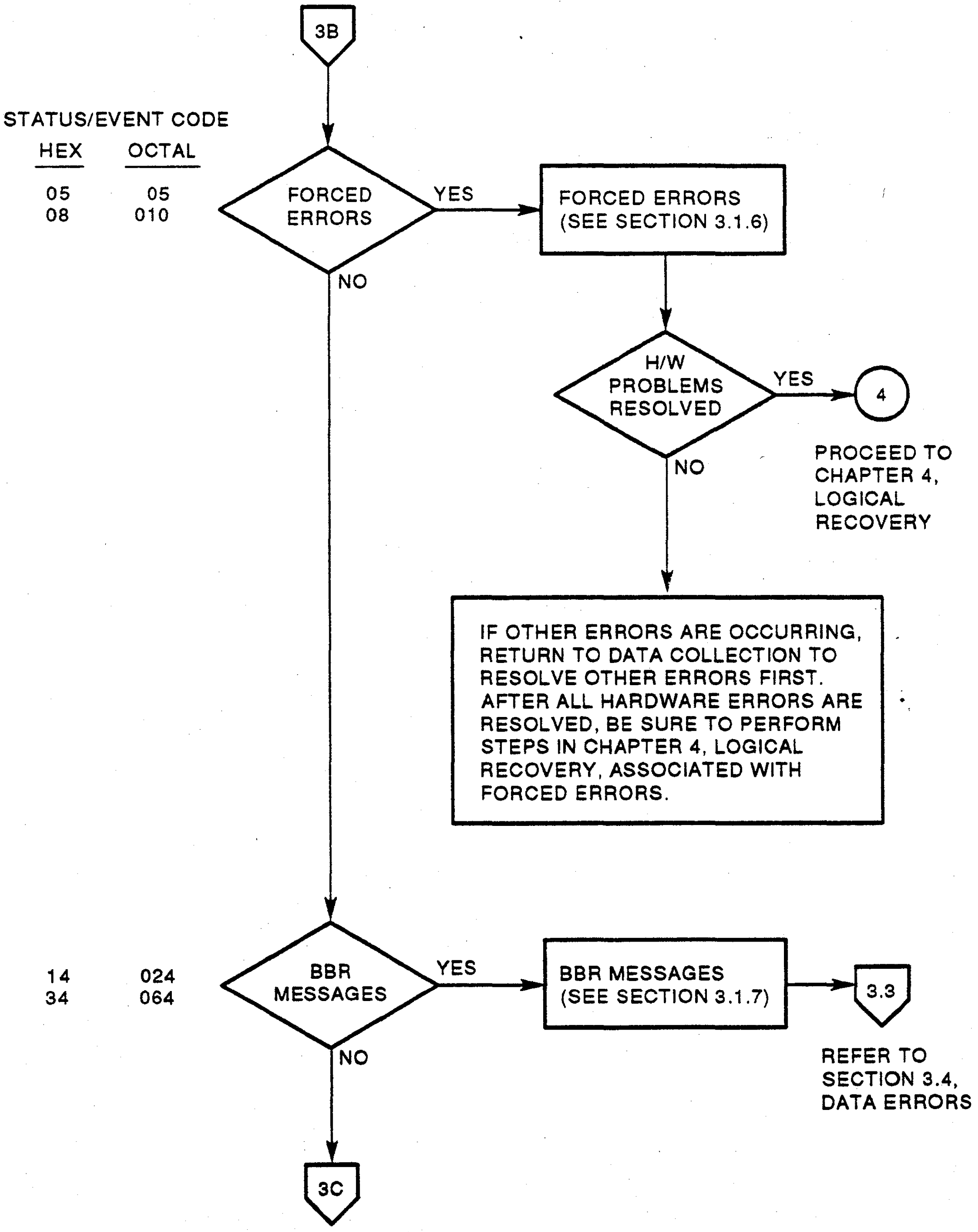
If many different blocks log only 6B errors (no other status/event codes), try reformatting the media. If other data errors are included with the 6B errors, proceed to Section 3.4, Data Errors, page 43.

NOTE

If you have previously reformatted the media for this problem and the 6Bs persist, replace the media (HDA or pack).

Fails to sync up on first revolution gets it on next revolution.
This is a soft event!

Figure 8: Flow – Data Analysis and Repair (3B)



CXO-2394A

3.1.6 Forced Errors

The FORCED ERROR flag is an indicator used to inform the host that corrupted data is "correctly written" into a sector. A forced error in a system file or executable file (.EXE) will cause problems not easily isolated (e.g., system crash, system hang, and so on). Any file being read on a user terminal with a forced error will result in a VMS message. This may or may not result in additional entries into the error log.

VMS produces the following error message when a forced error is in a block of a file that the user attempts to TYPE:

```
$ TYPE FILE.DAT
%TYPE-W-READERR, error reading $5$DUA230:[LOCAL.TEST]FILE.DAT;2
-RMS-F-RER, file read error
-SYSTEM-F-FORCEDERROR, forced error flagged in last sector read
```

When an uncorrectable ECC error is encountered in a block, several attempts are made to read and/or correct the data. If those attempts fail, the block causing the uncorrectable ECC error is assumed to be bad and becomes a candidate for replacement. During the replacement process (BBR), the bad block is read again (including retries) in an attempt to extract the data for relocation to a replacement block.

If the data is STILL uncorrectable, the BBR process writes "best guess" data into the replacement sector. The result is invalid data being correctly written to a good block. To inform the user that the data was at one time uncorrectable, the forced error flag is attached to the block.

It is the responsibility of the user to take the necessary steps to correct or replace the data and "clear" the forced error indicator. The only assured way to correct a forced error is to replace the file containing the affected LBN from a known good backup copy.

Forced errors are the result of another error or problem. Use previous history or further analysis of the data available to determine if the hardware problem that created the forced errors has been resolved. A status/event code indicating a forced error is associated with status/event code 14 or 34 (BBR status messages).

Status/event code (hex) 05 indicates a forced error in the RCT area.
Status/event code (hex) 08 indicates a forced error in the HOST area.

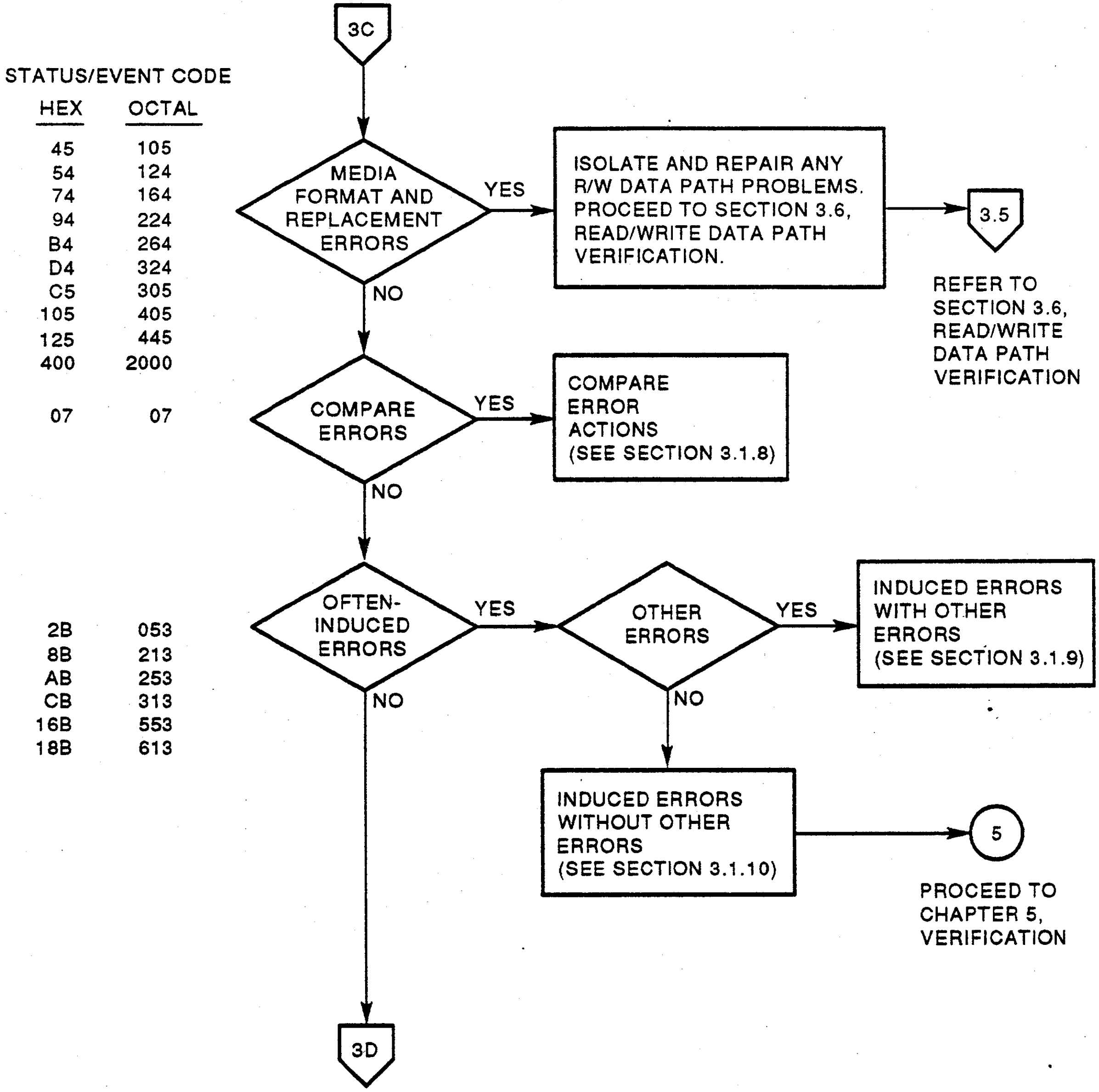
NOTE

Correct all known hardware problems first, then proceed to Chapter 4, Logical Recovery, page 53.

3.1.7 Bad Block Replacement Messages

These are NOT errors, but information entries reflecting the status of the Bad Block Replacement (BBR) process. A block read causes the data error to invoke BBR (set the BBR flag). The block is tested. The block either passes the test (possible transient error), which is status/event code = 34, or the block fails the test, which is status/event code = 14. Entries with the same LBNs can be associated with the BBR messages. Identify all entries with the same LBN to get the total picture. Note that the error log entries may be out of sequence.

Figure 9: Flow – Data Analysis and Repair (3C)



HEX	OCTAL
45	105
54	124
74	164
94	224
B4	264
D4	324
C5	305
105	405
125	445
400	2000
07	07
2B	053
8B	213
AB	253
CB	313
16B	553
18B	613

CXO-2395A

3.1.8 Compare Errors

A compare operation takes the data buffer in host memory as the result of the last I/O operation (read or write) and compares it to the data that is on the device. This compare is done in the controller.

NOTE

> **An error results if hardware or software modifies host memory before the compare operation is complete.**
>
> **Typically, any errors detected by the drive will result in other status/event codes or drive errors. Troubleshoot those errors first.**

The elements involved in this operation are controller memory, host memory, and the path between the controller and host memory.

3.1.9 Induced Errors with Other Errors

These status/event codes are often a result of:

- Drive–detected errors (status/event code = EB)
- Drive failures (hard faults)
- Controller failures
- SDI hardware problems

If other errors exist, troubleshoot those first. They will often isolate the root of the problem. For example, an error with a status/event code of EB (drive–detected error) is normally logged by the host AFTER the system has logged any number of induced errors.

3.1.10 Induced Errors Without Other Errors

Quite often one error in the DSA architecture may cause other errors to get reported to the error log. These "induced" errors are normally not the basis for primary troubleshooting. In the event that induced errors are the only available symptoms in the error log, use the following information for troubleshooting.

3.1.10.1 Controller Detected: Drive Command Timeout (Status/Event Code: 2B)

The controller timed out while waiting for the drive to complete an operation. If no other errors are associated with those errors, the problem may be due to a seek timeout. The drive internal error log or error silo may provide additional information.

Refer to Section 3.1.11, Suggestions for Troubleshooting Induced Errors, page 25.

3.1.10.2 Controller Detected: Loss of Read/Write Ready (Status/Event Code: 8B)

This error indicates R/W Ready (RTDS status bit) was negated when:

1. The controller attempted to initiate a transfer.

   or

2. A R/W Ready was found negated at the completion of a transfer **and** R/W ready had been previously asserted (indicating completion of a preceding seek).

This usually results from a drive–related error. The drive internal error log or silo may provide additional information.

Refer to Section 3.1.11, Suggestions for Troubleshooting Induced Errors, page 25.

3.1.10.3 Controller Detected: Drive Clock Dropout (Status/Event Code: AB)

Either data (Read/Response Line) or the state clock (RTDS) was missing when it should have been present. This is usually detected through a timeout.

A fatal drive condition can cause the drive to drop the drive clocks. The drive should reassert clocks after performing a drive INIT and establishing clocks to the controller to reestablish communications and state information between the drive and controller. The sequence of getting status and error information then occurs. Analysis of error log message packets usually indicates that the above sequence has occurred.

If such message packets are not being processed or received, the condition might not be detected by the drive. Possible causes for this problem include:

- Drive SDI logic and microprocessor logic
- Controller port module
- SDI bus (cables, connectors, and so on)

3.1.10.4 Controller Detected: Lost Receiver Ready (Status/Event Code: CB)

The Receiver Ready (RTDS status bit) was negated when the controller attempted to initiate a transfer, or RECEIVER READY was not asserted at the completion of a transfer. This includes all cases of the controller timeout expiring for a transfer operation (level–1 real–time command).

As a consequence of this condition, the controller performs an SDI INIT and then attempts to request a GET STATUS. The extended status error log entry returned in the GET STATUS command may indicate what the problem is.

If no information is being reported by the drive as a part of the error log sequence, approach the problem as a drive transmission–related error and proceed to Section 3.3, Communication Errors, page 39.

3.1.10.5 Controller Detected: Drive Failed Initialization (Status/Event Code: 16B)

The drive clock failed to resume following a controller–attempted drive initialization. This implies the drive encountered a fatal initialization error. It also can indicate the drive was attempting its own initialization or that the drive was looping in an initialization state or routine.

3.1.10.6 Controller Detected: Drive Ignored Initialization (Status/Event Code: 18B)

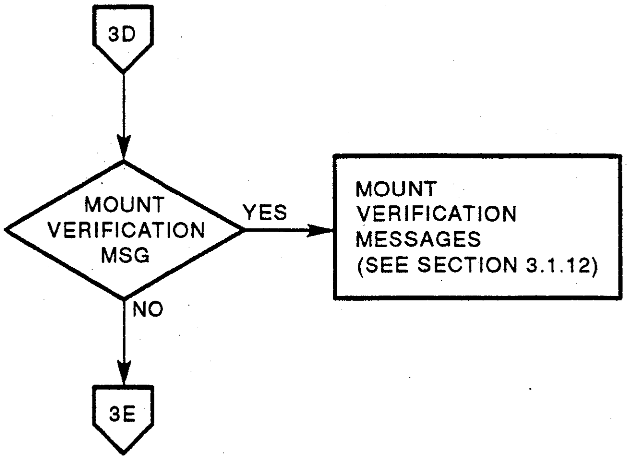
The drive clock continued running even though the controller attempted to perform a drive initialization. This implies the drive did not recognize the INIT command from the controller. It may also indicate the drive was performing an initialization caused by a drive–detected condition and, in the course of initialization, ignored the controller's attempt to initialize the drive.

3.1.11 Suggestions for Troubleshooting Induced Errors

If any of the previous (2B, 8B, AB, CB, 16B, 18B) errors are the **ONLY** errors, consider one or more of the following troubleshooting tips:

- Try to isolate the problem to a specific controller, drive, or SDI connection. Refer to Section 3.5, Device Isolation, page 48.
- Review the site history of previous service calls for any information that may allow further isolation of this problem (or other problems that might relate to this error).
- Verify all SDI cable connections.
- Investigate your system configuration. Could it be contributing to the problem?
- If these problems are common to one drive:
    1. Obtain any drive internal error log information and resolve any errors by using the appropriate drive service manual.
    2. Verify drive operation by using the drive internal diagnostics.
    3. Troubleshoot the drive SDI logic or microprocessor logic first.
- Contact the appropriate support resources for additional information or help.

Figure 10: Flow – Data Analysis and Repair (3D)



CXO-2396A

3.1.12 VMS Mount Verification Messages

Mount verification or mount verification timeout messages may or may not be the result of a hardware problem. The following are some of the reasons for mount verification messages:

1. The mount verification feature of Files–11 disk handling generally leaves users unaware that a mounted disk has gone offline and returned online.
2. A mounted disk has become unreachable and then restored. Mount verification is the default parameter for EXE$MOUNTVER.

3.1.12.1 Exceptions

Disks mounted /FOREIGN and disks mounted /NOMOUNTVERIFICATION do not undergo mount verification, except during cluster state transitions.
Dual–ported drives through HSCs should never be mounted using the /NOMOUNTVERIFICATION modifier, because it may prevent VMS from failing the drive over to the secondary HSC.

EXE$MOUNTVER sends status messages to OPCOM. Because there are cases when mount verification messages are needed at the operator console and OPCOM might not be able to provide them, mount verification also sends special messages with the prefix %SYSTEM–I–MOUNTVER to the operator console, OPA0.

3.1.12.2 VMS Problems of "Why a Drive Mount Verifies"

VMS calls EXE$MOUNTVER if a drive loses contact with the system (for example, the controller sends a command to the drive but does not get a successful response back within the controller–specific timeout period). It is a process to verify the disk with which VMS reestablished contact is the same disk to which VMS was originally connected.

Sending the drive to mount verify state involves:

1. Host initiating an MSCP ONLINE command to the drive modifier followed by a GET UNIT STATUS (GUS).
2. Reading the home block and comparing the volume information (serial number, name, and so on) of the drive before VMS lost contact with it and after VMS reestablishes contact with the drive during mount verification.

The sequence is repeated until success or timeout. During the sequence, the drive port light is on and the ready light blinks slowly as the controller accesses the LBN block and the RCT for the media ID, effectively doing full–stroke seeks.

The MVTIMEOUT system parameter defines the time (in seconds) that is allowed for a pending mount verification to complete before it is aborted. This dynamic parameter should be set to a reasonable value for the typical operations at the site.

NOTE

Do not use values less than the recommended default 600 seconds (10 minutes).

After a mount verification times out, the pending and future I/O requests to the volume fail. You may try to execute the DISMOUNT/ABORT command, which allows a subsequent mount to be successful if the MV timer has previously expired. In some extreme cases, drive failures may require the reboot of the controller or the system.

Entry to and exit from mount verify are time stamped. VAXcluster time–stamps may vary across the various cluster nodes due to differences in the Time of Year (TOY) clocks and the initial clock times. Slight variations in time stamps do not indicate multiple drive or controller failures causing mount verification, but rather one drive or controller failure causing every node to enter mount verification at their own locally specified time. The following reasons may explain why a drive enters mount verification:

- MANUAL intervention, such as:

- Change the state of the switches on the OCP of the disk
- Accidental release of the port buttons
- Accidental spin down of the drive
- Loss of power, trip the circuit breaker
- Loose cables
- Change volumes
- Change the unit number

Those reasons result in mount verification messages and are to be expected. They may also result in mount verification timeout messages if the timeout period is exceeded.

- **SYSTEM** level causes, such as:
  - Planned failovers
  - Mount verification timeout is incorrectly set (default is 600 seconds, less than 120 seconds is not appropriate).
  - Cluster state transition messages for drives mounted to the cluster. However, mount verification timeout messages might indicate a problem with the host, controller, or drive.
- **CONTROLLER** level causes, such as:
  - Controller failures, including an HSC crash
  - Unplanned failovers
  - A LAST–fail packet from an xDA/DB controller occurred shortly after the mount verification, meaning the controller faulted/initialized as well.

Those reasons result in mount verification messages associated with multiple drives. They could also result in mount verification timeout messages.

3.1.12.3 Actions

Locate the source of the controller symptoms. Refer to Data Collection Techniques in Chapter 2 to obtain the status/event codes, fault codes, and so on. Analyze and troubleshoot those errors according to Chapter 3. Refer to Section 3.5, Device Isolation, page 48, as appropriate.

By noting the time duration of the mount verification and other circumstances surrounding the mount verify status, you can determine some valuable troubleshooting information. Ask yourself the following questions:

How long did the mount verify take?

Less than MVTIMEOUT and the drive eventually succeeded.

A few seconds, implying a glitch or a recoverable fault.

Did it appear on another controller after the mount verification? If so, it could be a port–related problem.

Thirty seconds to a minute to remount probably means the drive was spun down and had to be spun back up. Was this due to a drive fault? Did the drive run its spinup diagnostics error free?

Infinite time probably means that along with the drive disappearing, it also:

- Changed its media_id
- Is a different drive
- Continually failed its spinup diagnostics
- Contained a hard fault

Were the mount verification messages associated with specific drives?

VMS does not log errors during the mount verify process, although it may log some before or after, depending on how the drive failed.

Were any errors logged to the host or HSC console log before or after the mount verify?

Do any drives that are nonexistent appear, characterizing a unit select problem?

Were all drives that failed on the same controller, K.sdi module, or controller port?

Does the error always appear on the same physical drive?

Did the drive see a fault during this period? (Examine the drive internal error log for error information.)

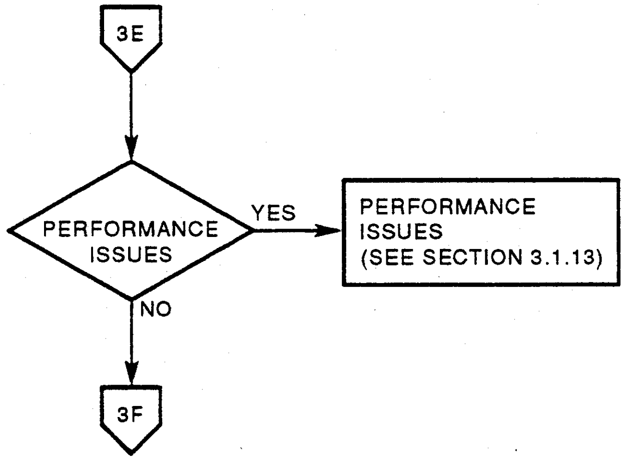
Mount verification messages are often associated with a single drive. Also, mount verification timeout may occur.

DRIVE level causes

- Intermittent OCP hardware
- SDI cables and connections
- Drive faults
- Drive communication (SDI) problems

Actions – Locate the cause of the drive problem. Use the data collection techniques to obtain any status/event codes, LED codes, drive internally logged information, and so on. Analyze and troubleshoot any errors found, according Section 3.5, Device Isolation, page 48, as appropriate.

Figure 11: Flow – Data Analysis and Repair (3E)



CXO-2397A

3.1.13 Performance Issues when No Errors Are Logged

Customer complaints of disk performance can take a fair amount of analysis. Often the performance complaints are quite subjective. Following is a list of questions that may help analyze performance complaints.

1. **QUESTION:** Do the performance issues relate to all or most of the disks?

   ANSWER: If most or all the disks are affected, ensure that the system parameters meet the suggested guidelines. Cluster size of disks, working set size parameters, paging parameters, and ACP/XQP–related parameters can all affect performance.

2. **QUESTION:** Do performance problems occur during image activation (when a large application program is initially started)?

   ANSWER: Many layered products require some time to fully activate. This is not a disk problem.

3. **QUESTION:** Is the performance problem noticed by users of the same image, layered product, or file on the (same) disk?

   ANSWER: If the disk is attached to a local controller (UDA/KDA/KDB) but is a VAX node member in a cluster where there is at least one HSC in the cluster, then request that the file/image/layered software product be moved to a disk on the HSC. Local serving of disks creates bus, VAX, and I/O overhead, which impacts performance.

4. **QUESTION:** Is the performance problem noticed by users of a file/image/layered product that resides on the same disk as the swap and page files?

   ANSWER: If so, request the system manager to monitor paging and swapping activity. High page/swap rates decrease VMS response and create an I/O bottleneck for the page/swap disk. Request the file/image/layered product be moved to another disk.

In addition to setting system parameters, this area of the architecture (hardware related) can contribute to loss of performance. These include nonprimary replacements in a critical file or directory structure.

Examples include:

- Nonprimary replacement in VMS disk [000000]INDEXF.SYS.
- Nonprimary replacement in a directory file that is frequently used.

NOTE

VMS uses virtual block file structures, not logical blocks. Virtual Block Numbers (VBNs) do not correlate to LBNs. To correlate an LBN to the affected file, you should understand the operating system file structure, such as VMS ODS–2 or contact support personnel. It is a very complicated procedure to identify affected files within ODS–2.

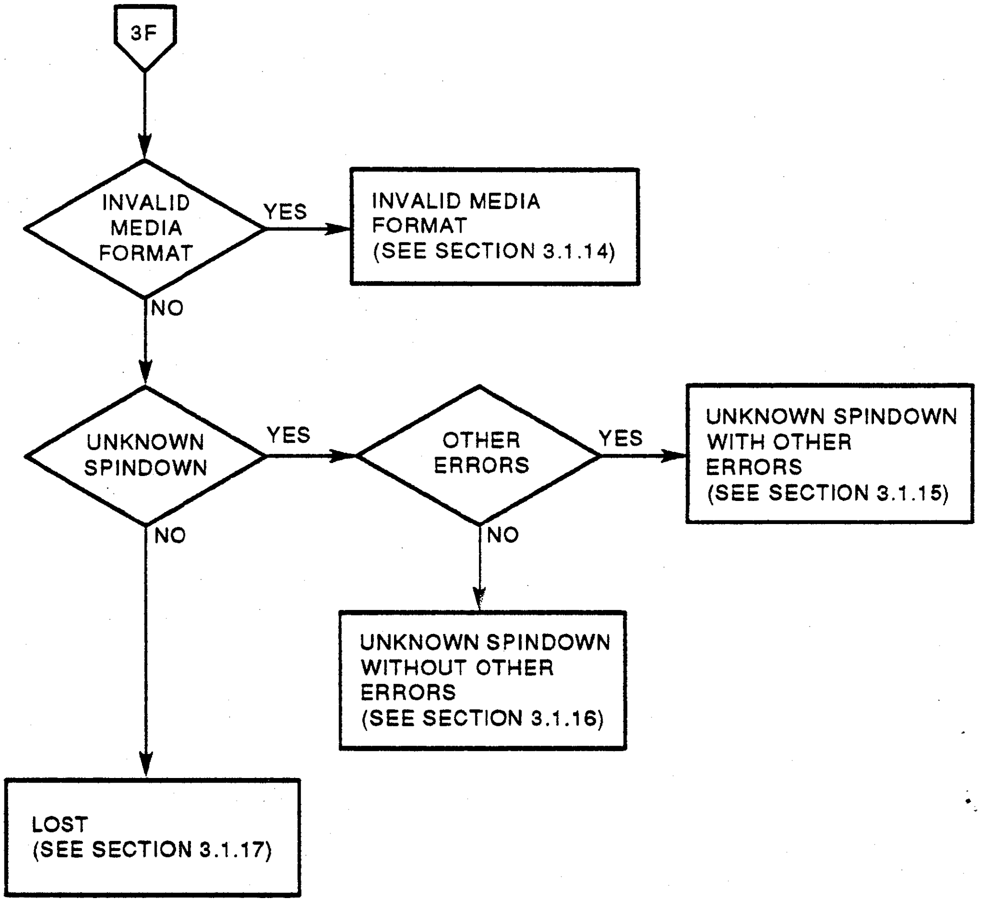
The two examples are files that may affect the perceived performance of a disk. However, the location of a block of data within a file and how the operating system is set up has an equal effect on nonprimary replacement, which in turn impacts system or disk drive performance.

A nonprimary replaced block in the INDEXF.SYS of a disk could be very significant if it is in the front of the file. However, if it is the last block within the file, it might not have as large an impact on system performance.

A nonprimary replacement in a block within SYS.EXE that is loaded once by VMS into memory (at startup) and stays resident in memory has no effect on performance. However, if the block is within a portion of SYS.EXE that is frequently brought in by VMS, it could impact performance. A solution is for the system manager to increase the VMS working set size.

A block within the swap or paging file that is nonprimary replaced generally does not have much impact. If the system is doing enough paging and swapping to notice the occurrence of nonprimary replacements, the real problem may be with the user or system working set size. Have the system manager adjust system parameters around paging and swapping and see if performance improves.

Figure 12: Flow – Data Analysis and Repair (3F)



CXO-2398A

3.1.14 Invalid Media Format

This message indicates the detection of a corrupted Format Control Table (FCT) on the media while trying to complete a mount operation. The media will most likely need to be reformatted to resolve this condition.

First, find the cause of the corruption. Plus, consider taking some actions to ensure that a hardware problem does not exist. For example:

- Look for other possible causes, using the data collection techniques. Troubleshoot any hardware errors found first.
- Verify the hardware data path, using EVRLF, EVRLG, ZUDH, ZUDI, NAKDAx, ILDISK, or drive internal R/W diagnostics.
- The problem may have been due to an incomplete format operation. In this case, a reformat will be the only resolution required.

Contact the appropriate support resources if additional assistance is needed.

3.1.15 Unknown Spin Downs with Other Errors

If other errors are present, troubleshoot those first. Go back to the beginning of this flowchart if needed.

3.1.16 Unknown Spin Downs Without Other Errors

Verify that no other errors or fault indications exist. The following are possible causes for unknown spin downs:

- Duplicate unit numbers
- Various host software programs or operating system commands (such as in VMS DISMOUNT without the /NOUNLOAD modifier)
- Invalid Media Format, see Section 3.1.14
- Power

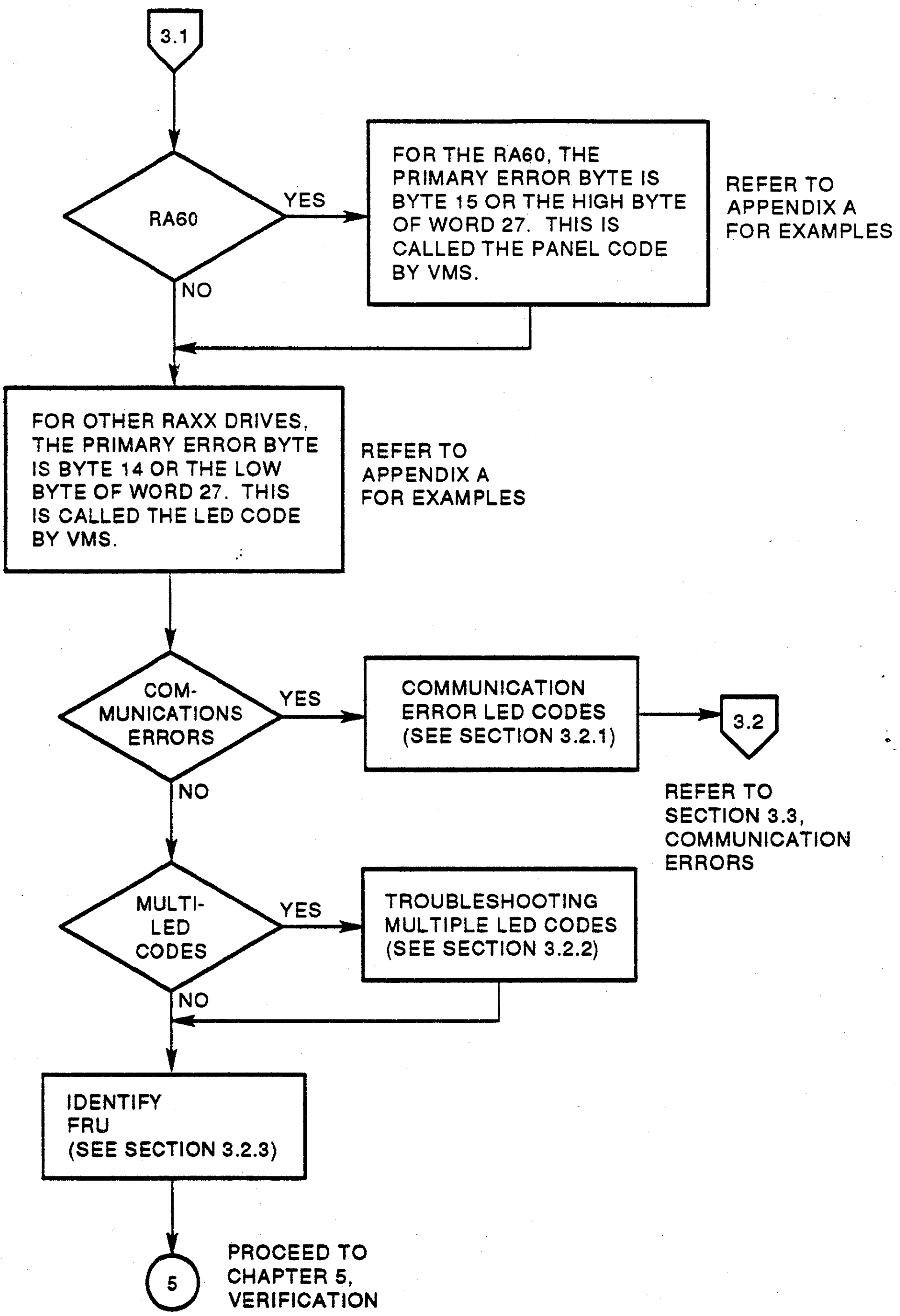
NOTE

Most RAxx drives will NOT spin back up after a power failure until the host software initiates the drive or the user toggles the run switch.

3.1.17 Lost

If the problem appears obscure, if too much time has been spent trying to isolate this problem, or if you have insufficient information from the data collection process, use the support resources available. Digital Field Service should operate with/by the guidelines of MAP within the respective areas.

Figure 13: Flow - Drive-Detected Errors (3.1)



CXO-2399A

SECTION 3.2

3.2 Drive-Detected Errors

3.2.1 Communication Error LED Codes

Communication errors most often cause other errors in the subsystem. Resolve communication errors first. Then continue troubleshooting any errors that are still occurring.

STATUS/EVENT codes of EB with LED codes of:

- In the RA60: 9C, A2, A3, A4, A5, A6
- In the RA80: 07, 08, 09, 0A, 0B, 0C, 1F, 20, 21, 22
- In the RA81: 07, 08, 09, 0A, 0B, 0C, 1F, 20, 21, 22, 41
- In the RA82: 07, 08, 09, 0A, 0B, 0C, 1F, 41, 4F
- In the RA70: 07, 08, 09, 0A, 0B, 0C, 0E, 17, 18, 1F, 41, 43, 44, 4F, ED
- In the RA90: 07, 08, 09, 0A, 0B, 0C, 0D, 0E, 10, 16, 17, 18, 19, 1A, 1F, 20, 21, 2A, 2B, 2C, 42

3.2.2 Troubleshooting Multiple LED Codes

When troubleshooting devices provides multiple LED codes, consider the following rules:

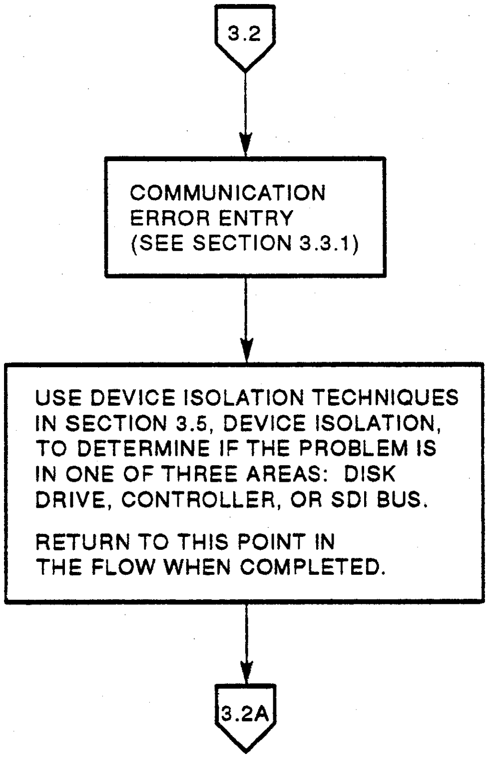
- Select error codes that occur during internal drive diagnostics. These should be given top priority.
- Select codes with the least FRU callout.
- Select codes occurring most often.
- Select codes with FRU(s) in common to most all the codes provided.

If in doubt, refer to the device service manual for details.

3.2.3 Identify FRU to Replace

As determined by the error codes received, identify the FRU for replacement. Refer to the device service manual for the correct procedures to replace and verify the identified FRU. After FRU replacement, proceed to Chapter 5, Verification, page 57.

Figure 14: Flow – Communication Errors (3.2)



CXO-2400A

SECTION 3.3

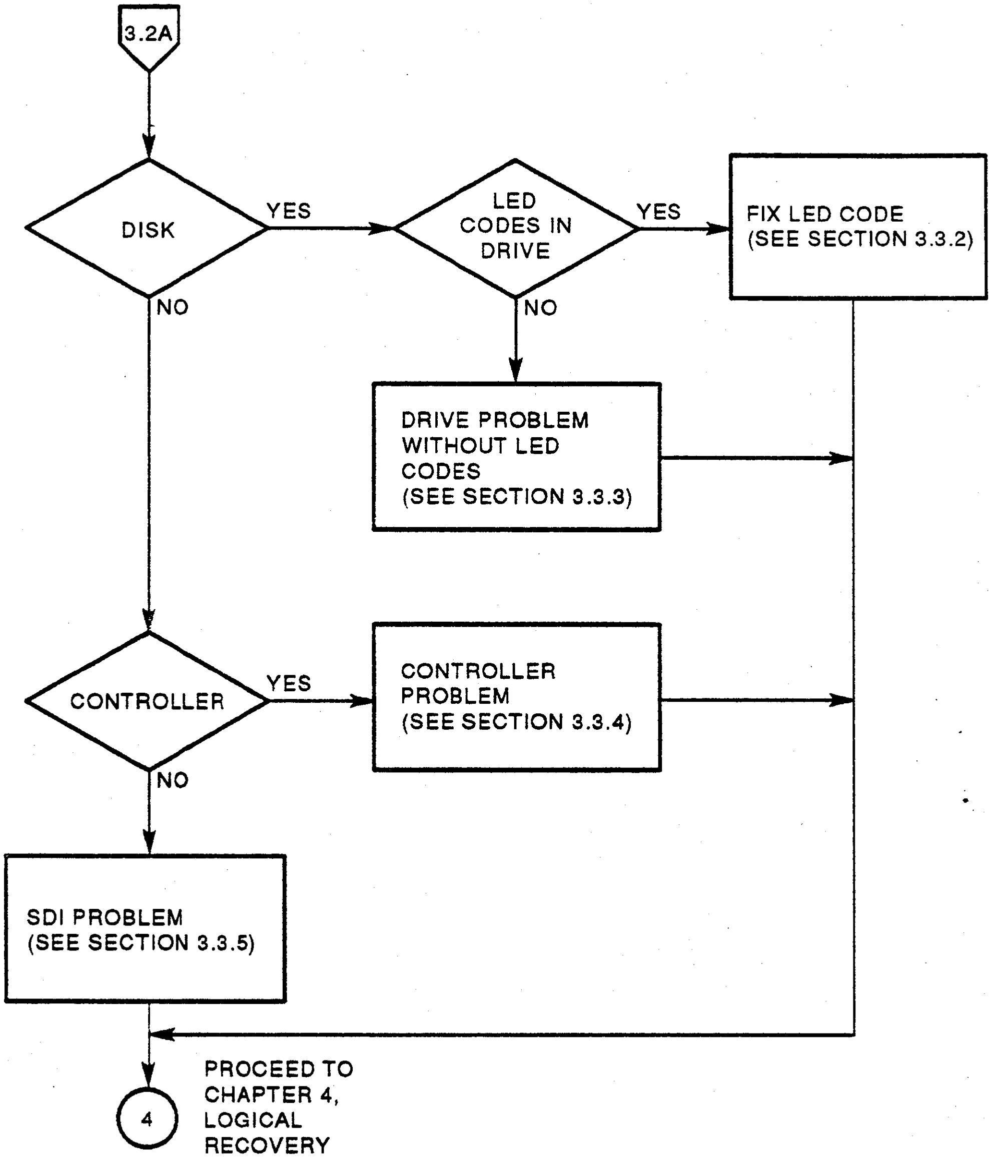
3.3 Communication Errors

3.3.1 Communication Errors Entry

You are here because of one of the following conditions:

- Status/event codes of 4B, 10B, 14B
- Excessive status/event codes of 1AB
- Status/event codes of EB with LED codes of:
  - In the RA60: 9C, A2, A3, A4, A5, A6
  - In the RA80: 07, 08, 09, 0A, 0B, 0C, 1F, 20, 21, 22
  - In the RA81: 07, 08, 09, 0A, 0B, 0C, 1F, 20, 21, 22, 41
  - In the RA82: 07, 08, 09, 0A, 0B, 0C, 1F, 41, 4F
  - In the RA70: 07, 08, 09, 0A, 0B, 0C, 0E, 17, 18, 1F, 41, 43, 44, 4F, ED
  - In the RA90: 07, 08, 09, 0A, 0B, 0C, 0D, 0E, 10, 16, 17, 18, 19, 1A, 1F, 20, 21, 2A, 2B, 2C, 42

Figure 15: Flow – Communication Errors (3.2A)



CXO-2401A

3.3.2 Disk Problem with LED Codes

You are here because drive–detected communication errors with drive LED codes indicate a drive problem. Troubleshoot the LED code and replace FRUs as per the drive service manual.

3.3.3 Disk Problem Without LED Codes

You are here because a controller STATUS/EVENT code indicates a drive problem. The problem could be in the following areas:

- Drive module with the SDI interface logic (for instance RA81 personality).
- Internal SDI cabling and connections.
- If STATUS/EVENT codes of 14B, 4B, or excessive 1AB, also include the module containing the microcode or microprocessor that controls the SDI functions as part of the FRU selection (e.g., in the RA81, this would be the microprocessor).

The drive service manual may provide additional information on identifying the FRUs associated with SDI communication logic.

3.3.4 Controller Problem

The problem has been isolated to the controller. The problem could be in one of the following areas:

- Controller module containing the SDI interface logic.
- Internal controller SDI cabling and connections.

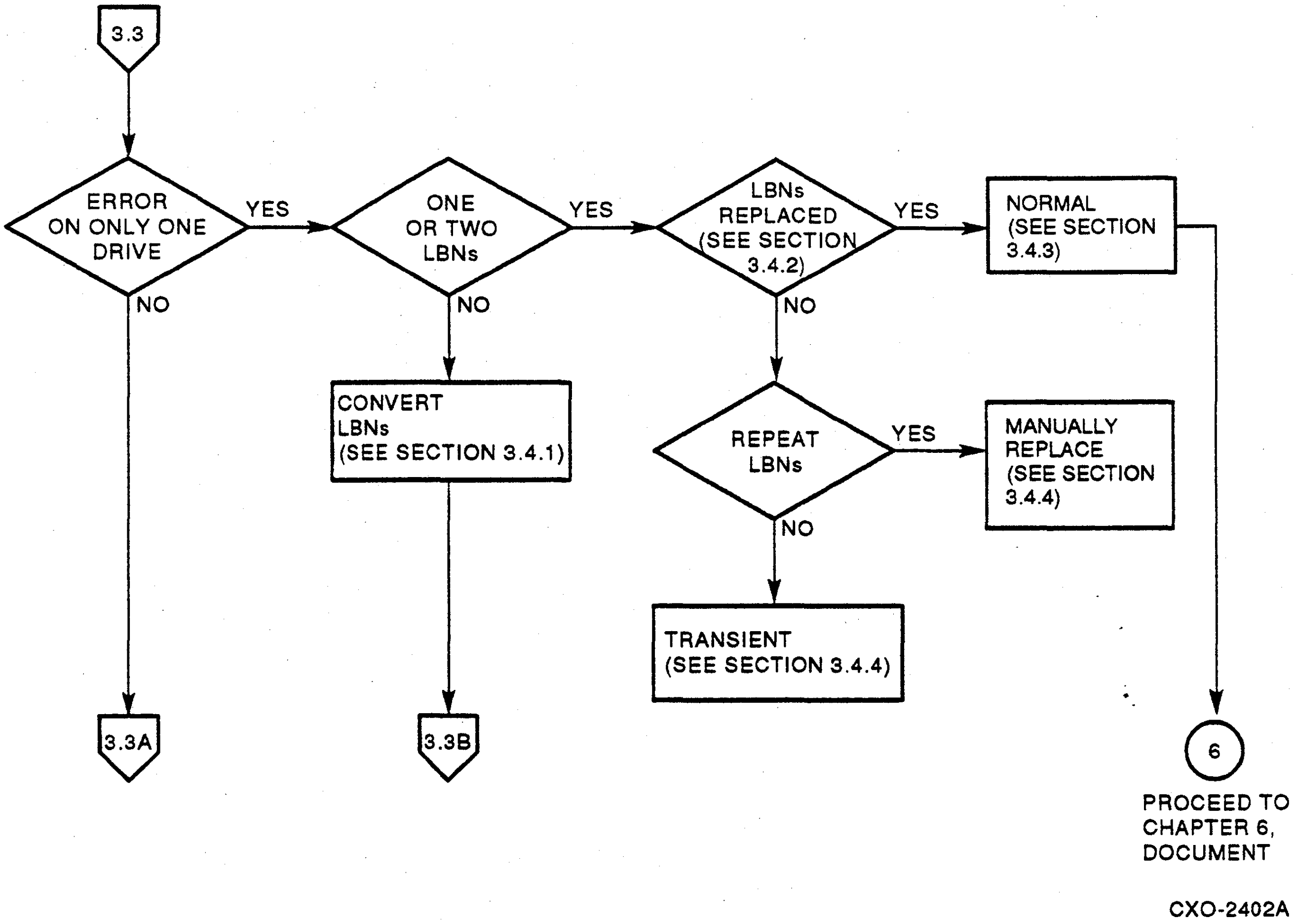
The controller service manual may provide additional information on identifying the proper FRU for replacement.

3.3.5 SDI Problem

The problem has been isolated to the external SDI. The problem could be in one of the following areas:

- External SDI cables
- SDI bulkhead/transition connections
- Poor grounding
- Environment

Figure 16: Flow – Data Errors (3.3)



SECTION 3.4

3.4 Data Errors

3.4.1 Convert LBNs

The conversion of LBNs to the physical drive characteristics allows further isolation of FRUs related to R/W transfer problems. This also allows the identification of media versus nonmedia problems. There are many online tools to assist with conversions. See Appendix B for a list of resource utilities that provide LBN conversions and Appendix C for the conversion formulas.

3.4.2 LBN Replaced

An additional event (BBR message) logged with a block being tested for replacement:

- Status/event code = 14 (Block replaced)
- Status/event code = 34 (Block not replaced)

NOTE

These are NOT errors but information entries reflecting the status of the BBR process. A block read causes a data error to invoke BBR (set the BBR flag). The block is tested. The block passes the test (possible transient error), status/event code = 34, or the block tested bad, status/event code = 14. Entries with the same LBNs can be associated with the BBR messages. Identify all entries with the same LBN to get the total picture.

You can use HSC VERIFY, DKUTIL, RAUTIL, or similar utilities to verify if the LBNs have been replaced.

3.4.3 Normal

Occasional LBN replacements are expected. This indicates a bad spot on the media. For future reference, document the LBNs that have been replaced.

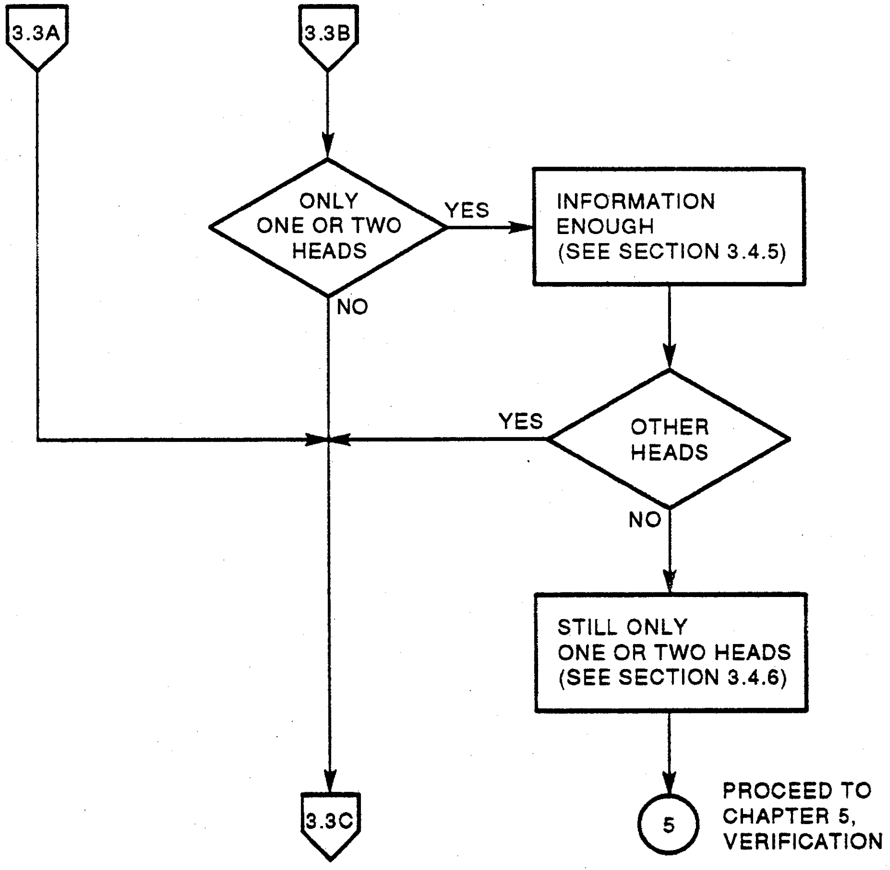
3.4.4 Transient/Manually Replace Repeating LBNs

Occasional transient blocks are acceptable for these products. Normally, they will get replaced (status/event = 14). LBN numbers that consistently recur within the host error log that are not replaced may be manually replaced with the utilities DKUTIL (HSC), EVRLK, ZUDLxx, or RAUTIL. This is a useful procedure for those blocks that are consistently reporting ECC/data errors.

This symptom can occur when the host BBR software does not utilize the user data as the pattern to test the suspect block. The block is initially flagged for replacement. The host executes a test of the block and finds nothing wrong with the block, does not revector, but restores the original data back into the block. The user then accesses the data again and may get another ECC error with severity to again invoke the BBR activity. An ECO made it a requirement to utilize user data as one of the test patterns when using the MSCP specification.

Blocks that are reporting consistent "header not found" problems or "positioner unintelligible" header problems will likely still report such a problem if the block is revectored because of the way a block is searched for in this architecture.

Figure 17: Flow – Data Errors (3.3A & 3.3B)



CXO-2403A

3.4.5 Enough Information

Since you are only here due to LBNs in the error log, you need to ensure visibility of all the LBNs with problems. Also, determine if the problems are isolated to one or two heads. The simplest technique is to perform an operation that reads all of the LBNs on the disk. The intent is to identify if all the information on the problem is known. If the problem is the result of a momentary failure (burst), the architecture is designed to handle these failures and no further actions should be needed.

Many system and subsystem utilities, commands, and diagnostics allow the Field Engineer to perform read–only functions, read/write I/O, scan, verify, and so on, that could provide additional data. Some of those tools are:

- ILDISK/ILEXER as appropriate
- EVRLK, ZUDL, BBR utilities (scrubbers) in verify mode
- NAKDAx (scrubber)
- RAUTIL
- EVRAE
- VMS command "$ ANALYZE/DISK/READ_CHECK DEVICE:"

3.4.6 Still Only on One or Two Heads

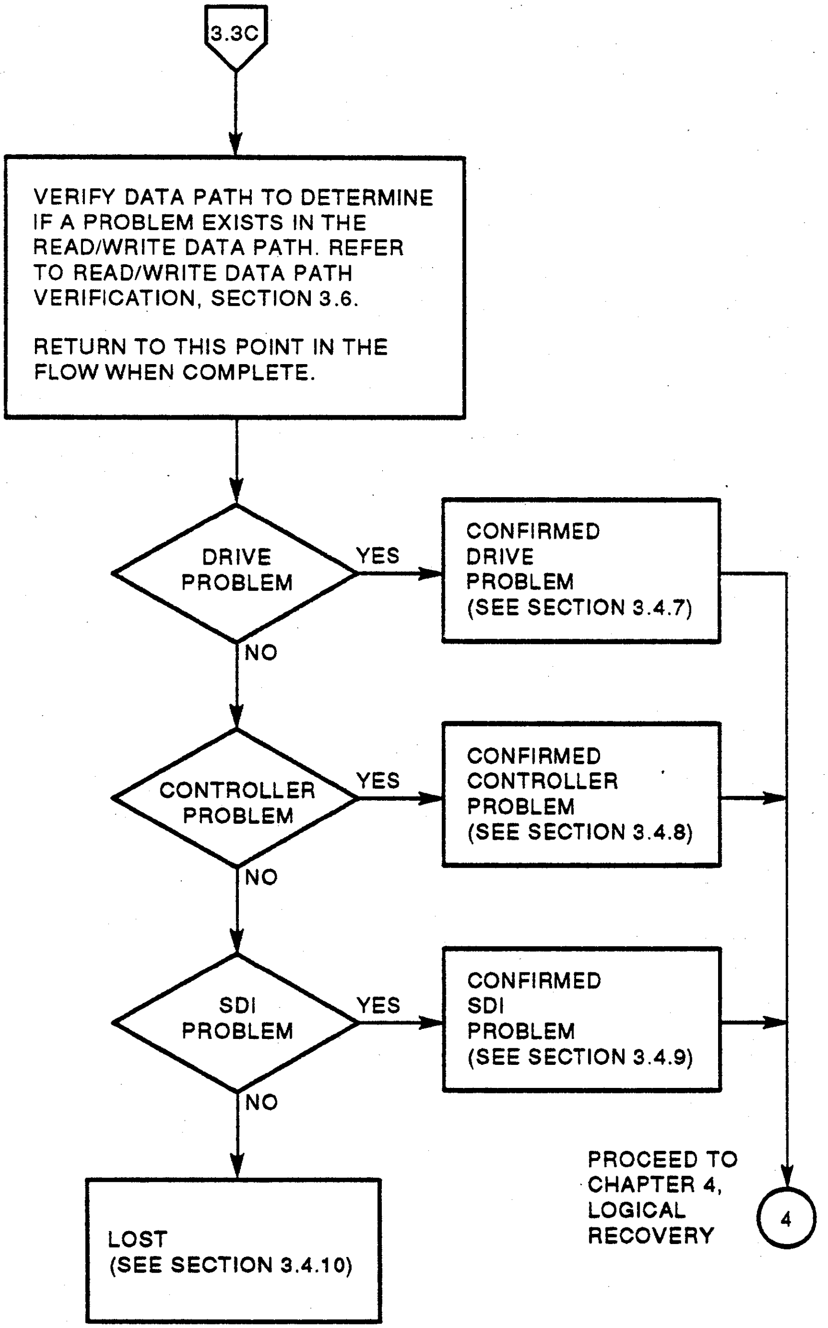
If

- The problem stabilizes (no further errors at this time), document and monitor.
- The symptoms from the previous history indicate a problem with the same one or two heads, replace the HDA.
- Continued testing or additional information results in a growing problem with the same one or two heads, replace the HDA.

NOTE

In the RA60, the HDA would equate to the heads or the media.

Figure 18: Flow – Data Errors (3.3C)



CXO-2404A

3.4.7 Confirmed Drive Problem

The problem has been confirmed to be a R/W data path problem in the drive. Most drive data path problems can be associated with the drive SDI logic, serial R/W data path, and head select logic. For example, in the RA81, the following items may cause problems:

- HDA ground brush assembly
- Read/write module
- Microprocessor module
- Personality module
- HDA
- Faulty spindle ground

Other less frequent items that could cause problems include the power supply, motor/brake assembly, worn belts, cabinet connections, servo module, and improper system/drive grounding. Knowledge of the drive is needed to determine the proper FRUs for replacement. The drive service manual may provide some assistance.

3.4.8 Confirmed Controller Problem

The problem has been confirmed to be a R/W data path problem in the controller. Most controller data path problems can be associated with the serial read/write data path, ECC generator, SDI logic, internal cables and connections. For example:

- In the UDA50 — M7486
- In the HSC50 — K.sdi

Other less frequent items that could cause problems include power supplies, cable connections, and improper grounding. Knowledge of the controller is needed to determine the proper FRUs for replacement. The controller service manual may provide some assistance.

3.4.9 Confirmed SDI Problem

The problem has been confirmed to be a R/W data path problem in the external SDI bus. Most external SDI bus problems can be associated with bulkhead/transition connections and the external SDI cables. Other less frequent items that could be problems are power, improper grounding, and poor environment.

The hardware that causes communication errors is the same hardware that will cause data path errors.

If any LED codes or status/event codes indicate communication errors or drive–detected errors and are encountered during the verification or troubleshooting process, these errors should be resolved first.

Refer to Section 3.2, Drive–Detected Errors, page 37, or Section 3.3, Communication Errors, page 39.

Any of the above actions may result in a STATUS/EVENT code with more definition and should be used to isolate problems per the main flow of this document.

3.4.10 Lost

If the problem appears obscure, too much time has been spent trying to isolate this problem, or you have insufficient information from the data collection process, use the support resources available. DIGITAL Field Service should operate with/by the guidelines of MAP within the respective areas.

SECTION 3.5

3.5 Device Isolation

Use the techniques in this section to isolate a problem to one of the following areas:

- A common drive
- A common controller port
- A common controller
- The SDI bus to include: SDI cables, bulkhead, internal cabling to the bulkhead, and grounding

Use these techniques

1. Perform a FAILOVER
2. Move the cable to an alternate drive
3. Move the cable to an alternate port at the controller
4. Move the cable to an alternate controller

Determine if the symptoms move to the alternate device or remain with the same (original) device.

Document your actions for clear traceability and to assure the isolation process has been completed (i.e., prevents confusion and getting lost).

EXIT with the device isolated and return to the flow chart.

SECTION 3.6

3.6 Read/Write Data Path Verification

You need to obtain enough evidence or data to prove or disprove that a data path problem exists by verifying the following symptoms. For example:

- Sufficient LBNs are translated to clearly prove that the errors are randomly distributed across several cylinders, sectors, and heads, thus proving a R/W data path problem exists in a drive.
- Sufficient data or errors have been logged to prove that a data path problem exists in a controller that is causing similar errors on multiple ports and/or multiple drives.

The read/write DATA PATH of a DSA subsystem consists of the following elements:

In the **CONTROLLER:**

- The logic in the controller containing the serial read/write data path, ECC generator, and the SDI logic
- Internal SDI cables and connections

In the **SDI** bus:

- Bulkhead/transition connections
- External SDI cables

In the **DRIVE:**

- The internal SDI cables and connections
- The logic in the drive containing the SDI, logic, serial read/write data path (R/W encode/decode, head select logic, and so on)

The following symptoms are common to many read/write data path problems:

- Read/write data path problems in **CONTROLLERS** often exhibit the following symptoms:
  - Errors for any drive connected to a port.
  - Errors for any drive connected to all ports.
  - Transmission errors.
  - Variety of drive LED codes.
  - Variety of status/event codes.
  - ECC errors, header errors, and so on, where the physical translation of logical blocks result in:
    - Random cylinders.
    - Random sectors.
    - Random R/W heads, usually errors on more than three heads.
  - Excessive status/event codes of 34. Usually a large number of entries with a code of 34 strongly indicates an intermittent or transient data path problem rather than a problem with the media. Status/event code 34 indicates LBNs were flagged for BBR testing but not replaced because the LBNs were marginal (transient errors).

- Read/write data path problems with a **DRIVE** often exhibit the following symptoms:
  - Transmission errors.
  - Variety of drive detected LED codes.
  - Variety of controller status/event codes.
  - ECC errors, header errors, and so on, where the physical translation of logical blocks result in:
    - Random cylinders.
    - Random sectors.
    - Random R/W heads, usually errors on more than three heads.
  - Excessive status/event codes of 34. Usually a large number of entries with a code of 34 strongly indicates an intermittent or transient data path problem rather than a problem with the media. Status/event code 34 indicates LBNs were flagged for BBR testing but not replaced because the LBNs were marginal (transient errors).

Refer to Section 3.5, Device Isolation Techniques, page 48, to assist in troubleshooting, if needed.

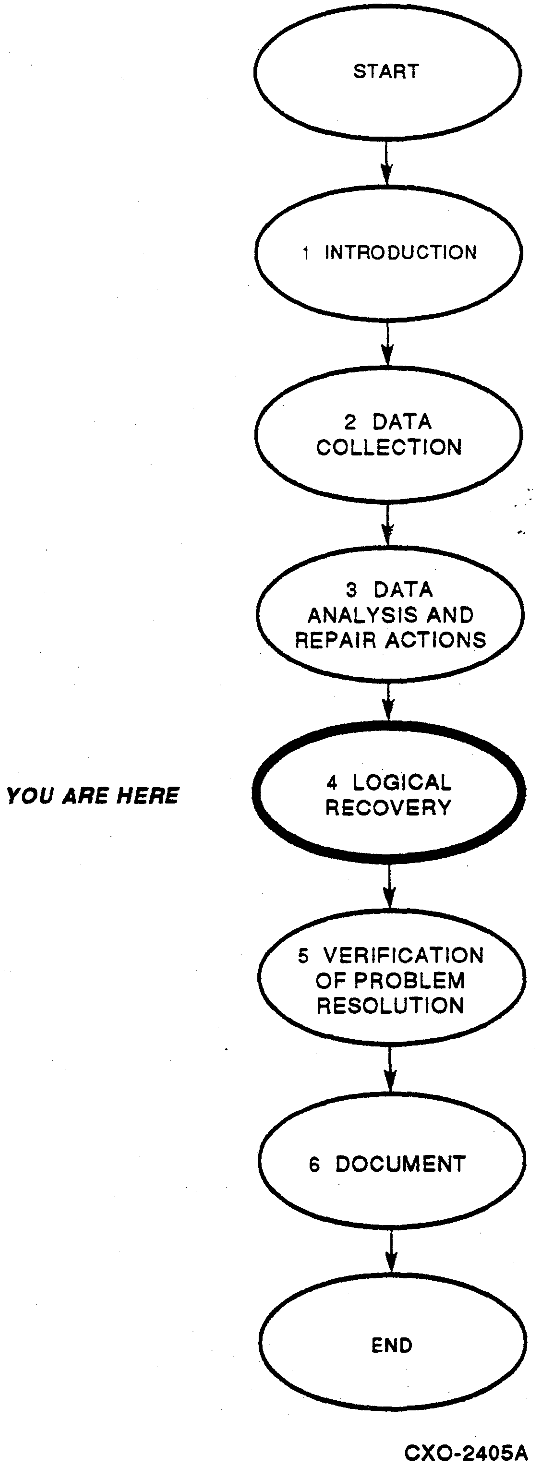
Obtaining the necessary information will often require you to employ one or more of the following techniques:

- Collect all available information from the error log.
  - Include data from a previous service call.
  - Include error log data that results from any of these techniques.
- Collect the HSC console information.
- Collect error information that results from any of the utilities or diagnostics used during this process.
- Use HSC VERIFY as appropriate.
- Use system and subsystem utilities, commands, and diagnostics that allow read only functions, R/W I/O, scan, verify, and so on.
  - ILDISK/ILEXER
  - EVRLK, ZUDL, BBR utilities (scrubbers) in verify mode
  - NAKDAx (scrubber)
  - RAUTIL
- Use drive internal diagnostics as appropriate.
- Use SDI loopback testing as appropriate.

Use a technique or process to reproduce the problem.

Once you have determined that a data path problem has been isolated to a controller, disk, or SDI path problem, return to the flow chart.

Figure 19: Flow Map – Logical Recovery



CHAPTER 4

LOGICAL RECOVERY

4.1 Overview

Read/write data path problems, transmission errors, and other hardware subsystem problems often affect everything that is read or written to the media, including files and structures. This corruption could render the hardware to "appear" as if the hardware is still broken. Logical recovery is the repair of those corrupted files or structures.

As such, no separate flows exist to further describe logical recovery. Logical recovery is a single step in the main flow, with notes describing the suggested actions to pursue. Not all problems result in corruption of structures nor require the need for logical recovery. The flow continuation pointers from Chapter 3, Data Analysis and Repair Actions, will skip this section when it is appropriate to do so.

NOTE

> **At times in the logical recovery process, reformatting is the recommended solution. It is essential that a known good copy of the data (file or volume) is available before the reformatting is done.**

Tools that can be used to perform logical recovery are:

- HSC VERIFY
- EVRLK
- ZUDL
- DKUTIL
- RAUTIL
- Formatters (HSC format, EVRLB, ZUDK, and so on)

See Appendix B for a description of what these tools can do.

4.1.1 Commonly Known Recovery Techniques

For files containing forced errors or EDC errors:

- If only a few files are affected, replace the files with known good copies of the files. Also, delete the damaged files with DELETE/ERASE.
- If several files are affected, consider restoring the volume from a known good backup copy of the volume.

The following example shows VMS messages that result from reading a file containing bad EDC as still written on the disk, even after the controller is repaired.

```
$ TYPE FILE.DAT
%TYPE-W-READERR, error reading $5$DUA230:[LOCAL.TEST]FILE.DAT;2
-RMS-F-RER, file read error
-SYSTEM-F-CTRLERR, fatal controller error
```

NOTE

Use a command under the operating system that will read all the host-allocated blocks on the disk to identify all the files with damage. For example, under VMS use the following command:

```
$ ANALYZE/DISK/READ device_name:
```

A file containing one forced error read many times will result in many forced errors being reported.

```
$ TYPE FILE.DAT
%TYPE-W-READERR, error reading $5$DUA230:[LOCAL.TEST]FILE.DAT;2
-RMS-F-RER, file read error
-SYSTEM-F-FORCEDERROR, forced error flagged in last sector read
```

4.1.2 Media Format and Replacement Errors

Errors that continue to produce any of the following status/event codes or the equivalent error messages should be resolved by reformatting the media and then restoring the volume from a known good backup copy of the volume:

45, 54, 74, 94, B4, D4, C5, 105, 125, 400

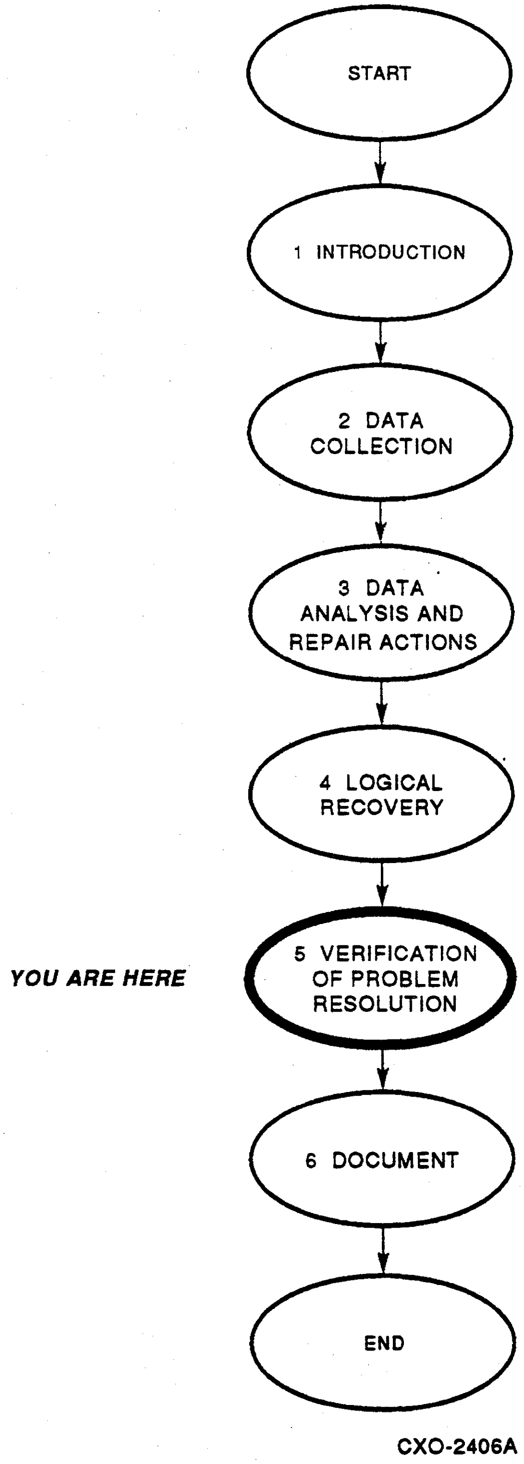
4.1.3 Excessive BBR (Status/Event Code = 14)

Read/write data path problems may cause the replacement of a high number of good blocks. This may lead to logical fragmentation of the disk. If this happens, the number of blocks in the RCT recorded as revectored differs substantially with FCT information. As an example, the RCT may show a doubling of replaced blocks occurring over a short period of time. If you have repaired a data path problem that has caused an excessive number of blocks to be replaced, consider reformatting the media and restoring the volume from a known good backup copy of the volume. Use EVRLB, NAKDAx, or ZUDKxx to reformat the disk and recover those good blocks. **Back up the user data *before* executing the format.** This can be done at the customer's convenience.

When all appropriate logical recovery actions have been performed, proceed to Chapter 5, Verification, page 57.

Chapter 5
Verification

Figure 20: Flow Map - Verification of Problem Resolution



CHAPTER 5

VERIFICATION

5.1 Device Verification

The following list can be used to determine the device level verification:

- Verify the new FRU as recommended in the device service manual.
- If the replacement of an FRU results in the same errors in the device, restore the original FRU.
    - This returns a probable good part that has had extensive testing (runtime).
    - This will reduce the probability of inducing a new problem over the long term.
- Run the device internal diagnostics, if appropriate.
- Verify that no new symptoms exist (no DOA).

5.2 Original Error Symptoms

Verify that the original error symptoms have been resolved.

- Event codes
- LED codes
- Fault indicators
- An operation that would reproduce the original symptoms

5.3 Verify the Problem Is Resolved to the Customer's Satisfaction

- Use a specific operation to verify the problem is resolved.
- Verify the customer symptoms are resolved.
- Use input received from the customer in the Chapter 2, Data Collection, page 3, as the exit criteria.

5.4 Verify That No New Problems Have Been Induced

New problems may be introduced by defective spares. Try an alternate spare if this seems to be the case. In some instances involving "multiple" problems, additional symptoms may appear only after more critical problems are resolved. Refer to Data Collection (Chapter 2, page 3) with the "new symptoms." If the same symptom persists, return the original FRU and try the next FRU per the appropriate service manual recommendation. Use the Data Collection step to assure you understand all the original symptoms.

5.5 Verify That No Residual Problems Remain

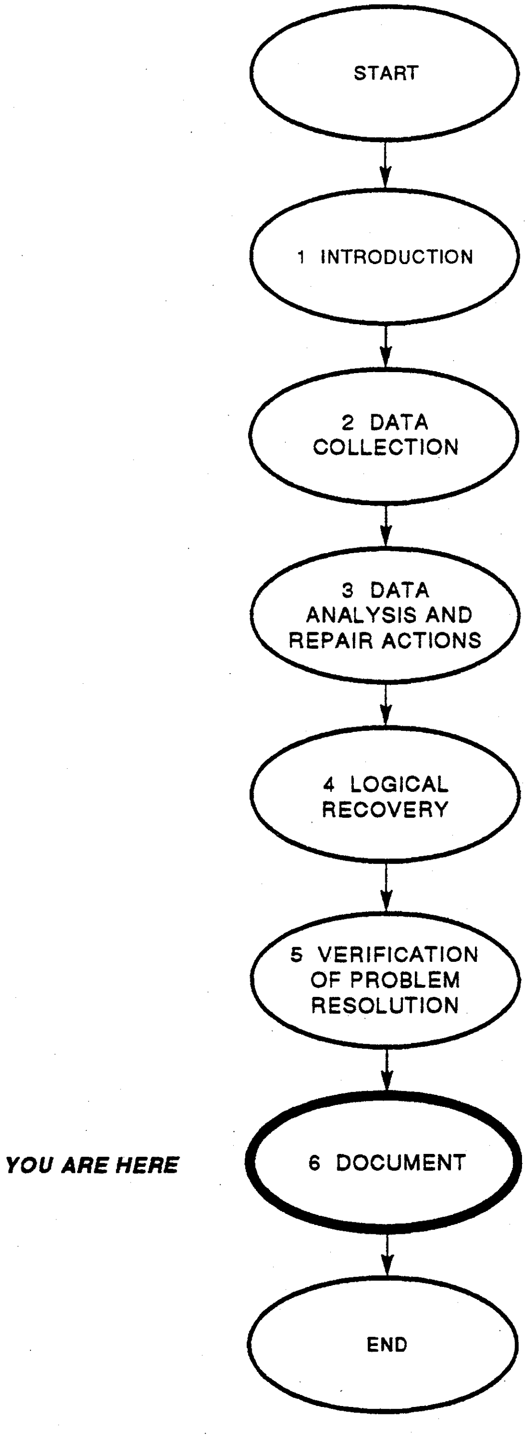
In the case of data type problems, verify that there are no residual "media" type problems after the original problem has been resolved. This may indicate the need for reviewing Chapter 4, Logical Recovery, page 53.

Many system and subsystem utilities, commands, and diagnostics exist for reading the blocks on the media. Refer to Appendix B, Utilities and Resources, for a list of some of these utilities.

If lost, or if the problem appears obscure, or if too much time has been spent trying to isolate this problem, utilize the support resources available. Digital Field Service should operate with/by MAP guidelines within the respective areas.

Proceed to Chapter 6, Document, page 61.

Figure 21: Flow Map – Document



CXO-2407A

CHAPTER 6

DOCUMENT

6.1 Overview

Documenting your actions is an important step in effective troubleshooting and problem resolution, which is often overlooked or skipped. Don't treat this step lightly. Take the time to document your steps.

Even though you may not forget what has happened, you may not be the next person to troubleshoot this system. How would you feel if you worked on a repeating problem with no information available from your predecessor? How does the customer feel when his/her system is still faulty because redundant actions failed to resolve a problem? Those situations occur due to lack of sufficient documentation to maintain "control" of the problems.

Document your actions during the service call in both of the following ways:

- LARS via CHAMP
- The log in the site guide

This information can be used in the following ways:

- To provide a history of the system for the next service call
- To verify the existence of new symptoms
- To isolate intermittent problems
- To determine the next step or FRU to change when troubleshooting an intermittent problem
- To determine if there are any repeat calls for this problem

If the service call resulted in changing the subsystem configuration (including device addresses), then note those changes in the *Site Management Guide* for future reference.

APPENDIX A

DRIVE-DETECED ERRORS

The following example shows how the HSC displays a drive–detected error. This example illustrates the location of the MSCP status/event code and the locations containing the master error code for the various types of disk drives.

```
ERROR-E Drive detected error at 8-apr-1986 15:11:44.37
        Command Ref #     00000000
        RA82 unit #       66.
        Error Flags       40
        Event             00EB --> MSCP STATUS/EVENT code                     <--
        Request           1B
        Mode              00
        Error             80
        Controller        00
        Retry/fail        00
        Extended Status   0C
                          0B
                          00
                          00
                          00
                          0C--> Master drive error code for RA70/80/81/82/90 <--
                          30--> Master drive error code for RA60              <--
        Requester #        7.
        Drive port #       0.
ERROR-I End of error.
```

The following examples illustrate how the same error would appear in a VMS formatted error log.

```
V A X / V M S        SYSTEM ERROR REPORT      COMPILED  8-APR-1986 16:41
                                                               PAGE   7.

**************************** ENTRY       3. ****************************
ERROR SEQUENCE 83.                                LOGGED ON SID 01380A4F

ERL$LOGMESSAGE ENTRY   8-APR-1986 15:11:44.37
                       KA780  REV# 7.  SERIAL# 2639.  MFG PLANT 0.

I/O SUB-SYSTEM, UNIT _HSC007$DUA66:

      MESSAGE TYPE        0001
                                    DISK MSCP MESSAGE
      MSLG$L_CMD_REF  00000000
      MSLG$W_UNIT         0042
                                    UNIT #66.
      MSLG$W_SEQ_NUM      01BC
                                    SEQUENCE #444.
      MSLG$B_FORMAT         03
                                    "SDI" ERROR
      MSLG$B_FLAGS          40                              * *************
                                    OPERATION CONTINUING -->* MSCP STATUS * <--
      MSLG$W_EVENT        00EB                              * /EVENT code *
                                    DRIVE ERROR             ***************
                                    DRIVE DETECTED ERROR
      MSLG$Q_CNT_ID   0000F807
                      01010000
                                    UNIQUE IDENTIFIER, 00000000F807
                                    MASS STORAGE CONTROLLER
                                    HSC70
      MSLG$B_CNT_SVR        02
                                    CONTROLLER SOFTWARE VERSION #2.
      MSLG$B_CNT_HVR        00
                                    CONTROLLER HARDWARE REVISION #0.
      MSLG$W_MULT_UNT     0050
      MSLG$Q_UNIT_ID  00000108
                      020B0000
                                    UNIQUE IDENTIFIER, 000000000108
                                    DISK CLASS DEVICE
                                    RA82
```

```
 V A X / V M S        SYSTEM ERROR REPORT      COMPILED  8-APR-1986 16:41
                                                                PAGE    8.
       MSLG$B_UNIT_SVR   01
                                   UNIT SOFTWARE VERSION #1.
       MSLG$B_UNIT_HVR   0F
                                   UNIT HARDWARE REVISION #15.
       MSLG$L_VOL_SER   03C769A2
                                   VOLUME SERIAL #63400354.
       MSLG$L_HEADER    00000000
                                   LBN #0.
                                   GOOD LOGICAL SECTOR
       MSLG$Z_SDI       REQUEST            1B
                                   RUN/STOP SWITCH IN
                                   PORT SWITCH IN
                                   LOG INFORMATION IN EXTENDED AREA
                                   SPINDLE READY
                                   PORT A RECEIVERS ENABLED
         MODE             00
                                   512-BYTE SECTOR FORMAT
         ERROR            80
                                   DRIVE ERROR
         CONTROLLER       00
                                   NORMAL DRIVE OPERATION
         RETRY            00
                                   0. RETRIES LEFT
CONTROLLER OR DEVICE DEPENDENT INFORMATION
                         ******************************************************
       LED CODE       C0 * --> Master drive error code for RA70/80/81/82/90<--*
       PANEL CODE     30 * --> Master drive error code for RA60            <--*
       LAST OPCODE    0C ******************************************************
                                   RUN
       PORT IMAGE         0B
                                   PORT B RTDS ENABLED
                                   PORT A RTDS ENABLED
                                   PORT A ENABLED
```

```
V A X / V M S        SYSTEM ERROR REPORT      COMPILED  8-APR-1986 16:41
                                                                PAGE   9.
      CUR CYLNDR            0000
                                       CURRENT CYLINDER, #0.
      CUR GROUP               00
                                       CURRENT GROUP, #0.
      REQUESTOR               07
                                       REQUESTOR #7.
      DRIVE PORT              00
                                       DRIVE PORT #0.
```

APPENDIX B

RESOURCES AND UTILITIES

There are significant concerns about running standalone diagnostics in troubleshooting DSA subsystem problems. The troubleshooting strategy takes into account user availability of the system, as well as the subsystem and the failing device. These strategies provide accurate diagnosis, fault correction, and verification to minimize the impact on the user.

NOTE

For transient disk subsystem errors, the running of available host loaded diagnostics via an xDA controller *seldom* isolates the errors without the necessary long run times. This is a *serious* availability impact. Heavy emphasis must be placed on utilizing the available error logs (hardware/software) that exist onsite. A list of these standalone diagnostics are provided for your reference.

B.0.1 Standalone Diagnostics

Table 1: VAX and PDP-11 Standalone Diagnostics

VAX	PDP-11	Comments
EVRLB	CZUDK	Formatter
EVRLF	CZUDH	Tests 1, 2, 3 1=UNIBUS Interrupt (Address tests) 2=Execute drive resident tests 3=Disk function test (R/W to DBN's)
EVRLG	CZUDI	Test 4, disk exerciser
EVRLJ	CZUDJ	Test 5, UDA/KDA subsystem
EVRLK	CZUDL	BBR utility (scrubber)
EVRLL	CZUDM	Disk resident error log dump utility
EVRAE		MSCP subsystem exerciser

Table 2: MicroVAX Standalone Diagnostics

MDM	HSC	Comments
NAKDAx		MicroVAX, Tests 1–4, Format, BBR utility, etc.
	ILDISK (inline)	HSC50/70 equivalent to tests 1–3
	ILEXER (inline)	HSC50/70 equivalent to test 4

B.0.2 What Can the Available Resources Do?

The following briefly describes the special utilities that are referenced in this document. The DSA troubleshooting course provides training in most of these areas.

B.0.2.1 EVRLK/ZUDL (VAX/PDP-11 Utility)

- Provides a list of all replacements (RCT)
- Scans host areas of the media. If used in verify mode, identifies blocks with Forced Error and/or bad EDC
- Provides auto replacement (scrubber)
- Provides a feature to manually replace LBNs (UDA/KDA subsystems)

B.0.2.2 HSC Verify (HSC Utility)

- Scans all blocks in host area
- Identifies corruption in structures (i.e., RCT, FCT)
- Identifies logical blocks written with forced errors and/or bad EDC
- Provides a list of all replacements (RCT)
- Provides a list of factory replaced blocks (FCT)
- Verifies the consistency of the RCT

B.0.2.3 DKUTIL (HSC Utility)

- Provides a list of all replacements (RCT)
- Provides a list of factory–replaced blocks (FCT)
- Provides a feature to manually replace LBNs (HSC based subsystems)

B.0.2.4 RAUTIL (VAX/MicroVAX VMS Utility)

- Provides media analysis on a "per head" basis
- Provides media analysis based on RCT replacements
- Provides a list of all replacements (RCT)
- Verifies all replacements
- Provides auto replacement as needed (scrubber)
- Verifies the consistency of the RCT (KDA/UDA–based subsystems)
- Identifies logical blocks written with forced errors and/or bad EDC
- Scans all blocks in the host area
- Provides a feature to manually replace LBNs (KDA/UDA–based subsystems)
- System/host program that runs online under VMS

B.0.2.5 VAXSIM$LBN.COM (LBN.COM or BLOCK.COM)(VAX/MicroVAX VMS Utility)

- Provides logical translation of LBNs, RBNs, DBNs, and XBNs into physical cylinder, track, head, and sector

B.0.2.6 DSAERR (DSA301.EXE, DSA303.EXE)(VAX/MicroVAX VMS Utility)

- Simplifies VMS and error log entries and provides the key information required by this document
- Provides customized error log sorting

NOTE

DSA303.EXE or higher is required for VMS V5.0 or higher.

APPENDIX C

CONVERSION FORMULAS FOR RA60

C.1 LBN to Physical and Logical Parameters

```
LC                         LBN
Logical                  -------
Cylinder        =          BPLC                 =  LC.LC_Rem

GP                    .LC_Rem * BPLC
Group               ------------------
                =          BPG                  =  GP.GP_Rem

TK
Track           =     .GP_Rem * BPG
(Logical)           ------------------
                           BPPT                 =  TK.TK_Rem

S
Sector          =     .GP_Rem * BPT             =  (Round result to
(Logical)                                          nearest whole number)

CYL60           =          LBN                  =  CYL60.Rem
                        ---------                  (discard)
                        4 * BPPC

Physical
Cylinder        =      (4 * CYL60) + GROUP

Physical        =   LBN - (CYL60 * 4 * BPPC)    =  HEAD.Rem
Head                -------------------------
                              BPLC

SFI
Physical
Sector          =     (GP * GP_Offset) + S      =  X.SFI_Rem
from                ----------------------         (discard X)
Index                         PSPT

                =       SFI_Rem * PSPT         =  SFI
                                                  (Round result to
                                                  nearest whole number)
```

NOTE

Refer to appendix E for the specific codes to use with these formulas.

C.1.1 Quick Algorithm for RA60 Head

If you know the LBN (Logical Block Number), first determine the Logical Cylinder:

```
LBN
---   =   Logical Cylinder . Fraction (discard fraction)
BPLC


Logical Cylinder
----------------    =   XXX.YYY
   6 (heads)

 PHYSICAL HEAD   =   6 * (.YYY)

 BPLC  =  168 (16-bit packs)        Blocks Per Logical Cylinder
          152 (18-bit packs)
```

APPENDIX D

CONVERSION FORMULAS FOR RA70/80/81/82/90

D.1 LBN to Physical and Logical Parameters

```
PC                        LBN
Physical                -------
Cylinder       =         BPPC                 =  PC.PC_Rem

PH                   .PC_Rem * BPPC
Physical           ------------------
Head           =         BPPT                 =  PH.PH_Rem

GP
Group          =     .PC_Rem * BPPC
(Logical)          ------------------
                         BPG                  =  GP.GP_Rem

TK
Track          =     .GP_Rem * BPG
(Logical)          ------------------         =  TK.TK_Rem
                         BPPT

S
Sector         =     .TK_Rem * BPPT           =  S (Round to
                                                 nearest whole number)

SFI
Physical            (GP * GP_Offset) + S
Sector         =  ------------------------    =  X.SFI_Rem
from                     PSPT                    (discard X)
Index

               =      SFI_Rem * PSPT          =  SFI
                                                 (Round result to
                                                 nearest whole number)
```

NOTE

Refer to Appendix E for the specific codes to use with these formulas.

APPENDIX E

TABLE OF CODES FOR CONVERSION FORMULAS

Table 3: Conversion Formulas for RAxx Drives

Table of Values	Disk	16-bit	18-bit
Blocks (LBNs)per physical cylinder (BPPC)	RA60	252	228
	RA70	363	
	RA80	434	406
	RA81	714	644
	RA82	855	
	RA90	897	
Blocks (LBNs) per physical track (BPPT)	RA60	42	38
	RA70	33	
	RA80	31	28
	RA81	51	46
	RA82	57	
	RA90	69	
Blocks (LBNs) per group (BPG)	RA60	42	38
	RA70	33	
	RA80	434	392
	RA81	51	46
	RA82	57	
	RA90	69	
Group offset (GP_Offset)	RA60	16	15
	RA70	08	
	RA80	16	16
	RA81	14	12
	RA82	11	
	RA90	14	
Physical sectors per track (PSPT)	RA60	43	39
	RA70	34	
	RA80	32	29

Table 3 (Cont.): Conversion Formulas for RAxx Drives

Table of Values	Disk	16–bit	18–bit
	RA81	52	47
	RA82	58	
	RA90	70	
Blocks (LBNs) per logical cylinder (BPLC)	RA60	168	152

PDP-11 SCRUB Information Handout

1 PDP-11 Media Package Contents

```
PACKAGE CONTENT REPORT          (CYCLE: 132)     REPORT DATE: 09-Jun-87

****************************************************************************
                DIAGNOSTIC
MEDIA PACKAGE   PACKAGE      ECO          MAINTAINER      COMPONENT PART
IDENTIFIER      IDENTIFIER   HISTORY      TITLE /GROUP    NUMBER
****************************************************************************

BB-FF66G-YC

* CZUDX                      G        CZUDXG0 F.S. 1600 BPI MT

                HMXM         F0       CHMXMF0 XXDP V2 EXTENDED MON
                HMSM         D0       CHMSMD0 XXDP V2 RESIDENT MON
                HSAX         C0       CHSAXC0 XXDP V2 DIAG SUPR EXT
                HSAA         G2       CHSAAG2 XXDP V2 SUPR SML
                HUDI         D0       CHUDID0 XXDP V2 DIRECTORY UT
                HDDB         C1       CHDDBC1 XXDP V2 DB DRVR/BOOT
                HDDD         D0       CHDDDD0 XXDP V2 DD DRVR/BOOT
                HDDL         D0       CHDDLD0 XXDP V2 DL DRVR/BOOT
                HDDM         C0       CHDDMC0 XXDP V2 DM DRVR/BOOT
                HDDR         C0       CHDDRC0 XXDP V2 DR DRVR/BOOT
                HDDU         E0       CHDDUE0 XXDP V2 DRVR/BOOT
                HDDY         D0       CHDDYD0 XXDP V2 DY DRVR/BOOT
                HDLP         B0       CHDLPB0 XXDP V2 LP DRVR
                HDMM         C0       CHDMMC0 XXDP V2 MM DRVR/BOOT
                HDMS         C0       CHDMSC0 XXDP V2 MS DRVR/BOOT
                HDMU         E0       CHDMUE0 XXDP V2 MU DRIVER
                HUDA         B0       CHUDAB0 XXDP V2 DATE UTILITY
                HZDU         C0       CHZDUC0 XXDP V2 DU SIZER
                HUP2         G1       CHUP2G1 XXDP V2 UPDATE UTIL
                HUTE         F1       CHUTEF1 XXDP V2 XTECO UTIL
                HUXC         I0       CHUXCI0 XXDP V2 DECX11 CNF/LN
                HUSU         F0       CHUSUF0 XXDP V2 SETUP UTIL
                HUPA         E1       CHUPAE1 XXDP V2 PATCH UTIL
                HQHL         G0       CHQHLG0 XXDP V2 HELP FILE
                ZUDL         A0       CZUDLA0 BAD BLK REPLACE UTIL
                ZUDM         A0       CZUDMA0 ERROR LOG UTILITY
```

EVRLL

Dumps + Formats drive errl
entries RA81, 82, 70, 90

2 PDP-11 TURBO SCRUBBER Patches for ZUDLA

CAUTION

Use this patch and the program at your own risk. This turbo patch should only apply to ZUDLA0. It will not work for any other version.

Purpose:

Will disable the drive ECC error threshold before requesting block replacement. Thus BBR will take place for any ECC error encountered regardless of the number of symbols in error.

Requirements:

1. Must be revision "A" of ZUDLA0
2. In order to prevent customer problems, only run this on scratch media. (Back up the customer data before and restore after using this Turbo Scrubber.)

Location	Was	Modify to	
23620	000000	001100	(Put in "Suppress ECC Command Modifier)
41154	000000	001100	---- Ditto ----
26304	010000	000000	Disable Forced Error Command Modifier
26312	000400	000000	Disable Write with Forced Error

TO SET "DEBUG" BIT

PURPOSE: The "debug" bit causes the program to ask for an address range to scrub. Will also cause program to continue running if a hard error (not hard ECC error) occurs.

Location	Was	Modify to	
2650	000000	000001	Set any 1 into this location (non-zero)

```
PROGRAM ASKS

Enter first LBN (A)  ?    --- GIVE IT THE "STARTING LBN" FOR SCANNING

Enter last LBN (A)  ?     --- Give it the "Ending LBN" for Scanning

                              (RA60 Max LBN = 400175)
                              (RA70 Max LBN = 540740)
                              (RA80 Max LBN = 237211)
                              (RA81 Max LBN = 891071)
                              (RA82 Max LBN = 1216664)
                              (RA90 Max LBN = 2376152)
```

3 PDP-11 TURBO SCRUBBER Patches for ZUDLB

CAUTION

Use this patch and the program at your own risk. This turbo patch should only apply to ZUDLB0. It will not work for any other version.

```
QUICK CHECK -- To ensure you have version "B", verify the following:

             Loc       Contents
             ---       --------
             23340     12737
             23342     0
             23344     2346

             40734     12737
             40736     0
             40740     2346

PATCH
-----
             Loc       From      To
             ---       ----      ---
             23342     0         1400
             40736     0         1400
```

Always run verify first

Turns off ECC in the controller so all errors are reported

Run 2 passes without patch first so bad ones get marked with forced error

When ECC is turned off do not allow replacements with forced error because any error will set written with EE

You can start ZUDLB0, then before answering the final question, halt and install the patch in memory, OR you can use the XXDP patch utility and install the patch permanently into a new file (ZUDLB0.TUR, for example).

Running the turbo will cause a lot of replacements that the non–turbo version does not make.

CAUTION

Running the turbo scrubber on a drive that is not in good working order can damage the integrity of the logical structures on the media. Use this turbo scrubber only on drives that are in good working order but have bad blocks that need to be replaced.

If, while running the turbo scrubber, you find that it is making far too many replacements (too many replacements for the turbo scrubber would be anything in excess of 150 replacements), then the drive or HDA, may not be in good working order. Fix or reformat the media, then try the turbo scrubber again.

If the drive is in good working order, use the turbo scrubber after the media (pack or HDA) has been replaced.

Preface

Intended Audience

This manual is intended for VMS system managers, operators, and system programmers.

Document Structure

This document consists of the following six sections:

- Description—Provides a full description of the Monitor Utility (MONITOR).
- Usage Summary—Outlines the following MONITOR information:

    –Invoking the utility
    –Exiting from the utility
    –Directing output
    –Restrictions or privileges required

- Qualifiers—Describes MONITOR qualifiers, including format, parameters, and examples.
- Commands—Describes MONITOR commands, including format, parameters, and examples.
- Examples—Provides additional MONITOR examples.
- Appendix A—Provides supplemental MONITOR information.

Associated Documents

For additional information on the topics covered in this document, refer to the *VMS DCL Dictionary* and the *Guide to VMS Performance Management*.

Conventions

Convention	Meaning
RET	In examples, a key name (usually abbreviated) shown within a box indicates that you press a key on the keyboard; in text, a key name is not enclosed in a box. In this example, the key is the RETURN key. (Note that the RETURN key is not usually shown in syntax statements or in all examples; however, assume that you must press the RETURN key after entering a command or responding to a prompt.)
CTRL/C	A key combination, shown in uppercase with a slash separating two key names, indicates that you hold down the first key while you press the second key. For example, the key combination CTRL/C indicates that you hold down the key labeled CTRL while you press the key labeled C. In examples, a key combination is enclosed in a box.
$ SHOW TIME 05-JUN-1988 11:55:22	In examples, system output (what the system displays) is shown in black. User input (what you enter) is shown in red.
$ TYPE MYFILE.DAT . . .	In examples, a vertical series of periods, or ellipsis, means either that not all the data that the system would display in response to a command is shown or that not all the data a user would enter is shown.
input-file, . . .	In examples, a horizontal ellipsis indicates that additional parameters, values, or other information can be entered, that preceding items can be repeated one or more times, or that optional arguments in a statement have been omitted.
[logical-name]	Brackets indicate that the enclosed item is optional. (Brackets are not, however, optional in the syntax of a directory name in a file specification or in the syntax of a substring specification in an assignment statement.)
quotation marks apostrophes	The term quotation marks is used to refer to double quotation marks ("). The term apostrophe (') is used to refer to a single quotation mark.

New and Changed Features

Version 5.0 of the Monitor Utility includes the following new functions:

- A new MONITOR MSCP_SERVER command that produces statistics useful in tuning an MSCP server.
- A new MONITOR RMS command that produces a variety of statistics related to VMS Record Management Services.
- The MONITOR MODES command now produces statistics that pertain to symmetric multiprocessor systems.
- The MONITOR IO command now produces data for split transfers.

MONITOR Description

The Monitor Utility (MONITOR) is a system management tool that enables you to obtain information on operating system performance. Using MONITOR, you can monitor classes of systemwide performance data (such as system I/O statistics, page management statistics, and time spent in each of the processor modes) at specifiable intervals, and produce several types of output.

To monitor a particular class of information, you specify the class name corresponding to the information class on the MONITOR command line. For example, to monitor page management statistics, specify the PAGE class name in the MONITOR command. MONITOR collects system performance data by class and produces the following three forms of optional output:

- A disk recording file in binary format
- Statistical terminal displays
- A disk file containing statistical summary information in ASCII format

The utility initiates a single MONITOR request for the classes of performance data specified each time you enter a command in the following form:

```
MONITOR [/qualifier[,...]] classname[,...] [/qualifier[,...]]
```

Regardless of the order in which you specify class-name parameters, MONITOR always executes requests in the following sequence:

PROCESSES
STATES
MODES
PAGE
IO
FCP
POOL
LOCK
DECNET
FILE_SYSTEM_CACHE
DISK
DLOCK
SCS
SYSTEM
CLUSTER
RMS
MSCP_SERVER

Depending on the command qualifiers specified, MONITOR collects system performance data from the running system or plays back data recorded previously in a recording file. When you play back data, you can display it, summarize it, and even rerecord it to reduce the amount of data in the recording file. The Examples section illustrates these operations in greater detail.

For additional information about interpreting the information the Monitor Utility provides, see the *Guide to VMS Performance Management.*

MONITOR Description

1 Class Types

Each MONITOR class consists of data items that, taken together, provide a statistical measure of a particular system performance category. The data items defined for individual classes are listed in the description of the MONITOR command in the Commands section.

There are two MONITOR class types, differentiated by the scope of the data items collected:

- *System Classes,* in which the data items provide statistics on resource utilization for the entire system (CLUSTER, DECNET, DLOCK, FCP, FILE_SYSTEM_CACHE, IO, LOCK, MSCP_SERVER, PAGE, POOL, STATES, SYSTEM).
- *Component Classes,* in which the data items provide statistics on the contribution of individual components to the overall system or cluster. These classes are DISK, MODES, PROCESSES, RMS (Record Management Services), and SCS (System Communication Services).

As an example of the distinction between MONITOR class types, the IO class includes a data item to measure all direct I/O operations for the entire system and is therefore a system class. The DISK class measures direct I/O operations for individual disks, and is therefore a component class.

2 Class-Name Qualifiers

The class-name qualifiers control the type of display and summary output format generated for each class name specified. They have no effect on the recording of binary data. Each of these qualifiers applies only to the immediately preceding class name. Class-name qualifiers must not appear as part of the command verb. Table MON–1 summarizes class-name qualifiers and defaults.

Table MON–1 MONITOR Class-Name Qualifiers

Class Name	Qualifiers	Defaults
ALL_CLASSES	/ALL /AVERAGE /CURRENT /MAXIMUM /MINIMUM	See Commands section.
CLUSTER	/ALL /AVERAGE /CURRENT /MAXIMUM /MINIMUM	/CURRENT
DECNET	/ALL /AVERAGE /CURRENT /MAXIMUM /MINIMUM	/ALL
DISK	/ALL /AVERAGE /CURRENT /ITEM /MAXIMUM /MINIMUM /[NO]PERCENT	/ALL /ITEM=OPERATION_RATE /NOPERCENT
DLOCK	/ALL /AVERAGE /CURRENT /MAXIMUM /MINIMUM	/ALL
FCP	/ALL /AVERAGE /CURRENT /MAXIMUM /MINIMUM	/ALL
FILE_SYSTEM_CACHE	/ALL /AVERAGE /CURRENT /MAXIMUM /MINIMUM	/ALL

Table MON–1 (Cont.) MONITOR Class-Name Qualifiers

Class Name	Qualifiers	Defaults
IO	/ALL /AVERAGE /CURRENT /MAXIMUM /MINIMUM	/ALL
LOCK	/ALL /AVERAGE /CURRENT /MAXIMUM /MINIMUM	/ALL
MODES	/ALL /AVERAGE /[NO]CPU /CURRENT /MAXIMUM /MINIMUM /[NO]PERCENT	/NOCPU /CURRENT /NOPERCENT
MSCP_SERVER	/ALL /AVERAGE /CURRENT /MAXIMUM /MINIMUM	/ALL
PAGE	/ALL /AVERAGE /CURRENT /MAXIMUM /MINIMUM	/ALL
POOL	/ALL /AVERAGE /CURRENT /MAXIMUM /MINIMUM	/ALL
PROCESSES	/TOPBIO /TOPCPU /TOPDIO /TOPFAULT	None
RMS	/ALL /AVERAGE /CURRENT /FILE /ITEM /MAXIMUM /MINIMUM	/ITEM=OPERATIONS /ALL
SCS	/ALL /AVERAGE /CURRENT /ITEM /MAXIMUM /MINIMUM /[NO]PERCENT	/ALL /ITEM=KB_MAP /NOPERCENT
STATES	/ALL /AVERAGE /CURRENT /MAXIMUM /MINIMUM /[NO]PERCENT	/CURRENT /NOPERCENT
SYSTEM	/ALL /AVERAGE /CURRENT /MAXIMUM /MINIMUM	/CURRENT

Following are the three categories of class-name qualifiers:

- Statistics qualifiers (/ALL, /AVERAGE, /CURRENT, /MAXIMUM, and /MINIMUM) specify which statistics appear in display and summary output. These are conflicting qualifiers; specify no more than one of these qualifiers with each class name in a MONITOR request. Statistics qualifiers cannot be used with the PROCESSES class name or for multifile summaries.
- The data transformation qualifier (/[NO]PERCENT) controls whether data for the selected class name is expressed as percentages of a whole. This qualifier can be used only with the STATES, DISK, MODES, and SCS class names, and is not allowed for multifile summaries.
- Class-specific qualifiers (/CPU, /ITEM, /FILE, /TOPBIO, /TOPCPU, /TOPDIO, and /TOPFAULT) control the output of a specific class.
  - /CPU is used with the MODES class name to produce information for specific CPUs in a multiprocessor configuration.
  - /ITEM is used with the component statistics class names DISK, RMS, and SCS to specify one or more data items for inclusion in display or summary output.
  - /FILE is used with the RMS class name to specify the RMS file to which a MONITOR RMS command applies.

- /TOP is used with the PROCESSES class name to produce bar graphs showing the top processes instead of the standard summary and display output. Top processes are the heaviest consumers of the resource being monitored. Up to eight processes can be shown in each display. Note that the /TOP qualifiers are mutually exclusive. Specify no more than one in a single request.

3 Output Types

MONITOR can produce any combination of three forms of output for any single MONITOR request. The forms are display output, recording file output, and summary output. Output forms are specified with the /DISPLAY, /RECORD, and /SUMMARY qualifiers, as follows:

- /DISPLAY produces output in the form of ASCII screen images. Screen images are written at a frequency governed by the /VIEWING_TIME qualifier.
- /RECORD produces a binary recording file containing data collected for requested classes; one record for each class is written per interval.
- /SUMMARY produces an ASCII file containing summary statistics for all requested classes over the duration of the MONITOR request.

If you specify /INPUT with any of these qualifiers, MONITOR collects performance data from one or more previously created recording files; otherwise, data is collected from counters and data structures on the running system. The MONITOR request begins and ends at times specified by the /BEGINNING and /ENDING qualifiers respectively.

3.1 Display Output

Display output consists of a series of terminal screen images. One screen image for each requested class for each requested viewing interval is produced. You can use any terminal supported by VMS with dimensions of at least 80 columns by 24 rows. (You might have to enter the DCL command SET TERMINAL to set the proper dimensions.) Display output can also be routed to a file for subsequent printing.

The amount of time between screen displays is determined by the /VIEWING_TIME value. Effective viewing time varies, however, depending on whether you are running MONITOR on your local system or on a remote node. (*Remote* in this context refers to use of the SET HOST command to access another node.) For remote access, the time required to display the screen is included in the viewing time, while for local access this time is not included. Therefore, use a larger viewing time than the 3-second default when running MONITOR on a remote system. The value appropriate for remote access depends on your terminal baud rate. For a 9600–baud terminal line, 6 seconds is a reasonable viewing time. For lower-speed lines, increase the viewing time appropriately.

By pressing CTRL/W, you can temporarily override the /VIEWING_TIME value and generate a new display immediately following the current one. This feature is useful when the MONITOR display area has been overwritten by an operator message. You can also use CTRL/W in conjunction with a large /VIEWING_TIME value to generate display events on demand.

3.2 Display Data

All displayable data items are rates or levels except in the PROCESSES class. Rates are shown in number of occurrences per second. A level is a value that indicates the size of the monitored data item.

MONITOR can display any of four different statistics for each data item, as follows:

- Current rate or level
- Average rate or level
- Minimum rate or level
- Maximum rate or level

Average, minimum, and maximum statistics are measured from the beginning of the MONITOR request. The current statistic is the most recently collected value for the rate or level. Any or all of the statistics can be requested. For the DISK, MODES, SCS, and STATES classes, all statistics can be expressed as percentages.

3.3 Screen Formats

There are two basic screen formats used for displaying MONITOR class data: the single-statistic screen and the multiple-statistic screen. The formats vary slightly depending on whether the class being displayed is a system or component class.

The following three characteristics occur in both screen formats:

- The date and time appearing in the heading of each screen refer to the time the displayed data was originally collected.
- The name of the node on which the data was originally collected also appears in the heading (except when playing back files that do not contain node name information or when displaying CLUSTER class data). The node name is obtained from the SCSNODE system parameter or, if SCSNODE is null, from the SYS$NODE logical name established by DECnet.
- The bottom line of the display is used for status information about the current MONITOR request.
  - If data collection is from a file of previously recorded monitor data, the word PLAYBACK appears at the left margin of the line. If the currently running system is being monitored, the word does not appear.
  - If a summary file has been requested, the word SUMMARIZING appears in the middle of the line. If not, it does not appear.
  - If creation of a recording file has been requested, the word RECORDING appears at the right margin of the line. If not, it does not appear.

The PROCESSES, SYSTEM, AND CLUSTER classes have unique screen formats.

3.3.1 Single-Statistic Screen for System Classes

This bar-graph style screen is used whenever one statistic (current, average, minimum, or maximum) is requested. Example MON–1 shows the maximum statistic for the STATES class. For other classes and statistics, the screen format remains the same with different heading and data item descriptions. If the display of percentages is requested, the percent symbol (%) appears in the title and next to the numbers along the top of the graph. All values in this screen format are rounded up or down to seven whole numbers (except percentages, which are rounded to three whole numbers).

Example MON–1 Single-Statistic Screen

```
                           VAX/VMS Monitor Utility
                                PROCESS STATES
             +------+           on node SAMPLE
             | MAX  |        31-DEC-1988 16:09:53
             +------+

                                    0         10        20        30        40
                                    + - - - - + - - - - + - - - - + - - - - -+
Collided Page Wait                1 |*
Mutex & Misc Resource Wait        3 |***
Common Event Flag Wait              |
Page Fault Wait                   2 |**
Local Event Flag Wait            28 |****************************
Local Evt Flg (Outswapped)        4 |****
Hibernate                        11 |***********
                                    |
Hibernate (Outswapped)            2 |**
Suspended                           |
Suspended (Outswapped)              |
Free Page Wait                      |
Compute                           4 |****
Compute (Outswapped)              1 |*
Current Process                   1 |*
                                    + - - - - + - - - - + - - - - + - - - - -+
```

3.3.2 Multiple-Statistic Screen for System Classes

This tabular-style screen is used whenever all four statistics are requested with the /ALL class-name qualifier. Example MON–2 shows a multiple-statistic screen. The precision of the data items is seven whole and two decimal places. For each class, the screen format remains the same with different heading and data item descriptions.

If you request the display of percentages, as in Example MON–3, the percent sign (%) appears in the title and the headings, and the figures consist of three whole and one decimal place.

Example MON–2 Sample Multiple-Statistic Screen

```
                       VAX/VMS Monitor Utility
                      PAGE MANAGEMENT STATISTICS
                            on node SAMPLE
                         31-DEC-1988 16:13:38

                                  CUR        AVE        MIN        MAX

Page Fault Rate                 58.00      38.33      18.66      58.00
Page Read Rate                  18.00      16.33      14.66      18.00
Page Read I/O Rate               3.33       3.16       3.00       3.33
Page Write Rate                 45.00      22.50       0.00      45.00

Page Write I/O Rate              1.66       0.83       0.00       1.66
Free List Fault Rate            26.33      15.66       5.00      26.33
Modified List Fault Rate         4.66       3.83       3.00       4.66
Demand Zero Fault Rate          12.00       7.66       3.33      12.00
Global Valid Fault Rate         11.33       7.83       4.33      11.33

Wrt In Progress Fault Rate       0.00       0.00       0.00       0.00
System Fault Rate               24.33      12.83       1.33      24.33
Free List Size                3356.00    3321.50    3287.00    3356.00
Modified List Size               1.00      70.00       1.00     139.00
```

Example MON–3 Sample Multiple-Statistic Screen (Data Expressed as Percentages)

```
                       VAX/VMS Monitor Utility
                       TIME IN PROCESSOR MODES (%)
                            on node SAMPLE
                         31-DEC-1988 16:13:38
                                  CUR%       AVE%       MIN%       MAX%
Interrupt Stack                  20.3       21.9       20.3       23.6

MP Synchronization                0.0        0.0        0.0        0.0

Kernel Mode                      23.0       23.8       23.0       24.6

Executive Mode                    3.0        3.5        3.0       24.6

Supervisor Mode                   0.0        0.0        0.0        0.6

User Mode                        51.3       46.9       42.6       51.3

Compatibility Mode                2.3        3.6        0.0        3.9

Idle Time                         0.0        0.0        0.0       94.9
```

3.3.3 Component Classes Screen

For all component classes except RMS and MODES, only one data item for each component is displayed on each screen. The item is identified in the upper left of the screen. Components for which statistics are reported appear in the left column of the screens. If more than one item keyword is specified with the /ITEM qualifier or if /ITEM=ALL is specified, a new screen appears for each item selected. For example, the following command would produce the output of the format shown in Example MON–4:

```
MONITOR DISK/ITEM=(OPERATION_RATE,QUEUE_LENGTH)
```

Example MON–4 Sample Component Statistics Screens

```
                                VAX/VMS Monitor Utility
                                  DISK I/O STATISTICS
                                     on node SAMPLE
                                  31-DEC-1988 20:08:42

I/O Operation Rate                                   CUR         AVE         MIN         MAX

DRA2:                      SAMPLEPAGE               0.00        0.03        0.00        0.33
DRB1:                      ACCREG                   0.00        0.00        0.00        0.00
DRC3:                      VMS_X20R                 1.99        0.19        0.00        1.99
DRC4:                      SAMPLESECD01             0.00        0.00        0.00        0.00
DBA3:                      UMASTER                  0.00        0.00        0.00        0.00
DBA5:                      MIDNITE                  0.00        0.00        0.00        0.00
DRA7:                      RES26APR                 0.00        0.00        0.00        0.00
DUA4:                      RES06AUG                 0.00        0.00        0.00        0.00
DUA5:                      VMSDOCLIB                0.00        0.00        0.00        0.00
DUA7:                      OLD_QVSS$                0.00        0.00        0.00        0.00

                                VAX/VMS Monitor Utility
                                  DISK I/O STATISTICS
                                     on node SAMPLE
                                  31-DEC-1988 20:08:45

I/O Request Queue Length                             CUR         AVE         MIN         MAX

DRA2:                      SAMPLEPAGE               0.00        0.00        0.00        0.00
DRB1:                      ACCREG                   2.00        1.43        0.00        4.00
DRC3:                      VMS_X20R                 0.00        0.00        0.00        0.00
DRC4:                      SAMPLESECD01             0.00        0.00        0.00        0.00
DBA3:                      UMASTER                  0.00        0.00        0.00        0.00
DBA5:                      MIDNITE                  0.00        0.00        0.00        0.00
DRA7:                      RES26APR                 0.00        0.00        0.00        0.00
DUA4:                      RES06AUG                 0.00        0.00        0.00        0.00
DUA5:                      VMSDOCLIB                0.00        0.00        0.00        0.00
DUA7:                      OLD_QVSS$                0.00        0.00        0.00        0.00
```

3.4 Recording File Output

A recording file is a VAX RMS sequential disk file that is created when a MONITOR request includes the /RECORD qualifier. A record of binary performance data is written to this file once per interval for each requested class; the record contains a predefined set of data for each of the requested performance classes. The file is created when a MONITOR request is initiated and closed when the request terminates. The resulting file can be used as a source file by later requests to format and display the data on a terminal, to create a summary file, or to record a new recording file with different characteristics.

All data pertaining to the class is recorded, even if you are concurrently displaying only a single statistic or a single item of a component statistics class.

3.4.1 Disk Space for Recording Files

When recording is active (or display output is being routed to a disk file), you can use large quantities of disk space in a short period of time. In particular, if disk quota is exceeded during execution of a MONITOR request, open files are closed, and the request is terminated prematurely. To avoid this situation, use the information provided in Appendix A to estimate the amount of disk space required.

When SYSTEM class data is recorded, the MODES, STATES, and PROCESSES classes are also recorded, even if not specifically requested. When CLUSTER class data is recorded, the MODES and DISK classes are also recorded. To estimate disk space requirements for CLUSTER recording files, multiply the totals for these classes by the number of nodes being monitored. After estimating disk space requirements, check the amount of disk quota available, and set appropriate values for /INTERVAL and /ENDING.

Refer to Appendix A for exact recording file record sizes.

3.4.2 Recording File Version Compatibility

Before Version 5.0 MONITOR can read recording files generated by previous MONITOR versions, you must convert the files to the current format. Use the CONVERT command described in the Commands section of this document. If you specify a list of recording files to produce a multifile summary, all recording files must have the same format.

3.5 Summary Output

Summary output is an ASCII disk file consisting of one display screen image for each requested class. The screen format for each class is based on the statistic requested. The only difference in format between a display screen and a summary screen image is that the word SUMMARY appears in the heading along with a beginning and ending time for the period covered by the summary. The data contained in the summaries is identical to that shown on the final display screen (if display output was also requested) for all except the PROCESSES/TOP, SYSTEM, and CLUSTER summaries.

Since the summary file reflects the accumulation of data throughout the MONITOR request, the average, minimum, and maximum statistics are of particular interest. For the TOP summaries of the PROCESSES and SYSTEM classes, the data represents the top users for the entire duration of the MONITOR request, subject to the following restriction. To be eligible for inclusion in the list of top users, a process must be present and swapped in at the beginning and end of the MONITOR request.

3.6 Multifile Summary Reports

Multifile summary reports provide a convenient method of combining data from a number of recording files to compare average performance statistics (excluding the PROCESSES and CLUSTER classes) for discrete time segments. Use the /BEGINNING and /ENDING command qualifiers to delimit the desired time segment (see the Qualifiers section).

To request a multifile summary, use the /SUMMARY command qualifier, and specify a list of recording files with the /INPUT qualifier. Note that since only AVERAGE statistics are collected, you should not specify class-name qualifiers. Note also that multifile summaries are static; that is, they do not provide continuously updated displays.

Caution: Version 5.0 MONITOR file structure must be common to all recording files in the list.

3.6.1 Interpreting Multifile Summary Reports

Multifile summary reports differ from regular (single-file) reports in both content (only AVERAGE statistics are collected) and format. MONITOR formats multifile report data as follows:

- By file—This is the default format. For each class requested, the report displays one column of AVERAGE statistics per input file, along with beginning and ending times for each file. For files that contain data for multiple nodes, there is one column per node per file.
- By node—To request this format, specify the /BY_NODE command qualifier (with the /SUMMARY and /INPUT qualifiers) when you create the summary file. The report combines data for a given node from all files into a single column that shows the average statistic for each data item. The contribution of the data from each file is weighted by the amount of time over which the data was collected (for rate items) or by the number of collections (for level items).

For both formats, MONITOR provides Row Sum, Row Average, Row Maximum, and Row Minimum statistics. These represent simple arithmetic operations performed on all averages in each row of the report.

Note: Because multifile summary reports frequently contain large amounts of data and use a 132-character format, you may want to print them. A single page can display only five columns of data. Depending on the number of recording files, nodes, and classes specified, a report may extend over many pages. In that event, Row values appear on the final page.

The following examples illustrate differences between single-file and multifile reports. In the first example, two fragments of single-file reports are generated on the same node from two different recording files, which cover, respectively, a two-hour and a ten-hour period:

```
                        VAX/VMS Monitor Utility
                      PAGE MANAGEMENT STATISTICS
                            on node BLUE           From: 31-DEC-1988 09:00:00
                               SUMMARY             To:   31-DEC-1988 11:00:00

                                  CUR        AVE         MIN         MAX
Page Fault Rate                  ----       90.00        ----        ----

                        VAX/VMS Monitor Utility
                      PAGE MANAGEMENT STATISTICS
                            on node BLUE           From: 31-DEC-1988 11:00:00
                               SUMMARY             To:   31-DEC-1988 21:00:00

                                  CUR        AVE         MIN         MAX
Page Fault Rate                  ----        6.00        ----        ----
```

In the next example, the corresponding "by node" multifile report fragment for the entire period shows that the average Page Fault Rate is weighted toward the figure that represents the larger elapsed time:

```
           +------+         VAX/VMS Monitor Utility
           | AVE |         PAGE MANAGEMENT STATISTICS
           +------+            Multifile SUMMARY

             Node:       BLUE (2)
             From: 31-DEC-1988 09:00       Row    Row       Row       Row
             To:   31-DEC-1988 21:00       Sum  Average   Minimum   Maximum

Page Fault Rate                20.00      20.0     20.0     20.00     20.00
```

The Row statistics provided in multifile reports are not time-weighted; they are meaningful only when all input files that contribute to the report cover a common time period, as in Example MON–6 ("by node" report for nodes MOE, CURLEY, and LARRY) in Section 3.6.2. Thus, as the following example shows, Row statistics would not be useful in a "by file" report for node BLUE because the contributing files covered *different* time periods:

```
           +------+         VAX/VMS Monitor Utility
           | AVE |         PAGE MANAGEMENT STATISTICS
           +------+            Multifile SUMMARY

             Node:        BLUE              BLUE
             From: 31-DEC-1988 09:00  31-DEC-1988 11:00       Row    Row       Row       Row
             To:   31-DEC-1988 11:00  31-DEC-1988 21:00       Sum  Average   Minimum   Maximum

Page Fault Rate                90.00               6.00      96.0     48.0      6.00     90.00
```

The preceding examples illustrate how MONITOR formats multifile report data. Each column of averages is headed by the name of the node on which the data was collected, the requested beginning and ending times, and (for reports by node) a parenthesized number signifying the number of input files that contributed data to the column. If no explicit beginning and ending times are specified in the summary request, the times stored in the files are listed.

Note that if a column in a report by node contains data from more than one input file, the earliest beginning time and the latest ending time among all the time stamps in those files are listed. Therefore, exercise caution in interpreting "by node" reports because they may include (in the listed time) the range of a period during which no data was collected.

3.6.2 Using Multifile Summary Reports in Single-Node Environments

In single-node environments, reports by file and reports by node have the following applications:

- By file—You can use this type of report (which contains one column of average statistics per input file) when you want to compare data from several different time segments. The input files selected must already contain the time segments of interest when the multifile summary is run; the /BEGINNING and /ENDING qualifiers cannot be used for this purpose because they can define only one time segment. If necessary, execute a preliminary MONITOR command to rerecord an input file, and write data for the desired time segment to a new file.

  Example MON–5 shows a multifile report generated from three input files, each of which contained data for the period 8 A.M. to noon on a different day. Some of these files may have been created as a result of rerecording larger files to extract only the 8 A.M. to noon time segment.

Note that the average page fault rate for a morning during the week of 11 May was 28.4, and that the daily averages for that time period ranged from 21.74 to 39.53.

- By node—If you specify the /BY_NODE qualifier, you can combine data from several input files to produce a single average statistic for each data item. If continuous recording is implemented, so that MONITOR data is recorded whenever the system is up, you can produce a report that shows averages for any time segment, regardless of how many input files are included (up to the maximum of 5,000). Use the /BEGINNING and /ENDING qualifiers to define the time segment of interest.

Example MON–5 shows a typical multifile summary report by file for three input files on a single node. The summary was requested with the following command:

```
MONITOR /INPUT=(11MAY,12MAY,13MAY)/NODISPLAY/SUMMARY PAGE
```

Example MON–5 Sample Single-Node Multifile Summary

```
           +------+        VAX/VMS Monitor Utility
           | AVE |       PAGE MANAGEMENT STATISTICS
           +------+          Multifile SUMMARY

             Node:        YELLOW              YELLOW              YELLOW
             From:  11-MAY-1988 08:00  12-MAY-1988 08:00  13-MAY-1988 08:00       Row      Row       Row       Row
             To:    11-MAY-1988 12:00  12-MAY-1988 12:00  13-MAY-1988 12:00       Sum  Average   Minimum   Maximum

Page Fault Rate                 39.53              23.98              21.74      85.2     28.4     21.74     39.53
Page Read Rate                  14.12               3.30               2.21      19.6      6.5      2.21     14.12
Page Read I/O Rate               1.73               0.52               0.43       2.7      0.9      0.43      1.73
Page Write Rate                 11.21               1.54               1.11      13.7      4.6      1.11     11.21
Page Write I/O Rate              0.11               0.01               0.01       0.1      0.0      0.00      0.11

Free List Fault Rate            10.28               5.00               4.73      20.0      6.6      4.73     10.28
Modified List Fault Rate         5.51               4.77               4.53      14.8      4.9      4.53      5.51
Demand Zero Fault Rate          11.58               5.06               4.21      20.8      6.9      4.21     11.58
Global Valid Fault Rate         10.25               8.54               7.76      26.5      8.8      7.76     10.25
Wrt In Progress Fault Rate       0.03               0.01               0.01       0.0      0.0      0.01      0.03
System Fault Rate                0.04               0.09               0.05       0.2      0.0      0.04      0.09

Free List Size                6829.60            7102.33            7271.85   21203.7   7067.9   6829.60   7271.85
Modified List Size             257.31             210.41             216.91     684.6    228.2    210.41    257.31
```

3.6.3 Using Multifile Summary Reports in Cluster Environments

In cluster or other multinode environments (DECnet or any set of noncommunicating nodes), reports by file and by node have the following applications:

- By file—You can use this type of report to compare several time segments as you would for the single-node case. By selecting the appropriate input files, you can include in the report data from any single node or set of nodes.
- By node—As in the single-node case, you can use this type of report to combine data from several input files for a given node in one column of averages. Note, however, that there is one column for each node. By specifying the /BEGINNING and /ENDING qualifiers to select a common time period, you can use the report to make node-to-node comparisons. In addition, you can examine the "Row" statistics to evaluate overall cluster performance.