

RABO-*

UDA FMTR DATA FILE
CZUDFAO

AH-SB40A-MC
FICHE 1 OF 1

OCT 1981
COPYRIGHT © 1981
MADE IN USA

00000000

DATA FILE FOR PDP-11 UDA DISK D MACRO X04.00 23-JUL-81 15:08:54 PAGE 2
USER DOCUMENTATION

.REM 8

IDENTIFICATION

PRODUCT CODE: AC-S839A-MC
PRODUCT NAME: CZUDFAO UDA FMTR DATA FILE
PRODUCT DATE: 10-JULY-81
MAINTAINER: DIAGNOSTIC ENGINEERING

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS DOCUMENT.

NO RESPONSIBILITY IS ASSUMED FOR THE USE OR RELIABILITY OF SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL OR ITS AFFILIATED COMPANIES.

COPYRIGHT (C) 1981 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL
DEC

PDP
DECUS

UNIBUS
DECTAPE

MASSBUS

TABLE OF CONTENTS

1.0	GENERAL INFORMATION
1.1	PROGRAM ABSTRACT
1.2	SYSTEM REQUIREMENTS
1.3	RELATED DOCUMENTS AND STANDARDS
2.0	OPERATING INSTRUCTIONS
2.1	COMMANDS
2.2	SWITCHES
2.3	FLAGS
2.4	HARDWARE QUESTIONS
2.5	SOFTWARE QUESTIONS
2.6	QUICK STARTUP PROCEDURE
3.0	MESSAGES
4.0	PERFORMANCE AND PROGRESS REPORTS

1.0 GENERAL INFORMATION

1.1 PROGRAM ABSTRACT

THIS DIAGNOSTIC HAS BEEN WRITTEN FOR USE WITH THE DIAGNOSTIC RUNTIME SERVICES SOFTWARE (SUPERVISOR). THESE SERVICES PROVIDE THE INTERFACE TO THE OPERATOR AND TO THE SOFTWARE ENVIRONMENT. THIS PROGRAM CAN BE USED WITH XXDP+, ACT, APT, SLIDE AND PAPER TAPE. FOR A COMPLETE DESCRIPTION OF THE RUNTIME SERVICES, REFER TO THE XXDP+ USER'S MANUAL. THERE IS A BRIEF DESCRIPTION OF THE RUNTIME SERVICES IN SECTION 2 OF THIS DOCUMENT.

THIS DOCUMENT DESCRIBES THE USAGE OF THE UNIBUS DISK ADAPTER (UDA) DISK FORMATTER FROM THE USER'S VIEWPOINT. THIS DOCUMENT DOES NOT DESCRIBE THE ACTUAL FORMATTING OF THE MEDIA, BUT RATHER THE POSSIBLE INTERACTIONS WITH A HUMAN OPERATOR.

THERE ARE FOUR GENERAL MODES OF FORMATTER OPERATION:

O BUILD - THIS MODE IS GENERALLY USED BY MANUFACTURING PERSONAL TO INITIALLY FORMAT A NEW MEDIA. IT USES DATA STORED ON THE MEDIA, FROM A LOCATION WHICH IS NORMALLY INACCESSABLE, TO LOCATE AND REVECTOR THE BAD BLOCKS ON THE MEDIA.

O REFORMAT - THIS MODE IS USED TO FORMAT A MEDIA WHICH HAS BEEN PREVIOUSLY FORMATTED, AND IS BEING REFORMATTED TO CLEAR EXISTING DATA OR TO CHANGE THE MODE OF THE MEDIA (512/576 BYTES PER SECTOR). IT IS ASSUMED THAT THE FACTORY CONTROL TABLE (FCT) IS STILL IN TACT.

O RESTORE - THIS MODE WILL ONLY BE RUN BY DIGITAL FIELD CIRCUS PERSONAL. IT PROVIDES AN EXTERNAL COPY OF THE FCT, PRODUCED AT BUILD TIME AND STORED OFFLINE BY DIGITAL, TO THE UDA FORMATTER.

O RECONSTRUCT - THIS MODE IS USED WHEN NONE OF THE OTHER MODES ARE POSSIBLE. IT DETECTS BAD BLOCKS BY PREFORMING REPETATIVE READ CHECKS OF EACH SECTOR. FOR THIS REASON, A RECONSTRUCT RUN TAKES CONSIDERABLY LONGER THAN THE OTHER MODES.

1.2 SYSTEM REQUIREMENTS

PDP-11 PROCESSOR
28K MEMORY
TERMINAL
PROGRAM LOAD DEVICE
UDASO CONTROLLER
1 OR MORE SDI COMPATIBLE DISK DRIVE

1.3 RELATED DOCUMENTS AND STANDARDS

DEC STD 166

2.0 OPERATING INSTRUCTIONS

THIS SECTION CONTAINS A BRIEF DESCRIPTION OF THE RUNTIME SERVICES.

FOR DETAILED INFORMATION, REFER TO THE XXDP+ USER'S MANUAL (CHQUS).

2.1 COMMANDS

THERE ARE ELEVEN LEGAL COMMANDS FOR THE DIAGNOSTIC RUNTIME SERVICES (SUPERVISOR). THIS SECTION LISTS THE COMMANDS AND GIVES A VERY BRIEF DESCRIPTION OF THEM. THE XXDP+ USER'S MANUAL HAS MORE DETAILS.

COMMAND	EFFECT
START	START THE DIAGNOSTIC FROM AN INITIAL STATE
RESTART	START THE DIAGNOSTIC WITHOUT INITIALIZING
CONTINUE	CONTINUE AT TEST THAT WAS INTERRUPTED (AFTER ^C)
PROCEED	CONTINUE FROM AN ERROR HALT
EXIT	RETURN TO XXDP+ MONITOR (XXDP+ OPERATION ONLY.)
ADD	ACTIVATE A UNIT FOR TESTING (ALL UNITS ARE CONSIDERED TO BE ACTIVE AT START TIME)
DROP	DEACTIVATE A UNIT
PRINT	PRINT STATISTICAL INFORMATION (IF IMPLEMENTED BY THE DIAGNOSTIC - SECTION 4.0)
DISPLAY	TYPE A LIST OF ALL DEVICE INFORMATION
FLAGS	TYPE THE STATE OF ALL FLAGS (SEE SECTION 2.3)
ZFLAGS	CLEAR ALL FLAGS (SEE SECTION 2.3)

A COMMAND CAN BE RECOGNIZED BY THE FIRST THREE CHARACTERS. SO YOU MAY, FOR EXAMPLE, TYPE 'STA' INSTEAD OF 'START'.

2.2 SWITCHES

THERE ARE SEVERAL SWITCHES WHICH ARE USED TO MODIFY SUPERVISOR OPERATION. THESE SWITCHES ARE APPENDED TO THE LEGAL COMMANDS. ALL OF THE LEGAL SWITCHES ARE TABULATED BELOW WITH A BRIEF DESCRIPTION OF EACH. IN THE DESCRIPTIONS BELOW, A DECIMAL NUMBER IS DESIGNATED BY 'DDDDD'.

SWITCH	EFFECT
/TESTS:LIST	EXECUTE ONLY THOSE TESTS SPECIFIED IN THE LIST. LIST IS A STRING OF TEST NUMBERS, FOR EXAMPLE - /TESTS:1:5:7-10. THIS LIST WILL CAUSE TESTS 1,5,7,8,9,10 TO BE RUN. ALL OTHER TESTS WILL NOT BE RUN.
/PASS:DDDDD	EXECUTE DDDDD PASSES (DDDDD = 1 TO 64000)
/FLAGS:FLGS	SET SPECIFIED FLAGS. FLAGS ARE DESCRIBED IN SECTION 2.3.
/EOP:DDDDD	REPORT END OF PASS MESSAGE AFTER EVERY DDDDD PASSES ONLY. (DDDDD = 1 TO 64000)
/UNITS:LIST	TEST/ADD/DROP ONLY THOSE UNITS SPECIFIED IN THE LIST. LIST EXAMPLE - /UNITS:0:5:10-12 USE UNITS 0,5,10,11,12 (UNIT NUMBERS = 0-63)

EXAMPLE OF SWITCH USAGE:

START/TESTS:1-5/PASS:1000/EOP:100

THE EFFECT OF THIS COMMAND WILL BE: 1) TESTS 1 THROUGH 5 WILL BE EXECUTED, 2) ALL UNITS WILL TESTED 1000 TIMES AND 3) THE END OF PASS MESSAGES WILL BE PRINTED AFTER EACH 100 PASSES ONLY. A

SWITCH CAN BE RECOGNIZED BY THE FIRST THREE CHARACTERS. YOU MAY, FOR EXAMPLE, TYPE "/TES:1-5" INSTEAD OF "/TESTS:1-5".

BELOW IS A TABLE THAT SPECIFIES WHICH SWITCHES CAN BE USED BY EACH COMMAND.

	TESTS	PASS	FLAGS	EOP	UNITS
START	X	X	X	X	X
RESTART	X	X	X	X	X
CONTINUE		X	X	X	
PROCEED			X		
DROP					X
ADD					X
PRINT					
DISPLAY					X
FLAGS					
ZFLAGS					
EXIT					

2.3 FLAGS

FLAGS ARE USED TO SET UP CERTAIN OPERATIONAL PARAMETERS SUCH AS LOOPING ON ERROR. ALL FLAGS ARE CLEARED AT STARTUP AND REMAIN CLEARED UNTIL EXPLICITLY SET USING THE FLAGS SWITCH. FLAGS ARE ALSO CLEARED AFTER A START COMMAND UNLESS SET USING THE FLAG SWITCH. THE ZFLAGS COMMAND MAY ALSO BE USED TO CLEAR ALL FLAGS. WITH THE EXCEPTION OF THE START AND ZFLAGS COMMANDS, NO COMMANDS AFFECT THE STATE OF THE FLAGS; THEY REMAIN SET OR CLEARED AS SPECIFIED BY THE LAST FLAG SWITCH.

FLAG	EFFECT
HOE	HALT ON ERROR - CONTROL IS RETURNED TO RUNTIME SERVICES COMMAND MODE
LOE	LOOP ON ERROR
IER*	INHIBIT ALL ERROR REPORTS
IBR*	INHIBIT ALL ERROR REPORTS EXCEPT FIRST LEVEL (FIRST LEVEL CONTAINS ERROR TYPE, NUMBER, PC, TEST AND UNIT)
IXR*	INHIBIT EXTENDED ERROR REPORTS (THOSE CALLED BY PRINTX MACRO'S)
PRI	DIRECT MESSAGES TO LINE PRINTER
PNT	PRINT TEST NUMBER AS TEST EXECUTES
BOE	'BELL' ON ERROR
UAM	UNATTENDED MODE (NO MANUAL INTERVENTION)
ISR	INHIBIT STATISTICAL REPORTS (DOES NOT APPLY TO DIAGNOSTICS WHICH DO NOT SUPPORT STATISTICAL REPORTING)
IDR	INHIBIT PROGRAM DROPPING OF UNITS
ADR	EXECUTE AUTODROP CODE
LOT	LOOP ON TEST
EVL	EXECUTE EVALUATION (ON DIAGNOSTICS WHICH HAVE EVALUATION SUPPORT)

*ERROR MESSAGES ARE DESCRIBED IN SECTION 3.1

SEE THE XXDP+ USER'S MANUAL FOR MORE DETAILS ON FLAGS. YOU MAY SPECIFY MORE THAN ONE FLAG WITH THE FLAG SWITCH. FOR EXAMPLE, TO CAUSE THE PROGRAM TO LOOP ON ERROR, INHIBIT ERROR REPORTS AND TYPE A 'BELL' ON ERROR, YOU MAY USE THE FOLLOWING STRING:

/FLAGS:LOE:IER:BOE

2.4 HARDWARE QUESTIONS

THE DM PROGRAM (CURRENTLY KDUBAO.PAK) AND THE HOST PORTION OF THE FORMATTER (CURRENTLY ZDUBAO.BIN) MUST EXIST ON THE BOOT DEVICE. THE FORMATTER IS STARTED BY ENTERING:

RUN ZDUBAO

WHEN THE HOST PORTION HAS BEEN LOADED, SEVERAL UNINFORMATIVE STATEMENTS WILL BE PRINTED, AND THEN THE FOLLOWING PROMPT WILL APPEAR:

DR>

RESPOND WITH:

START/PASS:1

THE NEXT QUESTION LOGGED WILL BE AN OPTION TO CHANGE THE DEFAULT RUN PARAMETERS OF THE FORMATTER. THE QUESTION IS

CHANGE HW (L)?

THE CURRENT DEFAULTS ARE FOR ONE UDA AT ADDRESS 172150, FORMAT THE DRIVE WITH A UNIT PLUG NUMBER OF 0 USING THE EXISTING FCT, AND STOP IF THAT FCT IS BAD. IF THESE DEFAULTS ARE ACCEPTABLE ENTER 'N' AND THE FORMAT WILL BEGIN. IF YOU ENTER 'Y' THEN YOU WILL HAVE TO ANSWER THE FOLLOWING QUESTIONS

UNITS (D)?

THE ANSWER TO THIS QUESTION DEFINES HOW MANY UDAS ARE TO RUN FORMATTERS CONCURRENTLY. A VALID RESPONSE IS A DECIMAL NUMBER FROM 1 TO 64 (MEMORY AVAILABILITY PERMITTING). THE FOLLOWING BLOCK OF QUESTIONS WILL BE REPEATED ONCE FOR EACH UNIT (UDA).

QUESTION 1: UNIBUS ADDRESS OF UDA (A)? XXXXXX

WHERE XXXXXX IS THE CURRENT DEFAULT (INITIALLY 172150). ENTER THE APPROPRIATE EVEN ADDRESS OR ENTER <CR> TO TAKE THE DEFAULT.

QUESTION 2: UNIT NUMBER TO FORMAT (D)? X

X IS THE CURRENT DEFAULT (INITIALLY 0). ENTER THE NUMBER ON THE UNIT PLUG OF THE DRIVE TO BE FORMATTED (DO NOT CONFUSE THE UNIT NUMBER, 0-63, ASSIGNED BY XXDP+ WITH THE UNIT NUMBER OF THE DRIVE).

QUESTION 3: USE RESIDENT FCT (L)? X

Y IS THE CURRENT DEFAULT (INITIALLY Y). IF THIS QUESTION IS ANSWERED 'Y' THEN A REFORMAT MODE FORMAT WILL BE PERFORMED AND THE NEXT QUESTION POSED IS NUMBER 4. IF THIS QUESTION IS ANSWERED 'N' THE NEXT QUESTION POSED IS NUMBER 5.

QUESTION 4: STOP IF RESIDENT FCT IS BAD (L)? X

X IS THE CURRENT DEFAULT (INITIALLY Y). IF THIS QUESTION IS ANSWERED 'Y' A REFORMAT MODE FORMAT WILL BE PERFORMED, AND NO FURTHER QUESTIONS WILL BE ASKED. THE FORMAT (OF THIS DRIVE ONLY) WILL BE ABORTED IF THE FCT IS UNUSABLE. IF THIS QUESTION IS ANSWERED 'N' THEN A RECONSTRUCT MODE FORMAT WILL BE PERFORMED EVEN IF THE FCT CAN NOT BE USED. QUESTIONING CONTINUES WITH QUESTION 6.

QUESTION 5: DOWN LINE LOAD FCT (L)? X

X IS THE CURRENT DEFAULT (INITIALLY N). ANSWERING THIS QUESTION 'Y' RESULTS IN A RESTORE MODE FORMAT BEING RUN, AND NO MORE QUESTIONS WILL BE ASKED. ANSWERING 'N' CAUSES QUESTION 6 TO BE ASKED.

QUESTION 6:

SERIAL NUMBER TO BE ASSIGNED (TOTAL 4 WORDS) WORD 1? (D)
WORD 2? (D)
WORD 3? (D)
WORD 4? (D)

THIS IS THE SERIAL NUMBER TO BE ASSIGNED TO THE DISK IF THE FCT WAS NOT USED. THERE IS NO DEFAULT VALUE, A NON-ZERO VALUE SHOULD BE CHOSEN AND IT SHOULD BE UNIQUE WITHIN THE INSTALLATION. NO FURTHER QUESTIONS ARE ASK.

NOTE THAT ONCE THE QUESTIONS HAVE BEEN ANSWERED FOR EACH UNIT, THOSE ANSWERS BECOME THE DEFAULT FOR THAT UNIT.

IF QUESTION 5 WAS ANSWERED 'Y', THEN QUESTION 7 IS ASKED.

QUESTION 7: ENTER FILE NAME FOR DOWN LINE LOAD (A)? XXXXXX.YYY

XXXXXX.YYY IS THE FILE NAME AND EXTENTION OF THE BAD SECTOR FILE IS USED BY THE FORMATTER. IF A NONEXISTENT FILE IS SPECIFIED, THE THE FORMATTER PRONTS A LOOK UP ERROR AND RETURNS TO THE MONITOR.

2.5 SOFTWARE QUESTIONS

AFTER YOU HAVE ANSWERED THE HARDWARE QUESTIONS OR AFTER A RESTART OR CONTINUE COMMAND, THE RUNTIME SERVICES WILL ASK FOR SOFTWARE PARAMETERS. THESE PARAMETERS WILL GOVERN SOME DIAGNOSTIC SPECIFIC OPERATION MODES. YOU WILL BE PROMPTED BY "CHANGE SW (L) ?"
IF YOU WISH TO CHANGE ANY PARAMFTERS, ANSWER BY TYPING 'Y'. THE SOFTWARE QUESTIONS AND THE DEFAULT VALUES ARE DESCRIBED IN THE NEXT PARAGRAPH(S).

THE FOLLOWING QUESTION MUST BE ANSWERED.

DATE VAX FORMAT (TOTAL 4 WORDS) WORD 1? (D)
WORD 2? (D)
WORD 3? (D)
WORD 4? (D)

THIS IS THE DATE TO BE ASSIGNED TO THE MOST RECENT DATE OF FORMAT
FACT OF EACH DRIVE FORMATTED. THE DEFAULT IS ZERO.

2.6 QUICK START-UP PROCEDURE (XXDP+)

TO START-UP THIS PROGRAM:

1. BOOT XXDP+
2. GIVE THE DATE AND ANSWER THE LSI AND 50HZ (IF THERE IS A CLOCK) QUESTIONS
3. TYPE 'R NAME', WHERE NAME IS THE NAME OF THE BIN OR BIC FILE FOR THIS PROGRAM
4. TYPE 'STAR'
5. ANSWER THE 'CHANGE HW' QUESTION WITH 'Y'
6. ANSWER ALL THE HARDWARE QUESTIONS
7. ANSWER THE 'CHANGE SW' QUESTION WITH 'N'

WHEN YOU FOLLOW THIS PROCEDURE YOU WILL BE USING ONLY THE
DEFAULTS FOR FLAGS AND SOFTWARE PARAMETERS. THESE DEFAULTS
ARE DESCRIBED IN SECTIONS 2.3 AND 2.5.

3.0 MESSAGES

3.1 ERROR MESSAGES

IF AN ERROR WAS ENCOUNTERED DURING FORMATTING, THE MESSAGE

UDA FORMATTER ERROR OCCURED
ERROR NUMBER REPORTED WAS XXX
ERROR CODE WAS YYYYYY DECIMAL
ERROR NUMBER MEANS:
EXPLANATION

UNIT X - REMOVED FROM TESTING FOR REMAINDER OF PASS.

IS LOGGED. XXX REPRESENTS THE INTERNAL FORMATTER ERROR NUMBER. IF
THIS ERROR NUMBER IS KNOWN BY THE HOST PORTION OF THE FORMATTER, THEN
THE ASCII MESSAGE DESCRIBING THIS ERROR IS ALSO LOGGED. YYYYYY IS AN
ADDITIONAL ERROR CODE. AN EXPLANATION IS REPORTED WITH EACH ERROR.

3.2 OTHER MESSAGES

ONCE THE FORMATTER BEGINS EXECUTING, THIS MESSAGE IS PRINTED.

FORMAT(S) BEGUN

ONCE A CYLINDER HAS BEEN FORMATTED, THIS MESSAGE APPEARS.

UNIT X - CYLINDER YYY COMPLETED

A CAUTIONARY MESSAGE APPEARS IF THE UDA50 HAS NOT COMMUNICATED WITH THE HOST.

THE UDA HAS NOT RESPONDED LATELY

THIS COULD OCCUR IF THE UDA IS DOING A LONG SERIES OF CALCULATIONS OR INDEED IS HUNG.

4.0 PERFORMANCE AND PROGRESS REPORTS

AT THE END OF EACH PASS, THE PASS COUNT IS GIVEN ALONG WITH THE TOTAL NUMBER OF ERRORS REPORTED SINCE THE DIAGNOSTIC WAS STARTED. THE 'EOP' SWITCH CAN BE USED TO CONTROL HOW OFTEN THE END OF PASS MESSAGE IS PRINTED. SECTION 2.2 DESCRIBES SWITCHES.

THE FORMAT HAS BEEN SUCCESSFULLY COMPLETED WHEN THE MESSAGE
FORMAT SUCCESSFULLY COMPLETED
IS LOGGED.

2
3

.TITLE DATA FILE FOR PDP-11 UDA DISK DRV FMTR
.IDENT /01.00/

COPYRIGHT (C) 1981
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY ON A
SINGLE COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION
OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE, OR ANY OTHER
COPIES THEREOF, MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE
TO ANY OTHER PERSON EXCEPT FOR USE ON SUCH SYSTEM AND TO ONE
WHO AGREES TO THESE LICENSE TERMS. TITLE TO AND OWNERSHIP OF
THE SOFTWARE SHALL AT ALL TIMES REMAIN IN DIGITAL.

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT
NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL
EQUIPMENT CORPORATION.

DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF
ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

VERSION 01.00

M. A. PARENTI 16-MAY-80

MODIFIED BY:

M. A. PARENTI 09-DEC-80
NEW DM INSTRUCTIONS (MEM -> MEM)
CODE OPTIMIZATION

31-MAR-81
M. A. PARENTI
ADDED DOUBLE WORD ADDRESSING FOR OVERLAY ADDRESSES

21-APR-81
M. A. PARENTI
FIX GROUP OFFSET CALCULATION

23-APR-81
M. A. PARENTI
FIX ZERO GROUP PROBLEM

24-APR-81
M. A. PARENTI
FIX LBN GROUP PROBLEM
FIX RECAL WAIT PROBLEM

58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114

05-MAY-81
M. A. PARENTI
FIX SIZE PROBLEM FOR RA81.
13-MAY-81
M. A. PARENTI
ADD LIMITED DUP FUNCTIONALITY
15-MAY-81
M. A. PARENTI
FIX SUBUNIT MASK PROBLEM
15-MAY-81
M. A. PARENTI
ONLY WRITE NON-PAD BLOCKS OF FCT
15-MAY-81
M. A. PARENTI
FIX COMPUTATION OF NON-PAD FCT BLOCKS
15-MAY-81
M. A. PARENTI
FIX BLOCK ZERO FCT PROBLEM
28-MAY-81
M. A. PARENTI
FIX SUBUNIT PROBLEMS
WRITE ONLY NON-PAD BLOCKS OF RCT
01-JUN-81
M. A. PARENTI
FIX DOUBLE COMPARE PROBLEM
08-JUN-81
M. A. PARENTI
FIX EXISTING FCT FORMAT PROBLEM
17-JUN-81
M. A. PARENTI
FIX SUBUNIT WRITE PROTECT PROBLEM
17-JUN-81
M. A. PARENTI
ADD STATUS UPDATES AT VARIOUS PLACES
22-JUN-81
M. A. PARENTI
FIX RBN STARTING BITS PROBLEM
22-JUN-81
M. A. PARENTI
FIX NON-PAD RCT INITIALIZE PROBLEM
06-JUL-81
M. A. PARENTI

115
116
117
118
119
120
121

FIX STATISTICS COUNT OF BAD BLOCKS TO NOT INCLUDE RBN
VERSION 1.0 FOR RELEASE
UDA50 DISK FORMATTER

```

1      .SBTTL  EQUATES
2
3      EQUATES
4
5
6
7      CONDITIONAL FLAGS
8
9      000001      LONG      -      1.      ;COMPLETE MESSAGE ON CYL (1)
10     000000      PROD      -      0      ;OR MOD 16 CYL (0)
11                                     ;PRODUCTION (1)
12                                     ;NOT PRODUCTION (0)
13
14     OFFSETS FOR FORMAT TRACK TABLE
15
16     0000C0      FT.BUF    -      0.      ;BUFFER POINTER OFFSET
17     000001      FT.LOW   -      1.      ;LOW ORDER HEADER OFFSET
18     000002      FT.HI    -      2.      ;HI ORDER HEADER OFFSET
19
20
21
22
23
24     ;
25     OFFSETS FOR READ/WRITE BUFFERS
26     000000      RW.STAT   =      0.      ; STATUS (12-15), NEXT BUFR PTR (0-14)
27     000000      RW.ER1    -      0.      ; ALSO USED AS ECC ERROR INDICATOR
28     000005      RW.DUM    =      5.      ; POINTER TO DUMMY SDI CONTROL BLOCK
29     000001      RW.BUF    -      1.      ; POINTER TO DATA BUFFER
30     000002      RW.LOW    =      2.      ; 1ST HEADER WORD (LO ORDER LBN)
31     000003      RW.HI     =      3.      ; 2ND HEADER WORD (HI ORDER LBN)
32     000004      RW.CMD    =      4.      ; SDI RT CMD (8-15), HEAD ADDR (0-7)
33     000000      RW.DAT    =      0.      ; 1ST WORD OF 256 WORD DATA BUFFER
34     000400      RW.EDC    =      256.    ; EDC
35     000401      RW.ER2    =      257.    ; 1ST ECC RESIDUE
36
37
38
39
40     ;
41     OFFSETS FOR CHECK PASS READ COMMANDS
42
43     000000      RB.BUF    =      0      ;BUFFER OFFSET
44     000001      RB.LOW    -      1.      ;LOW ORDER BLOCK NUMBER
45     000002      RB.HI     -      2.      ;HIGH ORDER BLOCK NUMBER
46     000003      RB.CMD    -      3.      ;READ COMMAND AND TRACK NUMBER
47     000004      RB.IM     -      4.      ;IMAGE COUNTER
48
49     ;
50     XFC DEFINITION EQUATES
51     000000      BREAK     -      0.      ;BREAKPOINT XFC CODE
52     000001      FORMAT    -      1.      ;FORMAT TRACK XFC CODE
53     000002      READ      -      2.      ;READ N SECTORS XFC CODE
54     000003      WRITE     =      3.      ;WRITE N SECTORS XFC CODE
55     000004      SEND      -      4.      ;SEND SDI COMMAND XFC CODE
56     000005      RCV       -      5.      ;RECEIVE SDI MESSAGE XFC CODE
57     000006      CMPDAT    -      6.      ;COMPARE DATA PATTERN XFC CODE

```

58	000007	STATUS	-	7.	:RETURN DRIVE STATUS XFC CODE
59	000010	ECHO	-	8.	:ECHO DATA TO DRIVE XFC CODE
60	000011	DINIT	-	9.	:DRIVE INITIALIZE XFC CODE
61	000012	SIP	-	10.	:WAIT FOR SECTOR/INDEX PULSE XFC CODE
62	000013	UREAD	-	11.	:READ UNIBUS MEMORY XFC CODE
63	000014	UWRITE	-	12.	:WRITE UNIBUS MEMORY XFC CODE
64	000015	ECC	=	13.	:DO ECC ON BUFFER XFC CODE
65	000016	MAINTR	-	14.	:SEND MAINT READ DATA XFC CODE
66	000017	MAINTW	=	15.	:RECEIVE MAINT WRITE DATA XFC CODE
67	000020	CVT	-	16.	:CONVERT TO PHYSICAL ADDRESS XFC CODE
68	000021	DONE	-	17.	:TERMINATE DM PROGRAM XFC CODE
69	000022	UPDATE	-	18.	:UPDATE DUP PROGRESS INDICATOR XFC
70					
71					
72					
73	000000	SHORTO	=	0.	:SHORT TIME OUT
74	000001	FRCPY	-	1.	:NUMBER OF F/RCT COPIES
75	000001	RTRY	-	1.	:NUMBER OF RETRIES
76	000001	LONGTO	-	1.	:LONG TIMEOUT
77	000002	ERRSYM	-	2.	:NUMBER OF ALLOWABLE ECC ERRORS
78	000002	FRCV	-	2.	:ERROR RECOVERY LEVELS SUPPORTED
79	000007	REVSEC	-	7.	:REVS/SECOND
80	000011	OFFS	-	9.	:GROUP OFFSET
81	000000	CYLBN	-	0.	:CYLINDERS IN LBN AREA
82	000002	STLBN	=	2.	:HIGH ORDER STARTING LBN
83	000003	STRBN	-	3.	:HIGH ORDER STARTING RBN
84	000002	STXBN	=	2.	:HIGH ORDER STARTING XBN
85	000000	STDBN	=	3.	:HIGH ORDER STARTING DBN
86	000001	STCYL	=	1.	:HIGH ORDER STARTING CYLINDER
87	000011	LBNTRK	=	9.	:NUMBER OF LBNS PER TRACK (512)
88	000004	RBNTRK	=	4.	:NUMBER OF RBNS PER TRACK
89	000021	XBNCYL	-	17.	:NUMBER OF CYLINDERS IN XBN AREA
90	000022	DBNCYL	=	18.	:NUMBER OF CYLINDERS IN DBN AREA
91	000012	LBNHOST	=	10.	:NUMBER OF LBN'S IN HOST AREA
92	000002	GRPCYL	=	2.	:GROUPS/CYLINDER
93	000003	TRKGRP	-	3.	:TRACKS/GROUP
94	000010	FCTS7	-	8.	:FCT SIZE IN SECTORS
95	000014	RCTS7	=	12.	:RCT SIZE IN LBN'S
96	000005	DATA	-	5.	:DATA PREAMBLE SIZE
97	000005	HEAD	-	5.	:HEADER PREAMBLE SIZE
98					
99					
100					
101					
102					
103	000001	BIT0	=	000001	
104	000002	BIT1	-	000002	
105	000004	BIT2	-	000004	
106	000010	BIT3	-	000010	
107	000020	BIT4	-	000020	
108	000040	BIT5	-	000040	
109	000100	BIT6	=	000100	
110	000200	BIT7	-	000200	
111	000400	BIT8	-	000400	
112	001000	BIT9	-	001000	
113	002000	BIT10	-	002000	
114	004000	BIT11	-	004000	

115	010000	BIT12	=	010000	
116	020000	BIT13	=	020000	
117	040000	BIT14	=	040000	
118	100000	BIT15	=	100000	
119		:			
120		:			
121		:			
122		:			
123		:			
124		:			
125	177400	HIBYTE	-	177400	;HIGH BYTE MASK
126	000377	LOBYTE	-	000377	;LOW BYTE MASK
127	177700	HI2BYTE	-	177700	;HIGH BYTE PLUS 2 BITS
128	177600	HI1BYTE	-	177600	;HIGH BYTE PLUS 1 BIT
129	007777	LO	-	007777	;ALL BUT HEADER CODE
130	177760	FCLR	-	177760	;CLEAR FOR FRCPY
131	170377	STCLR	-	170377	;CLEAR FOR STARTING BITS
132	007777	BUFMSK	-	007777	;BUFFER CLEAR MASK
133	000017	SIXTN	-	BIT0+BIT1+BIT2+BIT3	;FOR CHECK OF MOD 16 CYLINDER
134	000004	VLD	-	BIT2	;STATUS VALID BIT(1=VALID)
135	000010	VLD1	-	BIT3	;STATUS VALID BIT(1=VALID)
136	000200	PARITY	-	BIT7	;STATUS PARITY BIT(1=PARITY ERROR)
137	000400	PARIT1	-	BIT8	;REAL TIME ERROR(1=ERROR)
138		:			
139		:			
140		:			
141	000175	UNSEC	-	000175	;UNSUCCESSFUL COMPLETION
142		:			
143		:			
144		:			
145	000000	HD.LBN	-	000000	;GOOD LBN
146	060000	HD.RBN	-	060000	;GOOD RBN, PERHAPS UNUSED
147	030000	HD.REV	-	030000	;REVECTORED LBN
148	110000	HD.BAD	-	110000	;BAD BLOCK
149	050000	HD.PRIV	=	050000	;PRIMARY REVECTORED BLOCK
150	170000	HD.CLR	=	170000	;CLEAR HDR CODE
151	140000	HD.DBN	-	140000	;GOOD DBN
152	120000	HD.XBN	=	120000	;GOOD XBN
153	100000	PRMY	-	BIT15	;PRIMARY BIT IN FCT
154	010000	FBDHD	=	BIT12	;BAD HEADER CODE IN FCT
155		:			
156		:			
157		:			
158	000000	RC.FRE	=	000000	;FREE REPLACEMENT BLOCK
159	020000	RC.PRIV	=	020000	;PRIMARY REVECTOR
160	030000	RC.SND	=	030000	;SECONDARY REVECTOR
161	040000	RC.UNU	=	040000	;BAD REPLACEMENT BLOCK
162	100000	RC.NUL	=	100000	;NULL(FILL) BLOCK
163		:			
164		:			
165		:			
166	100000	RWRDY	-	BIT15	;READ/WRITE READY BIT POSITION
167	000002	ATTN	-	BIT1	;ATTENTION
168	000001	RCVRDY	-	BIT0	;RECEIVER READY
169		:			
170		:			
171		:			

```

172      :      GET STATUS BIT MASKS
173      :
174      000001      ST.RU      -      BIT0      :RUN/STOP SWITCH 1=IN
175      000002      ST.PS      =      BIT1      :PORT SWITCH 1=IN
176      000040      ST.DR      =      BIT5      :DIAGNOSTIC REQUESTED 1=YES
177      170000      ST.WP      -      BIT12+BIT13+BIT14+BIT15 :WRITE PROTECT SWITCH SU:0,1 1=IN
178      000020      ST.SR      =      BIT4      :SPINDLE READY 1=YES
179      001000      ST.DB      -      BIT9      :DIAG CYL ACCESS ENABLED 1=YES
180      002000      ST.FO      -      BIT10     :FORMAT CYL ACCESS ENABLED 1=YES
181      000004      ST.IN      -      BIT2      :DRIVE INITIALIZED 1=YES
182      000010      ST.WE      -      BIT3      :WRITE ERROR (WRITE LOCKED)
183      000020      ST.DF      -      BIT4      :DIAG FAILED - CANNOT DRIVE CLEAR
184      000374      ST.ERR      -      000374   :COMBINED CLEARABLE ERRORS BITS SET
185      000002      ST.ERB      -      2.       :ERROR BYTE OFFSET (3RD WORD)
186      :
187      :      OVERLAY TABLE OFFSETS
188      :
189      :
190      :
191      000021      OVCNT      -      17.       :NUMBER OF OVERLAYS
192      000003      OVLEN      -      3         :LENGTH OF 1 OVERLAY BLOCK
193      000000      LEN       -      0         :WORD COUNT OF OVERLAY
194      000001      HSTLO     =      1         :LOW ORDER UNIBUS ADDRESS
195      000002      HSTHI     =      2         :HI ORDER UNIBUS ADDRESS
196      000000      F1        =      0         :OFFSET INTO TABLE
197      000003      F2        =      3         :SECOND OVERLAY OFFSET INTO TABLE
198      000006      F3        =      6.        :THIRD OVERLAY OFFSET INTO TABLE
199      000011      F4        =      9.        :FOURTH OVERLAY OFFSET INTO TABLE
200      000014      F5        =      12.       :FIFTH OVERLAY OFFSET INTO TABLE
201      000017      F6        =      15.       :SIXTH OVERLAY OFFSET INTO TABLE
202      000022      F7        =      18.       :SEVENTH OVERLAY OFFSET INTO TABLE
203      000025      F8        =      21.       :EIGHTH OVERLAY
204      000030      F9        =      24.       :NINTH OVERLAY
205      000033      G2        =      27.       :ELEVENTH OVERLAY
206      000036      G3        =      30.       :TWELVTH OVERLAY
207      000041      G4        =      33.       :THIRTEENTH OVERLAY
208      000044      G5        =      36.       :FOURTEENTH OVERLAY
209      000047      G7        =      39.       :SIXTEENTH OVERLAY
210      000052      G8        =      42.       :SEVENTEENTH OVERLAY
211      000055      H1        -      45.       :NINETEENTH OVERLAY
212      000060      G1        -      48.       :TENTH OVERLAY
213      :
214      :      FLAG EQUATES
215      :
216      :
217      :
218      000001      FCTAVL     -      BIT0      :FCT AVAILABLE
219      000010      DBN        -      BIT3      :FORMAT DBN AREA
220      000100      REVECT     -      BIT6      :REVECTOR FLAG
221      001000      PRIM       -      BIT9      :PRIMARY FOUND FLAG
222      000002      FCTEMT     -      BIT11     :FCT EMPTY FLAG
223      000020      GOBAD      -      BIT4      :DO BEST GUESS IF FCT BAD
224      000040      RCINIT     -      BIT5      :RCT LAST BLOCK FIXED UP
225      000004      FCTBAD     -      BIT2      :FCT FOUND BAD (FOR STATS)
226      000200      MANU       -      BIT7      :MANUFACTURING FORMAT
227      000400      DLL        -      BIT8      :DOWN-LINE LOAD FLAG
228      002000      BSTGS      -      BIT10     :BEST GUESS FORMAT

```

229	004000	NDLL	=	BIT11	: ONLY WRITE FCT SCRATCH
230	020000	INIT	=	BIT13	: INIT RCT FLAG
231	040000	FIN	=	BIT14	: FORMAT FINISHED FLAG
232	010000	CHRDNE	=	BIT12	: CHARACTERISTICS DONE FLAG
233	100000	RTY	=	BIT15	: RETRY FLAG
234		:			
235		:			
236		FLAG1		EQUATES	
237		:			
238		:			
239	000001	WP	=	BIT0	: WRITE PROTECT FLAG
240	000002	RTYDN	=	BIT1	: RETRY DONE ON THIS SECTOR
241	000004	RPRIM	=	BIT2	: FLAG FOR PRIMARY GOOD EDC
242	000010	ERDN	=	BIT3	: FLAG FOR ERROR EXIT TRY
243	000020	DEAD	=	BIT4	: HOST GONE FLAG
244	000040	BDHD	=	BIT5	: BAD HEADER ON CHECK PASS READ
245	000100	RCINDN	=	BIT6	: RCT INIT DONE (WITH ONE FULL PAD BLK)
246		:			
247		:			
248				PHYSICAL CONVERSION XFC BLOCK EQUATES	
249		:			
250	000000	V1	=	0	: CYLINDER PARAMETER
251	000002	V2	=	2	: BLOCK NUMBER PARAMETER
252	000004	V3	=	4	: BLOCKS PER TRACK PARAMETER
253	000005	V4	=	5	: ONLY FOR RBN'S
254	000006	CYL	=	6	: CYLINDER RETURNED
255	000010	GRP	=	8	: GROUP RETURNED
256	000011	TRK	=	9	: TRACK RETURNED
257	000012	STSC	=	10	: STARTING SECTOR RETURNED
258	000013	INDSEC	=	11	: SECTOR FROM INDEX
259		:			
260		:			
261				DMBUF OFFSETS	
262		:			
263		:			
264	000001	DMOST	=	1	: STARTING OVERLAY ADDRESS
265	000003	DMUNIT	=	3	: DESIRED UNIT NUMBER
266	000004	DMFLG	=	4	: FLAG WORD
267	000016	DMBUFL	=	14	: BUFFER LENGTH
268		:			
269				FLAG FORMAT	
270		:			
271		:			
272		:			
273		:			
274		:			
275		:			
276		:			
277	000005	SER	=	5	: SERIAL NUMBER
278	000011	DAT	=	9	: DATE IN VAX/VMS FORMAT
279		:			
280	000002	LBD	=	2	: LBN'S BAD
281	000003	SND1	=	3	: SECONDARY REVECTORS
282	000004	RBD	=	4	: RCT BLOCKS BAD
283	000006	DBC	=	6	: DBN BLOCKS BAD
284	000010	XBD	=	8	: XBN BLOCKS BAD
285	000012	RTCNT	=	10	: RETRY COUNT

DO NOT USE FCT
 USE FCT
 CONTINUE IF FCT BAD
 STOP IF FCT BAD
 NO DLL OF FCT
 DLL FCT

```

286      000013      FCT      =      11.      ;FCT FLAG
287      :
288      :
289      :      FCT BLOCK OFFSETS
290      :
291      :
292      :
293      000002      FSER      =      2      ;SERIAL NUMBER
294      000001      INST      =      1      ;FORMATIN INSTANCE NUMBER
295      000016      C512      =      14.      ;COUNT OF USED 512 ENTRIES IN FCT
296      000012      FDAT      =      10.      ;MOST RECENT FORMAT DATE
297      000025      FCTFLG    =      21.      ;FCT FLAG FOR GOOD/BAD FCT
298      100000      NOFCT     =      BIT15    ;FLAG - 0 - FCT GOOD
299      :              ;      1 - FCT KNOW BAD
300      :
301      :
302      :      RCT BLOCK OFFSETS
303      :
304      :
305      000000      RSER      =      0      ;SERIAL NUMBER OFFSET
306      :
307      :      MISC      DEFINITIONS
308      :
309      :
310      :
311      000006      TWOB      =      6.      ;LENGTH OF 2 IMAGE ENTRIES
312      000011      THREB     =      9.      ;LENGTH OF 3 IMAGE ENTRIES
313      000005      RDLEN     =      5.      ;LENGTH OF CHECK PASS READ BLOCK
314      000002      ERLN     =      2.      ;LENGTH OF REVECTOR TABLE ENTRY
315      000004      REVLEN    =      4.      ;LENGTH OF SECONDARY TABLE
316      013400      RWCMD     =      013400  ;SDI READ COMMAND
317      122400      WRCMD     =      122400  ;SDI WRITE COMMAND
318      100000      RDCMD     =      100000  ;SIGNAL TO XFC NO MORE BLOCKS
319      000003      ERRLN     =      3.      ;LENGTH OF ERROR LIST ENTRY
320      040000      FULL      =      BIT14    ;FULL BBUFFER BIT
321      010000      ECCF      =      BIT12    ;ECC ERROR BIT
322      000200      RBNRPT    =      128.      ;NUMBER OF RBN COPIES IN REVECTOR
323      000003      IMLN     =      3.      ;LENGTH OF IMAGE BLOCK
324      020000      BD        =      BIT13    ;BAD FLAG IN IMAGE BUFFER
325      100000      LAST      =      BIT15    ;LAST FLAG IN IMAGE BUFFER
326      040000      RECIR     =      BIT14    ;RECIRCULATE IN FORMAT IMAGE BUFFER
327      126736      M512      =      126736  ;FCT MODE INDICATOR FOR 512
328      074161      M576      =      074161  ;FCT MODE INDICATOR FOR 576
329      100000      TIMVAL    =      32768.  ;TIMER LOOP VALUE
330      000010      MAXTRY    =      8.      ;FINAL SECONDARY WRITE RETRY LIMIT
331      007774      DUPOVL    =      7774    ;OVERLAY STARTING ADDRESS FROM DUP
332      :
333      :
334      :      STATUS OFFSETS
335      :
336      :
337      000000      MASK      =      0      ;SUBUNIT OFFSET MASK
338      000000      UID       =      0      ;UNIT NUMBER OFFSET
339      :
340      :
341      :      BUFFER DEFINITIONS
342      :      BUFFERS ARE 269 WORDS LONG AND ARE LOCATED AT LOC 4535(8)-7777(8)

```

343		:			
344	004535	BUF1	-	004535	;BUFFER 1 AT LOCATION 4535(8)
345	005152	BUF2	-	005152	;BUFFER 2 AT LOCATION 5152(8)
346	005567	BUF3	-	005567	;BUFFER 3 AT LOCATION 5567(8)
347	006204	BUF4	-	006204	;BUFFER 4 AT LOCATION 6204(8)
348	006621	BUF5	-	006621	;BUFFER 5 AT LOCATION 6621(8)
349	007275	BUF6	-	007275	;BUFFER 6 AT LOCATION 7236(8)
350		:			
351		:			
352		:			
353	004535	RDBUF	-	BUF1	;READ/WRITE BUFFER
354	005152	PBNBUF	-	BUF2	;BUFFER OF BAD PBN'S
355	005567	GDBLK	-	BUF3	;DATA FOR GOOD SECTOR
356	006204	PRMBUF	-	BUF4	;DATA PATTERN FOR PRIMARY REVECTOR
357	006204	REVBUFF	-	BUF4	;SECONDARY REVECTOR BUFFER
358	006621	CMDBUF	-	BUF5	;READ COMMAND BUFFERS
359	006621	RCTBUF	-	BUF5	;RCT BLOCK BUFFER
360	006621	RBNBUF	-	BUF5	;RBN FORMAT BUFFER
361	007275	IMAGE	-	BUF6	;FORMAT IMAGE BUFFER
362					;BUFFER EXCESS AFTER FORMAT IMAGE
363					;IS USED TO HOLD BLOCKS TO BE
364					;REVECTORED. MAX BLOCKS BEFORE
365					;REVECTOR ROUTINE IS CALLED VARIES
366					;WITH THE SIZE OF THE FORMAT BUFFER AREA
367	007775	BMAX	-	7775	;MAX BUFFER ADDRESS
368					
369					
370					

1		.SBTTL DATA STRUCTURES	
2		:	
3	000000	DMCODE KKDUB,0,714,13,255.	
4	000714 003047	ENTRY: JMP START	:JUMP TO START LOCATION
5		:	
6		DATA STRUCTURES	
7		:	
8		RETRY COUNTERS	
9		:	
10	000715 000000	UN.ERR: .WORD 0	:UNSUCCESSFUL CMD RETRY CNTR
11	000716 000000	UN.ERT: .WORD 0	:TRANSMISSION ERROR RETRY CNTR
12	000717 000000	UN.ERI: .WORD 0	:INITIALIZATION ERROR RETRY CNTR
13	000720 000000	UN.SEK: .WORD 0	:SEEK RETRY COUNT
14		:	
15		READ COMMAND BLOCK	
16		:	
17		:	
18		WRBLK:	
19	000721	RDBLK: .WORD 100000	:STATUS POINTER
20	000721 100000	.WORD 0	:POINTER TO DATA BUFFER
21	000722 000000	.WORD 0	:FIRST WORD OF EXPECTED HDR
22	000723 000000	.WORD 0	:SECOND WORD "
23	000724 000000	.WORD 0	:REAL-TIME SDI COMMAND
24	000725 000000	.WORD 0	:POINTER TO SDI BLOCK
25	000726 000000	.WORD 0	
26		:	
27		DUMMY DOUBLE WORDS AND DUMMY SDI COMMAND	
28		:	
29	000727 000200	HSLIM: .WORD 200	:HEADER CMP LIMIT
30	000730 001045	.WORD SCR-5	:POINTER TO SUBUNIT CHAR
31	000731 000000	DDUMMY: .WORD 0	:DUMMY DOUBLE WORD FOR ONE
32	000732 000000	.WORD 0	:BYTE OPERAND CONVERSION
33	000733	TEMP2:	:ALSO USE AS TEMP
34	000733	MULPC: .BLKW 2	:MULTIPLICATION BUFFER
35	000735 000000	.WORD 0	:RESERVED LOCATION (A+7)
36	000736	OFFSET:	:FOR EASIER REFERENCE
37	000736	TEMP: .BLKW 2	:USED FOR COMPUTATIONS
38		:	
39		:	
40		CURRENT UDA PORT	
41		:	
42		UNIT: .WORD 0	:SDI INTERCONNECT
43	000740 000000	:	
44		MESSAGE TABLES	
45		:	
46		:	
47	000741	CR.GST: MSG GST,1,ST,7	:GET STATUS
48	000745	CR.GCR: MSG GCR,1,CR,11.	:GET CHARACTERISTICS
49	000751	CR.GSR: MSG GSR,2,SCR,19.	:GET SUBUNIT CHARACTERISTICS
50	000755	CR.DIS: MSG DIS,2,ST,6	:UNLOAD DRIVE
51	000761	CR.RUN: MSG RUN,1,ST,6	:LOAD DRIVE
52	000765	CR.ACC: MSG ACC,3,ST,6	:SET FORMAT ACCESS
53	000771	CR.CLR: MSG DCLR,2,ST,6	:DRIVE CLEAR
54	000775	CR.SEK: MSG ISEEK,6,ST,6	:SEEK
55	001001	CR.RCL: MSG IRECAL,1,ST,6	:RECALIBRATE
56	001005	CR.ERV: MSG ERECOV,2,ST,6	:ERROR RECOVERY COMMAND
57	001011	CR.ONL: MSG ONLINE,2,ST,6	:ONLINE COMMAND

58					
59					
60				MESSAGES AND COMMANDS	
61	001015	004400	GST:	.WORD 000011*256.	:GET STATUS COMMAND
62	001016	103400	GCR:	.WORD 000207*256.	:GET CHARACTERISTICS
63	001017	104000	GSR:	.WORD 00210*256.	:GET SUBUNIT CHARACTERISTICS
64	001020	000000		.WORD 0	:SUBUNIT MASK
65	001021	102000	DIS:	.WORD 000204*256.	:UNLOAD DRIVE
67	001022	000000		.WORD 0	:NO SPIN DOWN MODIFIER
71	001023	006000	RUN:	.WORD 000014*256.	:INITIATE LOAD
72	001024	100400	ACC:	.WORD 000201*256.	:ACCESS DIAG AND FMT CYL
73	001025	003006		.WORD 3006	:MASK BYTE/MODE BYTE
74	001026	002400	DCLR:	.WORD 000005*256.	:DRIVE CLEAR
75	001027	000374		.WORD 374	:BITS TO CLEAR
76	001030		ST:	.BLKW 7	:STATUS MESSAGE BUFFER
77	001037		CR:	.BLKW 11.	:CHARACTERISTICS MESSAGE BUFF
78	001052		SCR:	.BLKW 19.	:SUBUNIT CHARACTERISTICS BUFF
79	001075	005000	ISEEK:	.WORD 000012*256.	:INITIATE SEEK
80	001076	000000		.WORD 0	:
81	001077	000000		.WORD 0	:
82	001100	000000		.WORD 0	:
83	001101	107000	IRECAL:	.WORD 000216*256.	:INITIATE RECAL
84	001102	003000	ERECOV:	.WORD 000006*256.	:ERROR RECOVERY COMMAND
85	001103	000000		.WORD 0	:RECOVERY LEVEL
86	001104	105400	ONLINE:	.WORD 000213*256.	:ONLINE COMMAND
87	001105	000377		.WORD 377	:COMMAND TIMEOUT (SECS)
88					
89				DISK LOCATION POINTERS	
90					
91	001106	000000	CURRBN:	.WORD 0	:CURRENT RBN
92	001107	000000		.WORD 0	:
93	001110	000000	CURPBN:	.WORD 0	:CURRENT PBN
94	001111	000000		.WORD 0	:
95	001112	000000	CURTRK:	.WORD 0	:CURRENT TRACK
96	001113	000000	CURBN:	.WORD 0	:CURRENT BLOCK NUMBER
97	001114	000000		.WORD 0	:
98	001115		CURLBN:		:FOR RCT INIT
99	001115	000000	CURXBN:	.WORD 0	:CURRENT XBN NUMBER
100	001116	000000		.WORD 0	:
101	001117	000000	HOLDBN:	.WORD 0	:BLOCK NUMBER OF FIRST BLOCK ON CYL
102	001120	000000		.WORD 0	:
103	001121	000000	HOLRBN:	.WORD 0	:BLOCK NUM OF FIRST RBN ON CYLINDER
104	001122	000000		.WORD 0	:
105	001123	000000	HOLDPN:	.WORD 0	:PBN OF FIRST SECOTR ON TRACK
106	001124	000000		.WORD 0	:
107	001125	000000	CYLNUM:	.WORD 0	:CURRENT CYLINDER NUMBER
108	001126	000000		.WORD 0	:
109	001127	000000	SECTRK:	.WORD 0	:SECTORS/TRACK
110	001130	000000		.WORD 0	:
111	001131	000000	SECTCY:	.WORD 0	:SECTORS/CYLINDER
112	001132	000000		.WORD 0	:
113	001133		LBNLBN:	.BLKW 2	:LBN'S IN LBN AREA
114	001135		RBNLBN:	.BLKW 2	:RBN'S IN LBN AREA
115	001137		XBNSEC:	.BLKW 2	:SECTORS IN LBN AREA
116	001141		TRKCYL:	.BLKW 2	:TRACKS/CYLINDER
117	001143		LBNCYL:	.BLKW 2	:NUMBER OF LBN CYLINDERS
118	001145		LBNPCY:	.BLKW 2	:LBN'S/CYLINDER

119	001147		RBNPCY: .BLKW	2	:RBN'S/CYLINDER
120	001151	000000	REVRBN: .WORD	0	:REVECTORED RBN NUMBER
121	001152	000000		0	:
122	001153	000000	CURFOVL: .WORD	0	:CURRENT OVERLAY
123	001154		HGHPBN: .BLKW	2	:HIGHEST PBN IN LBN AREA
124			:		
125			:		
126			:		
127			STACK		
128			:		
129			:		
130			:		
131	001156			31.	:STACK
132	001215	000000	STACK: .WORD	0	:TOP OF STACK
133	001216	000000	STCKSV: .WORD	0	:STACK PTR TEMP SAVE
134			:		
135			:		
136			MISC	DEFINITIONS	
137			:		
138			:		
139	001217	000000	FLAG: .WORD	0	:FLAG WORD
140	001220	000000	FLAG1: .WORD	0	:FLAG WORD
141	001221	000000	ERFLAG: .WORD	0	:RE-FORMAT FLAG
142	001222	000000	WRFLG: .WORD	0	:RCT WRAP FLAG
143	001223	000000	BADPBN: .WORD	0	:POINTER TO PBN TAB ENTRY
144	001224	000000	ERRBUF: .WORD	0	:POINTER TO BEGINNING OF REVECTOR BUFFER
145	001225	000000	EMAX: .WORD	0	:MAX NUMBER OF REVECTORS BEFORE
146					:RCT UPDATE ROUTINE IS CALLED
147	001226	000000	ERR: .WORD	0	:NUMBER OF SECTORS IN ERROR
148	001227	000000	HOLD: .WORD	0	:DOUBLE WORD TEMP STORAGE
149	001230	000000		0	
150	001231	000000	EIMAGE: .WORD	0	:ADDRESS OF END IMAGE BLOCK
151	001232	000000	STARIT: .WORD	0	:STARTING ADDRESS OF THIS PASS
152	001233	000000	SKPCNT: .WORD	0	:OFFSET FOR FIRST READ CHECK
153	001234	000000	TBLK: .WORD	0	:INTERLEAVE
154					:6 - B!-LEAVE
155					:9 - TRI-LEAVE
156	001235		RCTTOT: .WORD	2950.	:ALSO RCT TOTAL HOLDING AREA
157	001235	005606	CUTOFF: .WORD	0	:SECT/SECOND CUTOFF
158	001236	000000		0	:DOUBLE WORD
159	001237	000000	FCTNT: .WORD	0	:COUNT OF USED FCT ENTRIES FOR FORMATTING
160	001240		FCTFMT: .BLKW	2	:SIZE OF ONE FCT COPY
161	001242		RCTFMT: .BLKW	2	:SIZE OF ONE RCT COPY
162	001244	000000	FCTCPY: .WORD	0	:NUMBER OF FCT COPIES
163	001245	000000	NEXT1: .WORD	0	:MULTI-COPY COUNTER
164	001246	000105	INI: .WORD	69.	:INITIAL VALUE FOR EDC
165	001247	000400	CNT: .WORD	256.	:COUNT FOR EDC
166	001250	000100	LTO: .WORD	100	:LONG TIMEOUT
167	001251	002000	STO: .WORD	1024.	:SHORT TIMEOUT (IN MILLESECS)
168	001252	000000	ERPNT: .WORD	0	:REVECTOR LIST POINTER
169	001253	000000	BUFPNT: .WORD	0	:BUFFER POINTER FOR FCT READ
170	001254	000000	REVCNT: .WORD	0	:REVECTOR COUNT
171	001255	000000	FCTPTR: .WORD	0	:POINT TO CURRENT LOCATION IN FCT BLOCK
172	001256	000001	FCTCNT: .WORD	1	:CURRENT FCT BLOCK
173	001257	000000		0	
174	001260	000000	FCTNPD: .WORD	0	:NON-PAD FCT BLOCKS
175	001261	000000	RCTLBN: .WORD	0	:LBN'S IN RCT

176	001262	000000			MNCNT: .WORD 0	:USED FCT ENTRIES
177	001263				DMBUF: .BLKW 14.	:MAINTENANCE BUFFER
178	001301				DATE: .BLKW 4	:DATE BUFFER
179	001305				SERNUM: .BLKW 4	:SERIAL NUMBER
180	001311	000000			FCTREV: .WORD 0	:FCT ENTRIES AT CERTAIN POINTS
181	001312	000000			LBNBAD: .WORD 0	:TOTAL REVECTORED LBN'S
182	001313	000000			RCTBAD: .WORD 0	:TOTAL BAD RCT BLOCKS
183	001314	000000			DBBAD: .WORD 0	:TOTAL DBN BAD BLOCKS
184	001315	000000			XBBAD: .WORD 0	:TOTAL LBN BAD BLOCKS
185	001316	030000			CYLMSG: .WORD 30000	:DUP CODE
186	001317	103	040	000	.ASCIZ 'C'	:SIGNAL CYLINDER COMPLET MESSAGE
187	001321	010000			FMSG: .WORD 10000	:DUP CODE
188	001322	106	040	000	.ASCIZ 'F'	:FCT DLL MSG
189	001324	050000			EMSG: .WORD 50000	:DUP CODE
190	001325	105	040	000	.ASCIZ 'E'	:ERROR MESSAGE
191	001327	040000			DONMSG: .WORD 40000	:DUP CODE
192	001330	104	040	000	.ASCIZ 'D'	:DONE MESSAGE
193	001332	000000			IMSTAR: .WORD 0	:POINTER TO START OF IMAGE
194	001333	000000			HPREA: .WORD 0	:HEADER PREAMBLE LENGTH
195	001334	000000			DPREA: .WORD 0	:DATA PREAMBLE LENGTH
196	001335	000000			ST.LBN: .WORD 0	:STARTING LBN BITS
197	001336	000000			ST.RBN: .WORD 0	:STARTING RBN BITS
198	001337	000000			ST.XBN: .WORD 0	:STARTING XBN BITS
199	001340	000000			ST.DBN: .WORD 0	:STARTING DBN BITS
200					:	
201					:	
202					:	
203					:	
204					:	
205					:	
206					:	
207					:	
208	001341	001103			OVLTBL: .WORD OVL.F1	:LENGTH OF FIRST OVERLAY
209	001342	010466			.WORD OVS.F1	:LOW ORDER HOST ADDRESS
210	001343	000000			.WORD 0	:HIGH ORDER HOST ADDRESS
211	001344	001416			.WORD OVL.F2	:LENGTH OF SECOND OVERLAY
212	001345	013570			.WORD OVS.F2	:LOW ORDER HOST ADDRESS
213	001346	000000			.WORD 0	:HIGH ORDER HOST ADDRESS
214	001347	000423			.WORD OVL.F3	:LENGTH OF THIRD OVERLAY
215	001350	020746			.WORD OVS.F3	:LOW ORDER HOST ADDRESS
216	001351	000000			.WORD 0	:HIGH ORDER HOST ADDRESS
217	001352	000647			.WORD OVL.F4	:LENGTH OF FOURTH OVERLAY
218	001353	022014			.WORD OVS.F4	:LOW ORDER HOST ADDRESS
219	001354	000000			.WORD 0	:HIGH ORDER HOST ADDRESS
220	001355	001202			.WORD OVL.F5	:LENGTH OF FIFTH OVERLAY
221	001356	024336			.WORD OVS.F5	:LOW ORDER HOST ADDRESS
222	001357	000070			.WORD 0	:HIGH ORDER HOST ADDRESS
223	001360	000320			.WORD OVL.F6	:LENGTH OF SIXTH OVERLAY
224	001361	030054			.WORD OVS.F6	:LOW ORDER HOST ADDRESS
225	001362	000000			.WORD 0	:HIGH ORDER HOST ADDRESS
226	001363	000445			.WORD OVL.F7	:LENGTH OF SEVENTH OVERLAY
227	001364	026742			.WORD OVS.F7	:LOW ORDER HOST ADDRESS
228	001365	000000			.WORD 0	:HIGH ORDER HOST ADDRESS
229	001366	000625			.WORD OVL.F8	:LENGTH OF EIGHTH OVERLAY
230	001367	016624			.WORD OVS.F8	:LOW ORDER HOST ADDRESS
231	001370	000000			.WORD 0	:HIGH ORDER HOST ADDRESS
232	001371	000202			.WORD OVL.F9	:LENGTH OF NINTH OVERLAY

OVERLAY POINTERS
 NOTE:
 WHEN ADDING AN ENTRY TO THIS TABLE EQUATE
 OVCNT MUST BE INCREMENTED

233 001372 035330
 234 001373 000000
 235 001374 000052
 236 001375 030714
 237 001376 000000
 238 001377 000214
 239 001400 031040
 240 001401 000000
 241 001402 001720
 242 001403 031470
 243 001404 000000
 244 001405 000237
 245 001406 035734
 246 001407 000000
 247 001410 000336
 248 001411 012674
 249 001412 000000
 250 001413 000224
 251 001414 020276
 252 001415 000000
 253 001416 000302
 254 001417 023532
 255 001420 000000
 256 001421 001376
 257 001422 005472
 258 001423 000000
 259 001424

.WORD OVS.F9
 .WORD 0
 .WORD OVL.G2
 .WORD OVS.G2
 .WORD 0
 .WORD OVL.G3
 .WORD OVS.G3
 .WORD 0
 .WORD OVL.G4
 .WORD OVS.G4
 .WORD 0
 .WORD OVL.G5
 .WORD OVS.G5
 .WORD 0
 .WORD OVL.G7
 .WORD OVS.G7
 .WORD 0
 .WORD OVL.G8
 .WORD OVS.G8
 .WORD 0
 .WORD OVL.H1
 .WORD OVS.H1
 .WORD 0
 .WORD OVL.G1
 .WORD OVS.G1
 .WORD 0
 OVLBLK: .BLKW 4

:LOW ORDER HOST ADDRESS
 :HIGH ORDER HOST ADDRESS
 :LENGTH OF EIGHTH OVERLAY
 :LOW ORDER HOST ADDRESS
 :HIGH ORDER HOST ADDRESS
 :LENGTH OF EIGHTH OVERLAY
 :LOW ORDER HOST ADDRESS
 :HIGH ORDER HOST ADDRESS
 :LENGTH OF EIGHTH OVERLAY
 :LOW ORDER HOST ADDRESS
 :HIGH ORDER HOST ADDRESS
 :LENGTH OF FOURTEENTH OVERLAY
 :LOW ORDER HOST ADDRESS
 :HIGH ORDER HOST ADDRESS
 :LENGTH OF SIXTEENTH OVERLAY
 :LOW ORDER HOST ADDRESS
 :HIGH ORDER HOST ADDRESS
 :LENGTH OF SEVENTEENTH OVERLAY
 :LOW ORDER HOST ADDRESS
 :HIGH ORDER HOST ADDRESS
 :LENGTH OF SEVENTEENTH OVERLAY
 :LOW ORDER HOST ADDRESS
 :HIGH ORDER HOST ADDRESS
 :LENGTH OF SEVENTEENTH OVERLAY
 :LOW ORDER HOST ADDRESS
 :HIGH ORDER HOST ADDRESS
 :FOR BUFFER OVERLAYS

```

1
2
3
4
5
6 001430 001045      CONBLK: .WORD 1045      ;STARTING CYL PLUS LBN CYL
7 001431 000000      .WORD 0      ;HIGH ORDER
8 001432 000000      .WORD 0      ;LOW ORDER XBN-X
9 001433 000000      .WORD 0      ;SECTORS PER TRACK
10 001434 000040      .WORD 40     ;HIGH - BOTH FILLED IN DYNAMICALLY
11 001435              .BLKW 6     ;FILLED IN BY XFC
12
13
14
15
16 001443 000377      NUM: .WORD 255.      ;NUMBER OF WORDS IN PATTERN
17 001444 004536      CBUF: .WORD RDBUF+1  ;BUFFER TO COMPARE(NOT FIRST WORD)
18 001445 155555      FWRD: .WORD 155555   ;FIRST WORD OF PATTERN
19 001446 133333      SWRD: .WORD 133333   ;SECOND WORD OF PATTERN
20 001447 155555      TWRD: .WORD 155555   ;THIRD WORD OF PATTERN
21 001450 167356      DWRD: .WORD 167356   ;DIAGNOSTIC WORD(FIRST IN SECTOR)
22
23
24 001451 125677      EDC: .WORD 125677     ;EDC FOR ABOVE DATA PATTERN
25 001452 052100      BADED: .WORD 52100    ;BAD EDC FOR RBN BLOCKS
26
27
28
29
30 001453 000000      ERRCNT: .WORD 0      ;FOR TESTING VERIFICATION
31 001454 000000      SECCNT: .WORD 0      ;SECTOR COUNT
32 001455 000000      N: .WORD 0          ;NUMBER OF ORIGINAL CHECK PASS READ
33 001456 000000      N1: .WORD 0         ;NUMBER OF ERROR READS
34 001457 000000      NN1: .WORD 0        ;DITTO
35 001460 000000      CNTCYL: .WORD 0      ;NUMBER OF CYLINDERS TO FORMAT
36 001461 000000      .WORD 0
37 001462 000000      HD.CUR: .WORD 0     ;CURRENT HEADER
38 001463 000000      CURGRP: .WORD 0     ;CURRENT GROUP
39 001464 000000      GRPCNT: .WORD 0     ;NUMBER OF GROUPS TO DO
40 001465 000000      TRKCNT: .WORD 0     ;NUMBER OF TRACKS TO DO
41 001466 000001      ONE: .WORD 1        ;WORD CONSTANT OF 1
42 001467 000000      .WORD 0          ;DOUBLE WORD
43 001470 000002      TWOC: .WORD 2       ;WORD CONSTANT OF 2
44 001471 000000      .WORD 0          ;DOUBLE WORD
45 001472 000000      SNDCNT: .WORD 0     ;COUNT OF SECONDARY REVECTORS
46 001473 000000      RTYCNT: .WORD 0     ;COUNT OF SECTORS RETRYED
47 001474 000000      UPDPNT: .WORD 0     ;POINTER FOR RCT UPDATE
48 001475 000000      TOTRCT: .WORD 0     ;TOTAL LBN'S IN RCT'S
49 001476 000000      .WORD 0
50 001477 000000      RCTCNT: .WORD 0     ;CURRENT RCT BLOCK
51 001500 000000      PCNT: .WORD 0       ;PBN BLOCK COUNTER
52 001501 000000      COUNT: .WORD 0     ;COUNT FOR XBN DLL
53 001502 000005      RETRY: .WORD 5     ;RETRIES FROM SDI
54 001503 000000      RECOV: .WORD 0     ;RECOVERY LEVELS SUPPORTED BY DRIVER
55 001504 000000      TMPTRY: .WORD 0     ;TEMP FOR RETRY COUNT
56 001505 000000      RECTMP: .WORD 0     ;TEMP FOR ERROR RECOVERY LEVEL

```

```

1      .SBTTL  MATH SUBROUTINES
2      :
3      :
4      SUBROUTINES
5      :
6      :
7      DOUBLE ADD ROUTINE
8      :
9      INPUT PARAMETERS
10     :
11     R3      CONTAINS POINTER TO OPERAND 1
12     :
13     R4      CONTAINS POINTER TO OPERAND 2
14     :
15     OUTPUT PARAMETER
16     :
17     R4      CONTAINS THE RESULT
18     :
19 001506 100465 DADD:  MOV    R5,-(SP)      ;SAVE A SCRATCH REGISTER
20 001507 100461      MOV    R1,-(SP)      ;SAVE ANOTHER
21 001510 104235      MOV    (R3)+,R5      ;GET LOW ORDER OPERAND
22 001511 104131      MOV    (R3),R1      ;GET HIGH ORDER OPERAND
23 001512 105245      ADD     (R4)+,R5      ;ADD LOW ORDER OPERAND
24 001513 071515      BCC     DADD1        ;BRANCH IF NO CARRY
25 001514 115401      INC     R1           ;ADD ONE TO HIGH IF CARRY
26 001515 105141 DADD1:  ADD     (R4),R1      ;ADD OP 2
27 001516 100141      MOV    R1,(R4)      ;SAVE HIGH ORDER
28 001517 100445      MOV    R5,-(R4)      ;SAVE LOW ORDER
29 001520 104261      MOV    (SP)+,R1      ;RESTORE R1
30 001521 104265      MOV    (SP)+,R5      ;RESTORE R5
31 001522 117403      DEC     R3           ;RESTORE R3
32 001523 000000      RETURN
33     :

```

```

1
2 001524      DSUB:      ; **
3                  ; DOUBLE PRECISION FIXED POINT SUBTRACT ROUTINE
4
5                  ; INPUTS:
6                  ; R3  POINTER TO OPERAND 1 (SUBTRAHEND)
7                  ; R4  POINTER TO OPERAND 2 (MINUEND)
8
9                  ; OUTPUT:
10                 ; R4  POINTER TO RESULT WHERE (R4) = (R4) - (R3)
11                 ; --
12
13
14 001524      PUSH      R1,R5          ; SAVE REGISTERS
15 001526      MOV       (R4)+,R5      ; GET LO ORDER MINUEND
16 001527      MOV       (R4),R1       ; GET HI ORDER MINUEND
17 001530      SUB       (R3),R5       ; SUBTRACT LOW ORDER OPERANDS
18 001531      BCC      10$           ; POSITIVE RESULT
19 001532      DEC       R1            ; BORROW FROM HI ORDER OPERAND
20 001533      SUB       1(R3),R1      ; SUBTRACT HI ORDER OPERANDS
21 001535      MOV       R1,(R4)       ; STORE HI ORDER RESULT
22 001536      MOV       R5,-(R4)     ; STORE LO ORDER RESULT
23 001537      POP       R5,R1        ; RESTORE REGISTERS
24 001541      RETURN
25
26 001542      DMUL:      ; **
27                  ; DOUBLE PRECISION FIXED POINT MULTIPLY ROUTINE
28
29                  ; INPUTS:
30                  ; R3 = POINTER TO MULTIPLIER (SINGLE PRECISION)
31                  ; R4 = POINTER TO MULTIPLICANT (DOUBLE PRECISION)
32
33                  ; OUTPUT:
34                  ; R4  POINTER TO RESULT WHERE (R4) = (R4) * (R3)
35                  ; --
36
37 001542      PUSH      R0,R3          ; SAVE R0 & R3
38 001544      MOV       (R3),R0       ; GET MULTIPLIER
39 001545      BNE      5$              ; MULTIPLIER NOT = 0
40 001546      MOV       R0,(R4)       ; LOAD LO ORDER RESULT
41 001547      MOV       R0,1(R4)      ; LOAD HI ORDER RESULT
42 001551      BR       20$           ; RETURN
43 001552      MOV       (R4),MULPC    ; COPY MULTIPLICANT FOR DADD
44 001554      MOV       1(R4),MULPC+1
45 001557      MOV       #MULPC,R3
46 001561      DEC       R0            ; ADJUST MULTIPLIER FOR *1
47 001562      BEQ      20$           ; MULTIPLIER = 0, EXIT
48 001563      CAL      DADD          ; PERFORM ITERATIVE ADDS
49 001564      BR       10$
50 001565      POP      R3,R0        ; RESTORE R0 & R3
51 001567      RETURN
52
53 001570      DDIV:      ; **
54                  ; DOUBLE PRECISION FIXED POINT DIVIDE
55                  ; INPUTS:
56                  ; R3 = POINTER TO DIVISOR (SINGLE PRECISION)
57

```

```

58                                     :      HIGH ORDER WORD MUST BE ZERO
59                                     :      R4 = POINTER TO DIVIDENT (DOUBLE PRECISION)
60                                     :
61                                     :      OUTPUT:
62                                     :      R3 = POINTER TO REMAINDER
63                                     :      R4 = POINTER TO QUOTIENT
64                                     :
65                                     :      NOTE - THE CASES WHERE EITHER THE DIVISOR OR DIVIDENT ARE ZERO,
66                                     :      ARE NOT CONSIDERED IN THIS ROUTINE.
67                                     :      --
68
69 001570      PUSH      R0,R1,R2,R5      : SAVE REGISTERS
70 001574      CLR      R0              : CLR LO ORDER QUOTIENT REG
71 001575      CLR      R1              : CLR HI ORDER QUOTIENT REG
72 001576      MOV      (R3),R2         : GET DIVISOR
73 001577      MOV      1(R4),R5        : GET HI ORDER DIVIDENT
74 001601      BNE      20$             : DIVISOR NOT = 0
75 001602      MOV      (R4),R5        : GET LO ORDER DIVIDENT
76 001603      CMP      R5,R2          : IS DIVIDENT < DIVISOR ?
77 001604      BCC      20$            : NO, CONTINUE
78 001605      BR       30$            : YES, STOP
79 001606      CALL     DSUB           : SYNTHESIZE DIVIDE
80 001607      ADD      #1,R0          : INCR LO ORDER QUOTIENT
81 001611      BCC      10$            : DID NOT OVERFLOW
82 001612      INC      R1             : ADJUST HI ORDER QUOTIENT
83 001613      BR       10$
84 001614      MOV      (R4),R5        : GET REMAINDER
85 001615      MOV      R0,(R4)        : LOAD LO ORDER QUOTIENT
86 001616      MOV      R1,1(R4)      : LOAD HI ORDER QUOTIENT
87 001620      MOV      R5,(R3)      : LOAD REMAINDER
88 001621      POP      R5,R2,R1,R0   : RESTORE REGISTERS
89 001625      RETURN
90

```

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14 001626 10046 '
15 001627 10046
16 001630 104171
17 001631 104447 000001
18 001633 104637 000001
19 001635 011651
20 001636 041644
21 001637 104261
22 001640 104267
23 001641 106204 000000
24 001643 000000
25 001644 104261
26 001645 104267
27 001646 106204 077777
28 001650 000000
29 001651 106131
30 001652 051657
31 001653 104261
32 001654 104267
33 001655 16044
34 001656 000000
35 001657 106131
36 001660 041644
37 001661 001637

:
: DOUBLE COMPARE
:
: INPUT PARAMETERS
:
: R3     CONTAINS A POINTER TO THE FIRST OPERAND
:
: R4     CONTAINS A POINTER TO THE SECOND OPERAND
:
: OUTPUT PARAMETERS
:
: THE FLAGS ARE SET AS IF A SINGLE PRECISION 'CMP' HAD OCCURED
:
DCMP:  MOV     R0,-(SP)      ;SAVE R0 FOR USE AS SCRATCH
      MOV     R1,-(SP)      ;SAVE R1 FOR USE AS SCRATCH
      MOV     (R4),R1       ;GET LOW ORDER DEST OPERAND
      MOV     1(R4),R0      ;GET HIGH ORDER DEST OPERAND
      CMP     1(R3),R0      ;DO ACTUAL HIGH ORDER TEST
      BEQ     DCMP1         ;GO DO ADDITIONAL TESTING
      BCC     DCMP2         ;SRC HI, CLEAN UP AND RTN
DCMP4:  MOV     (SP)+,R1     ;RESTORE R1
      MOV     (SP)+,R0      ;RESTORE R0
      CMP     #0,R4        ;SET CONDITION CODES - SRC LSS
      RETURN                    ;AND RETURN
DCMP2:  MOV     (SP)+,R1     ;RESTORE R1
      MOV     (SP)+,R0      ;RESTORE R0
      CMP     #077777,R4    ;SET CONDITION CODES - DST LSS
      RETURN                    ;AND RETURN
DCMP1:  CMP     (R3),R1      ;TEST LOW ORDER
      BNE     DCMP3         ;BRANCH IF NOT EQUAL
      MOV     (SP)+,R1      ;RESTORE R1
      MOV     (SP)+,R0      ;RESTORE R0
      CMP     R4,R4        ;SET CONDITION CODES - EQUAL
      RETURN                    ;AND RETURN
DCMP3:  CMP     (R3),R1      ;COMPARE AGAIN
      BCC     DCMP2         ;BRANCH ON SRC HI
      BR      DCMP4         ;BRANCH ON SRC LOW

```

```

1      .SBTTL SDI SUBROUTINES
2
3
4      GET STATUS
5
6      OUTPUT PARAMETERS
7
8      CLEARS DRIVE STATUS AND GETS CHARACTERISTICS
9      IF NOT ALREADY RECEIVED
10
11
12 001662 GSTATS: PUSH R3 ;PUSH R3
13 001663      PUSH R5 ;PUSH R5
14 001664 104203 000741 STATST: MOV #CR.GST,R3 ;POINT TO GET STATUS TABLE
15 001666 022016      CALL TALK ;GET STATUS
16 001667 104207 001030      MOV #ST,R0 ;POINT TO SUBUNIT CHARACTERISTICS
17 001671 104673 000002      MOV ST.ERB(R0),R3 ;GET ERROR BYTE
18 001673 103203 177420      BIC #ST.DF+HIBYTE,R3 ;CLEAR HIGH BYTE AND DF BIT
19 001675 115003      TST R3 ;ANY NEED TO ISSUE DRIVE CLEAR ?
20 001676 011702      BEQ STSK1 ;NOPE - SKIP IT
21 001677 104030 001027      MOV R3,DCLR+1 ;STORE MASK IN DRIVE CLR COMMAND
22 001701 022243      CALL CLEAR ;DO A DRIVE CLEAR
23 001702 104205 001030 STSK1: MOV #ST,R5 ;POINT TO STATUS BLOCK
24 001704 105205 000001      ADD #1,R5 ;POINT TO 2ND WORD
25 001706 104253      MOV (R5)+,R3 ;GET FIRST WORD OF STATUS
26 001707 104202 000001      MCV #1,R2 ;ERROR SUBCODE IN CASE
27 001711 102203 000040      BIT #ST.DR,R3 ;IS DRIVE IN DIAGNOSTIC MODE
28 001713 052013      BNE STPNIC ;YES, WE LOSE
29 001714 115402      INC R2 ;ERROR SUBCODE 2
30 001715 102203 000001      BIT #ST.RU,R3 ;IS RUN STOP SWITCH OUT
31 001717 012013      BEQ STPNIC ;YES, LOSE AGAIN
32 001720 104202 000004      MOV #4,R2 ;SUBCODE
33 001722 102203 000002      BIT #ST.PS,R3 ;PORT SWITCH OUT ?
34 001724 012013      BEQ STPNIC ;YES - DIE PAINFULLY
35 001725 104032      MOV R3,R2 ;GET STATUS MODE BYTE
36 001726 110702      SWAB R2 ;SWITCH WRITE PROTECT TO LOW BYTE
37 001727 102302 001020      BIT GSR+1,R2 ;WRITE PROTECTED ?
38 001731 011745      BEQ SRCK ;IF NOT CHECK IF SPINNING
39 001732 104202 000003      MOV #3,R2 ;IN CASE IT'S FATAL
40 001734 102200 000001 001220      BIT #WP,FLAG1 ;BEEN HERE ONCE ?
41 001737 052013      BNE STPNIC ;YUP - GIVE UP
42 001740 101200 000001 001220      BIS #WP,FLAG1 ;SET BEEN HERE FLAG
43 001743 022235      CALL ACCESS ;TRY TO RESET IT
44 001744 001664      BR STATST ;AND SEE IF IT WORKED
45 001745 102203 000020      SRCK: BIT #ST.SR,R3 ;IS PACK SPINNING?
46 001747 051752      BNE STFORM ;YES, TEST FOR FORMAT ENABLE
47 001750 022160      CALL LOAD ;NO, SPIN PACK
48 001751 001664      JMP STATST ;SEE IF ANYTHING CHANGED
49 001752 102203 002000      STFORM: BIT #ST.FO,R3 ;IS FORMATTING ENABLED?
50 001754 051757      BNE STDIAG ;YES, TEST FOR DIAG ACCESS
51 001755 022235      CALL ACCESS ;NO, SET UP DIAG/FORM ACCESS
52 001756 001664      JMP STATST ;SEE IF ANYTHING CHANGED
53 001757 102203 001000      STDIAG: BIT #ST.DB,R3 ;IS DIAG CYL ACCESS ALLOWED
54 001761 051764      BNE STWLK ;YES, CHECK FOR ERRORS
55 001762 022235      CALL ACCESS ;NO, SET UP DIAG/FORM ACCESS
56 001763 001664      JMP STATST ;SEE IF ANYTHING CHANGED
57 001764 104153      STWLK: MOV (R5),R3 ;GET SECOND STATUS WORD

```

58	001765	102203	000010	.	BIT	#ST.WE,R3	:ANY WRITE ENABLE ERRORS
59	001767	011772			BEQ	CHAR	:NO, GET CHARACTERISTICS
60	001770	022235			CALL	ACCESS	:TRY ENABLING LOGICAL WRITE
61	001771	001664			BR	STATST	:AND CHECK WORLD AGAIN
62	001772	102200	010000	001217	CHAR: BIT	#CHRDNE,FLAG	:CHARACTERISTICS ALREADY RECEIVED
63	001775	052007			BNE	STSKP	:YUP - NO NEED TO GET AGAIN
64	001776	104203	000745		MOV	#CR.GCR,R3	:POINT TO GET CHAR CMD TABLE
65	002000	022016			CALL	TALK	:GET CHARACTERISTICS
66	002001	104203	000751		MOV	#CR.GSR,R3	:GET SUBUNIT CHARACTERISTICS
67	002003	022016			CALL	TALK	:GET THEM
68	002004	101200	010000	001217	BIS	#CHRDNE,FLAG	:SET CHAR DONE BIT
69	002007				STSKP: POP	R5	:RESTORE R5
70	002010				POP	R3	:RESTORE R3
71	002011	114001			CLR	R1	:CLEAR ERROR INDICATOR
72	002012	000000			RETURN		:RETURN TO CALLER
73	002013	104201	000001		STONIC: MOV	#1,R1	:INDICATE STATUS FAILURE
74	002015	022552			CALL	FRRMNT	:SEND ERROR MSG AND QUIT

```

1
2
3
4
5
6
7
8
9
10
11
12
13 002016
14 002017
15 002020 104663 000001
16 002022 104237
17 002023 104231
18 002024 104302 000740
19 002026 060004
20 002027 115001
21 002030 012036
22 002031 115400 000716
23 002033 104201 000002
24 002035 002103
25 002036 102200 100000 001217 MSG1:
26 002041 052044
27 002042 114000 000716
28 002044 104231
29 002045 104137
30 002046 104302 000740
31 002050 060005
32 002051 115001
33 002052 012060
34 002053 115400 000715
35 002055 104201 000004
36 002057 002103
37 002060 106207 000175
38 002062 052070
39 002063 115400 000715
40 002065 104201 000003
41 002067 002103
42 002070 102200 100000 001217 TALKRT:
43 002073 052076
44 002074 114000 000715
45 002076
46 002077
47 002100 000000
48 002101 114002
49 002102 022552
50 002103 102200 100000 001217 TCLEAR:
51 002106 052115
52 002107 101200 100000 001217
53 002112 104060 001216
54 002114 002117
55 002115 104306 001216 TALKIP:
56 002117 106300 001502 000716 TALIP1:
57 002122 072101

TALK ROUTINE
INPUT PARAMETERS
R2 CONTAINS UNIT NUMBER CODE
R3 CONTAINS A POINTER TO A COMMAND/RESPONSE ADDRESS/SIZE TABLE
OUTPUT PARAMETERS

TALK: PUSH R3
      PUSH R4
LOOP1: MOV 1(SP),R3
      MOV (R3)+,R0
      MOV (R3)+,R1
      MOV UNIT,R2
      XFC SEND
      TST R1
      BEQ MSG1
      INC UN.ERT
      MOV #2,R1
      BR TCLEAR
MSG1:  BIT #RTY,FLAG
      BNE LOOP2
      CLR UN.ERT
LOOP2: MOV (R3)+,R1
      MOV (R3),R0
      MOV UNIT,R2
      XFC RCV
      TST R1
      BEQ TALKDN
      INC UN.ERR
      MOV #4,R1
      BR TCLEAR
TALKDN: CMP #UNSEC,R0
      BNE TALKRT
      INC UN.ERR
      MOV #3,R1
      BR TCLEAR
TALKRT: BIT #RTY,FLAG
      BNE TALKP
      CLR UN.ERR
TALKP: POP R4
      POP R3
      RETURN
ERRT:  CLR R2
      CALL ERRMNT
TCLEAR: BIT #RTY,FLAG
      BNE TALKIP
      BIS #RTY,FLAG
      MOV SP,STCKSV
      BR TALIP1
TALKIP: MOV STCKSV,SP
TALIP1: CMP RETRY,UN.ERT
      BMI ERRT

;RESTORE R3 FOR RETRIES
;GET COMMAND ADDRESS
;GET COMMAND SIZE
;MAKE SURE HAVE INTERCONNECT
;SEND GET STATUS COMMAND
;SUCCESSFUL?
;YES, BRANCH
;INCREMENT ERROR COUNT
;ERROR NUMBER IN CASE
;DO RECOVERY
;IN A RETRY?
;YES - DON'T CLEAR COUNTER
;FOR RESET
;POINT TO RCV BUFFER
;SET SIZE OF REPLY
;MAKE SURE HAVE INTERCONNECT
;RCV REPLY TO GET STATUS
;SUCCESSFUL?
;YES, CHECK STATUS
;INCREMENT ERROR COUNT
;ERROR CODE IN CASE
;DO RECOVERY
;WAS CMD UNSUCCESSFUL?
;YES, DONE
;INCREMENT ERROR COUNT
;ERROR CODE IN CASE
;NO, TRY AGAIN
;IN A RETRY?
;YUP - SKIP CLEAR
;CLEAR FOR REST
;RESTORE R4
;RESTORE R3

;CLEAR SUBCODE
;ERROR EXIT
;IN A RETRY?
;YUP - SKIP FLAG SET AND STACK SAVE
;SET FLAG
;SAVE STACK POINTER
;SKIP RETRY HANDLING
;RESTORE STACK POINTER
;DONE RETRIES?
;YUP - CAN IT

```

58	002123	106300	001502	000715	CMP	RETRY,UN.ERR	:OVER THE LIMIT ?
59	002126	072101			BMI	ERRT	:YUP
60	002127	104302	000740		MOV	UNIT,R2	:GET UNIT
61	002131	060011			XFC	DINIT	:INIT THE DRIVE
62	002132	022671			CALL	STATVL	:TST DRIVER STATUS VALIDITY
63	002133	052143			BNE	TERR	:IF NOT ZERO - NO GOOD
64	002134	022656			CALL	TIMER	:WAIT ANOTHER 2 SECONDS
65	002135	022656			CALL	TIMER	:TO MAKE SURE DRIVER HAS ENOUGH TIME
66	002136	022671			CALL	STATVL	:GET VALID STATUS AGAIN
67	002137	052143			BNE	TERR	:IF NO GOOD - ERROR
68	002140	102201	000001		BIT	#RCVRDY,R1	:IS RECEIVER READY SET
69	002142	052147			BNE	TATTN1	:YES - ALL SET
70	002143	104201	000024		MOV	#20.,R1	:SET ERROR CODE
71	002145	114002			CLR	R2	:CLEAR SUBCODE
72	002146	022552			CALL	ERRMNT	:DIE PEACEFULLY
73	002147	021662			CALL	GSTATS	:GET STATUS AND CLEAR ERRORS
74	002150	103200	100000	001217	BIC	#RTY,FLAG	:CLEAR RETRY FLAG
75	002153	002020			BR	LOOP1	:AND TRY AGAIN
76							
77							
78						RECAL ROUTINE	
79	002154	100463			RECAL:	MOV R3,-(SP)	:SAVE R3
80	002155	104203	001001		MOV	#CR.RCL,R3	:POINT TO RECAL TABLE
81	002157	002165			JMP	LOAD5	:SEND CMD VIA LOAD ROUTINE
82							
83						LOAD ROUTINE	
84							
85	002160	100463			LOAD:	MOV R3,-(SP)	:SAVE R3
86	002161	114000	000717		CLR	UN.ERI	:FOR INIT
87	002163	104203	000761		LOAD1:	MOV #CR.RUN,R3	:POINT TO LOAD DRIVE TABLE
88	002165				LOAD5:	PUSH R4	:SAVE R4
89	002166	104137			MOV	(R3),R0	:GETCOMMAND ADDRESS
90	002167	104631	000001		MOV	1(R3),R1	:GET COMMAND SIZE
91	002171	104302	000740		MOV	UNIT,R2	:GET INTERCONNECT
92	002173	060004			XFC	SEND	:ISSUE GET STATUS COMMAND
93	002174	115001			TST	R1	:SUCCESSFUL ?
94	002175	012205			BEQ	LOAD2	:YUP - SKIP RETRY
95	002176	115400	000717		INC	UN.ERI	:INC COUNT
96	002200	106300	001502	000717	CMP	RETRY,UN.ERI	:DONE ALL RETIES ?
97	002203	072231			BMI	LOADER	:YUP
98	002204	002165			BR	LOAD5	
99	002205	114000	000717		LOAD2:	CLR UN.ERI	:FOR ERROR CLEAR
100	002207	104304	001250		MOV	LTO,R4	:LONG TIMEOUT VALUE (SECONDS)
101	002211	104631	000002		LOAD3:	MOV 2(R3),R1	:GET RECEIVE BUFFRE
102	002213	104637	000003		MOV	3(R3),R0	:GET BUFFER LENGTH
103	002215	104302	000740		MOV	UNIT,R2	:GET INTERCONNECT
104	002217	060005			XFC	RCV	:RECEIVE SDI RESPONSE
105	002220	115001			TST	R1	:SUCCESSFUL ?
106	002221	012226			BEQ	LOAD4	:YUP - SKIP RETRY
107	002222	022656			CALL	TIMER	:WAIT 1 SECOND
108	002223	117404			DEC	R4	:DECREMENT COUNTER
109	002224	052211			BNE	LOAD3	:LOOP TILL DONE
110	002225	002231			BR	LOADER	:IF NOT DONE YET - THEN ERROR
111	002226				LOAD4:	POP R4	:RESTORE R4
112	002227	104263			MOV	(SP)+,R3	:RESTORE R3
113	002230	000000			RETURN		:RETURN TO CALLER
114	002231	104201	000003		LOADER:	MOV #3,R1	:ERROR CODE

115	002233	114002		CLR	R2		:CLEAR SUBCODE
116	002234	022552		CALL	ERRMNT		:DIE
117				:			
118				:	ACCESS ROUTINE		
119				:			
120	002235	100463		ACCESS:	MOV	R3, -(SP)	:SAVE R3
121	002236	104203	000765		MOV	#CR.ACC, R3	:POINT TO ACCESS TABLE
122	002240	022016			CALL	TALK	:SEND ACCESS CMD
123	002241	104263			MOV	(SP)+, R3	:RESTORE R3
124	002242	000000			RETURN		:RETURN TO CALLER
125				:			
126				:	CLEAR ROUTINE		
127				:			
128	002243	100463		CLEAR:	MOV	R3, -(SP)	:SAVE R3
129	002244	104203	000771		MOV	#CR.CLR, R3	:POINT TO CLEAR TABLE
130	002246	022016			CALL	TALK	:SEND CLEAR CMD
131	002247	104263			MOV	(SP)+, R3	:RESTORE R3
132	002250	000000			RETURN		:RETURN TO CALLER
133				:			
134				:	SEEK ROUTINE		
135				:			
136	002251	100463		SEEK:	MOV	R3, -(SP)	:SAVE R3
137	002252	100467			MOV	R0, -(SP)	:SAVE R0
138	002253	104302	000740	SEEK0:	MOV	UNIT, R2	:MAKE SURE HAVE UNIT
139	002255	104203	000775		MOV	#CR.SEK, R3	:POINT TO SEEK TABLE
140	002257	022016			CALL	TALK	:SEND SEEK COMMAND
141	002260	104303	001251		MOV	STO, R3	:SHORT TIMEOUT
142	002262	022671		SEEK1:	CALL	STATVL	:CHECK FOR STATUS VALIDITY
143	002263	052307			BNE	SEEK5	:IF NOT ZERO - DIE
144	002264	102201	000002		BIT	#ATTN, R1	:ANY PROBLEMS
145	002266	052310			BNE	SEEK2	:YES, BRANCH
146	002267	102201	100000		BIT	#RWRDY, R1	:NO, DONE?
147	002271	052301			BNE	SEEK6	:ALL DONE
148	002272	117403			DEC	R3	:DECREMENT COUNTER
149	002273	012321			BEQ	SEEK3	:IF ZERO THEN DEAD
150	002274	104207	000240		MOV	#160, R0	:1MS DELAY
151	002276	117407		SEEK7:	DEC	R0	:DECREMENT COUNTER
152	002277	052276			BNE	SEEK7	:DELAY LOOP
153	002300	002262			BR	SEEK1	:TRY AGAIN
154	002301	114001		SEEK6:	CLR	R1	:CLEAR ERROR FLAG
155	002302	114000	000720	SEEK4:	CLR	UN.SEK	:FOR RESET
156	002304	104267			MOV	(SP)+, R0	:YES, RESTORE R0
157	002305	104263			MOV	(SP)+, R3	:RESTORE R3
158	002306	000000			RETURN		:RETURN TO CALLER
159	002307	022371		SEEK5:	CALL	INITPT	:INIT THE DRIVE
160	002310	115400	000720	SEEK2:	INC	UN.SEK	:INCREMENT RETRY COUNTER
161	002312	106300	001502		CMP	RETRY, UN.SEK	:HAVE WE DONE ALL RETRIES?
162	002315	012321			BEQ	SEEK3	:YES, PANIC
163	002316	021662			CALL	GSTATS	:PANIC AND CALL GET STATUS
164	002317	022154			CALL	RECAL	:RECAL DRIVE
165	002320	002253			BR	SEEK0	:AND TRY AGAIN
166	002321	104201	177775	SEEK3:	MOV	#-3, R1	:SET ERROR CODE
167	002323	002302			BR	SEEK4	:RESTORE REGS AND RETURN
168				:			
169				:	UNLOAD ROUTINE		
170				:			
171	002324	100463		DISCON:	MOV	R3, -(SP)	:SAVE R3

172	002325	104203	000755	MOV	#CR.DIS,R3	:DISCONNECT WITH
173	002327	022016		CALL	TALK	:SEND UNLOAD CMD
174	002330	104263		MOV	(SP)+,R3	:RESTORE R3
175	002331	000000		RETURN		:RETURN TO CALLER
176				:		
177				:	ONLINE ROUTINE	
178				:		
179	002332	100463		ONLIN:	MOV R3,-(SP)	:SAVE R3
180	002333	104203	001011		MOV #CR.ONL,R3	:ONLINE COMMAND
181	002335	022016			CALL TALK	:BRING DRIVE ONLINE
182	002336	104263			MOV (SP)+,R3	:RESTORE R3
183	002337	000000			RETURN	:RETURN TO CALLER
184				:		
185				:	INITIALIZE ROUTINE	
186				:		
187	002340	100461		INITIT:	MOV R1,-(SP)	:SAVE R1
188	002341				PUSH R3,R4	:SAVE R3 AND R4
189	002343	104204	000001		MOV #1,R4	:START WITH PORT 0
190	002345	104203	000004		MOV #4,R3	:INIT PORT COUNTER
191	002347	104042		INIT5:	MOV R4,R2	:SET UP INTERCONNECT
192	002350	060011			XFC DINIT	:INIT DRIVE
193	002351	104207	066540		MOV #28000.,R0	:TIMER (APPROX 2 SECS)
194	002353	022671		ATTN1:	CALL STATVL	:CHECK STATUS VALIDITY
195	002354	052362			BNE AOUT	:IF NOT ZERO - NO GOOD
196	002355	117407			DEC R0	:DEC COUNT
197	002356	012362			BEQ AOUT	:IF ZERO THEN DEAD
198	002357	102201	000001		BIT #RCVRDY,R1	:IS RECEIVER READY SET ?
199	002361	012353			BEQ ATTN1	:NO, TRY AGAIN
200	002362	110204		AOUT:	ROL R4	:NEXT PORT
201	002363	117403			DEC R3	:DECREMENT COUNTER
202	002364	052347			BNE INIT5	:IF NOT DONE DO NEXT PORT
203	002365				POP R4,R3	:RESTORE R3 AND R4
204	002367	104261			MOV (SP)+,R1	:RESTORE R1
205	002370	000000			RETURN	:AND RETURN TO CALLER
206				:		
207				:	INIT GIVEN PORT	
208				:		
209				:		
210	002371	104302	000740	INITPT:	MOV UNIT,R2	:GET PORT NUMBER
211	002373	060011			XFC DINIT	:DO THE INIT
212	002374	104207	066540		MOV #28000.,R0	:1 SECOND TIMER
213	002376	022671		INITP1:	CALL STATVL	:VALIDATE STATUS
214	002377	052406			BNE INITDD	:DEAD IF NOT VALID
215	002400	117407			DEC R0	:DECREMETN COUTNER
216	002401	012406			BEQ INITDD	:DEAD IF COUNT EXPIRED
217	002402	102201	000001		BIT #RCVRDY,R1	:DONE INIT ?
218	002404	012376			BEQ INITP1	:NOPE - KEEP TRYING
219	002405	000000			RETURN	:EXIT
220	002406	104201	000024	INITDD:	MOV #20.,R1	:ERROR CODE
221	002410	114002			CLR R2	:NO SUBCODE
222	002411	022552			CALL ERRMNT	:ERROR EXIT

```

1
2
3
4
5
6
7
8 002412 104204 001341
9 002414 105014
10 002415 104203 003047
11 002417 022526
12 002420 115007
13 002421 052426
14 002422 114000 000715
15 002424
16 002425 003047
17 002426 106300 001502 000715
18 002431 012435
19 002432 115400 000715
20 002434 002417
21 002435 104012
22 002436 104201 000005
23 002440 101200 000020 001220
24 002443 022552

.SBTTL OVERLAY PROCESSING ROUTINES
OVERLAY PROCESSING ROUTINES
R1 = OFFSET INTO TABLE

NEXT CALLS OVERLAY FOR NEXT CODE OVERLAY

NEXT: MOV #OVL TBL,R4 ;GET POINTER TO OVERLAY TABLE
      ADD R1,R4 ;INDEX INTO TABLE
      MOV #START,R3 ;UDA ADDRESS TO LOAD AT
NEXT5: CALL OVLAY ;CALL ROUTINE TO DO OVERLAY
      TST R0 ;CHECKSUM O.K. ??
      BNE OERR ;YES - RETRY IF POSSIBLE
      CLR UN.ERR ;CLEAR ERROR COUNT
      POP R1 ;POP CURRENT RETURN ADDRESS
      BR START ;GO TO OVERLAY
OERR: CMP RETRY,UN.ERR ;DONE ALL RETRIES ?
      BEQ OERR2 ;YUP
      INC UN.ERR ;INC ERROR AND
      BR NEXT5 ;TRY AGAIN
OERR2: MOV R1,R2 ;GET ERROR CODE FROM XFC
      MOV #5,R1 ;SET UNIBUS READ ERROR
      BIS #DEAD,FLAG1 ;INDICATE HOST GONE
      CALL ERRMNT ;ERROR RETURN

```

```

1
2
3
4
5
6
7
8
9
10 002444 104303 001153
11 002446
12 002447 104204 001341
13 002451 105014
14 002452 104203 003047
15 002454 022526
16 002455 115007
17 002456 012461
18 002457 022510
19 002460 002454
20 002461 114000 000715
21 002463 104203 002472
22 002465 100463
23 002466 104203 003047
24 002470 100463
25 002471 000000
26
27
28 002472
29 002473 104010 001153
30 002475 104204 001341
31 002477 105014
32 002500 104203 003047
33 002502 022526
34 002503 115007
35 002504 012507
36 002505 022510
37 002506 002502
38 002507 000000
39 002510 106300 001502 000715
40 002513 012517
41 002514 115400 000715
42 002516 000000
43 002517 101200 000020 001220
44 002522 114002
45 002523 104201 000005
46 002525 022552

:
:
: PAGE BRINGS IN AN OVERLAY AND CALLS IT
: UPON RETURN OF THE OVERLAY PAGE BRINGS IN
: THE PREVIOUS OVERLAY AND BEGINS EXECUTION
: AT THE POINT THE CALL TO PAGE WAS MADE
:
: R1 - OVERLAY TO BE BROUGHT IN
:
:
: PAGE:  MOV    CUROVL,R3      ;GET 'CALLING' OVERLAY
:        PUSH   R3            ;PUSH FOR LATER RETURN
:        MOV    #OVL TBL,R4    ;POINT TO OVERLAY TABLE
:        ADD    R1,R4          ;POINT TO ENTRY FOR NEW OVERLAY
:        MOV    #START,R3      ;POINT TO UDA LOAD ADDRESS
: PALP1:  CALL   OVRLAY        ;BRING IN NEW OVERLAY
:        TST    R0             ;EDC O.K. ?
:        BEQ    PALP2          ;YUP
:        CALL   PAERR          ;CALL ERROR HANDLER
:        BR     PALP1          ;TRY AGAIN
: PALP2:  CLR     UN.ERR        ;CLEAR ERROR COUNT
:        MOV    #PAGER,R3      ;ADDRESS FOR OVERLAY TO RETURN TO
:        MOV    R3,-(SP)        ;PUSH ON STACK
:        MOV    #START,R3      ;STARTING ADDRESS OF NEW OVERLAY
:        MOV    R3,-(SP)        ;PUSH ON STACK FOR 'CALL'
:        RETURN                ;'CALL' OVERLAY
:
:        ;NEXT ADDRESS ON STACK IS RETURN
:        ;ADDRESS TO PAGE
:        ;POP OLD OVERLAY NUMBER
:        ;MAKE IT CURRENT
:        ;POINT TO OVERLAY TABLE
:        ;POINT TO OLD OVERLAY BLOCK
:        ;POINT TO UDA LOAD ADDRESS
: PALP4:  CALL   OVRLAY        ;BRING IT IN
:        TST    R0             ;EDC O.K. ?
:        BEQ    PALP3          ;YUP
:        CALL   PAERR          ;ERROR HANDLER
:        BR     PALP4          ;TRY AGAIN
: PALP3:  RETURN                ;RETURN TO ADDRESS PAGE CALLED FROM
: PAERR:  CMP     RETRY,UN.ERR  ;DONE ALL RETRIES ?
:        BEQ    PAERR1          ;YUP
:        INC     UN.ERR         ;INC ERROR COUNT
:        RETURN                ;RETURN AND TRY AGAIN
:
: PAERR1: BIS     #DEAD,FLAG1   ;SET HOST GONE
:        CLR     R2            ;NO SUBCODE
:        MOV     #5,R1         ;UNIBUS READ ERROR
:        CALL    ERRMNT        ;DIE

```

```
1  
2  
3  
4  
5  
6  
7 002526 104647 000001  
8 002530 104641 000002,  
9 002532 104642 000000  
10 002534 060013  
11 002535 000000
```

OVERLAY ROUTINE
ISSUES UNIBUS READ TO GET OVERLAY
R3 -> UDA ADDRESS TO LOAD

```
OVRLAY: MOV      HSTLO(R4),R0      :LOW ORDER UNIBUS ADDRESS  
        MOV      HSTHI(R4),R1      :HI ORDER UNIBUS ADDRESS  
        MOV      LEN(R4),R2        :WORD COUNT TO OVERLAY  
        XFC      UREAD              :ISSUE UNIBUS READ  
        RETURN                      :RETURN TO CALLING ROUTINE
```

```

1
2
3
4
5 002536 104207 001263
6 002540 104201 000016
7 002542 060016
8 002543 000000
9
10
11
12
13
14 002544 104207 001263
15 002546 104201 000016
16 002550 060017
17 002551 000000
18
19
20
21
22
23
24 002552 102200 000010 001220
25 002555 052603
26 002556 101200 000010 001220
27 002561 104207 001263
28 002563 100671 000002
29 002565 100672 000003
30 002567 114002
31 002570 100672 000004
32 002572 104301 001324
33 002574 100171
34 002575 104301 001325
35 002577 100671 000001
36 002601 022536
37 002602 022324
38 002603 114007
39 002604 102200 000020 001220
40 002607 012611
41 002610 115407
42 002611 060021
43
44
45
46 002612 115000 001505
47 002614 012624
48 002615 104300 001505 001103
49 002620 104203 001005
50 002622 022016
51 002623 000000
52 002624 022154
53 002625 022251
54 002626 000000

.SBTTL MISCELLANEOUS COMMON ROUTINES

:
: SEND MAINTENANCE READ COMMAND
:
SNDMNT: MOV #DMBUF,R0 ;POINT TO BUFFER
MOV #DMBUFL,R1 ;LENGTH
XFC MAINTR ;ISSUE COMMAND
RETURN

:
: RECEIVE MAINTENANCE WRITE DATA
:
RCVMNT: MOV #DMBUF,R0 ;POINT TO BUFFER
MOV #DMBUFL,R1 ;SIZE OF BUFFER
XFC MAINTW ;RECEIVE MAINT WRITE DATA
RETURN

:
: ERROR RETURN TO HOST
:
ERRMNT: BIT #ERDN,FLAG1 ;TRIED TO QUIT YET ?
BNE ALLOVR ;YUP - DISCON FAILED
BIS #ERDN,FLAG1 ;SET FLAG
MOV #DMBUF,R0 ;POINT TO MAINT BUFFER
MOV R1,2(R0) ;PUT ERROR NUMBER IN MSG
MOV R2,3(R0) ;PUT IN ERROR SUBCODE
CLR R2 ;CLEAR *** TEMP ***
MOV R2,4(R0) ;ONLY ONE WORD RIGHT NOW ***
MOV EMSG,R1 ;GET DUP CODE
MOV R1,(R0) ;STORE IT
MOV EMSG+1,R1 ;GET 'E' IDENTIFIER
MOV R1,1(R0) ;STORE IT
CALL SNDMNT ;SEND TO HOST
CALL DISCON ;DISCONNECT/SPINDOWN DRIVE
ALLOVR: CLR R0 ;IN CASE O.K
BIT #DEAD,FLAG1 ;DIE OR JUST QUIT ?
BEQ ALLOV1 ;JUST QUIT
INC R0 ;MAKE NON ZERO
ALLOV1: XFC DONE ;EXIT DM MODE

:
: ERROR RETRY SEQUENCER
:
ERRHND: TST RECTMP ;RECOVERY LEVEL 0 ?
BEQ ERRLST ;YES - DO LAST DITCH EFFORT
MOV RECTMP,ERECOV+1 ;STORE LEVEL IN COMMAND
MOV #CR.ERV,R3 ;POINT TO COMMAND
CALL TALK ;DO ERROR RECOVERY
RETURN ;RETURN
ERRLST: CALL RECAL ;DO A RECALIBRATE
CALL SEEK ;RESEEK
RETURN

```

1				:		
2				:		
3				:	COMPUTE EDC	
4				:	R2 -> BUFFER	
5				:		
6	002627			CEDC:	PUSH R2,R4,R5	
7	002632	104303	001246		MOV INI,R3	: INITIAL VALUE (69)
8	002634	104305	001247		MOV CNT,R5	: SECTOR SIZE (258)
9	002636	104224		EAGAIN:	MOV (R2)+,R4	: MOVE A WORD INTO R4
10	002637				XOR R4,R3	: XOR INTO EDC
11	002643	105203	000000		ADD #0,R3	: CLEAR CARRY
12	002645	110203			ROL R3	: CIRCLE LEFT
13	002646	042650			BCC NOCRY	: IF NO CARRY SKIP
14	002647	115403			INC R3	: INCREMENT IF CARRY
15	002650	117405		NOCRY:	DEC R5	: DECREMENT COUNTER
16	002651	052636			BNE EAGAIN	: IF NOT 0 DO NEXT WORD
17	002652				POP R5,R4,R2	
18	002655	000000			RETURN	; R3 = EDC
19				:		
20				:		
21				:		
22				:	ONE SECOND DELAY ROUTINE	
23				:		
24				:		
25	002656			TIMER:	PUSH R3	: SAVE R3
26	002657	104203	100000		MOV #TIMVAL,R3	: LOOP COUNTER (32768)
27	002661	104011		TIMLP:	MOV R1,R1	: DELAY
28	002662	104011			MOV R1,R1	: DELAY
29	002663	104011			MOV R1,R1	: DELAY
30	002664	104011			MOV R1,R1	: DELAY
31	002665	117403			DEC R3	: DECRMENT COUNTER
32	002666	052661			BNE TIMLP	: LOOP TILL DONE
33	002667				POP R3	: RESTORE R3
34	002670	000000			RETURN	: RETUR TO CALLER
35				:		
36				:		
37				:		
38				:	VALIDATE XFC STATUS RESPONSE	
39				:		
40				:		
41	002671			STATVL:	PUSH R5	: SAVE R5
42	002672	104205	077777		MOV #77777,R5	: COUNTER (1 SECOND)
43	002674	060007		STATRE:	XFC STATUS	: GET DRIVE STATUS
44	002675	117405			DEC R5	: INCREMENT COUNTER
45	002676	012710			BEQ STATFR	: IF ZERO - NO GOOD
46	002677	102201	000004		BIT #VLD,R1	: LOW VALID BIT GOOD ?
47	002701	012674			BEQ STATRE	: NO - RETRY
48	002702	102201	000400		BIT #PARIT1,R1	: PARITY ERROR ?
49	002704	052674			BNE STATRE	: YES - RETRY
50	002705				POP R5	: RESTORE R5
51	002706	106011			CMP R1,R1	: SET COND CODE EQ ZERO
52	002707	000000		STATRT:	RETURN	: RETURN
53	002710			STATFR:	POP R5	: RESTORE R5
54	002711	106201	177777		CMP #177777,R1	: MAKE SURE NOT EQUAL
55	002713	002707			BR STATRT	: RETURN

3				COMPUTE ECC SYMBOLS IN ERROR	
4					
5					
6	003027	104202	001037	FCCCK: MOV	#CR,R2 :POINT TO CHARACTERISTICS
7	003031	104623	000002	MOV	ERRSYM(R2),R3 :GET THRESHOLD
8	003033	103203	177400	BIC	#4IBYTE,R3 :CLAR HIGH GARBAGE
9	003035	104207	000721	MOV	#RDBLK,R0 :POINT TO COMMAND BLOCK
10	003037	060015		XFC	ECC :PERFORM ECC CORRECTION
11	003040	115001		TST	R1 :SUCCESSFUL ?
12	003041	053044		BNE	105\$:NOPE
13	003042	106073		CMP	R0,R3 :WITHIN BOUNDS ?
14	003043	073046		BMI	GDECC :YES CONSIDER GOOD
15	003044	104201	177777	105\$: MOV	#-1,R1 :ELSE SIGNAL BAD
16	003046	0000C0		GDECC: RETURN	:RETURN

1	003047	003063			START: JMP	START3		:SKIP LOCAL DATA STORAGE
2					:			
3					:			
4					:			
5	003050				DESC: .BLKW	4		:HOST BUFFER DESCRIPTOR
6	003054	010000			DMSG1: .WORD	10000		:TYPE AND SEQUENCE NUMBER
7	003055	123	040		.ASCII	'S'		:START UP
8	003056	125	120	040	.ASCIIZ	'UP YOURS'		:MESSAGE ZERO
	003057	131	117	125				
	003061	122	123	000				
9		000007			DMSG1L -	.-DMSG1		:LENGTH OF MESSAGE ZERO
10					:			
11					:			
12	003063	104206	001215		START3: MOV	#STACK,SP		:SET UP STACK
13	003065	104207	003054		MOV	#DMSG1,R0		:MESSAGE
14	003067	104201	000007		MOV	#DMSG1L,R1		:LENGTH
15	003071	060016			XFC	MAINTR		:SEND MESSAGE
16	003072	104207	001263		MOV	#DMBUF,R0		:ADDRESS OF BUFFER
17	003074	104201	000016		MOV	#DMBUFL,R1		:LENGTH OF BUFFER
18	003076	060017			XFC	MAINTW		:GET ANSWER
19	003077	023103			CALL	SETOVL		:SET UP OVERLAY ADDRESSES
20	003100	104201	000060		MOV	#G1,R1		:NEXT OVERLAY
21	003102	022412			CALL	NEXT		:BRING IT IN

1					
2					
3					
4	003103	104205	000021		
5	003105	104204	007774		
6	003107	104200	005472	000731	
7	003112	114000	000732		
8	003114	104203	000731		
9	003116	021524			
10	003117	104043			
11	003120	104204	001342		
12	003122	021506			
13	003123	105204	000003		
14	003125	117405			
15	003126	053122			
16	003127	000000			

	SET UP OVERLAY TABLE	
SETOVL:	MOV #OVCNT,R5	:GET COUNT OF OVERLAYS
	MOV #DUPOVL,R4	:POINT TO OVERLAY ADDRESS (2 WORDS)
	MOV #OVS.G1,DDUMMY	:RELATIVE START OF FIRSTOVERLAY
	CLR DDUMMY+1	:CLEAR HIGH ORDER
	MOV #DDUMMY,R3	:FOR SUB
	CALL DSUB	:GET OFFSET (MUST ADD TO RELATIVE START ADDRESS OF E
	MOV R4,R3	:CHANGE POINTER FOR ADDS
	MOV #OVL.TBL+HSTLO,R4	:POINT TO LOW HOST ADD OF FIRST ENTRY
SLOOP:	CALL DADD	:ADD OFFSET
	ADD #OVLEN,R4	:POINT TO LOW HOST ADD. OF NEXT ENTRY
	DEC R5	:DECREMENT COUNTER
	BNE SLOOP	:IF NON-ZERO THEN CONTINUE
	RETURN	:ELSE DONE

1				.SBTTL	INITIALIZATION OVERLAY	
2				:		
3				:	INITIALIZATION OVERLAY	
4				:		
5	003130			DMOVLY	G1,START	
6	003047	024430		CALL	INITL	:INITIALIZE DISK
7	003050	115001		TST	R1	:ANY ERRORS ?
8	003051	053064		BNE	BUMER	:YUP - QUIT
9	003052	102200	000001	BIT	#FCTAVL,FLAG	:USE RESIDENT FCT ?
10	003055	053061		BNE	DOLBN	:YES - ONLY DO LBN
11	003056	104201	000000	MOV	#F1,R1	:ELSE DO D/XBN FIRST
12	003060	003063		BR	DXBN	:SKIP LBN FLAGGING
13	003061	104201	000003	DOLBN: MOV	#F2,R1	:SIGNAL LBN FORMAT
14	003063	022412		DXBN: CALL	NEXT	:BRING IN NEXT OVERLAY
15	003064	104201	000006	BUMER: MOV	#6,R1	:SIGNAL INIT
16	003066	022552		CALL	ERRMNT	:ERROR RETURN
17				:		
18				:		
19				:	COMPUTE D.SK CONSTANTS FROM CHARACTERISTICS	
20				:		
21				:		
22				:		
23				:	COMPUTE TRACKS/CYLINDER	
24				:		
25	003067	104207	001052	CONINT: MOV	#SCR,R0	:POINT TO SUB CHARACTERISTICS
26	003071	104673	000003	MOV	TRKGRP(R0),R3	:LOAD TRACKS/GROUP
27	003073	103203	177400	BIC	#HIBYTE,R3	:CLEAR HIGH BYTE
28	003075	104030	000731	MOV	R3,DDUMMY	:STORE IN DUMMY AREA
29	003077	114000	000732	CLR	DDUMMY+1	:CLEAR FOR STORE
30	003101	104673	000002	MOV	GRPCYL(R0),R3	:GET GROUPS/CYLINDER
31	003103	103203	177400	BIC	#HIBYTE,R3	:CLEAR HIGH BYTE
32	003105	104030	000736	MOV	R3,TEMP	:STORE IN TEMP AREA
33	003107	114000	000737	CLR	TEMP+1	:CLEAR HIGH ORDER
34	003111	104203	000731	MOV	#DDUMMY,R3	:SETR UP FOR MULT
35	003113	104204	000736	MOV	#TEMP,R4	:DITTO
36	003115	021542		CALL	DMUL	:COMPUTE IT
37	003116	104140	001141	MOV	(R4),TRKCYL	:LOAD FOR STORE
38	003120	104640	000001	MOV	1(R4),TRKCYL+1	:LOAD FOR STORE
39				:		
40				:	COMPUTE LBN'S/CYLINDER	
41				:		
42	003123	104673	000011	MOV	LBNTRK(R0),R3	:GET LBN'S/TRACK
43	003125	103203	177400	BIC	#HIBYTE,R3	:CLEAR HIGH BYTE
44	003127	104030	000736	MOV	R3,TEMP	:FOR MULT
45	003131	114000	000737	CLR	TEMP+1	:FOR STORE
46	003133	104204	000736	MOV	#TEMP,R4	:FOR MULTIPLY
47	003135	104203	001141	MOV	#TRKCYL,R3	:DITTO
48	003137	021542		CALL	DMUL	:GET LBN'S/CYL
49	003140	104140	001145	MOV	(R4),LBNPCY	:GET LOW ORDER
50	003142	104640	000001	MOV	1(R4),LBNPCY+1	:GET HIGH ORDER
51				:		
52				:	COMPUTE RBN'S/CYLINDER	
53				:		
54	003145	104673	000004	MOV	RBNTRK(R0),R3	:GET RBN'S/TRACK
55	003147	103203	177600	BIC	#HIBYTE,R3	:CLEAR OUT GARBAGE
56	003151	104030	000736	MOV	R3,TEMP	:STORE FOR MULT
57	003153	114000	000737	CLR	TEMP+1	:FOR STORE

58	003155	104204	000736		MOV	#TEMP,R4	:FOR MULTIPLY
59	003157	104203	001141		MOV	#TRKCYL,R3	:DITTO
60	003161	021542			CALL	DMUL	:GET RBN'S/CYL
61	003162	104140	001147		MOV	(R4),RBNPCY	:GET LOW ORDER
62	003164	104640	000001	001150	MOV	1(R4),RBNPCY+1	:GET HIGH ORDER
63							
64							
65						COMPUTE LBN'S IN LBN AREA	
66							
67							
68	003167	104207	001052		MOV	#SCR,R0	:POINT TO CHARACTERISTICS
69	003171	104670	000000	000736	MOV	CYLBN(R0),TEMP	:GET LBN CYLINDERS
70	003174	104673	000001		MOV	CYLBN+1(R0),R3	:GET HIGH ORDER
71	003176	103203	170000		BIC	#HD.CLR,R3	:CLEAR STARTING CYLINDER BITS
72	003200	104030	000737		MOV	R3,TEMP+1	:STORE IT
73	003202	104204	000736		MOV	#TEMP,R4	:FOR MULT
74	003204	104203	001145		MOV	#LBNPCY,R3	:POINT TO LBN'S/CYLINDER
75	003206	021542			CALL	DMUL	:GET LBN'S IN LBN AREA
76	003207	104140	001133		MOV	(R4),LBNLBN	:GET LOW ORDER
77	003211	104640	000001	001134	MOV	1(R4),LBNLBN+1	:GET HIGH ORDER
78	003214	104670	000012	000731	MOV	LBNHOST(R0),DDUMMY	:GET LBN'S IN HOST AREA
79	003217	104670	000013	000732	MOV	LBNHOST+1(R0),DDUMMY+1	:GET HIGH ORDER
80	003222	104203	000731		MOV	#DDUMMY,R3	:FOR SUB
81	003224	021524			CALL	DSUB	:SUBTRACT TO GET LBN'S IN RCT
82	003225	104140	001475		MOV	(R4),TOTRCT	:GET LOW ORDER
83	003227	104640	000001	001476	MOV	1(R4),TOTRCT+1	:GET HIGH ORDER
84							
85						COMPUTE RBN'S IN LBN AREA	
86							
87							
88	003232	104207	001052		MOV	#SCR,R0	:POINT TO CHARACTERISTICS
89	003234	104670	000000	000736	MOV	CYLBN(R0),TEMP	:GET LBN CYLINDERS
90	003237	104673	000001		MOV	CYLBN+1(R0),R3	:GET HIGH ORDER
91	003241	103203	170000		BIC	#HD.CLR,R3	:CLEAR STARTING CYLINDER BITS
92	003243	104030	000737		MOV	R3,TEMP+1	:STORE IT
93	003245	104204	000736		MOV	#TEMP,R4	:FOR MULT
94	003247	104203	001147		MOV	#RBNPCY,R3	:POINT TO RBN'S/CYLINDER
95	003251	021542			CALL	DMUL	:GET LBN'S IN LBN AREA
96	003252	104140	001135		MOV	(R4),RBNLBN	:GET LOW ORDER
97	003254	104640	000001	001136	MOV	1(R4),RBNLBN+1	:GET HIGH ORDER
98							
99						COMPUTE SECTORS/TRACK	
100							
101	003257	104673	000004		MOV	RBNTRK(R0),R3	:LOAD RBN'S/TRACK
102	003261	103203	177600		BIC	#HI1BYTE,R3	:CLEAR OUT GARBAGE
103	003263	104674	000011		MOV	LBNTRK(R0),R4	:LOAD LBN'S/TRACK(512)
104	003265	103204	177400		BIC	#HI1BYTE,R4	:CLEAR OUT HIGH BYTE
105	003267	105043			ADD	R4,R3	:ADD FOR SECT/TRACK
106	003270	104030	001127		MOV	R3,SECTRK	:STORE IT
107	003272	114000	001130		CLR	SECTRK+1	:CLEAR FOR STORE
108							
109						COMPUTE SECTORS/CYLINDER	
110							
111	003274	104300	001127	000731	MOV	SECTRK,DDUMMY	:LOW ORDER
112	003277	104300	001130	000732	MOV	SECTRK+1,DDUMMY+1	:HIGH ORDER
113	003302	104204	000731		MOV	#DDUMMY,R4	:SET UP FOR MULT
114	003304	104203	001141		MOV	#TRKCYL,R3	:DITTO

115	003306	021542			CALL	DMUL	: COMPUTE IT
116	003307	104140	001131		MOV	(R4),SECTCY	: LOAD FOR STORE
117	003311	104640	000001	001132	MOV	1(R4),SECTCY+1	: LOAD FOR STORE
118							
119						COMPUTE SECTORS IN XBN AREA (R4 -> SECTORS/CYL)	
120							
121	003314	104670	000021	000736	MOV	XBN CYL(R0),TEMP	: LOAD NUMBER OF XBN CYLINDERS
122	003317	114000	000737		CLR	TEMP+1	: CLEAR FOR STORE
123	003321	104203	000736		MOV	#TEMP,R3	: SET UP FOR MULT
124	003323	021542			CALL	DMUL	: GET XBN SECTORS
125	003324	104140	001137		MOV	(R4),XBNSEC	: LOAD FOR STORE
126	003326	104640	000001	001140	MOV	1(R4),XBNSEC+1	: LOAD FOR STORE
127							
128						COMPUTE LAST CYLINDER NUMBER ON SUBUNIT	
129							
130	003331	104207	001052		MOV	#SCR,R0	: POINT TO CHARACTERISTICS
131	003333	104670	000000	001143	MOV	CY LBN(R0),LBNCYL	: GET LOW ORDER LBN CYLINDERS
132	003336	104670	000000	000736	MOV	CY LBN(R0),TEMP	: ALSO FOR MATH
133	003341	104674	000001		MOV	CY LBN+1(R0),R4	: GET HIGH ORDER
134	003343	104040	001144		MOV	R4,LBNCYL+1	: STORE HI ORDER
135	003345	104040	000737		MOV	R4,TEMP+1	: ALSO FOR MATH
136	003347	104204	000736		MOV	#TEMP,R4	: SET UP FOR ADD
137	003351	104670	000021	000731	MOV	XBN CYL(R0),DDUMMY	: LOAD XBN CYLINDERS
138	003354	114000	000732		CLR	DDUMMY+1	: CLEAR HIGH ORDER
139	003356	104203	000731		MOV	#DDUMMY,R3	: SET UP FOR ADD
140	003360	021506			CALL	DADD	: GET LBN+XBN CYLINDERS
141	003361	104673	000022		MOV	DBN CYL(R0),R3	: LOAD DBN CYLINDERS
142	003363	110703			SWAB	R3	: GET INTO LO BYTE
143	003364	103203	177400		BIC	#HIBYTE,R3	: CLEAR GARBAGE
144	003366	104030	000731		MOV	R3,DDUMMY	: FOR DIVIDE
145	003370	114000	000732		CLR	DDUMMY+1	: CLEAR HIGH ORDER
146	003372	104203	000731		MOV	#DDUMMY,R3	: SET UP FOR ADD
147	003374	021506			CALL	DADD	: GET LBN+XBN+DBN CYLINDERS
148	003375	104642	000001		MOV	1(R4),R2	: GET HIGH ORDER
149	003377	104673	000001		MOV	STCYL(R0),R3	: GET CYLINDER BITS
150	003401	103203	007777		BIC	#LO,R3	: CLEAR OUT REST OF WORD
151	003403	101032			BIS	R3,R2	: OR IN CYLINDER HIGH BITS
152	003404	100642	000001		MOV	R2,1(R4)	: STORE BACK
153	003406	104203	001466		MOV	#ONE,R3	: DECREMENT TO GET NUM OF LAST CYL
154	003410	021524			CALL	DSUB	: DO IT
155	003411	104140	001125		MOV	(R4),CYLNUM	: MAKE IT CURRENT CYLINDER
156	003413	104640	000001	001126	MOV	1(R4),CYLNUM+1	: LOAD HI ORDER
157							
158						COMPUTE NON-PAD RCT	
159							
160	003416	104300	001135	000736	MOV	RBN LBN,TEMP	: GET LOW ORDER LBN'S IN HOST AREA
161	003421	104300	001136	000737	MOV	RBN LBN+1,TEMP+1	: GET HIGH ORDER
162	003424	104200	000177	000731	MOV	#127,DDUMMY	: ADD 127 FOR DIV FUNCTION
163	003427	114000	000732		CLR	DDUMMY+1	: FOR CLEAR
164	003431	104204	000736		MOV	#TEMP,R4	: FOR ADD
165	003433	104203	000731		MOV	#DDUMMY,R3	: DITTO
166	003435	021506			CALL	DADD	: ADD
167	003436	104200	000200	000731	MOV	#128,DDUMMY	: FOR DIVIDE (128 RBN/RCT BLOCK)
168	003441	114000	000732		CLR	DDUMMY+1	: FOR STORE
169	003443	104203	000731		MOV	#DDUMMY,R3	: POINT TO IT
170	003445	021570			CALL	DDIV	: DO DIVIDE
171	003446	104140	001261		MOV	(R4),RCT LBN	: GET LOW ORDER QUOTIENT

172	003450	105200	000002	001261	ADD	#2,RCTLBN	:FOR CONTROL BLOCKS
173							
174					COMPUTE	NON-PAD FCT	
175							
176	003453	104300	001135	000736	MOV	RBNLBN,TEMP	:GET LOW ORDER LBN'S IN HOST AREA
177	003456	104300	001136	000737	MOV	RBNLBN+1,TEMP+1	:GET HIGH ORDER
178	003461	104200	000002	000731	MOV	#2,DDUMMY	:FOR DIVIDE BY 2
179	003464	114000	000732		CLR	DDUMMY+1	:DITTO
180	003466	104204	000736		MOV	#TEMP,R4	:SETUP
181	003470	104203	000731		MOV	#DDUMMY,R3	:SETUP
182	003472	021570			CALL	DDIV	:CALL DIVIDE
183	003473	104200	000177	000731	MOV	#127.,DDUMMY	:ADD 127 FOR DIV FUNCTION
184	003476	114000	000732		CLR	DDUMMY+1	:FOR CLEAR
185	003500	104204	000736		MOV	#TEMP,R4	:FOR ADD
186	003502	104203	000731		MOV	#DDUMMY,R3	:DITTO
187	003504	021506			CALL	DADD	:ADD
188	003505	104200	000200	000731	MOV	#128.,DDUMMY	:FOR DIVIDE (128 RBN/RCT BLOCK)
189	003510	114000	000732		CLR	DDUMMY+1	:FOR STORE
190	003512	104203	000731		MOV	#DDUMMY,R3	:POINT TO IT
191	003514	021570			CALL	DDIV	:DO DIVIDE
192	003515	104200	000002	000731	MOV	#2,DDUMMY	:FOR MULT
193	003520	114000	000732		CLR	DDUMMY+1	:CLEAR HIGH WORD
194	003522	104203	000731		MOV	#DDUMMY,R3	:FOR DIVIDE
195	003524	021542			CALL	DMUL	:DO MULTIPLY
196	003525	104140	001260		MOV	(R4),FCTNPD	:NON-PAD FCT BLOCKS
197	003527	115400	001260		INC	FCTNPD	:FOR NON-PAD FCT BLOCKS
198							
199							
200					INITIALIZE	GOOD DATA BLOCK	
201	003531	104203	005567		MOV	#GDBLK,R3	:POINT TO BUFFER
202	003533	104302	001450		MOV	DWRD,R2	:DIAGNOSTIC WORD
203	003535	100232			MOV	R2,(R3)+	:STORE IT
204	003536	104204	000125		MOV	#85.,R4	:SET COUNTER
205	003540	104302	001445		MOV	FWRD,R2	:FIRST WORD OF PATTERN
206	003542	100232			MOV	R2,(R3)+	:STORE IT
207	003543	104302	001446		MOV	SWRD,R2	:SECOND WORD OF PATTERN
208	003545	100232			MOV	R2,(R3)+	:STORE IT
209	003546	104302	001447		MOV	TWRD,R2	:THIRD WORD OF PATTERN
210	003550	100232			MOV	R2,(R3)+	:STORE IT
211	003551	117404			DEC	R4	:DECREMENT COUNTER
212	003552	053540			BNE	OVER	:REPEAT TILL DONE
213	003553	104302	001451		MOV	EDC,R2	:EDC FOR PATTERN
214	003555	100232			MOV	R2,(R3)+	:STORE IT
215							
216							
217							
218					SET UP	END OF IMAGE POINTER	
219	003556	104203	001127		MOV	#SECTRK,R3	:SEC/TRACK
220	003560	104200	000003	000736	MOV	#3,TEMP	:FOR MULT
221	003563	114000	000737		CLR	TEMP+1	:FOR STORE
222	003565	104204	000736		MOV	#TEMP,R4	:SET UP FOR MULT
223	003567	021542			CALL	DMUL	:GET LENGTH OF IMAGE BLOCK
224	003570	104200	007275	000731	MOV	#IMAGE,DDUMMY	:FOR ADD
225	003573	114000	000732		CLR	DDUMMY+1	:CLEAR HIGH BYTE
226	003575	104203	000731		MOV	#DDUMMY,R3	:SET UP FOR ADD
227	003577	021506			CALL	DADD	:ADD TO GET ADDRESS
228	003600	104140	001231		MOV	(R4),EIMAGE	:GET ADDRESS

```

1
2
3
4
5
6 003602 104207 001037
7 003604 104673 000007
8 003606 110703
9 003607 103203 177400
10 003611 104030 000736
11 003613 114000 000737
12 003615 104204 000736
13 003617 104203 001127
14 003621 021542
15 003622 106300 001235 000736
16 003625 033635
17 003626 104200 000011 001234
18 003631 104200 000044 001233
19 003634 003643
20 003635 104200 000006 001234 TWO:
21 003640 104200 000033 001233 ISKIP:
22 003643

```

COMPUTE INTERLEAVE FACTOR
 MOV #CR,R0 ;POINT TO CHARACTERISTICS BLOCK
 MOV REVSEC(R0),R3 ;GET REVS/SECOND
 SWAB R3 ;GET INTO LOW BYTE
 BIC #HIBYTE,R3 ;CLEAR HIGH BYTE
 MOV R3,TEMP ;FOR MULTIPLY
 CLR TEMP+1 ;CLEAR FOR STORE
 MOV #TEMP,R4 ;SET UP FOR MULTIPLY
 MOV #SECTRK,R3 ;SECTORS/TRACK
 CALL DMUL ;GET SECTORS/SECOND
 CMP CUTOFF,TEMP ;WITHIN LIMITS ?
 BPL TWO ;DO BI-LEAVE
 MOV #THREB,TBLK ;ELSE DO TRI-LEAVE
 MOV #44,SKPCNT ;INIT CHECK PASS OFFSET
 BR ISKIP ;SKIP BI-LEAVE SETUP
 MOV #TWOB,TBLK ;DO BI-LEAVE
 MOV #33,SKPCNT ;INIT CHECK PASS OFFSET

LINE	ADDRESS	OPERAND	COMMENT
1			
2			
3			
4	003643	104207 001037	MOV #CR,R0 ;POINT TO CHARACTERISTICS BLK
5	003645	104673 000001	MOV FRCPY(R0),R3 ;GET F/RCT COPIES
6	003647	110703	SWAB R3 ;GET INTO LOW BYTE
7	003650	103203 177760	BIC #FCLR,R3 ;CLEAR OUT REST OF GARBAGE
8	003652	104030 001244	MOV R3,FCTCPY ;STORE
9	003654	104207 001052	MOV #SCR,R0 ;POINT TO SUBUNIT CHARACTERISTICS
10	003656	104670 000010 001240	MOV FCTSZ(R0),FCTFMT ;GET FCT SIZE IN SECTORS
11	003661	114000 001241	CLR FCTFMT+1 ;CLEAR HIGH ORDER
12	003663	104670 000014 001242	MOV RCTSZ(R0),RCTFMT ;GET RCT SIZE
13	003666	114000 001243	CLR RCTFMT+1 ;CLEAR HIGH ORDER
14			
15			
16			
17			
18			
19			
20	003670	104300 001133 000736	MOV LBNLBN,TEMP ;GET LOW ORDER
21	003673	104300 001134 000737	MOV LBNLBN+1,TEMP+1 ;GET HIGH ORDER
22	003676	104204 000736	MOV #TEMP,R4 ;FOR SUB
23	003700	104203 001135	MOV #RBNLBN,R3 ;POINT TO RBN'S IN LBN AREA
24	003702	021506	CALL DADD ;ADD TO GET HIGHEST PBN
25	003703	104203 001466	MOV #ONE,R3 ;DITTO
26	003705	021524	CALL DSUB ;TO GET ACTUAL PBN NUMBER
27	003706	104642 000001	MOV 1(R4),R2 ;GET HIGH ORDER
28	003710	107302 001335	SUB ST.LBN,R2 ;SUBTRACT TO GET RELATIVE LAST BLOCK
29	003712	104020 001155	MOV R2,HGHPBN+1 ;STORE IT
30	003714	104140 001154	MOV (R4),HGHPBN ;GET HIGH ORDER

LINE	ADDRESS	DATA	COMMENT
1			COMPUTE REVECTOR BUFFER ADDRESS AND MAX REVECTOR COUNT
2			
3			
4	003716	104200 000003 000736	MOV #IMLEN,TEMP ;GET LENGTH OF FORMAT IMAGE BLOCK
5	003721	114000 000737	CLR TEMP+1 ;CLEAR FOR STORE
6	003723	104204 000736	MOV #TEMP,R4 ;FOR MULT
7	003725	104203 001127	MOV #SECTRK,R3 ;SECTORS/TRACK
8	003727	021542	CALL DMUL ;GET LENGTH OF FORMAT BUFFER TABLE
9	003730	104303 000736	MOV TEMP,R3 ;GET LENGTH
10	003732	104204 007275	MOV #IMAGE,R4 ;GET IMAGE BUFFER START ADDRESS
11	003734	105034	ADD R3,R4 ;GET START ADDRESS OF REVECTOR BUFFER
12	003735	104040 001224	MOV R4,ERRBUF ;STORE IT
13	003737	104040 001252	MOV R4,ERPNT ;INIT POINTER
14	003741	104200 007775 000736	MOV #BMAX,TEMP ;GET MAX BUFFER ADDRESS
15	003744	114000 000737	CLR TEMP+1 ;FOR CLEAR
16	003746	104040 000731	MOV R4,DDUMMY ;STORE BEGINNING ADDRESS
17	003750	114000 000732	CLR DDUMMY+1 ;CLEAR HIGH WORD
18	003752	104204 000736	MOV #TEMP,R4 ;POINT TO END ADDRESS
19	003754	104203 000731	MOV #DDUMMY,R3 ;POINT TO BEGINNING ADDRESS
20	003756	021524	CALL DSUB ;SUBTRACT TO GET LENGTH
21	003757	104304 000736	MOV TEMP,R4 ;GET LENGTH
22	003761	110604	ROR R4 ;DIVIDE BY 2 (LENGTH OF 1 ENTRY)
23	003762	104040 001225	MOV R4,EMAX ;STORE AS MAX NUMBER
24			
25			STORE RETRY AND RECOVERY LEVELS
26			
27	003764	104207 001037	MOV #CR,R0 ;POINT TO CHARACTERISTICS
28	003766	104673 000002	MOV ERCV(R0),R3 ;GET RECOVERY LEVELS
29	003770	103203 177400	BIC #HI BYTE,R3 ;CLEAR HIGH ORDER
30	003772	104030 001503	MOV R3,RECOV ;STORE IT
31	003774	104030 001505	MOV R3,RECTMP ;INIT COUNTER
32	003776	104673 000001	MOV RTRY(R0),R3 ;GET RETRY NUMBER
33	004000	110603	ROR R3 ;SHIFT BY FOUR TO GET INTO LOW ORDER NIBBLE
34	004001	110603	ROR R3
35	004002	110603	ROR R3
36	004003	110603	ROR R3
37	004004	103203 177700	BIC #HI2BYTE,R3 ;CLEAR HIGH ORDER JUNK
38	004006	115403	INC R3 ;ONE MORE BECAUSE OF WRONG INC
39	004007	104030 001502	MOV R3,RETRY ;STORE IT
40	004011	114000 001504	CLR TMPTRY ;FOR STORE
41			
42			SET UP LONG TIMEOUT
43			
44			
45			
46	004013	104207 001037	MOV #CR,R0 ;POINT TO COMMON CHARACTERISTICS
47	004015	104673 000001	MOV LONGTO(R0),R3 ;GET LONG TIMEOUT IN LOG2
48	004017	103203 177760	BIC #FCLR,R3 ;CLEAR ALL BUT TIMEOUT
49	004021	117403	DEC R3 ;ALREADY HAVE SHIFTED ONCE
50	004022	104201 000001	MOV #1,R1 ;INIT COUNTER
51	004024	105201 000000	ADD #0,R1 ;CLEAR CARRY
52	004026	110201	TIMLOP: ROL R1 ;SHIFT
53	004027	117403	DEC R3 ;DECREMENT SHIFT COUNT
54	004030	054026	BNE TIMLOP ;CONTINUE TILL DONE
55	004031	104010 001250	MOV R1,LTO ;STORE IT FOR US LATER
56			
57			

```

58      :      SET UP SHORT TIMEOUT
59      :
60      :
61 004033 104207 001037      MOV      #CR,R0      ;POINT TO COMM CHAR
62 004035 104673 000000      MOV      SHORTO(R0),R3      ;GET SHORT TIMEOUT WORD
63 004037 103203 177760      BIC      #FCLR,R3      ;CLEAR WORD
64 004041 117403      DEC      R3      ;FOR LOG CALCULATION
65 004042 104201 000001      MOV      #1,R1      ;INIT COUNTER
66 004044 105201 000000      ADD      #0,R1      ;CLEAR CARRY
67 004046 110201      TILOP:  ROL      R1      ;ROTATE (MULT BY 2)
68 004047 117403      DEC      R3      ;DECREMENT COUNTER
69 004050 054046      BNE      TILOP      ;KEEP GOING TILL DONE
70 004051 104203 000012      MOV      #10.,R3      ;SHIFT COUNT FOR MILSEC CONVERSION
71 004053 110201      TILOP1: ROL      R1      ;GET NUMBER IN MILSECS
72 004054 117403      DEC      R3      ;GO TILL DONE
73 004055 054053      BNE      TILOP1      ;AGAIN
74 004056 104010 001251      MOV      R1,STO      ;STORE IT
75      :
76      :
77      :      GET SERIAL NUMER AND DATE FROM DMBUF
78      :
79      :
80 004060 104205 000004      MOV      #4,R5      ;COUNT OF WORDS IN SERIAL NUMBRE
81 004062 104207 001263      MOV      #DMBUF,R0      ;POINT TO BUFFER
82 004064 105207 000005      ADD      #SER,R0      ;POINT TO SERIAL NUMBER
83 004066 104203 001305      MOV      #SERNUM,R3      ;POINT TO BUFFER
84 004070 104272      1$:  MOV      (R0)+,R2      ;GET WORD
85 004071 100232      MOV      R2,(R3)+      ;STORE WORD
86 004072 117405      DEC      R5      ;DECREMENT COUNT
87 004073 054070      BNE      1$      ;CONTINUE TILL DONE
88 004074 104205 000004      MOV      #4,R5      ;RESET COUNT FOR DATE
89 004076 104207 001263      MOV      #DMBUF,R0      ;POINT TO DMBUF
90 004100 105207 000011      ADD      #DAT,R0      ;POINT TO DATE
91 004102 104203 001301      MOV      #DATE,R3      ;POINT TO DATE BUFFER
92 004104 104272      2$:  MOV      (R0)+,R2      ;GET WORD
93 004105 100232      MOV      R2,(R3)+      ;STORE WORD
94 004106 117405      DEC      R5      ;DECREMENT COUTNER
95 004107 054104      BNE      2$      ;CONTINUE TILL DONE
96      :
97      :
98      :      SET UP HEADER AND DATA PREAMBLE LENGTHS
99      :
100      :
101      :
102 004110 104207 001052      MOV      #SCR,R0      ;POINT TO CHARACTERISTICS BLK
103 004112 104670 000005 001334      MOV      DATA(R0),DPREA      ;DATA PREAMBLE LENGTH
104 004115 103200 177400 001334      BIC      #HIBYTE,DPREA      ;CLEAR OUT HIGH BYTE GARBAGE
105 004120 104673 000005      MOV      HEAD(R0),R3      ;HEADER PREAMBLE LENGTH
106 004122 110703      SWAB      R3      ;GET INTO LOW BYTE
107 004123 103203 177400      BIC      #HIBYTE,R3      ;CLEAR OUT HIGH BYTE GARBAGE
108 004125 104030 001333      MOV      R3,HPREA      ;STORE IT
109 004127 114001      CLR      R1      ;FOR NO ERROR

```

1				:	SET UP STARTING ADDRESS BITS	
2				:		
3				:		
4	004130	104203	001052	:	MOV #SCR,R3	;POINT TO SUBUNIT CHAR BLOCK
5				:		
6				:	LBN	
7				:		
8	004132	104637	000002	:	MOV STLBN(R3),R0	;GET THE WORD
9	004134	103207	170377	:	BIC #STCLR,R0	;CLEAR THE REST
10	004136	104070	001335	:	MOV R0,ST.LBN	;STORE IT
11				:		
12				:	RBN	
13				:		
14	004140	104637	000003	:	MOV STRBN(R3),R0	;GET RBN WORD
15	004142	103207	170377	:	BIC #STCLR,R0	;CLEAR THE REST
16	004144	104070	001336	:	MOV R0,ST.RBN	;STORE IT
17				:		
18				:	XBN	
19				:		
20	004146	104637	000002	:	MOV STXBN(R3),R0	;GET THE WORD
21	004150	110607		:	ROR R0	
22	004151	110607		:	ROR R0	
23	004152	110607		:	ROR R0	
24	004153	110607		:	ROR R0	;GET INTO THE RIGHT NIBBLE
25	004154	103207	170377	:	BIC #STCLR,R0	;CLEAR THE REST
26	004156	104070	001337	:	MOV R0,ST.XBN	;STORE IT
27				:		
28				:	DBN	
29				:		
30	004160	104637	000003	:	MOV STDBN(R3),R0	;GET THE WORD
31	004162	110607		:	ROR R0	
32	004163	110607		:	ROR R0	
33	004164	110607		:	ROR R0	
34	004165	110607		:	ROR R0	;GET INTO THE RIGHT NIBBLE
35	004166	103207	170377	:	BIC #STCLR,R0	;CLEAR THE REST
36	004170	104070	001340	:	MOV R0,ST.DBN	;STORE IT


```

58 004320 074352
59 004321 104653 000003
60 004323 104641 000000
61 004325 103201 170000
62 004327 107013
63 004330 104202 000001
64 004332 105202 000000
65 004334 110202
66 004335 117403
67 004336 054334
68 004337 105202 000000
69 004341 110202
70 004342 110202
71 004343 110202
72 004344 110202
73 004345 104020 001020
74 004347 101020 001025
75 004351 000000
76 004352 104302 000740
77 004354 105022
78 004355 104020 000740
79 004357 114000 000717
80 004361 114000 001501
81 004363 106202 000010
82 004365 034212
83 004366 104652 000002
84 004370 104201 000007
85 004372 022552

```

```

BMI NOTHER
MOV DMUNIT(R5),R3
MOV UID(R4),R1
BIC #H,CLR,R1
SUB R1,R3
MOV #1,R2
ADD #0,R2
SFTRPT: ROL R2
DEC R3
BNE SFTRPT
GDONE: ADD #0,R2
ROL R2
ROL R2
ROL R2
ROL R2
MOV R2,GSR+1
BIS R2,ACC+1
RETURN
NOTHER: MOV UNIT,R2
ADD R2,R2
MOV R2,UNIT
CLR UN.ER1
CLR COUNT
CMP #8,R2
BPL GOVER
MOV 2(R5),R2
MOV #7,R1
CALL ERRMNT

```

```

;IF NEGATIVE THEN NOT IN THE RANGE
;GET DESIRED
;GET ORIGINAL
;CLEAR SUBUNIT MASK
;FIGURE OUT WHICH SUBUNIT
;INIT SUBUNIT MASK
;TO CLEAR CARRY
;SHIFT LEFT
;DECREMTN COUNTER
;REPEAT SHIFT
;CLEAR CARRY
;HAVE TO ROTATE
;4 TIMES TO DO A
;SWAP NIBBLE
;FOR SUBUNIT MASK
;STORE MASK IN SUBUNIT CHAR COMMAND
;SET IN FOR CHANGE MODE COMMAND
;RETURN
;GET CURRENT INTERCONNECT
;NEXT PORT
;SAVE BACK
;FOR RESTORE
;CLEAR ERROR COUNT
;ALL DONE ?
;NOPE - TRY THIS INTERCONNECT
;UNIT SEARCHING FOR
;SIGNAL NON-EXISTANT UNIT
;ERROR RETURN

```

```

1
2
3
4 004373 104203 001263      SET UP FLAGS
5 004375 104634 000004      FLGSET: MOV    #DMBUF,R3      ;POINT TO BUFFER
6 004377 102204 000002      MOV    DMFLG(R3),R4    ;GET FLAG WORD
7 004401 054405      BIT    #BIT1,R4      ;STOP IF BAD FLAG ?
8 004402 101200 000020 001217  BNE    FLGSKP      ;YES - SKIP SET
9 004405 102204 000001      BIS    #GOBAD,FLAG    ;SET IT
10 004407 054417      FLGSKP: BIT    #BIT0,R4      ;USE RESIDENT FCT ?
11 004410 102204 000004      BNE    RFACT      ;YUP - SET IT UP
12 004412 054423      BIT    #BIT2,R4      ;DOWN-LINE LOAD FCT ?
13 004413 101200 002000 001217  BNE    DFCT      ;YUP - SET IT UP
14 004416 004427      BIS    #BSTGS,FLAG    ;DO BEST GUESS
15 004417 101200 000001 001217  BR     FLGDON      ;ELS USE BEST GUESS
16 004422 004427      RFACT: BIS    #FCTAVL,FLAG   ;SET FLAG
17 004423 101200 000400 001217  BR     FLGDON      ;EXOT
18 004426 004427      DFCT:  BIS    #DLL,FLAG     ;SET FOR DLL
19 004427 000000      BR     FLGDON      ;DONE
20
21
22
23      INITIALIZATION ROUTINE
24 004430 104302 000740      INITL: MOV    UNIT,R2      ;SELECT UNIT
25 004432 022340      CALL    INITIT      ;INIT DRIVE
26 004433 024203      CALL    GETUNT      ;GET THE SDI INTERCONNECT
27 004434 024373      CALL    FLGSET      ;SET UP FLAGS
28 004435 022332      CALL    ONLIN      ;BRING IT ONLINE
29 004436 021662      CALL    GSTATS     ;GET STATUS
30 004437 115001      TST     R1          ;ANY ERRORS ?
31 004440 054443      BNE    OOPS        ;YUP - QUIT
32 004441 022154      CALL    RECAL      ;RECALIBRATE
33 004442 023067      CALL    CONINT     ;COMPUTE DISK CONSTANTS
34 004443 000000      OOPS:  RETURN    ;AND STOP

```

DBN/XBN FORMAT OVERLAY (F1)

FORMAT DBN AND XBN AREAS

8	004444				DMOVLY	F1,START	
10	003047	104200	000000	001153	MOV	#F1,CUROVL	:OVERLAY #1
11	003052	101200	000010	001217	BIS	#DBN,FLAG	:SET DBN FORMAT
12	003055	023114			CALL	DXFORM	:FORMAT DBN AREA
13	003056	104303	001311		MOV	FCTREV,R3	:STARTING FCT ENTRY COUNT
14	003060	107303	001237		SUB	FCNT,R3	:TOTAL BAD IN DBN AREA
15	003062	105303	001453		ADD	ERRCNT,R3	:GET FINAL TOTAL
16	003064	104030	001314		MOV	R3,DBBAD	:STORE IT FOR STATS
17	003066	114000	001453		CLR	ERRCNT	:FOR CLEAR
18	003070	104300	001237	001311	MOV	FCNT,FCTREV	:FOR BAD BLOCK COUNT
19	003073	103200	000010	001217	BIS	#DBN,FLAG	:DO XBN AREA
20	003076	023114			CALL	DXFORM	:FORMAT XBN AREA
21	003077	104303	001311		MOV	FCTREV,R3	:STARTING FCT ENTRY COUNT
22	003101	107303	001237		SUB	FCNT,R3	:TOTAL BAD IN XBN AREA
23	003103	105303	001453		ADD	ERRCNT,R3	:GET FINAL TOTAL
24	003105	104030	001315		MOV	R3,XBBAD	:STORE IT FOR STATS
25	003107	114000	001453		CLR	ERRCNT	:FOR CLEAR
26	003111	104201	000006		MOV	#F3,R1	:FCT DLL OVERLAY
27	003113	022412			CALL	NEXT	:BRING IN NEXT OVERLAY
28							
29	003114	104207	001052		DXFORM: MOV	#SCR,R0	:POINT TO CHARACTERISTICS BLOCK
30	003116	102200	000010	001217	BIT	#DBN,FLAG	:DO DBN AREA ?
31	003121	013171			BEQ	XBNIT	:NO - DO XBN AREA
32	003122	104673	000022		MOV	DBNCYL(R0),R3	:GET NUMBER OF CYLS TO FM
33	003124	110703			SWAB	R3	:GET INTO LOW BYTE
34	003125	103203	177400		BIC	#HIBYTE,R3	:CLEAR HI BYTE
35	003127	104030	001460		MOV	R3,CNTCYL	:SET UP COUNTER
36	003131	102200	002000	001217	BIT	#BSTGS,FLAG	:DOING BEST GUESS ???
37	003134	053164			BNE	SKIP4	:YES - SKIP FCT SET UP
38	003135	104200	000200	001500	MOV	#128,PCNT	:FOR PBN COUNT INIT
39	003140	114000	001256		CLR	FCTCNT	:FOR INIT FCT READ
40	003142	023407			CALL	DXFCPG	:READ IT IN
41	003143	104200	005152	001223	MOV	#PBNBUF,BADPBN	:ADDR OF BAD PBN LIST
42	003146	104300	005170	001237	MOV	PBNBUF+C512,FCNT	:GET COUNT OF USED ENTRIES
43	003151	104300	005170	001311	MOV	PBNBUF+C512,FCTREV	:STORE IT FOR STAT COMPUTATION
44	003154	013161			BEQ	SKIP19	:IF ZERO - THEN NO ENTRIES
45	003155	115400	001256		INC	FCTCNT	:START WITH SECOND FCT BLOCK
46	003157	023407			CALL	DXFCPG	:BRING IT IN
47	003160	003164			BR	SKIP4	:SKIP NO ENTRY STUFF
48	003161	101200	000002	001217	SKIP19: BIS	#FCTEMT,FLAG	:SET EMPTY FLAG
49	003164	104200	140000	001462	SKIP4: MOV	#HD,DBN,HD.CUR	:GET DBN HEADER CODE
50	003167	023423			CALL	NUMDBN	:GET DBN OF FIRST BLOCK ON LAST CYLINDER
51	003170	003200			BR	SKIP1	:SKIP XBN SETUP
52	003171	104200	120000	001462	XBNIT: MOV	#HD,XBN,HD.CUR	:GET XBN HEADER CODE
53	003174	104670	000021	001460	MOV	XBNCYL(R0),CNTCYL	:GET CYLINDERS IN XBN AREA
54	003177	023453			CALL	NUMXBN	:GET XBN OF FIRST BLOCK ON LAST XBN CYL
55	003200	104140	001117		SKIP*: MOV	(R4),HOLDBN	:LO ORDER FIRST BLOCK NUM TO DO
56	003202	104240	001113		MOV	(R4)+,CURBN	:AND MAKE IT CURRENT NUMBER
57	003204	104140	001120		MOV	(R4),HOLDBN+1	:HI ORDER FIRST BLOCK NUM TO DO

58	003206	104140	001114		MOV	(R4),CURBN+1	:AND MAKE IT CURRENT NUMBER
59	003210	104207	001052		MOV	#SCR,R0	:POINT TO CHARACTERISTICS BLK
60	003212	104670	000002	001464	MOV	GRPCYL(R0),GRPCNT	:LOAD GROUPS/CYL
61	003215	103200	177400	001464	BIC	#HIBYTE,GRPCNT	:CLEAR OUT HIGH GARBAGE
62	003220	104300	001464	001463	MOV	GRPCNT,CURGRP	:GET GROUP NUMBER BY
63	003223	117400	001463		DEC	CURGRP	:DECREMENTING
64	003225	104300	001125	001076	MOV	CYLNUM,ISEEK+1	:GET LO ORDER WORD OF CYLINDER NUMBER
65	003230	104300	001126	001077	MOV	CYLNUM+1,ISEEK+2	:LOAD HIGH ORDER WORD OF CYL NUM
66	003233	104300	001463	001100	MOV	CURGRP,ISEEK+3	:LOAD GROUP NUMBER
67	003236	022251			CALL	SEEK	:SEEK TO CURRENT CYL NUM
68	003237	115001			TST	R1	:ANY ERRORS ?
69	003240	073402			BMI	SKERR	:YUP - QUIT
70	003241	104207	001052		MOV	#SCR,R0	:POINT TO CHARACTERISTICS BLOCK
71	003243	104673	000003		MOV	TRKGRP(R0),R3	:LOAD TRACKS/GROUP
72	003245	103203	177400		BIC	#HIBYTE,R3	:CLEAR OUT HIGH GARBAGE
73	003247	104030	001465		MOV	R3,TRKCNT	:STORE IN COUNTER
74	003251	117403			DEC	R3	:TRACK NUMBER IS ONE LESS
75	003252	104030	001112		MOV	R3,CURTRK	:RESET CURRENT TRACK NUMBER
76	003254	104201	000047		MOV	#G7,R1	:FORMAT SETUP OVERLAY
77	003256	022444			CALL	PAGE	:DO IT
78	003257	104304	001334		MOV	DPREA,R4	:DATA PREAMBLE LENGTH
79	003261	104303	001333		MOV	HPREA,R3	:HEADER PREAMBLE LENGTH
80	003263	104307	001332		MOV	IMSTAR,R0	:POINT TO FORMAT IMAGE START POINT
81	003265	104301	001112		MOV	CURTRK,R1	:TRACK TO FORMAT
82	003267	104302	000740		MOV	UNIT,R2	:SDI INTERCONNECT
83	003271	104205	007275		MOV	#IMAGE,R5	:RECIRCULATION POINTER
84	003273	060001			XFC	FORMAT	:FORMAT THE TRACK
85	003274	115001			TST	R1	:ANY ERRORS ?
86	003275	013310			BEQ	SKIP6	:NO - DO CHECK PASS
87	003276	115400	000717		INC	UN.ERI	:INCREMENT IT
88	003300	106300	001502	000717	CMF	RETRY,UN.ERI	:DONE ALL RETRIES ?
89	003303	073376			BMI	FERR	:YUP - ERROR
90	003304	022371			CALL	INITPT	:REINIT
91	003305	022243			CALL	CLEAR	:DRIVE CLEAR
92	003306	022251			CALL	SEEK	:RE-SEEK AND GROUP SELECT
93	003307	003257			BR	SKIP7	:RETRY FORMAT
94	003310	114000	000717		CLR	UN.ERI	:FOR STORE
95	003312	023477			CALL	DXCHEC	:DO CHECK PASS
96	003313	117400	001112		DEC	CURTRK	:DECREMENT IT
97	003315	104204	001117		MOV	#HOLDBN,R4	:PREPARE FOR BEGINNING BLOCK DECREMENT
98	003317	104203	001127		MOV	#SECTRK,R3	:DECREMENT BY SECTORS/TRACK
99	003321	021524			CALL	DSUB	:DO DECREMENT
100	003322	104300	001117	001113	MOV	HOLDBN,CURBN	:LO ORDER NEW BLOCK NUMBER
101	003325	104300	001120	001114	MOV	HOLDBN+1,CURBN+1	:HI ORDER NEW BLOCK NUMBER
102	003330	117400	001465		DEC	TRKCNT	:DECREMENT IT
103	003332	053254			BNE	SKIP3	:NO - DO NEXT TRACK
104	003333	117400	001463		DEC	CURGRP	:DECREMENT GROUP NUMBER
105	003335	117400	001464		DEC	GRPCNT	:DECREMENT IT
106	003337	053225			BNE	SLEEK2	:NO - DO NEXT GROUP
107	003340	117400	001460		DEC	CNTCYL	:DECREMENT IT
108	003342	060022			XFC	UPDATE	:UPDATE PROGRESS INDICATOR
109	003343	104207	001263		MOV	#DMBUF,R0	:POINT TO MAINT BUFFER
110	003345	104303	001316		MOV	CYLMSG,R3	:GET DUP CODE
111	003347	100173			MOV	R3,(R0)	:STORE IT
112	003350	104303	001317		MOV	CYLMSG+1,R3	:GET CYLINDER MSG FLAG
113	003352	100673	000001		MOV	R3,1(R0)	:STORE IT
114	003354	104303	001125		MOV	CYLNUM,R3	:GET LOW ORDER CYLINDER NUMBER

119	003356	100673	000003		MOV	R3,3(R0)	;PUT IN MESSAGE
120	003360	104303	001126		MOV	CYLNUM+1,R3	;GET HIGH ORDER
121	003362	100673	000002		MOV	R3,2(R0)	;STORE IT
122	003364	022536			CALL	SNDMNT	;LET HOST KNOW WE'RE ALIVE
123	003365	104204	001125	SKIP12:	MOV	#CYLNUM,R4	;PREPARE FOR CYL NUM DECREMENT
124	003367	104203	001466		MOV	#ONE,R3	;DECREMENT BY ONE
125	003371	021524			CALL	DSUB	;DO SUBTRACT
126	003372	115000	001460		TST	CNTCYL	;ARE WE DONE ?
127	003374	053210			BNE	SLEEK	;DONE ? NO - DO NEXT CYLINDER
128	003375	000000			RETURN		;YES - ALL DONE
129	003376	104012		FERR:	MOV	R1,R2	;GET XFC FAILURE CODE
130	003377	104201	000010		MOV	#8.,R1	;SET FORMAT ERROR
131	003401	003406			BR	DXERR	
132	003402	104302	001125	SKERR:	MOV	CYLNUM,R2	;CYLINDER WHICH FAILED ON
133	003404	104201	000012		MOV	#10.,R1	;SEEK ERROR
134	003406	022552		DXERR:	CALL	ERRMNT	;ERROR RETURN

```

1
2
3
4 003407
5 003410 104201 000033
6 003412 022444
7 003413 104200 000200 001501
8 003416 104200 005152 001223
9 003421
10 003422 000000
11
12
13
14
15
16
17 003423 104673 000022
18 003425 110703
19 003426 103203 177400
20 003430 104030 000736
21 003432 114000 000737
22 003434 104204 000736
23 003436 104203 001131
24 003440 021542
25 003441 104641 000001
26 003443 105301 001340
27 003445 100641 000001
28 003447 104203 001127
29 003451 021524
30 003452 000000
31
32
33
34
35
36
37 003453 104670 000021 000736
38 003456 114000 000737
39 003460 104204 000736
40 003462 104203 001131
41 003464 021542
42 003465 104641 000001
43 003467 105301 001337
44 003471 100641 000001
45 003473 104203 001127
46 003475 021524
47 003476 000000
48
49
50
51
52
53 003477
54 003500 114000 001226
55 003502 114000 001221
56 003504 102200 000600 001217
57 003507 013522

:
: PAGE IN NEW FCT BLOCK
:
: DXFCPG: PUSH R0
: MOV #G2,R1
: CALL PAGE
: MOV #128.,COUNT
: MOV #PBNBUF,BADPBN
: POP R0
: RETURN
: DLL OVERLAY
: EXECUTE OVERLAY
: FOR INIT
: FOR POINTER RESET
: RESTORE R0
: RETURN

:
: COMPUTE NUMBER OF FIRST DBN ON LAST DBN TRACK
: R0 -> CHARACTERISTICS BLOCK
:
:
: NUMDBN: MOV DBN(CYL(R0),R3
: SWAB R3
: BIC #HIBYTE,R3
: MOV R3,TEMP
: CLR TEMP+1
: MOV #TEMP,R4
: MOV #SECTCY,R3
: CALL DMUL
: MOV 1(R4),R1
: ADD ST.DBN,R1
: MOV R1,1(R4)
: MOV #SECTRK,R3
: CALL DSUB
: RETURN
: GET NUMBER OF CYLINDERS IN DBN AREA
: GET INTO LOW BYTE
: CLEAR OUT OTHER INFO
: MOVE TO TEMP AREA
: CLEAR FOR STORE
: POINT R4 AT TEMP AREA
: POINT TO NUM OF SECTORS/CYLINDER
: MULTIPLY TO GET SECTORS BEFORE LAST CYL
: GET HIGH ORDER
: ADD HIGH ORDER STARTING DBN
: STORE BACK
: WANT FIRST DN OF LAST TRACK
: SUB TO GET IT

:
: COMPUTE NUMBER OF FIRST XBN ON LAST XBN TRACK
: R0 -> CHARACTERISTICS BLOCK
:
:
: NUMXBN: MOV XBN(CYL(R0),TEMP
: CLR TEMP+1
: MOV #TEMP,R4
: MOV #SECTCY,R3
: CALL DMUL
: MOV 1(R4),R1
: ADD ST.XBN,R1
: MOV R1,1(R4)
: MOV #SECTRK,R3
: CALL DSUB
: RETURN
: GET NUMBER OF CYLINDERS IN XBN AREA
: CLEAR FOR STORE
: POINT TO TEMP AREA
: POINT TO SECTORS/CYLINDER
: MULTIPLY TO GET SECTORS IN XBN AREA
: GET HIGH ORDER
: ADD HIGH ORDER STARTING XBN
: STORE BACK
: WANT XBN OF LAST TRACK
: SUB TO GET IT

:
: CHECK PASS
:
: DXCHEC: PUSH R0
: DXCH: CLR ERR
: CLR ERFLAG
: BIT #MANU+DLL,FLAG
: BEQ CSKIP
: SAVE PTR TO CHARACTERISTICS BLK
: FOR ERROR COUNT RESET
: CLEAR RE-FORMAT FLAG
: FCT AVAILABLE ?
: NO - DO EXTENSIVE READ

```

58	003510	104200	000001	001455	MOV	#1,N	:SET UP FOR STORE
59	003513	104200	000005	001456	MOV	#5,N1	:SET UP
60	003516	104300	001456	001457	MOV	N1,NN1	:SAVE FOR LATER RESET
61	003521	003533			BR	CSKIP2	:SKIP EXTENSIVE READ SETUP
62	003522	104200	000003	001455	CSKIP: MOV	#3,N	:EXTENSIVE REGULAR READ
63	003525	104200	000024	001456	MOV	#20,N1	:EXTENSIVE ERROR READS
64	003530	104300	001456	001457	MOV	N1,NN1	:SAVE FOR LATER RESET
65	003533	024046			CSKIP2: CALL	FIXIT	:DO IT
66	003534	104302	000740		CSKIP1: MOV	UNIT,R2	:SDI INTERCONNECT
67	003536	060012			XFC	SIP	:SYNCH WITH SECTOR/INDEX PULSE
68	003537	104300	001127	001454	MOV	SECTRK,SECCNT	:LOAD SECTORS/TRACK
69	003542	104205	006621		MOV	#CMDBUF,R5	:POINT TO COMMAND BUFFER
70	003544	104207	000721		AGAIN: MOV	#RDBLK,R0	:POINT TO READ COMMAND BLOCK
71	003546	104653	000000		MOV	RB.BUF(R5),R3	:ZERO BUFFER POINTER ?
72	003550	013626			BEQ	NOERR	:YES - SKIP CHECKS
73	003551	100673	000001		MOV	R3,RW.BUF(R0)	:ELSE STORE IT
74	003553	104653	000001		MOV	RB.LOW(R5),R3	:GET LOW ORDER BLOCK NUMBER
75	003555	100673	000002		MOV	R3,RW.LOW(R0)	:STORE IN COMMAND BLOCK
76	003557	104653	000002		MOV	RB.HI(R5),R3	:LOAD HIGH ORDER BLOCK NUMBER
77	003561	100673	000003		MOV	R3,RW.HI(R0)	:STORE IN COMMAND BLOCK
78	003563	104653	000003		MOV	RB.CMD(R5),R3	:GET READ COMMAND AND TRACK NUMBER
79	003565	100673	000004		MOV	R3,RW.CMD(R0)	:STORE IN COMMAND BLOCK
80	003567	104203	000726		MOV	#HSLIM-1,R3	:POINTER TO DUMMY SDI BLOCK
81	003571	100673	000005		MOV	R3,RW.DUM(R0)	:STORE IT IN READ BLOCK
82	003573	104207	100721		READ1: MOV	#<BIT15!RDBLK>,R0	:MAKE SURE POINTING AT BLOCK
83	003575	104203	100000		MOV	#RDCMD,R3	:RESET STATUS POINTER
84	003577	100673	000000		MOV	R3,RW.STAT(R0)	:STORE IT BACK
85	003601	060002			XFC	READ	:READ 1 SECTOR
86	003602	115001			TST	R1	:ANY ERRORS ?
87	003603	053611			BNE	RRERR	:YES - UH OH
88	003604	104207	001443		MOV	#NUM,R0	:POINT TO COMPARE BLOCK
89	003606	060006			XFC	CMPCAT	:DO DATA COMPARE
90	003607	115001			TST	R1	:ANY ERROR IN COMPARE ?
91	003610	013626			BEQ	NOERR	:NOPE - CONTINUE LOOP
92	003611				RRERR:		
93	003611	104653	000004		MOV	RB.IM(R5),R3	:GET POINTER TO IMAGE
94	003613	104134			MOV	(R3),R4	:GET BUFFER POINTER WORD
95	003614	102204	020000		BIT	#BD,R4	:ALREADY MARKED BAD ??
96	003616	053626			BNE	NOERR	:YUP - DON'T COUNT AGAIN
97	003617	101204	020000		BIS	#BD,R4	:FLAG AS BAD
98	003621	100134			MOV	R4,(R3)	:STORE BACK
99	003622	115400	001226		INC	ERR	:INCREMENT ERROR COUNT
100	003624	115400	001473		INC	RTYCN	:INC COUNTER
101	003626	105205	000005		NOERR: ADD	#RDLEN,R5	:POINT TO NEXT READ CMD BLOCK
102	003630	114000	001504		CLR	TMPTRY	:FOR RESET
103	003632	104300	001503	001505	MOV	RECOV,RECTMP	:GET RECOVERY LEVELS
104	003635	103200	000002	001220	BIC	#RTYDN,FLAG1	:CLEAR RETRY DONE FLAG
105	003640	117400	001454		DEC	SFCCNT	:DECREMENT COUNTER
106	003642	053544			BNE	AGAIN	:NO - DO NEXT SECTOR
107	003643	117400	001455		DEC	N	:DECREMENT COUNTER
108	003645	053534			BNE	CSKIP1	:NO - REPEAT TRACK READ AND COMPARE
109	003646	115000	001226		TST	ERR	:ANY ERRORS ?
110	003650	014044			BEQ	CDONE	:NO - ALL DONE CHECK PASS
111	003651	104204	007275		MOV	#IMAGE,R4	:POINT TO IMAGE BUFFER
112	003653	104143			HERE: MOV	(R4),R3	:GET BUFFER POINTER WORD
113	003654	102203	020000		BIT	#BD,R3	:IS IT BAD
114	003656	014000			BEQ	CSKIP7	:NO - SKIP IT

115	003657	104643	000002	MOV	FT.HI(R4),R3	:GET HIGH ORDER
116	003661	103203	007777	BIC	#LO,R3	:CLEAR OUT ALL BUT HEADER
117	003663	106303	001462	CMP	HD.CUR,R3	:IS IT A 'GOOD X/D BN' ?
118	003665	054000		BNE	CSKIP7	:NOPE - ALREADY REVECTORED IT
119	003666	104302	000740	MOV	UNIT,R2	:SDI INTERCONNECT
120	003670	060012		XFC	SIP	:WAIT FOR PULSE
121	003671	104207	000721	MOV	#RDBLK,R0	:PREPARE FOR READ SECTORS
122	003673	104203	000726	MOV	#HSLIM-1,R3	:POINTER TO DUMMY SDI BLOCK
123	003675	100673	000005	MOV	R3,RW.DUM(R0)	:STORE IN COMMAND BLOCK
124	003677	104643	000001	MOV	1(R4),R3	:LO ORDER BLOCK NUMBER
125	003701	100673	000002	MOV	R3,RW.LOW(R0)	:STORE IN READ CMD BLOCK
126	003703	104643	000002	MOV	2(R4),R3	:HI ORDER BLOCK NUM AND CODE
127	003705	100673	000003	MOV	R3,RW.HI(R0)	:STORE IN READ CMD BLOCK
128	003707	104203	004535	MOV	#RDBUF,R3	:LOAD ADDRESS OF DATA BUFFER
129	003711	100673	000001	MOV	R3,RW.BUF(R0)	:STORE IN COMMAND BUFFER
130	003713	104203	013400	MOV	#RWCMD,R3	:LOAD SDI READ COMMAND
131	003715	104301	001112	MOV	CURTRK,R1	:GET CURRENT HEAD NUMBER IN R1
132	003717	101013		BIS	R1,R3	:SET IT IN COMMAND
133	003720	100673	000004	MOV	R3,RW.CMD(R0)	:STORE BACK
134	003722	104207	100721	READ2: MOV	#<BIT15!RDBLK>,R0	:MAKE SURE POINTING AT BLOCK
135	003724	104203	100000	MOV	#RDCMD,R3	:MARK AS ONLY REQUEST
136	003726	100173		MOV	R3,(R0)	:STORE IN CMD BLOCK
137	003727	104302	000740	MOV	UNIT,R2	:SDI INTERCONNECT
138	003731	060002		XFC	READ	:READ 1 SECTOR
139	003732	115001		TST	R1	:ANY ERROR IN READ ?
140	003733	053753		BNE	ER1	:YES - CONSIDER BAD
141	003734	104173		MOV	(R0),R3	:LOAD ECC ERROR INDICATOR FOR TEST
142	003735	102203	010000	BIT	#ECCF,R3	:ERROR ?
143	003737	013743		BEQ	CSKIP6	:NO - CHECK EDC
144	003740	023027		CALL	ECCCK	:ELSE FIND HOW MANY SYMBOLS IN ERROR
145	003741	115001		TST	R1	:WITHIN BOUNDS ?
146	003742	073753		BMI	ER1	:NOPE - CONSIDER BAD
147	003743	104202	004535	CSKIP6: MOV	#RDBUF,R2	:POINT TO BUFFER
148	003745	022627		CALL	EDC	:COMPUTE EDC - RETURNED IN R3
149	003746	104205	004535	MOV	#RDBUF,R5	:POINT TO BUFFER
150	003750	106653	000400	CMP	RW.EDC(R5),R3	:EDC O.K. ?
151	003752	013770		BEQ	OK	:NO ERROR
152	003753			ER1:		
153	003753	104643	000002	MOV	FT.HI(R4),R3	:GET HI ORDER BLOCK NUM AND HDR CODE
154	003755	103203	170000	BIC	#HD.CLR,R3	:CLEAR THE HEADER
155	003757	101203	110000	BIS	#HD.BAD,R3	:MARK AS BAD
156	003761	100643	000002	MOV	R3,FT.HI(R4)	:STORE BACK IN IMAGE
157	003763	115400	001221	INC	ERFLAG	:SET RE-FORMAT FLAG
158	003765	115400	001453	INC	ERRCNT	:UP COUNTER OF BAD BLOCKS
159	003767	003773		BR	CSKIP3	:NO NEED TO RE-READ ANY MORE THIS SECTOR
160	003770	117400	001456	OK: DEC	N1	:DECREMENT COUNTER - DONE ?
161	003772	053653		BNE	HERE	:NO - RE-READ SECTOR IN ERROR
162	003773	104300	001457	001456 CSKIP3: MOV	NN1,N1	:GET SAVED VALUE
163	003776	117400	001226	DEC	ERR	:DECREMENT IT
164	004000	105204	000003	CSKIP7: ADD	#IMLEN,R4	:POINT TO NEXT ERROR ENTRY
165	004002	115000	001226	TST	ERR	:ALL DONE ERROR SECTORS
166	004004	053653		BNE	HERE	:NO - DO NEXT SECTOR
167	004005	115000	001221	TST	ERFLAG	:WERE THERE ANY BAD SECTORS FOUND
168	004007	014044		BEQ	CDONE	:NOPE - ALL DONE
169	004010	104304	001334	OVER2: MOV	DPREA,R4	:DATA PREAMBLE LENGTH
170	004012	104303	001333	MOV	HPREA,R3	:HEADER PREAMBLE LENGTH
171	004014	104307	001332	MOV	IMSTAR,R0	:POINT TO TRACK IMAGE START POINT

172	004016	104301	001112		MOV	CURTRK,R1	:CURRENT TRACK NUMBER
173	004020	104302	000740		MOV	UNIT,R2	:SDI INTERCONNECT
174	004022	104205	007275		MOV	#IMAGE,R5	:RECIRCULATION ADDRESS
175	004024	060001			XFC	FORMAT	:RE-FORMAT
176	004025	115001			TST	R1	:ANY PROBLEMS ??
177	004026	014041			BEQ	OVER1	:NO -REDO CHECK PASS
178	004027	115400	000717		INC	UN.ERI	:INCREMENT IT
179	004031	106300	001502	000717	CMP	RETRY UN.ER1	:DONE ALL RETRIES ?
180	004034	073376			BMI	FERR	:YUP - ERROR
181	004035	022371			CALL	INITPT	:REINIT
182	004036	022243			CALL	CLEAR	:DRIVE CLEAR
183	004037	022251			CALL	SEEK	:RE-SEEK AND GROUP SELECT
184	004040	004010			BR	OVER2	:RETRY FORMAT
185	004041	114000	000717	OVER1:	CLR	UN.ERI	:CLEAR RETRY COUNT
186	004043	003500		(DONE:	BR	DXCH	:RE-CYCLE CHECK PASS
187	004044				POP	RO	:RESTORE CHARACTERISTICS PTR
188	004045	000000			RETURN		

```

1
2
3
4
5
6 004046 104300 001127 001454 FIXIT: MOV SECTRK,SECCNT ;INIT COUNTER
7 004051 104207 006621 MOV #CMDBUF,R0 ;COMMAND BUFFER
8 004053 104205 007275 MOV #IMAGE,R5 ;POINT TO TRACK IMAGE
9 004055 104303 001233 MOV SKPCNT,R3 ;GET STARTING OFFSET(TUNED)
10 004057 105035 ADD R3,R5 ;POINT TO FIRST ENTRY
11 004060 104050 001232 MOV R5,STARIT ;MARK STARTING ADDRESS
12 004062 104653 000002 MORE: MOV 2(R5),R3 ;SET UP FOR HSR CODE COMPARE
13 004064 103203 007777 BIC #LO,R3 ;ISOLATE HI 4 BITS(HDR CODE)
14 004066 106203 120000 CMP #HD,XBN,R3 ;GOOD XBN ?
15 004070 014102 BEQ FKIP2 ;YES - MARK AS GOOD TO CHECK
16 004071 106203 140000 CMP #HD,DBN,R3 ;GOOD DBN ?
17 004073 014102 BEQ FKIP2 ;YES - MARK AS GOOD TO CHECK
18 004074 114003 CLR R3 ;CLEAR FOR STORE
19 004075 100673 000000 MOV R3,RB.BUF(R0) ;STORE AS BAD SECTOR FLAG
20 004077 105207 000005 ADD #RDLEN,R0 ;POINT PAST BLOCK
21 004101 004121 BR FKIP1 ;SKIP GOOD MARK
22 004102 104203 004535 FKIP2: MOV #RDBUF,R3 ;LOAD ADDRESS OF DATA BUFFER
23 004104 100273 MOV R3,(R0)+ ;STORE BUFFER ADDRESS
24 004105 104653 000001 MOV 1(R5),R3 ;LO ORDER BLOCK NUMBER
25 004107 100273 MOV R3,(R0)+ ;STORE IN READ CMD BLOCK
26 004110 104653 000002 MOV 2(R5),R3 ;HI ORDER BLOCK NUM AND CODE
27 004112 100273 MOV R3,(R0)+ ;STORE IN READ CMD BLOCK
28 004113 104203 013400 MOV #RWCMD,R3 ;LOAD SDI READ COMMAND
29 004115 101303 001112 BIS CURTRK,R3 ;SET IN CURRENT TRACK NUMBER
30 004117 100273 MOV R3,(R0)+ ;STORE IN BLOCK
31 004120 100275 MOV R5,(R0)+ ;SAVE PTR TO IMAGE BLK ENTRY
32 004121 105305 001234 FKIP1: ADD TBLK,R5 ;ADD TO GET NEXT SECTOR
33 004123 106305 001231 CMP EIMAGE,R5 ;SEE IF HAVE TO LOOP BACK TO TOP
34 004125 014134 BEQ REDO ;NEED TO RESET
35 004126 034136 BPL FKP1 ;NO NEED - JUST CONTINUE
36 004127 107305 001231 SUB EIMAGE,R5 ;SUBTRACT TO GET LOOP AMOUNT
37 004131 105205 007275 ADD #IMAGE,R5 ;AND ADD OFFSET
38 004133 004136 BR FKP1 ;SKIP ZERO CONDITION
39 004134 104205 007275 REDO: MOV #IMAGE,R5 ;IF ZERO SIMPLY MOVE TO FRONT
40 004136 106305 001232 FKP1: CMP STARIT,R5 ;AT BEGINNING ADDRESS ?
41 004140 054145 BNE FKIP10 ;NO - JUST CONTINUE
42 004141 105205 000003 ADD #IMLEN,R5 ;ELSE POINT TO NEXT ENTRY
43 004143 104050 001232 MOV R5,STARIT ;MAKE IT NEW STARTING ADDRESS
44 004145 117400 001454 FKIP10: DEC SECCNT ;DECREMENT
45 004147 054062 BNE MORE ;NO - DO NEXT SECTOR
46 004150 000000 RETURN

```

```

1
2
3
4
5
6
7
8
9
10 G04151
11 003047 104200 000047 001153
12 003052 104300 001127 001454 DXTRK:
13 003055 104302 001223
14 003057 104205 007275
15 003061 102200 000600 001217
16 003064 013074
17 003065 102200 000010 001217
18 003070 013073
19 003071 023307
20 003072 003074
21 003073 023341 TKIP1:
22 003074 104203 005567 TKIP2:
23 003076 100253
24 003077 104303 001113
25 003101 100253
26 003102 102200 002002 001217
27 003105 053146
28 003106 104302 001223
29 003110 104121
30 003111 106010 001110
31 003113 053146
32 003114 104621 000001
33 003116 103201 170000
34 003120 106010 001111
35 003122 053146
36 003123 117400 001501
37 003125 117400 001237
38 003127 053133
39 003130 101200 000002 001217
40 003133 104303 001114 TKIP7:
41 003135 103203 170000
42 003137 101203 110000
43 003141 100253
44 003142 105200 000002 001223
45 003145 003156
46 003146 104303 001114 TKIP3:
47 003150 103203 170000
48 003152 104301 001462
49 003154 101013
50 003155 100253
51 003156
52 003163 102200 000600 001217 TKIP4:
53 003166 013202 TKIP10:
54 003167
55 003174 104303 001501
56 003176 053202
57 003177 115400 001256

.SBTTL DBN/XBN TRACK FORMAT OVERLAY (G7)

SET UP TRACK FORMAT

DMOVLY G7, START
MOV #G7, CUROVL ;GET OVERLAY INDICATOR
MOV SECTRK, SECCNT ;MOVE SECTOR COUNT INTO R3
MOV BADPBN, R2 ;MOVE PTR TO BAD LIST INTO R2
MOV #IMAGE, R5 ;POINT TO FORMAT TRACK IMAGE
BIT #MANU+DLL, FLAG ;SEE IF FCT AVAILABLE
BEQ TKIP2 ;NO - SKIP PBN COMPUTATION
BIT #DBN, FLAG ;DO DBN AREA ??
BEQ TKIP1 ;NO - DO XBN AREA
CALL DPBN ;COMPUTE PBN FOR STARTING DBN
BR TKIP2 ;SKIP XBN COMPUTATION
CALL XPBN ;COMPUTE PBN FOR STARTING XBN
MOV #GDBLK, R3 ;POINT R3 AT GOOD DATA BLOCK
MOV R3, (R5)+ ;AND STORE PTR IN IMAGE BLOCK
MOV CURBN, R3 ;GET LOW ORDER BLOCK NUMBER
MOV R3, (R5)+ ;AND STORE IN IMAGE BLOCK
BIT #BSTGS+FCTEMT, FLAG ;IS FCT AVAILABLE AND NON-EMPTY ?
BNE TKIP3 ;NO - CONSIDER GOOD
MOV BADPBN, R2 ;GET POINTER TO PBN
MOV (R2), R1 ;GET LOW ORDER PBN
CMP R1, CURPBN ;ARE THEY EQUAL ?
BNE TKIP3 ;NO - SKIP REST OF COMPARE
MOV 1(R2), R1 ;GET HIGH ORDER BAD
BIC #HD.CLR, R1 ;CLEAR HEADER FOR COMPARE
CMP R1, CURPBN+1 ;EQUAL ?
BNE TKIP3 ;NO - MARK AS GOOD
DEC COUNT ;DECREMENT IT
DEC FCT ;DEC IT
BNE TKIP7 ;IF NOT EMPTY THEN CONTINUE
BIS #FCTEMT, FLAG ;SET FCT EMPTY FLAG
MOV CURBN+1, R3 ;HI ORDER BLOCK NUM AND HDR CODE
BIC #HD.CLR, R3 ;CLEAR THE HEADER CODE
BIS #HD.BAD, R3 ;SET TO BAD HEADER CODE
MOV R3, (R5)+ ;AND STORE IN IMAGE BLOCK
ADD #2, BADPBN ;MOVE PTR TO NEXT BAD BLOCK
BR TKIP4 ;SKIP GOOD MARKING
MOV CURBN+1, R3 ;HI ORDER BLOCK NUM AND HDR CODE
BIC #HD.CLR, R3 ;CLEAR HEADER CODE
MOV HD.CUR, R1 ;GET CURRENT HEADER CODE (XBN OR DBN)
BIS R1, R3 ;SET TO GOOD HEADER CODE
MOV R3, (R5)+ ;AND STORE IN IMAGE BLOCK
DUBINC CURBN ;INCREMENT IT
BIT #MANU+DLL, FLAG ;IS FCT AVAILABLE ?
BEQ SKIP5 ;NO - SKIP PBN INCREMENT
DUBINC CURPBN ;INCREMENT IT
MOV COUNT, R3 ;GET PBN COUNT
BNE SKIP5 ;IF NOT DONE SKIP
INC FCTCNT ;GET NEXT BLOCK NUMBER

```

Line	Address	Offset	Label	Instruction	Comment	
58	003201	02337C		CALL	:ELSE PAGE IN NEW FCT BLOCK	
59	003202	117400	001454	DEC	:DECREMENT IT	
60	003204	013206		BEQ	:YUP - RETURN	
61	003205	003074		BR	:NO - DO NEXT SECTOR	
62	003206		DONE1:			
63	003206	107205	000003	SUB	:POINT TO FLAG OF LAST ENTRY	
64	003210	104153		MOV	:GET FLAG WORD	
65	003211	101203	040000	BIS	:SET RECIRCULATION FLAG	
66	003213	100153		MOV	:STORE IT BACK	
67	003214	115000	001463	TST	:IS GROUP ZERO ???	
68	003216	013230		BEQ	:YES - NO OFFSET	
69	003217	104203	001052	MOV	:POINT TO CHARACTERISTICS	
70	003221	104632	000011	MOV	:GET GROUP OFFSET	
71	003223	110702		SWAB	:GET INTO LOWBYTE	
72	003224	103202	177400	BIC	:CLEAR HIGH GARBAGE	
73	003226	115002		TST	:ANY OFFSET ?	
74	003227	053236		BNE	:YUP - HANDLE IT	
75	003230	104200	007275	MOV	:ELSE START AT BEGINNING	
76	003233	104304	001332	MOV	:GET START	
77	003235	003302		BR	:SKIP OFFSET COMPUTATION	
78	003236	104020	000736	MOV	:STORE IT	
79	003240	114000	000737	CLR	:FOR STORE	
80	003242	104300	001463	MOV	:GET CURRENT GROUP	
81	003245	114000	000732	CLR	:CLEAR HIGH WORD	
82	003247	104203	000736	MOV	:FOR MUL	
83	003251	104204	000731	MOV	:DITTO	
84	003253	021542		CALL	:MULTIPLY TO GET OFFSET FOR THIS GROUP	
85	003254	106300	001127	000731	TKIP8: CMP	:IS TOTAL OFFSET MORE THAN NUMBER OF SECTORS ?
86	003257	033264		BPL	:NO - ALL IS FINE	
87	003260	107300	001127	000731	SUB	:YES - SUBTRACT TILL IT IS
88	003263	003254		BR	:CHECK AGAIN	
89	003264	104200	000003	000736	TKIP9: MOV	:GET LENGTH OF IMAGE BLOCK
90	003267	114000	000737	CLR	:FOR STORE	
91	003271	104203	000736	MOV	:FOR MULT	
92	003273	021542		CALL	:GET LENGTH TO OFFSET	
93	003274	104143		MOV	:GET RESULT	
94	003275	104304	001231	MOV	:GET ADDRESS OF END OF IMAGE	
95	003277	107034		SUB	:SUBTRACT TO GET STARTING LOCATION	
96	003300	104040	001332	MOV	:STORE IT	
97	003302	104143		TKIP6: MOV	:GET BUFF POINTER	
98	003303	101203	100000	BIS	:SIGNAL AS LAST	
99	003305	100143		MOV	:STORE IT BACK	
100	003306	000000		RETURN		

```

1
2
3
4
5
6
7
8 003307 104300 001137 000736 DPBN:  MOV    XBNSEC,TEMP           ;GET NUMBER OF SECTORS IN XBN AREA
9 003312 104300 001140 000737      MOV    XBNSEC+1,TEMP+1       ;GET HI ORDER
10 003315 104204 000736      MOV    #TEMP,R4           ;POINT R4 AT TEMP AREA
11 003317 104203 001133      MOV    #LBNLBN,R3          ;POINT AT NUM OF LBN'S IN LBN AREA
12 003321 021506      CALL    DADD                ;ADD
13 003322 104203 001135      MOV    #RBNLBN,R3          ;POINT TO NUM OF RBN'S IN LBN AREA
14 003324 021506      CALL    DADD                ;ADD TO GET SECTORS IN LBN + XBN AREA
15 003325 104203 001113      MOV    #CURBN,R3          ;POINT TO CURRENT BLOCK NUMBER (DBN)
16 003327 021506      CALL    DADD                ;GET RELATIVE PBN
17 003330 104641 000001      MOV    1(R4),R1          ;GET HIGH ORDER
18 003332 107301 001340      SUB     ST,DBN,R1           ;SUBTRACT HIGH ORDER STARTING DBN
19 003334 104140 001110      MOV    (R4),CURPBN        ;GET LO ORDER PBN
20 003336 104010 001111      MOV    R1,CURPBN+1       ;STORE HIGH ORDER
21 003340 000000      RETURN
22
23
24
25
26
27
28
29
30
31 003341 104300 001133 000736 XPBN:  MOV    LBNLBN,TEMP           ;GET NUMBER OF LBN'S IN LBN AREA
32 003344 104300 001134 000737      MOV    LBNLBN+1,TEMP+1       ;GET HIGH ORDER
33 003347 104204 000736      MOV    #TEMP,R4           ;POINT R4 TO TEMP AREA
34 003351 104203 001135      MOV    #RBNLBN,R3          ;POINT R3 AT RBN'S IN LBN AREA
35 003353 021506      CALL    DADD                ;ADD TO GET TOTAL SECTORS IN LBN AREA
36 003354 104203 001113      MOV    #CURBN,R3          ;POINT R3 AT CURRENT BLOCK NUMBER
37 003356 021506      CALL    DADD                ;ADD TO GET RELATIVE PBN
38 003357 104641 000001      MOV    1(R4),R1          ;GET HIGH ORDER
39 003361 107301 001337      SUB     ST,XBN,R1           ;SUBTRACT HIGH ORDER STARTING XBN
40 003363 104140 001110      MOV    (R4),CURPBN        ;GET LO ORDER OF PBN
41 003365 104010 001111      MOV    R1,CURPBN+1       ;SAVE HIGH ORDER
42 003367 000000      RETURN
  
```

```
1  
2  
3  
4 003370  
5 003371 104201 000033  
6 003373 022444  
7 003374 104200 000200 001501  
8 003377 104200 005152 001223  
9 003402  
10 003403 000000
```

;;PAGE IN NEW FCT BLOCK
DXFCP1: PUSH R0
MOV #G2,R1
CALL PAGE
MOV #128,COUNT
MOV #PBNBUF,BADPBN
POP R0
RETURN

:DLL OVERLAY
:EXECUTE OVERLAY
:FOR INIT
:FOR POINTER RESET
:RESTORE R0
:RETURN

1					.SBTTL LBN FORMATTING OVERLAY (F2)	
2						
3					LBN FORMATTING	
4						
5	003404				DMOVLY F2,START	
6						
7						
8	003047	104200	000003	001153	LFORM: MOV #F2,CUROVL	:OVERLAY #2
9	003052	102200	000001	001217	BIT #FCTAVL,FLAG	:FCT AVAILAABLE ?
10	003055	013061			BEQ XSKIP1	:NO - SKIP SET UP
11	003056	104201	000036		MOV #G3,R1	:OVERLAY TO GET RIGHT FCT BLOCK
12	003060	022444			CALL PAGE	:EXECUTE IT
13	003061	104207	001052		XSKIP1: MOV #SCR,R0	:POINT TO CHARACTERISTICS BLOCK
14	003063	104300	001143	001125	MOV LBNCYL,CYLNUM	:GET LO ORDER CYLINDER COUNT
15	003066	104300	001143	001460	MOV LBNCYL,CNTCYL	:MAKE LO ORDER COUNT
16	003071	104300	001144	001126	MOV LBNCYL+1,CYLNUM+1	:GET HIGH ORDER
17	003074	104300	001144	001461	MOV LBNCYL+1,CNTCYL+1	:STORE IT
18	003077	103200	170000	001461	BIC #HD.CLR,CNTCYL+1	:CLEAR STARTING CYLINDER BITS
19	003102	104204	001125		MOV #CYLNUM,R4	:SUBTRACT TO GET CYLINDER NUMBER
20	003104	104203	001466		MOV #ONE,R3	:1 - BECAUSE START AT 0
21	003106	021524			CALL DSUB	:DO SUBTRACT
22	003107	104300	001475	001235	MOV TOTRCT,RCTTOT	:GET TOTAL RCT LBN'S
23	003112	104201	000052		MOV #G8,R1	:POINT TO OVERLAY
24	003114	022444			CALL PAGE	:COMPUTE VARIOUS CONSTANTS
25	003115	104207	001052		XSLEEK: MOV #SCR,R0	:POINT TO CHARACTERISTICS
26	003117	104673	000002		MOV GRPCYL(R0),R3	:GET GRUPS/CYLINDER
27	003121	103203	177400		BIC #HIBYTE,R3	:CLEAR OUT GARBAGE
28	003123	104030	001464		MOV R3,GRPCNT	:USE AS COUNTER
29	003125	104030	001463		MOV R3,CURGRP	:GROUP NUMBER
30	003127	117400	001463		DEC CURGRP	:DECREMENT TO GET ACTUAL NUMBER
31	003131	104300	001 25	001076	XSLEK2: MOV CYLNUM,ISEEK+1	:GET CURRENT CYLINDER NUMBER
32	003134	104300	001126	001077	MOV CYLNUM+1,ISEEK+2	:GET HIGH ORDER
33	003137	104300	001463	001100	MOV CURGRP,ISEEK+3	:LOAD GROUP NUMBER
34	003142	022251			CALL SEEK	:DO THE SEEK
35	003143	115001			TST R1	:ANY ERROR ?
36	003144	073435			BMI SEEKER	:YUP - CUT OUT
37	003145	104207	001052		MOV #SCR,R0	:POINT TO CHARACTERISTICS
38	003147	104673	000003		MOV TRKGRP(R0),R3	:GET TRACKS/GROUP
39	003151	103203	177400		BIC #HIBYTE,R3	:CLEAR OUT GARBAGE
40	003153	104030	001465		MOV R3,TRKCNT	:MAKE COUNTER
41	003155	117403			DEC R3	:WANT LAST TRACK NUMBER
42	003156	104030	001112		MOV R3,CURTRK	:MAKE CURRENT TRACK 0
43	003160	104201	000025		XSKIP3: MOV #F8,R1	:TRACK SET UP OVERLAY
44	003162	022444			CALL PAGE	:SET UP TRACK FORMAT
45	003163	104304	001334		XSKIP2: MOV DPREA,R4	:GET DATA PREAMBLE LENGTH
46	003165	104303	001333		MOV HPREA,R3	:GET HEADER PREAMBLE LENGTH
47	003167	104307	001332		MOV IMSTAR,R0	:POINT TO TRACK IMAGE START POINT
48	003171	104301	001112		MOV CURTRK,R1	:TRACK TO FORMAT
49	003173	104302	000740		MOV UNIT,R2	:SDI INTERCONNECT
50	003175	104205	007275		MOV #IMAGE,R5	:RECIRCULATION ADDRESS
51	003177	060001			XFC FORMAT	:DO FORMAT
52	003200	115001			TST R1	:ANY ERROR ?
53	003201	013214			BEQ LSKIP4	:NO - DO CHECK PASS
54	003202	115400	000717		INC UN.ERI	:INCREMENT IT
55	003204	106300	001502	000717	CMP RETRY,UN.ERI	:DONE ALL RETRIES ?
56	003207	073431			BMI FORERR	:YUP - ERROR
57	003210	022371			CALL INITPT	:REINIT

58	003211	022243			CALL	CLEAR	:DRIVE CLEAR
59	003212	022251			CALL	SEEK	:RE-SEEK AND GROUP SELECT
60	003213	003163			BR	XSKIP2	:NOPE - RETRY
61	003214	114000	000717		LSKIP4: CLR	UN.ERI	:FOR STORE
62	003216	104301	001225		MOV	EMAX,R1	:GET MAX REVECTORS
63	003220	107301	001254		SUB	REVCNT,R1	:SUBTRACT CURRENT ENTRIES
64	003222	106301	001127		CMP	SECTRK,R1	:ENOUGH LEFT FOR WHOLE TRACK ??
65	003224	073235			BMI	XSKIP4	:YES - CONTINUE
66	003225	104201	000011		MOV	#F4,R1	:SIGNAL RCT UPDATE OVERLAY
67	003227	022444			CALL	PAGE	:PAGE IT IN
68	003230	114000	001254		CLR	REVCNT	:FOR STORE
69	003232	104300	001224	001252	MOV	ERRBUF,ERPNT	:FOR RESET
70	003235	023442			XSKIP4: CALL	LCHEC	:DO CHECK PASS
71	003236	102200	020000	001217	BIT	#INIRCT,FLAG	:TIME TO INIT RCT ?
72	003241	013260			BEQ	XSKIP5	:NOPE
73	003242	103200	020000	001217	BIC	#INIRCT,FLAG	:REST FLAG
74	003245	102200	002000	001217	BIT	#BSTGS,FLAG	:DOING BEST GUESS ?
75	003250	013255			BEQ	XSKIP6	:NO - GO ALL THE WAY
76	003251	104201	000022		MOV	#F7,R1	:RCT INIT OVERLAY
77	003253	022444			CALL	PAGE	:EXECUTE IT
78	003254	003260			BR	XSKIP5	:SKIP OTHER
79	003255	104201	000014		XSKIP6: MOV	#F5,R1	:DO FCT->RCT AND INIT
80	003257	022444			CALL	PAGE	:EXECUTE IT
81	003260	117400	001112		XSKIP5: DEC	CURTRK	:DECREMENT IT
82	003262	104204	001117		MOV	#HOLDBN,R4	:GET STARTING BLOCK NUMBER
83	003264	104207	001052		MOV	#SCR,R0	:POINT TO CHARACTERISTICS
84	003266	104673	000011		MOV	LBNTRK(R0),R3	:GET LBN/TRACK
85	003270	103203	177400		BIC	#HIBYTE,R3	:CLEAR HIGH BYTE
86	003272	104030	000731		MOV	R3,DDUMMY	:STORE IT
87	003274	114000	000732		CLR	DDUMMY+1	:FOR STORE
88	003276	104203	000731		MOV	#DDUMMY,R3	:LBN/TRACK
89	003300	021524			CALL	DSUB	:GET STARTING LBN FOR NEW TRACK
90	003301	104300	001117	001113	MOV	HOLDBN,CURBN	:GET LOW ORDER
91	003304	104300	001120	001114	MOV	HOLDBN+1,CURBN+1	:GET HIGH ORDER
92	003307	104204	001121		MOV	#HOLRBN,R4	:GET STARTING RBN NUMBER
93	003311	104673	000004		MOV	RBNTRK(R0),R3	:GET RBN/TRACK
94	003313	103203	177600		BIC	#HIBYTE,R3	:CLEAR OUT GARBAGE
95	003315	104030	000731		MOV	R3,DDUMMY	:STORE IT
96	003317	114000	000732		CLR	DDUMMY+1	:FOR STORE
97	003321	104203	000731		MOV	#DDUMMY,R3	:RBN'S/TRACK
98	003323	021524			CALL	DSUB	:GET STARTING RBN FOR NEW TRACK
99	003324	104300	001121	001106	MOV	HOLRBN,CURRBN	:GET LOW ORDER
100	003327	104300	001122	001107	MOV	HOLRBN+1,CURRBN+1	:GET HI ORDER
101	003332	117400	001465		DEC	TRKCN	:DECREMENT IT
102	003334	053160			BNE	XSKIP3	:NO - DO NEXT TRACK
103	003335	117400	001463		DEC	CURGRP	:DECREMENT GROUP NUMBER
104	003337	117400	001464		DEC	GRPCNT	:DECREMENT IT
105	003341	053131			BNE	XSLEK2	:NO - DO NEXT GROUP
106	003342	104204	001460		MOV	#CNTCYL,R4	:GET READY TO DEC CYLINDER CNT
107	003344	104203	001466		MOV	#ONE,R3	:CONSTANT WORD OF 1
108	003346	021524			CALL	DSUB	:DECREMENT IT
109	003347	060022			XFC	UPDATE	:UPDATE PROGRESS INDICATOR
110	003350	104207	001263		MOV	#DMBUF,R0	:POINT TO MAINT BUFFER
111	003352	104303	001316		MOV	CYLMSG,R3	:GET CYLINDER MSG FLAG
112	003354	100173			MOV	R3,(R0)	:STORE IT
113	003355	104303	001317		MOV	CYLMSG+1,R3	: 'C' CODE
114	003357	100673	000001		MOV	R3,1(R0)	:STORE IT

115	003361	104303	001125		MOV	CYLNUM,R3	:GET LOW ORDER CYLINDER NUMBER
120	003363	100673	000003		MOV	R3,R3(R0)	:PUT IN MESSAGE
121	003365	104303	001126		MOV	CYLNUM+1,R3	:GET HIGH ORDER
122	003367	100673	000002		MOV	R3,R3(R0)	:STORE IT
123	003371	022536			CALL	SNDMNT	:LET HOST KNOW WE'RE ALIVE
124	003372	104204	001125		LSKIP5: MOV	#CYLNUM,R4	:GET CURRENT CYLINDER NUMBER
125	003374	104203	001466		MOV	#ONE,R3	:FOR DECREMENT
126	003376	021524			CALL	DSUB	:DECREMENT FOR NEW CYLINDER NUM
127	003377	104304	001460		MOV	CNTCYL,R4	:LOW ORDER ZERO ?
128	003401	053115			BNE	XSLEEK	:NO - CONTINUE
129	003402	104304	001461		MOV	CNTCYL+1,R4	:HIGH ORDER ZERO ?
130	003404	053115			BNE	XSLEEK	:NO - CONTINUE
131	003405	104303	001254		MOV	REVCNT,R3	:ANY LEFTOVER REVECTORS ?
132	003407	013416			BEQ	XDONE	:NOPE
133	003410	101200	040000	001217	BIS	#FINI,FLAG	:SIGNAL NOT TO SEEK
134	003413	104201	000011		MOV	#F4,R1	:SIGNAL RCT UPDATE OVERLAY
135	003415	022444			CALL	PAGE	:UPDATE IT
136	003416	104303	001312		XDONE: MOV	LBNBAD,R3	:GET BAD BLOCKS FROM FCT
137	003420	104304	001453		MOV	ERRCNT,R4	:GET CHECK PASS BAD
138	003422	105043			ADD	R4,R3	:ADD TO GET TOTAL
139	003423	104030	001312		MOV	R3,LBNBAD	:STORE BACK
140	003425	104201	000041		MOV	#G4,R1	:RCT CLEANUP
141	003427	022412			CALL	NEXT	:GET NEXT OVERLAY
142	003430	000000			RETURN		
143	003431	104012			FORERR: MOV	R1,R2	:XFC ERROR CODE
144	003432	104201	000014		MOV	#12,R1	:SIGNAL LBN FORMAT ERROR
145	003434	003441			BR	LFERR	
146	003435	104302	001125		SEEKER: MOV	CYLNUM,R2	:CYLINDER FAILED ON
147	003437	104201	000012		MOV	#10,R1	:SEEK ERROR
148	003441	022552			LFERR: CALL	ERRMNT	:ERROR RETURN

1									
2									
3									
4									
5									
6	003442	114000	001226		LCHEC:	CLR	ERR		:FOR ERROR COUNT RESET
7	003444	114000	001221			CLR	ERFLAG		:CLEAR REFORMAT FLAG
8	003446	102200	002000	001217		BIT	#BSTGS,FLAG		:BEST GUESS ?
9	003451	053464				BNE	LSKIP		:YES - DO EXTENSIVE READ
10	003452	104200	000001	001455		MOV	#1,N		:SET UP FOR STORE
11	003455	104200	000005	001456		MOV	#5,N1		:SET UP
12	003460	104300	001456	001457		MOV	N1,NN1		:SAVE FOR LATER RESET
13	003463	003475				BR	LSKIP2		:SKIP EXTENSIVE READ SETUP
14	003464	104200	000003	001455	LSKIP:	MOV	#3,N		:EXTENSIVE REGULAR READ
15	003467	104200	000024	001456		MOV	#20,N1		:EXTENSIVE ERROR READS
16	003472	104300	001456	001457		MOV	N1,NN1		:SAVE FOR LATER RESET
17	003475	024361			LSKIP2:	CALL	LFIXIT		:EXECUTE IT
18	003476	104302	000740		LSKIP1:	MOV	UNIT,R2		:GET SDI INTERCONNECT
19	003500	060012				XFC	SIP		:WAIT FOR PULSE
20	003501	104300	001127	001454		MOV	SECTRK,SECCNT		:LOAD SECTORS/TRACK
21	003504	104205	006621			MOV	#CMDBUF,R5		:POINT TO COMMAND BUFFER
22	003506	104207	000721		LAGAIN:	MOV	#RDBLK,R0		:POINT TO READ COMMAND BLOCK
23	003510	104653	000000			MOV	RB.BUF(R5),R3		:BUFFER PTR ZERO ?
24	003512	013570				BEQ	LNOERR		:SECTOR BAD - SKIP CHECKS
25	003513	100673	000001			MOV	R3,RW.BUF(R0)		:ELSE STORE IN COMMAND BLOCK
26	003515	104653	000001			MOV	RB.LOW(R5),R3		:LOAD LOW ORDER SECTOR NUMBER
27	003517	100673	000002			MOV	R3,RW.LOW(R0)		:STORE IN COMMAND BLOCK
28	003521	104653	000002			MOV	RB.HI(R5),R3		:LOAD HIGH ORDER BLOCK NUMBER
29	003523	100673	000003			MOV	R3,RW.HI(R0)		:STORE IN COMMAND BLOCK
30	003525	104653	000003			MOV	RB.CMD(R5),R3		:GET READ COMMAND AND TRACK NUMBER
31	003527	100673	000004			MOV	R3,RW.CMD(R0)		:STORE IN COMMAND BLOCK
32	003531	104203	000726			MOV	#HSLIM-1,R3		:POINTER TO DUMMY SDI BLOCK
33	003533	100673	000005			MOV	R3,RW.DUM(R0)		:STORE IN READ BLOCK
34	003535	104207	100721		RFAD3:	MOV	#<BIT15!RDBLK>,R0		:MAKE SURE POINTING AT BLOCK
35	003537	104203	100000			MOV	#RDCMD,R3		:RESET STATUS POINTER
36	003541	100673	000000			MOV	R3,RW.STAT(R0)		:STORE IT BACK
37	003543	060002				XFC	READ		:READ 1 SECTOR
38	003544	115001				TST	R1		:ANY ERRORS ?
39	003545	053553				BNE	LERR		:YES - UH OH
40	003546	104207	001443			MOV	#NUM,R0		:POINT TO COMPARE BLOCK
41	003550	060006				XFC	CMPCAT		:DO DATA COMPARE
42	003551	115001				TST	R1		:ANY ERROR IN COMPARE ?
43	003552	013570				BEQ	LNOERR		:NOPE - CONTINUE LOOP
44	003553				LERR:				
45	003553	104653	000004			MOV	RB.IM(R5),R3		:GET IMAGE POINTER
46	003555	104134				MOV	(R3),R4		:GET BUFFER POINTER WORD
47	003556	102204	020000			BIT	#BD,R4		:ALREADY BEEN HERE ??
48	003560	053570				BNE	LNOERR		:YUP - DON'T COUNT AGAIN
49	003561	101204	020000			BIS	#BD,R4		:MARK AS BAD
50	003563	100134				MOV	R4,(R3)		:STORE BACK
51	003564	115400	001473			INC	RTYCNT		:INC IT
52	003566	115400	001226			INC	ERR		:INCREMENT ERROR COUNT
53	003570	105205	000005		LNOERR:	ADD	#RDLEN,R5		:POINT TO NEXT READ CMD BLOCK
54	003572	114000	001504			CLR	TMPTRY		:FOR RESET
55	003574	104300	001503	001505		MOV	RECOV,RECTMP		:GET RECOVERY LEVELS
56	003577	103200	000002	001220		BIC	#RTYDN,FLAG1		:CLEAR RETRY DONE FLAG
57	003602	117400	001454			DEC	SECCNT		:DECREMENT COUNTER

58	003604	053506				BNE	LAGAIN		:NO - DO NEXT SECTOR
59	003605	117400	001455			DEC	N		:DECREMENT COUNTER
60	003607	073476				BNE	LSKIP1		:NO - REPEAT TRACK READ AND COMPARE
61	003610	115000	001226			TST	ERR		:ANY ERRORS ON FIRST PASS ?
62	003612	014360				BEQ	LDONE		:NO - ALL DONE CHECK PASS
63	003613	104204	007275			MOV	#IMAGE,R4		:POINT TO IMAGE BUFFER
64	003615	104143			HERE:	MOV	(R4),R3		:GET BUFFER POINTER WORD
65	003616	102203	020000			BIT	#BD,R3		:IS IT BAD ?
66	003620	014230				BEQ	LSKIP7		:NO - SKIP IT
67	003621	104643	000002			MOV	FT.HI(R4),R3		:GET HI ORDER BLOCK NUM AND HDR CODE
68	003623	103203	007777			BIC	#LO,R3		:CLEAR LOW ORDER
69	003625	106203	000000			CMP	#HD.LBN,R3		:IS IT A 'GOOD' LBN
70	003627	013633				BEQ	LSKIP8		:YES - DO IT
71	003630	106203	060000			CMP	#HD.RBN,R3		:IS IT AN RBN ???
72	003632	054230				BNE	LSKIP7		:NOPE - ALREADY PROCESSED SKIP IT
73	003633	104302	000740		LSKIP8:	MOV	UNIT,R2		:SDI INTERCONNECT
74	003635	060012				XFC	SIP		:WAIT FOR PULSE
75	003636	104207	000721			MOV	#RDBLK,R0		:PREPARE FOR READ SECTORS
76	003640	104203	000726			MOV	#HSLIM-1,R3		:POINTER TO DUMMY SDI BLOCK
77	003642	100673	000005			MOV	R3,RW.DUM(R0)		:STORE IN COMMAND BLOCK
78	003644	104643	000001			MOV	1(R4),R3		:LO ORDER BLOCK NUMBER
79	003646	100673	000002			MOV	R3,RW.LOW(R0)		:STORE IN READ CMD BLOCK
80	003650	104643	000002			MOV	2(R4),R3		:HI ORDER BLOCK NUM AND CODE
81	003652	100673	000003			MOV	R3,RW.HI(R0)		:STORE IN READ CMD BLOCK
82	003654	104203	004535			MOV	#RDBUF,R3		:LOAD ADDRESS OF DATA BUFFER
83	003656	100673	000001			MOV	R3,RW.BUF(R0)		:STORE IN COMMAND BUFFER
84	003660	104203	013400			MOV	#RWCMD,R3		:LOAD SDI READ COMMAND
85	003662	101303	001112			BIS	CURTRK,R3		:SET CURRENT HEAD ADDRESS IN COMMAND
86	003664	100673	000004			MOV	R3,RW.CMD(R0)		:STORE BACK
87	003666	104207	100721		HEAD4:	MOV	#<BIT15!RDBLK>,R0		:MAKE SURE POINTING AT BLOCK
88	003670	104203	100000			MOV	#RDCMD,R3		:MARK AS ONLY REQUEST
89	003672	100173				MOV	R3,(R0)		:STORE IN CMD BLOCK
90	003673	104302	000740			MOV	UNIT,R2		:SDI INTERCONNECT
91	003675	060002				XFC	READ		:READ 1 SECTOR
92	003676	115001				TST	R1		:ANY ERROR IN READ ?
93	003677	053715				BNE	LER3		:YES - CONSIDER BAD
94	003700	104173				MOV	(R0),R3		:LOAD ECC ERROR INDICATOR FOR TEST
95	003701	102203	010000			BIT	#ECCF,R3		:TEST FOR ECC ERROR
96	003703	013707				BEQ	LSKIP6		:NO - CHECK EDC
97	003704	023027				CALL	ECCCK		:FIND OUT HOW MANY SYMBOLS IN ERROR
98	003705	115001				TST	R1		:TOO MANY ?
99	003706	073720				BMI	LER1		:YUP - CONSIDER BAD
100	003707				LSKIP6:				
101	003707	104207	001443			MOV	#NUM,R0		:POINT TO COMPARE BLOCK
102	003711	060006				XFC	CMPCAT		:DO DATA COMPARE
103	003712	115001				TST	R1		:ANY ERROR IN COMPARE ?
104	003713	014220				BEQ	LOK		:NO ERROR
105	003714	003720				BR	LER1		:SKIP BAD HEADER FLAGGING
106	003715	101200	000040	001220	LER3:	BIS	#BDHD,FLAG1		:FLAG AS BAD HEADER
107	003720	102200	000100	001217	LER1:	BIT	#REVECT,FLAG		:IN RCT ???
108	003723	053727				BNE	LER2		:NOPE - SKIP NEXT STUFF
109	003724	115400	001313			INC	RCTBAD		:INC BAD COUNTER
110	003726	004136				BR	BDIRCT		:THEN MARK BAD
111	003727	104643	000002		LER2:	MOV	FT.HI(R4),R3		:GET HI ORDER BLOCK NUM AND HDR CODE
112	003731	103203	007777			BIC	#LO,R3		:CLEAR LOW ORDER
113	003733	106203	060000			CMP	#HD.RBN,R3		:IS IT A BAD RBN ?
114	003735	014034				BEQ	BRBN		:YUP - GO HANDLE IT

115	003736	104643	000002		MOV	FT.HI(R4),R3	:RELOAD HEADER
116	003740	103203	170000		BIC	#HD.CLR,R3	:CLEAR THE HEADER
117	003742	102200	001000	001217	BIT	#PRIM,FLAG	:ANY PRIMARY YET ?
118	003745	054153			BNE	SND	:YUP - THIS ONE SECONDARY
119	003746	104030	000732		MOV	R3,DDUMMY+1	:STORE HIGH ORDER FOR RBN COMPUTATION
120	003750	101200	001000	001217	BIS	#PRIM,FLAG	:SET PRIMARY FLAG
121	003753	101200	000004	001220	BIS	#RPRIM,FLAG1	:SET GOOD RBN EDC NEEDED
122	003756	101203	050000		BIS	#HD.PRIV,R3	:MARK AS PRIMARY
123	003760	100643	000002		MOV	R3,FT.HI(R4)	:STORE BACK IN IMAGE
124	003762	104640	000001	000731	MOV	FT.LOW(R4),DDUMMY	:STORE LOW ORDER FOR RBN COMPUTATION
125	003765	104042			MOV	R4,R2	:SAVE IMAGE POINTER
126	003766	104207	001052		MOV	#SCR,R0	:MAKE SURE POINT TO CHAR BLOCK
127	003770	104204	000731		MOV	#DDUMMY,R4	:POINT TO BLOCK NUMBER
128	003772	022757			CALL	PRIMR3	:GET PRIMARY RBN NUMBER
129	003773	104307	001151		MOV	REVRBN,R0	:GET NUMBER OF REVECTORED RBN
130	003775	104301	001152		MOV	REVRBN+1,R1	:GET HIGH ORDER
131	003777	101201	060000		BIS	#HD.RBN,R1	:SET IN RBN HEADER CODE
132	004001	104205	006204		MOV	#PRMBUF,R5	:USE RDBUF TO HOLD 128 COPIES OF RBN
133	004003	104203	000200		MOV	#RBNRPT,R3	:COUNT OF REPLICATED RBN'S
134	004005	100257		RPT1:	MOV	R0,(R5)+	:STORE A COPY
135	004006	100251			MOV	R1,(R5)+	:AND HIGH ORDER
136	004007	117403			DEC	R3	:DECREMENT COUNTER - DONE ?
137	004010	054005			BNE	RPT1	:NO - STORE ANOTHER COPY
138	004011	104024			MOV	R2,R4	:RESTORE IMAGE POINTER
139	004012	104205	006204		MOV	#PRMBUF,R5	:POINT TO BEGINNING OF BUFFER
140	004014	104642	000000		MOV	FT.BUF(R4),R2	:GET BUFFER POINTER
141	004016	103202	007777		BIC	#BUFMSK,R2	:CLEAR ONLY BUFFER POINTER
142	004020	101052			BIS	R5,R2	:OR IN NEW BUFFER POINTER
143	004021	100642	000000		MOV	R2,FT.BUF(R4)	:STORE IT BACK
144	004023	104202	006204		MOV	#PRMBUF,R2	:FOR EDC COMPUTATION
145	004025	022627			CALL	CEDC	:COMPUTE IT
146	004026	100623	000400		MOV	R3,RW.EDC(R2)	:STORE IT
147	004030	103200	000040	001220	BIC	#BDHD,FLAG1	:WAN'T TO STAY PRIMARY
148	004033	004165			BR	LSND	:BRANCH AROUND SECONDARY
149	004034	117400	001453	BRBN:	DEC	ERRCNT	:DEC ERR CNT SO PRIMARY STATS WILL BE RIGHT
150	004036	102200	001000	001217	BIT	#PRIM,FLAG	:IS THERE A PRIMARY ON THIS TRACK ?
151	004041	014130			BEQ	7\$	:NO - SKIP HEADER RESET
152	004042	104203	007275		MOV	#IMAGE,R3	:POINT TO FORMAT TABLE
153	004044	104632	000002	5\$:	MOV	FT.HI(R3),R2	:GET HEADER WORD
154	004046	103202	007777		BIC	#LO,R2	:CLEAR ALL BUT HEADER
155	004050	106202	050000		CMP	#HD.PRIV,R2	:IS IT THE PRIMARY ?
156	004052	014056			BEQ	6\$	:YES - DONE
157	004053	105203	000003		ADD	#IMLEN,R3	:NO - POINT TO NEXT ENTRY
158	004055	004044			BR	5\$	:CHECK NEXT ENTRY
159	004056	104632	000002	6\$:	MOV	FT.HI(R3),R2	:RESET TO HI ORDER
160	004060	103202	170000		BIC	#HD.CLR,R2	:CLEAR HEADER
161	004062	101202	030000		BIS	#HD.REV,R2	:MARK AS SECONDARY
162	004064	100632	000002		MOV	R2,FT.HI(R3)	:STORE BACK
163	004066	104202	005567		MOV	#GDBLK,R2	:POINT TO GOOD BLOCK
164	004070	104635	000000		MOV	FT.BUF(R3),R5	:GET BUFFER POINTER AND FLAGS
165	004072	103200	007777	000014	BIC	#BUFMSK,R5	:CLEAR ONLY BUFFER POINTER
166	004075	101025			BIS	R2,R5	:OR IN NEW BUFFER POINTER
167	004076	100635	000000		MOV	R5,FT.BUF(R3)	:MOVE IN BUFFER POINTER AND FLAGS
168	004100	104305	001252		MOV	ERPNT,R5	:GET REVECTOR POINTER
169	004102	107205	000002	8\$:	SUB	#ERLEN,R5	:LOOK FOR PRIMARY BACKWARDS
170	004104	104652	000001		MOV	1(R5),R2	:GET HIGH ORDER
171	004106	103202	007777		BIC	#LO,R2	:CLEAR LO STUFF

172	004110	106202	050000		CMP	#HD.PRIV,R2	:IS IT THE PRIMARY ?
173	004112	054102			BNE	8\$	:NO - KEEP LOOKING
174	004113	104652	000001		MOV	1(R5),R2	:GET HIGH ORDER AGAIN
175	004115	103202	170000		BIC	#HD.CLR,R2	:CLEAR THE HEADER
176	004117	101202	030000		BIS	#HD.REV,R2	:MAKE IT SECONDARY
177	004121	100652	000001		MOV	R2,1(R5)	:STORE IT BACK
178	004123	115400	001472		INC	SND CNT	:INC SECONDARY COUNTER
179	004125	103200	000004	001220	BIC	#RPRIM,FLAG1	:DON'T NEED GOOD EDC ANY LONGER
180	004130	103200	000040	001220	7\$: BIC	#BDHD,FLAG1	:CLEAR SO WILL PUT IN AS RBN
181	004133	101200	001000	001217	BIS	#PRIM,FLAG	:SET SO NONE WILL BE PRIMARY
182	004136	104643	000002		BDIRECT: MOV	FT.HI(R4),R3	:RELOAD HEADER
183	004140	103203	170000		BIC	#HD.CLR,R3	:CLEAR THE HEADER
184	004142	101203	110000		BIS	#HD.BAD,R3	:MARK AS BAD
185	004144	100643	000002		MOV	R3,FT.HI(R4)	:STORE BACK IN IMAGE
186	004146	102200	000100	001217	BIT	#REVECT,FLAG	:IN RET ?
187	004151	054165			ENE	LSND	:NO - PUT IN TO REVECTOR
188	004152	004223			BR	LSKIP3	:ELSE DO NEXT ENTRY
189	004153	102200	000040	001220	SND: BIT	#BDHD,FLAG1	:BAD HEADER ?
190	004156	054136			BNE	BDIRECT	:YUP - MARK AS BAD
191	004157	101203	030000		BIS	#HD.REV,R3	:MARK AS SECONDARY
192	004161	100643	000002		MOV	R3,FT.HI(R4)	:STORE BACK IN IMAGE
193	004163	115400	001472		INC	SND CNT	:INC IT
194	004165	115400	001221		LSND: INC	ERFLAG	:SET RE-FORMAT FLAG
195	004167	104303	001252		MOV	ERPNT,R3	:STORE BACK
196	004171	104642	000001		MOV	FT.LOW(R4),R2	:GET LOW ORDER BLOCK NUMBER
197	004173	100232			MOV	R2,(R3)+	:STORE FOR RCT UPDATE
198	004174	104642	000002		MOV	FT.HI(R4),R2	:GET HIGH ORDER
199	004176	102200	000040	001220	BIT	#BDHD,FLAG1	:BAD HEADER ?
200	004201	014210			BEQ	LSKIP9	:NO - HANDLE AS USUAL
201	004202	103202	170000		BIC	#HD.CLR,R2	:ELSE CLEAR BAD HEADER CODE
202	004204	101202	030000		BIS	#HD.REV,R2	:AND PUT IN SECONDARY CODE
203	004206	115400	001472		INC	SND CNT	:INC SECONDARY COUNT
204	004210	100232			LSKIP9: MOV	R2,(R3)+	:STORE FOR RCT UPDATE
205	004211	104030	001252		MOV	R3,ERPNT	:STORE BACK
206	004213	115400	001254		INC	REVCNT	:INCREMENT IT
207	004215	115400	001453		INC	ERRCNT	:UP COUNTER OF BAD BLOCKS
208	004217	004223			BR	LSKIP3	:NO NEED TO RE-READ ANY MORE THIS SECTOR
209	004220	117400	001456		LOK: DEC	N1	:DECREMENT COUNTER
210	004222	053615			BNE	LHERE	:NO - RE-READ SECTOR IN ERROR
211	004223	104300	001457	001456	LSKIP3: MOV	NN1,N1	:GET SAVED VALUE
212	004226	117400	001226		DEC	ERR	:DECREMENT IT
213	004230	103200	000040	001220	LSKIP7: BIC	#BDHD,FLAG1	:CLEAR BAD HEADER FLAG
214	004233	105204	000003		ADD	#IMLEN,R4	:POINT TO NEXT ENTRY
215	004235	115000	001226		TST	ERR	:DONE ALL SECTORS ?
216	004237	053615			BNE	LHERE	:NO - DO NEXT SECTOR
217	004240	115000	001221		TST	ERFLAG	:WERE THERE ANY BAD SECTORS FOUND
218	004242	014360			BEQ	LDONE	:NOPE - ALL DONE
219	004243	104207	006621		MOV	#RBNBUF,R0	:POINT TO RBN BUFFER
220	004245	104301	001450		MOV	DWRD,R1	:DIAGNOSTIC WORD
221	004247	100271			MOV	R1,(R0)+	:STORE IT
222	004250	104204	000125		MOV	#85,R4	:SET COUNTER OF TRIPLE WORDS
223	004252	104301	001445		MOV	FWRD,R1	:FIRST WORD OF PATTERN
224	004254	104302	001446		MOV	SWRD,R2	:SECOND WORD OF PATTERN
225	004256	104303	001447		MOV	TWRD,R3	:THIRD WORD OF PATTERN
226	004260	100271			LOVER: MOV	R1,(R0)+	:STORE IT
227	004261	100272			MOV	R2,(R0)+	:STORE IT
228	004262	100273			MOV	R3,(R0)+	:STORE IT

229	004263	117404		DEC	R4	:DECREMENT COUNTER
230	004264	054260		BNE	LOVER	:REPEAT TILL DONE
231	004265	104302	001452	MOV	BADEDC,R2	:EDC FOR PATTERN (FORCED ERROR IND)
232	004267	100272		MOV	R2,(R0)+	:STORE IT
233	004270	102200	000004 001220	BIT	#RPRIM,FLAG1	:NEED GOOD RBN EDC ???
234	004273	014321		BEQ	LOVER2	:NOPE
235	004274	104203	007275	MOV	#IMAGE,R3	:POINT TO IMAGE
236	004276	104205	005567	MOV	#GDBLK,R5	:POINT TO GOOD BLOCK
237	004300	104632	000002	LOVER4: MOV	FT.HI(R3),R2	:GET HI ORDER
238	004302	103202	007777	BIC	#LO,R2	:CLEAR JUNK
239	004304	106202	060000	CMP	#HD.RBN,R2	:IS IT THE PRIMARY ?
240	004306	014312		BEQ	LOVER3	:YUP - HANDLE IT
241	004307	105203	000003	ADD	#IMLEN,R3	:CHECK NEXT ENTRY
242	004311	004300		BR	LOVER4	:TRY AGAIN
243	004312	104632	000000	LOVER3: MOV	FT.BUF(R3),R2	:GET BUFFER POINTER
244	004314	103202	007777	BIC	#BUFMSK,R2	:CLEAR ONLY BUFFER POINTER
245	004316	101052		BIS	R5,R2	:SET IN NEW BUFFER POINTER
246	004317	100632	000000	MOV	R2,FT.BUF(R3)	:STORE IT
247	004321	104304	001334	LOVER2: MOV	DPREA,R4	:DATA PREAMBLE LENGTH
248	004323	104303	001333	MOV	HPREA,R3	:HEADER PREAMBLE LENGTH
249	004325	104307	001332	MOV	IMSTAR,R0	:POINT TO TRACK IMAGE START POINT
250	004327	104301	001112	MOV	CURTRK,R1	:CURRENT TRACK NUMBER
251	004331	104302	000740	MOV	UNIT,R2	:SDI INTERCONNECT
252	004333	104205	007275	MOV	#IMAGE,R5	:RECIRCULATION ADDRESS
253	004335	060001		XFC	FORMAT	:RE-FORMAT
254	004336	115001		TST	R1	:ANY PROBLEMS ??
255	004337	014355		BEQ	LOVER1	:NO - DO CHECK PASS
256	004340	115400	000717	INC	UN.ERI	:INCREMENT IT
257	004342	106300	001502 000717	CMP	RETRY,UN.ERI	:DONE ALL RETRIES ?
258	004345	073431		BMI	FORERR	:YUP - ERROR
259	004346	022371		CALL	INITPT	:REINIT
260	004347	022243		CALL	CLEAR	:DRIVE CLEAR
261	004350	104300	001463 001100	MOV	CURGRP,ISEEK+3	:GROUP
262	004353	022251		CALL	SEEK	:RE-SEEK AND GROUP SELECT
263	004354	004321		BR	LOVER2	:NOPE - RETRY
264	004355	114000	000717	LOVER1: CLR	UN.ERI	:CLEAR RETRY COUNT
265	004357	003442		BR	LCHEC	:RE-CYCLE CHECK PASS
266	004360	000000		LDONE: RETURN		

```

1
2
3
4
5
6 004361 104300 001127 001454 LFIXIT: MOV SECTRK,SECCNT ;INIT COUNTER
7 004364 104207 006621 MOV #CMDBUF,R0 ;COMMAND BUFFER
8 004366 104205 007275 MOV #IMAGE,R5 ;POINT TO TRACK IMAGE
9 004370 104303 001233 MOV SKPCNT,R3 ;GET STARTING OFFSET(TUNED)
10 004372 105035 ADD R3,R5 ;POINT TO FIRST ENTRY
11 004373 104050 001232 MOV R5,STARIT ;MARK STARTING ADDRESS
12 004375 104653 000002 LMORE: MOV 2(R5),R3 ;SET UP FOR HSR CODE COMPARE
13 004377 103203 007777 BIC #LO,R3 ;ISOLATE HI 4 BITS(HDR CODE)
14 004401 106203 000000 CMP #HD.LBN,R3 ;GOOD LBN ?
15 004403 014415 BEQ FLKIP2 ;YES - MARK AS GOOD TO CHECK
16 004404 106203 060000 CMP #HD.RBN,R3 ;GOOD RBN ?
17 004406 014415 BEQ FLKIP2 ;YES - MARK AS GOOD TO CHECK
18 004407 114003 CLR R3 ;CLEAR FOR STORE
19 004410 100673 000000 MOV R3,RB.BUF(R0) ;STORE AS BAD SECTOR FLAG
20 004412 105207 000005 ADD #RDLEN,R0 ;POINT TO NEXT BLOCK
21 004414 004434 BR FLKIP1 ;SKIP GOOD MARK
22 004415 104203 004535 FLKIP2: MOV #RDBUF,R3 ;LOAD ADDRESS OF DATA BUFFER
23 004417 100273 MOV R3,(R0)+ ;STORE IN BLOCK
24 004420 104653 000001 MOV 1(R5),R3 ;LO ORDER BLOCK NUMBER
25 004422 100273 MOV R3,(R0)+ ;STORE IN READ CMD BLOCK
26 004423 104653 000002 MOV 2(R5),R3 ;HI ORDER BLOCK NUM AND CODE
27 004425 100273 MOV R3,(R0)+ ;STORE IN READ CMD BLOCK
28 004426 104203 013400 MOV #RWCMND,R3 ;LOAD SDI READ COMMAND
29 004430 101303 001112 BIS CURTRK,R3 ;SET IN CURRENT TRACK NUMBER
30 004432 100273 MOV R3,(R0)+ ;STORE IN BLOCK
31 004433 100275 MOV R5,(R0)+ ;SAVE PTR TO IMAGE BLK ENTRY
32 004434 105305 001234 FLKIP1: ADD TBLK,R5 ;ADD TO GET NEXT SECTOR
33 004436 106305 001231 CMP EIMAGE,R5 ;SEE IF HAVE TO LOOP BACK TO TOP
34 004440 014447 BEQ LREDO ;NEED TO RESET
35 004441 034451 BPL FLKP1 ;NO NEED - JUST CONTINUE
36 004442 107305 001231 SUB EIMAGE,R5 ;SUBTRACT TO GET LOOP AMOUNT
37 004444 105205 007275 ADD #IMAGE,R5 ;AND ADD OFFSET
38 004446 004451 BR FLKP1 ;SKIP ZERO CONDITION
39 004447 104205 007275 LREDO: MOV #IMAGE,R5 ;IF ZERO SIMPLY MOVE TO FRONT
40 004451 106305 001232 FLKP*: CMP STARIT,R5 ;AT BEGINNING ADDRESS ?
41 004453 054460 BNE FKIP9 ;NO - JUST CONTINUE
42 004454 105705 000003 ADD #IMLEN,R5 ;ELSE POINT TO NEXT ENTRY
43 004456 104050 001232 MOV R5,STARIT ;MAKE IT NEW STARTING ADDRESS
44 004460 117400 001454 FKIP9: DEC SECCNT ;DECREMENT
45 004462 054375 BNE LMORE ;NO - DO NEXT SECTOR
46 004463 000000 RETURN

```

```

1      .SBTTL  LBN FORMAT IMAGE SETUP OVERLAY (F8)
2 004464  DMOVL  F8,START
3
4      SET UP TRACK FORMAT
5
6 003047 104200 000025 001153 LTRK:  MOV  #F8,CUROVL      ;GET OVERLAY OFFSET
7 003052 104207 001052      MOV  #SCR,R0      ;POINT TO CHARACTERISTICS BLOCK
8 003054 104673 000011      MOV  LBNTRK(R0),R3    ;GET LBN'S/TRACK
9 003056 103203 177400      BIC   #HIBYTE,R3      ;CLEAR HIGH BYTE
10 003060 104030 001454      MOV  R3,SECCNT      ;USE AS COUNTER
11 003062 104302 001223      MOV  BADPBN,R2      ;POINT TO CURRENT BAD PBN ENTRY
12 003064 104205 007275      MOV  #IMAGE,R5      ;POINT TO TRACK IMAGE BUFFER
13 003066 104300 001123 001110  MOV  HOLDPN,CURPBN    ;GET LOW ORDER PBN
14 003071 104300 001124 001111  MOV  HOLDPN+1,CURPBN+1 ;GET HIGH ORDER
15 003074 104203 005567      MOV  #GDBLK,R3      ;POINT TO DATA BLOCK
16 003076 104304 001451      MOV  EDC,R4        ;GET GOOD EDC
17 003100 100634 000400      MOV  R4,RW.EDC(R3)    ;STORE IN BUFFER
18 003102 103200 001000 001217  BIC   #PRIM,FLAG    ;CLEAR PRIMARY FLAG
19 003105 104203 005567      MOV  #GDBLK,R3      ;POINT TO GOOD DATA BLOCK
20 003107 100653 000000      MOV  R3,FT.BUF(R5)    ;STORE IN IMAGE BLOCK
21 003111 104303 001113      MOV  CURBN,R3      ;GET LOW ORDER BLOCK NUMBER
22 003113 100653 000001      MOV  R3,FT.LOW(R5)    ;STORE IN IMAGE BLOCK
23 003115 102200 000001 001217  BIT   #FCTAVL,FLAG    ;IS FCT AVAILABLE ?
24 003120 013301      BEQ   LKIP3      ;NO - ASSUME BLOCK GOOD
25 003121 102200 000002 001217  BIT   #FCTEMT,FLAG    ;FCT EMPTY ?
26 003124 053301      BNE   LKIP3      ;YUP - BLOCK IS GOOD
27 003125 104302 001223      MOV  BADPBN,R2      ;GET FCT POINTER
28 003127 104121      MOV  (R2),R1      ;GET LOW ORDER BAD PBN
29 003130 106010 001110      CMP   R1,CURPBN      ;ARE THEY EQUAL ?
30 003132 053301      BNE   LKIP3      ;NO - SKIP REST OF COMPARE
31 003133 104621 000001      MOV  1(R2),R1      ;GET HIGH ORDER
32 003135 103201 170000      BIC   #HD.CLR,R1      ;CLEAR THE HEADER
33 003137 106010 001111      CMP   P,CURPBN+1    ;ARE THEY EQUAL ?
34 003141 053301      BNE   LKIP3      ;NO - MUST BE GOOD
35 003142 117400 001500      DEC   PCNT      ;DECREMENT IT
36 003144 117400 001237      DEC   FCNT      ;DECREMENT FCT COUNT
37 003146 053152      BNE   LKIP12     ;IF NON - ZERO SKIP FLAG SET
38 003147 101200 000002 001217  BIS   #FCTEMT,FLAG    ;SET EMPTY FLAG
39 003152      LKIP12:
40 003152 102200 000100 001217  BIT   #REVECT,FLAG    ;IN RCT ?
41 003155 053161      BNE   6$      ;NO - SKIP RCT STUFF
42 003156 115400 001313      INC   RCTBAD      ;INCREMENT IT
43 003160 003265      BR    MARBAD      ;GO MARK BAD
44 003161 104623 000001      MOV  1(R2),R3      ;GET BAD PBN HDR
45 003163 102203 100000      BIT   #PRMY,R3      ;IS IT SECONDARY ?
46 003165 013244      BEQ   LKIP5      ;YES - GO DO IT
47 003166 101200 001000 001217  BIS   #PRIM,FLAG    ;SET FLAG FOR PRIMARY FOUND
48 003171 101200 000004 001220  BIS   #RPRIM,FLAG1   ;SET GOOD EDC NEEDED
49 003174 104303 001114      MOV  CURBN+1,R3    ;GET HIGH ORDER HEADER
50 003176 103203 170000      BIC   #HD.CLR,R3      ;CLEAR HEADER CODE
51 003200 101203 050000      BIS   #HD.PRV,R3      ;MARK AS PRIMARY REVECTOR
52 003202 100653 000002      MOV  R3,FT.HI(R5)    ;STORE IN IMAGE BLOCK
53 003204 104207 001052      MOV  #SCR,R0      ;MAKE SURE POINT TO CHAR BLOCK
54 003206 104204 001113      MOV  #CURBN,R4      ;POINT TO BLOCK NUMBER
55 003210 022757      CALL  PRIMRB      ;GET PRIMARY RBN NUMBER
56 003211 104307 001151      MOV  REVRBN,R0      ;GET NUMBER OF REVECTORED RBN
57 003213 104301 001152      MOV  REVRBN+1,R1    ;GET HIGH ORDER

```

58	003215	101201	060000		BIS	#HD.RBN,R1	:SET IN RBN HEADER CODE
59	003217	104204	006204		MOV	#PRMBUF,R4	:USE RDBUF TO HOLD 128 COPIES OF RBN
60	003221	104203	000200		MOV	#RBNRPT,R3	:COUNT OF REPLICATED RBN'S
61	003223	100247		RPT:	MOV	R0,(R4)+	:STORE A COPY
62	003224	100241			MOV	R1,(R4)+	:AND HIGH ORDER
63	003225	117403			DEC	R3	:DECREMENT COUNTER - DONE ?
64	003226	053223			BNE	RPT	:NO - STORE ANOTHER COPY
65	003227	104204	006204		MOV	#PRMBUF,R4	:POINT TO BEGINNING OF BUFFER
66	003231	100654	000000		MOV	R4,FT.BUF(R5)	:STORE NEW BUFFER PTR IN IMAGE
67	003233	105200	000002	001223	ADD	#2,BADPBN	:INCREMENT BADPBN POINTER
68	003236	104202	006204		MOV	#PRMBUF,R2	:POINT TO BUFFER
69	003240	022627			CALL	CEDC	:COMPUTE EDC - RETURNED IN R3
70	003241	100623	060400		MOV	R3,RW.EDC(R2)	:STORE IT
71	003243	003311			BR	LKIP4	:SKIP SECONDARY REVECTOR
72	003244			LKIP5:			
73	003244	115400	001472		INC	SND CNT	:INC IT
74	003246	102202	010000		BIT	#FBDHD,R2	:HEADER IN ERROR CODE IN FCT ?
75	003250	053265			BNE	MARBAD	:YUP - MARK BAD
76	003251	104303	001114		MOV	CURBN+1,R3	:GET HIGH ORDER HEADER
77	003253	103203	170000		BIC	#HD.CLR,R3	:CLEAR HEADER CODE
78	003255	101203	030000		BIS	#HD.REV,R3	:SET HEADER TO SECONDARY REVECTOR
79	003257	100653	000002		MOV	R3,FT.HI(R5)	:STORE IN IMAGE
80	003261	105200	000002	001223	ADD	#2,BADPBN	:INCREMENT BADPBN POINTER
81	003264	003311			BR	LKIP4	:SKIP GOOD MARK
82	003265			MARBAD:			
83	003265	104303	001114		MOV	CURBN+1,R3	:GET HIGH ORDER HEADER
84	003267	103203	170000		BIC	#HD.CLR,R3	:CLEAR HEADER CODE
85	003271	101203	110000		BIS	#HD.BAD,R3	:MARK AS BAD
86	003273	100653	000002		MOV	R3,FT.HI(R5)	:STORE IN IMAGE
87	003275	105200	000002	001223	ADD	#2,BADPBN	:UPDATE COUNTER
88	003300	003311			BR	LKIP4	:SKIP GOOD MARKING
89	003301	104303	001114		MOV	CURBN+1,R3	:GET HIGH ORDER HEADER
90	003303	103203	170000		BIC	#HD.CLR,R3	:CLEAR HEADER
91	003305	101203	000000		BIS	#HD.LBN,R3	:MARK AS GOOD LBN
92	003307	100653	000002		MOV	R3,FT.HI(R5)	:STORE IN IMAGE
93	003311	105205	000003		ADD	#IMLEN,R5	:GO TO NEXT IMAGE BLOCK
94	003313				DUBINC	CURBN	:DO THE INCREMENT
95	003320	102200	000100	001217	BIT	#REVECT,FLAG	:STILL IN RCT AREA ?
96	003323	053335			BNE	LKIP9	:NO - NO NEED TO DECREMENT
97	003324	117400	001235		DEC	RCTTOT	:DECREMENT IT
98	003326	053335			BNE	LKIP9	:OUT OF RCT ? - NO
99	003327	101200	020100	001217	BIS	#REVECT+INIRCT,FLAG	:SET TO REVECTOR
100	003332	104300	001237	001312	MOV	FCNT,LBNBAD	:GET FCT ENTRY COUNT - AFTER RCT
101	003335	102200	002002	001217	BIT	#BSTGS+FCTEMT,FLAG	:DOING BEST GUESS OR FCT EMPTY
102	003340	053352			BNE	LKIP10	:YUP - SKIP FCT STUFF
103	003341				DUBINC	CURPBN	:DO THE INCREMENT
104	003346	104303	001500		MOV	PCNT,R3	:GET PBN COUNT
105	003350	053352			BNE	LKIP10	:IF NOT DONE SKIP
106	003351	023657			CALL	FCPG	:ELSE PAGE IN NEW FCT BLOCK
107	003352	117400	001454		DEC	SECCNT	:DECREMENT IT
108	003354	053105			BNE	LKIP2	:NO - DO NEXT LBN
109							
110							
111							
112							
113							
114	003355	104207	001052		MOV	#SCR,R0	:POINT TO CHARACTERISTICS

RBN FORMATTING

15	00335	104673	000004		MOV	RBNTRK(R0),R3	:GET RBN'S/TRACK
116	003361	103203	177600		BIC	#HI1BYTE,R3	:CLEAR HIGH GARBAGE
117	003363	104030	001454		MOV	R3,SECCNT	:USE AS COUNTER
118	003365	104207	006621	LKIP8:	MOV	#RBNBUF,R0	:POINT TO RBN BUFFER
119	003367	104301	001450		MOV	DWRD,R1	:DIAGNOSTIC WORD
120	003371	100271			MOV	R1,(R0)+	:STORE IT
121	003372	104204	000125		MOV	#85.,R4	:SET COUNTER
122	003374	104301	001445		MOV	FWRD,R1	:FIRST WORD OF PATTERN
123	003376	104302	001446		MOV	SWRD,R2	:SECOND WORD OF PATTERN
124	003400	104303	001447		MOV	TWRD,R3	:THIRD WORD OF PATTERN
125	003402	100271		LLOVER:	MOV	R1,(R0)+	:STORE IT
126	003403	100272			MOV	R2,(R0)+	:STORE IT
127	003404	100273			MOV	R3,(R0)+	:STORE IT
128	003405	117404			DEC	R4	:DECREMENT COUNTER
129	003406	053402			BNE	LLOVER	:REPEAT TILL DONE
130	003407	104302	001452		MOV	BADEDC,R2	:EDC FOR PATTERN (FORCED ERROR IND)
131	003411	100272			MOV	R2,(R0)+	:STORE IT
132	003412	104203	006621		MOV	#RBNBUF,R3	:POINT TO BUFFER
133	003414	100653	000000		MOV	R3,FT.BUF(R5)	:STORE IN IMAGE
134	003416	104303	001106		MOV	CURRBN,R3	:GET LOW ORDER RBN
135	003420	100653	000001		MOV	R3,FT.LOW(R5)	:STORE IN IMAGE
136	003422	102200	002002	001217	BIT	#BSTGS+FCTEMT,FLAG	:IS FCT AVAILABLE ?
137	003425	053471			BNE	LKIP6	:NO - CONSIDER GOOD
138	003426	104203	001110		MOV	#CURPBN,R3	:CURRETN PBN
139	003430	104121			MOV	(R2),R1	:GET LOW ORDER BAD
140	003431	104137			MOV	(R3),R0	:GET LOW ORDER CURRENT
141	003432	106071			CMP	R0,R1	:ARE THEY EQUAL ?
142	003433	053471			BNE	LKIP6	:NO - SKIP REST OF COMPARE
143	003434	104621	000001		MOV	1(R2),R1	:GET HIGH ORDER BAD
144	003436	104637	000001		MOV	1(R3),R0	:GET HIGH ORDER CURRENT
145	003440	106071			CMP	R0,R1	:ARETHEY EQUAL ?
146	003441	053471			BNE	LKIP6	:NO - MUST BE GOOD
147	003442	117400	001500		DEC	PCNT	:DECREMENT IT
148	003444	117400	001237		DEC	FCNT	:DECREMENT IT
149	003446	053452			BNE	LKIP13	:IF NON - ZERO SKIP FLAG SET
150	003447	101200	000002	001217	BIS	#FCTEMT,FLAG	:SET EMPTY FLAG
151	003452	104303	001107	LKIP*3:	MOV	CURRBN+1,R3	:GET HIGH ORDR RBN
152	003454	103203	170000		BIC	#HD.CLR,R3	:CLERA THE HEADER
153	003456	101203	110000		BIS	#HD.BAD,R3	:MARK AS BAD
154	003460	100653	000002		MOV	R3,FT.HI(R5)	:STORE IN IMAGE
155	003462	105200	000002	001223	ADD	#2,BADPBN	:POINT TO NEXT BAD
156	003465	101200	001000	001217	BIS	#PRIM,FLAG	:MAKE SURE NO PRIMARIES
157	003470	003514			BR	LKIP7	:SKIP GOOD MARKING
158	003471	104303	001107	LKIP6:	MOV	CURRBN+1,R3	:GET HIGH ORDER RBN
159	003473	103203	170000		BIC	#HD.CLR,R3	:CLEAR HEADER
160	003475	101203	060000		BIS	#HD.RBN,R3	:MARK AS GOOD RBN
161	003477	100653	000002		MOV	R3,FT.HI(R5)	:STORE IN IMAGE
162	003501	102200	000004	001220	BIT	#RPRIM,FLAG1	:NEED GOOD EDC ?
163	003504	013514			BEQ	LKIP7	:NOPE
164	003505	103200	000004	001220	BIC	#RPRIM,FLAG1	:CLEAR FLAG
165	003510	104203	005567		MOV	#GDBLK,R3	:GET GOOD EDC BLOCK
166	003512	100653	000000		MOV	R3,FT.BUF(R5)	:STORE IT IN BUFFER POINTER
167	003514			LKIP7:	DUBINC	CURRBN	:DO THE INCREMENT
168	003521	105205	000003		ADD	#IMLEN,R5	:POINT TO NEXT IMAGE BLOCK
169	003523	102200	002002	001217	BIT	#BSTGS+FCTEMT,FLAG	:DOING BEXT GUESS ?
170	003526	053546			BNE	LKIP14	:YUP - SKIP FCT STUFF
171	003527	104204	001110		MOV	#CURPBN,R4	:PREPARE FOR INC

172	003531	104141			MOV	(R4),R1	:GET LOW ORDER OPERAND
173	003532	105201	000001		ADD	#1,R1	:INCREMENT WITH CARRY
174	003534	100241			MOV	R1,(R4)+	:SAVE LOW ORDER RESULT
175	003535	043542			BCC	RINC1	:BRANCH IF DONE
176	003536	104141			MOV	(R4),R1	:GET HIGH ORDER RESULT
177	003537	105201	000001		ADD	#1,R1	:INCREMENT
178	003541	100141			MOV	R1,(R4)	:SAVE HIGH ORDER
179	003542	104303	001500	RINC1:	MOV	PCNT,R3	:GET PBN COUNT
180	003544	053546			BNE	LKIP14	:IF NOT DONE SKIP
181	003545	023657			CALL	FCPG	:ELSE PAGE IN NEW FCT BLOCK
182	003546	117400	001454	LKIP14:	DEC	SECCNT	:DECREMENT IT
183	003550	053365			BNE	LKIP8	:NO - DO NEXT RBN
184	003551	107205	000003		SUB	#IMLEN,R5	:POINT TO LAST ENTRY BUFF POINTER
185	003553	104153			MOV	(R5),R3	:GET IT
186	003554	101203	040000		BIS	#RECIR,R3	:MARK IT AS RECIRC
187	003556	100153			MOV	R3,(R5)	:STORE BACK
188	003557	115000	001463		TST	CURGRP	:IS IT GROUP ZERO ?
189	003561	013573			BEQ	LKIP19	:YES - NO OFFSET
190	003562	104203	001052		MOV	#SCR,R3	:POINT TO CHARACTERISTICS
191	003564	104632	000011		MOV	OFFS(R3),R2	:GET GROUP OFFSET
192	003566	110702			SWAB	R2	:GET IN LOW ORDER
193	003567	103202	177400		BIC	#HIBYTE,R2	:CLEAR HIGH ORDER
194	003571	115002			TST	R2	:ANY GROUPS ?
195	003572	053601			BNE	LKIP15	:YUP - HANDLE IT
196	003573	104200	007275	001332	LKIP19:	MOV	#IMAGE,IMSTAR
197	003576	104304	001332		MOV	IMSTAR,R4	:ELSE START AT VERY BEGINNING
198	003600	003645			BR	LKIP16	:POINT TO START FOR LATER
199	003601	104020	000736		LKIP15:	MOV	R2,TEMP
200	003603	114000	000737		CLR	TEMP+1	:STORE IT
201	003605	104300	001463	000731	MOV	CURGRP,DDUMMY	:FOR STORE
202	003610	114000	000732		CLR	DDUMMY+1	:GET CURRENT GROUP
203	003612	104203	000736		MOV	#TEMP,R3	:CLEAR HIGH ORDER
204	003614	104204	000731		MOV	#DDUMMY,R4	:FOR MUL
205	003616	021542			CALL	DMUL	:DITTO
206	003617	106300	001127	000731	LKIP17:	CMP	SECTRK,DDUMMY
207	003622	033627			BPL	LKIP18	:MULTIPLY TO GET OFFSET FOR THIS GROUP
208	003623	107300	001127	000731	SUB	SECTRK,DDUMMY	:IS TOTAL OFFSET MORE THAN NUMBER OF SECTORS ?
209	003626	003617			BR	LKIP17	:NO - ALL IS FINE
210	003627	104200	000003	000736	LKIP18:	MOV	#IMLEN,TEMP
211	003632	114000	000737		CLR	TEMP+1	:YES - SUBTRACT TILL IT IS
212	003634	104203	000736		MOV	#TEMP,R3	:CHECK AGAIN
213	003636	021542			CALL	DMUL	:MUST MULTIPLY BY ENTRY LENGTH
214	003637	104143			MOV	(R4),R3	:FOR CLEAR
215	003640	104304	001231		MOV	EIMAGE,R4	:FOR MULT
216	003642	107034			SUB	R3,R4	:DO MULTIPLY
217	003643	104040	001332		MOV	R4,IMSTAR	:GET RESULT
218	003645	104143			LKIP16:	MOV	(R4),R3
219	003646	101203	100000		BIS	#LAST,R3	:GET ADDRESS OF END OF IMAGE
220	003650	100143			MOV	R3,(R4)	:SUBTRACT TO GET STARTING LOCATION
221	003651	104204	001123		MOV	#HOLDPN,R4	:STORE IN IMSTAR
222	003653	104203	001127		MOV	#SECTRK,R3	:GET BUFF POINTER
223	003655	021524			CALL	DSUB	:SIGNAL AS LAST
224	003656	000000			RETURN		:STORE IT BACK
							:FOR DECREMENT
							:DITTO
							:SUBTRAC TO GET NEXT TRACK

1						
2						
3						
4	003657					
5	003660	104200	005152	001253		
6	003663	104201	000017			
7	003665	022444				
8	003666	104200	000200	001500		
9	003671					
10	003672	000000				

FCPG:	PUSH	R5	
	MOV	#PBNBUF,BUFPNT	
	MOV	#F6,R1	
	CALI	PAGE	
	MOV	#128.,PCNT	
	POP	R5	
	RETURN		

	:SAVE R5
	:POINT TO BUFFER
	:OVERLAY F6 DOES IT
	:EXECUTE IT
	:RESET COUNT
	:RESTORE R5
	:RETURN

```

1
2
3
4
5
6 003673
7
8
9 003047 104200 000052 001153
10 003052 104207 001052
11 003054 023104
12 003055 104140 001113
13 003057 104140 001117
14 003061 104640 000001 001114
15 003064 104640 000001 001120
16 003067 023144
17 003070 104140 001106
18 003072 104640 000001 001107
19 003075 104140 001121
20 003077 104640 000001 001122
21 003102 023204
22 003103 000000
  
```

.SBTTL L/RBN COMPUTE OVERLAY (G8)

THIS OVERLAY COMPUTES LBN AND RBN OF THE LAST TRACK ON LAST LBN CYLINDER
 AND COMPUTES THE PBN OF THAT LBN

DMOVLY G8,START

```

MOV    #G8,CUROVL          ;FOR RECCRDING
MOV    #SCR,R0              ;POINT TO CHARACTERISTICS BLOCK
CALL   NUMLBN               ;GET NUMBER OF FIRST LBN ON LAST LBN CYL
MOV    (R4),CURBN           ;GET LOW ORDER
MOV    (R4),HOLDBN          ;SAVE FOR LATER
MOV    1(R4),CURBN+1        ;GET HIGH ORDER
MOV    1(R4),HOLDBN+1       ;SAVE FOR LATER
CALL   NUMRBN               ;GET NUM OF FIRST RBN ON LAST LBN CYLINDER
MOV    (R4),CURRBN          ;GET LOW ORDER
MOV    1(R4),CURRBN+1       ;GET HIGH ORDER
MOV    (R4),HOLRBN          ;SAVE LOW FOR LATER
MOV    1(R4),HOLRBN+1       ;SAVE HIGH FOR LATER
CALL   LPBN                 ;GET PBN OF FIRST SECTOR ON LAST TRACK
RETURN
  
```

```

1
2
3
4
5 003104 104300 001143 000736 NUMLBN: MOV LBNCYL,TEMP ;GET LOW ORDER NUM OF LBN CYLINDERS
6 003107 104300 001144 000737 MOV LBNCYL+1,TEMP+1 ;GET HIGH ORDER
7 003112 103200 170000 000737 BIC #HD.CLR,TEMP+1 ;CLEAR STARTING CYLINDER BITS
8 003115 104204 000736 MOV #TEMP,R4 ;DITTO
9 003117 104203 001145 MOV #LBNPCY,R3 ;GET LBN'S/CYLINDER
10 003121 021542 CALL DMUL ;GET FIRST LBN ON LAST CYLINDER
11 003122 104641 000001 MOV 1(R4),R1 ;GET LBN
12 003124 105301 001335 ADD ST.LBN,R1 ;ADD STARTING LBN TO GET ABSOLUTE LBN
13 003126 100641 000001 MOV R1,1(R4) ;STORE BACK
14 003130 104673 000011 MOV LBNTRK(R0),R3 ;GET LBN/TRK
15 003132 103203 177400 BIC #HIBYTE,R3 ;CLERA HIGH BYTE
16 003134 104030 000731 MOV R3,DDUMMY ;STORE IT
17 003136 114000 000732 CLR DDUMMY+1 ;FOR STORE
18 003140 104203 000731 MOV #DDUMMY,R3 ;LBN/TRACK
19 003142 021524 CALL DSUB ;WANT LBN ON LAST TRACK
20 003143 000000 RETURN
  
```

1						
2						
3						
4						
5	003144	104300	001143	000736	NUMRBN:	MOV LBNCYL,TEMP ;GET LOW ORDER NUMBER OF LBN CYLINDER
6	003147	104300	001144	000737	MOV LBNCYL+1,TEMP+1 ;GET HIGH ORDER	
7	003152	103200	170000	000737	BIC #HD.CLR,TEMP+1 ;CLEAR STARTING CYLINDER BITS	
8	003155	104204	000736		MOV #TEMP,R4 ;DITTO	
9	003157	104203	001147		MOV #RBNPCY,R3 ;GET RBN'S/CYLINDER	
10	003161	021542			CALL DMUL ;GET FIRST RBN ON LAST CYLINDER	
11	003162	104641	000001		MOV 1(R4),R1 ;GET HIGH ORDER	
12	003164	105301	001336		ADD ST.RBN,R1 ;ADD TO GET ABSOLUTE LBN	
13	003166	100641	000001		MOV R1,1(R4) ;STORE BACK	
14	003170	104673	000004		MOV RBNTRK(R0),R3 ;GET RBN/TRACK	
15	003172	103203	177600		BIC #HI1BYTE,R3 ;CLEAR OUT GARBAGE	
16	003174	104030	000731		MOV R3,DDUMMY ;STORE IT	
17	003176	114000	000732		CLR DDUMMY+1 ;FOR STORE	
18	003200	104203	000731		MOV #DDUMMY,R3 ;WANT LAST TRACK	
19	003202	021524			CALL DSUB ;GET IT	
20	003203	000000			RETURN	

1									
2									
3									
4									
5	003204	104300	001113	000736	L PBN:	MOV	CURBN,TEMP		:GET LOW ORDER
6	003207	104300	001114	000737		MOV	CURBN+1,TEMP+1		:GET HIGH ORDER
7	003212	104204	000736			MOV	#TEMP,R4		:FOR SUBTRACT
8	003214	104641	000001			MOV	1(R4),R1		:GET HIGH ORDER
9	003216	107301	001335			SUB	ST.LBN,R1		:SUB STARTING LBN TO GET ABSOLUTE LBN
10	003220	100641	000001			MOV	R1,1(R4)		:STORE BACK
11	003222	104673	000011			MOV	LBNTRK(R0),R3		:GET LBN'S/TRACK
12	003224	103203	177400			BIC	#HI1BYTE,R3		:CLEAR HIGH WORD
13	003226	104030	000731			MOV	R3,DDUMMY		:STORE FOR COMPUTATION
14	003230	114000	000732			CLR	DDUMMY+1		:CLEAR FOR STORE
15	003232	104203	000731			MOV	#DDUMMY,R3		:FOR DIVIDE
16	003234	021570				CALL	DDIV		:GET NUMBER OF TRACKS
17	003235	104673	000004			MOV	RBNTRK(R0),R3		:GET RBN'S/TRACK
18	003237	103203	177600			BIC	#HI1BYTE,R3		:CLEAR GARBAGE
19	003241	104030	000731			MOV	R3,DDUMMY		:FOR COMPUTATION
20	003243	114000	000732			CLR	DDUMMY+1		:CLEAR HIGH WORD
21	003245	104203	000731			MOV	#DDUMMY,R3		:FOR MULTIPLY
22	003247	021542				CALL	DMUL		:GET NUMBER OF RBN'S
23	003250	104300	001113	000731		MOV	CURBN,DDUMMY		:GET LOW ORDER CURRENT BLOCK NUMBER
24	003253	104300	001114	000732		MOV	CURBN+1,DDUMMY+1		:GET HIGH ORDER
25	003256	107300	001335	000732		SUB	ST.LBN,DDUMMY+1		:SUBTRACT STARTING
26	003261	104203	000731			MOV	#DDUMMY,R3		:FOR ADD
27	003263	021506				CALL	DADD		:ADD TO GET PBN
28	003264	104140	001123			MOV	(R4),HOLDON		:GET LOW ORDER
29	003266	104640	000001	001124		MOV	1(R4),HOLDON+1		:STORE HIGH ORDER
30	003271	000000				RETURN			

```

1      .SBTTL FCT DOWN-LINE LOAD OVERLAY (F3)
2
3      :
4      : DOWNLINE LOADER FOR FCT
5      FDLL: DMOVL F3,START          ;OVERLAY #3
6
7      :
8      : MOV #F3,CUROVL          ;OVERLAY #3
9      : MOV #1,COUNT            ;INIT COUNT TO 1
10     : MOV ST,XBN,CURXBN+1      ;ALSO INITIALIZE XBN COUNTER
11     : MOV ST,XBN,CURBN+1      ;HIGH ORDER
12     : CLR CURXBN              ;LOW ORDER IS ZERO
13     : CLR CURBN               ;DITTO
14     : MOV #CONBLK,R0          ;POINT TO CONVERT BLOCK
15     : MOV #SCR,R3             ;POINT TO CHARACTERISTICS
16     : MOV CYLBN(R3),R2        ;GET LBN CYLINDERS
17     : MOV R2,V1(R0)           ;STORE IN CONVERT BLOCK
18     : MOV CYLBN+1(R3),R2      ;GET HIGH ORDER
19     : MOV R2,V1+1(R0)         ;STORE IT
20     : MOV SECTR,R3            ;GET SECTORS/TRACK
21     : MOV R3,V3(R0)           ;STORE IN CONVERT BLOCK
22     : BIT #BSTGS,FLAG         ;IN BEST GUESS MODE ?
23     : BNE NODLL              ;YUP - FIX UP FIRST BLOCK
24     : BR LOOP                ;START LOOP
25     :
26     : NODLL: MOV #RDBUF,R3     ;POINT TO BUFFER
27     : CLR R2                 ;SET MEDIA MODE TO 0 (IN FORMAT)
28     : MOV R2,(R3)            ;STORE IT
29     : MOV #SERNUM,R4          ;POINT TO SERIAL NUMBER
30     : ADD #FSER,R3           ;POINT TO ENTRY IN FCT BLOCK
31     : MOV #4,R5              ;INIT COUNTER
32     : 9$: MOV (R4)+,R2        ;GET WORD
33     : MOV R2,(R3)+          ;STORE WORD
34     : DEC R5                 ;DECRMENT COUNTER
35     : BNE 9$                 ;CONTINUE TILL DONE
36     : CLR R2                 ;FOR INSTANCE NUMBER
37     : MOV #RDBUF,R3          ;RESET POINTER
38     : MOV R2,INST(R3)        ;STORE INSTANCE NUMBER IN BLOCK
39     : BIS #NOFCT,R2          ;SET NO FCT AVAILABLE BIT
40     : MOV R2,FCTFLG(R3)      ;STORE IT IN FCT INFO BLOCK
41     : BR LOOPP2              ;SKIP DLL STUFF
42     :
43     : LOOP: MOV #DMBUF,R5     ;POINT TO MAINT BUFFER
44     : MOV FCMMSG,R3           ;GET DUP CODE
45     : MOV R3,(R5)            ;STORE IT IN MESSAGE
46     : MOV FCMMSG+1,R3        ;GET 'F' IDENTIFIER
47     : MOV R3,1(R5)           ;STORE IT
48     : MOV CURBN,R3           ;GET BLOCK NUMBER DESIRED
49     : MOV R3,2(R5)           ;STORE IT
50     : CALL SNDMNT             ;SEND REQUEST
51     : CALL RCVMT              ;RECEIVE ANSWER
52     : MOV 2(R5),R3           ;GET BLOCK COUNT RECEIVED
53     : BEQ DONDLL             ;DONE DLL OF ACTUAL FCT
54     : MOV 3(R5),OVLBLK+1     ;GET HOST ADDRESS
55     : CLR OVLBLK+2           ;ZERO HIGH ORDER
56     : MOV #257,OVLBLK       ;NUMBER OF WORDS TO TRANSFER
57     : MOV #OVLBLK,R4        ;FOR OVERLAY ROUTINE
58     : MOV #RDBUF,R3         ;POINT TO BUFFER
59     : CALL OVLAY             ;GET THE SECTOR
60     : CLR R5                 ;CLEAR WRITE ERROR COUNT
61     :
62     : LOOPP2: CLR R5

```

58	003204	104050	001245		MOV	R5,NEXT1	:CLEAR REPEAT COUNT
59	003206	106200	000001	001501	CMP	#1,COUNT	:IS IT THE FIRST ONE ?
60	003211	053231			BNE	LOOPP	:NO - SKIP THIS STUFF
61	003212	104204	001301		MOV	#DATE,R4	:POINT TO SERIAL NUMBER
62	003214	104203	004535		MOV	#RDBUF,R3	:POINT TO BUFFER
63	003216	114005			CLR	R5	:FOR MEDIA FORMAT UPDATE
64	003217	100135			MOV	R5,(R3)	:SET FORMAT IN PROGRESS
65	003220	105203	000012		ADD	#FDAT,R3	:POINT TO ENTRY IN FCT BLOCK
66	003222	104205	000004		MOV	#4,R5	:INIT COUNTER
67	003224	104242		10\$:	MOV	(R4)+,R2	:GET WORD
68	003225	100232			MOV	R2,(R3)+	:STORE WORD
69	003226	117405			DEC	R5	:DECRMENT COUNTER
70	003227	053224			BNE	10\$	:CONTINUE TILL DONE
71	003230	114005			CLR	R5	:CLEAR R5 (ERROR COUNTER)
72	003231	104202	004535		MOV	#RDBUF,R2	:POINT TO BUFFER
73	003233	022627		LOOPP:	CALL	CEDC	:COMPUTE EDC
74	003234	100623	000400		MOV	R3,RW.EDC(R2)	:STORE IT
75	003236	104300	001113	000736	MOV	CURBN,TEMP	:GET LOW ORDER
76	003241	104300	001114	000737	MOV	CURBN+1,TEMP+1	:GET HIGH ORDER
77	003244	104204	000736		MOV	#TEMP,R4	:FOR SUB
78	003246	104641	000001		MOV	1(R4),R1	:GET HIGH ORDER
79	003250	107301	001337		SUB	ST.XBN,R1	:SUBTRACT STARTING XBN
80	003252	100641	000001		MOV	R1,1(R4)	:STORE BACK
81	003254	022714			CALL	CVTSK	:CONVERT AND SEEK
82	003255	104207	000721		MOV	#WRBLK,R0	:POINT TO COMMAND BLOCK
83	003257	104203	122400		MOV	#WRCMD,R3	:GET WRITE COMMAND
84	003261	104302	001112		MOV	CURTRK,R2	:GET CURRENT TRACK
85	003263	101023			BIS	R2,R3	:SET TRACK FOR WRITE
86	003264	100673	000004		MOV	R3,RW.CMD(R0)	:STORE IN COMMAND BLOCK
87	003266	104203	004535		MOV	#RDBUF,R3	:POINT TO BUFFER
88	003270	100673	000001		MOV	R3,RW.BUF(R0)	:STICK IN COMMAND BLOCK
89	003272	104303	001113		MOV	CURBN,R3	:GET LOW ORDER HEADER
90	003274	100673	000002		MOV	R3,RW.LOW(R0)	:STORE IN WRITE BLOCK
91	003276	104303	001114		MOV	CURBN+1,R3	:GET HIGH ORDER
92	003300	101203	120000		BIS	#HD.XBN,R3	:SET HEADER
93	003302	100673	000003		MOV	R3,RW.HI(R0)	:STORE IN WRITE BLOCK
94	003304	104203	000726		MOV	#HSLIM-1,R3	:GET DUMMY SDI POINTER
95	003306	100673	000005		MOV	R3,RW.DUM(R0)	:POINT IN COMMAND BLOCK
96	003310	104303	001333	WRITE1:	MOV	HPREA,R3	:GET HEADER PREAMBLE
97	003312	104304	001334		MOV	DPREA,R4	:GET DATA PREAMBLE
98	003314	104302	000740		MOV	UNIT,R2	:SET UNIT
99	003316	104207	000721		MOV	#WRBLK,R0	:MAKE SURE POINTING AT BLOCK
100	003320	101207	100000		BIS	#BIT15,R0	:SET NO REVECTORING
101	003322	060012			XFC	SIP	:WAIT FOR SECTOR PULSE
102	003323	060003			XFC	WRITE	:WRITE SECTOR
103	003324	115001			TST	R1	:ANY ERROR ?
104	003325	013347			BEQ	NO	:NOPE
105	003326	106300	001502	001504	CMP	RETRY,IMPTRY	:MAX ?
106	003331	013335			BEQ	1\$	:YES - TRY SOME RECOVERY
107	003332	115400	001504		INC	IMPTRY	:INC RETRY COUNT
108	003334	003310			BR	WRITE1	:DO RETRY
109	003335	104303	001505	1\$:	MOV	RECTMP,R3	:GET CURRENT ERROR RECOVERY LEVEL
110	003337	073346			BMI	2\$	:IF NEGATIVE THEN FRIED
111	003340	022612			CALL	ERRHND	:TRY RECOVERY
112	003341	114000	001504		CLR	IMPTRY	:FOR INIT
113	003343	117400	001505		DEC	RECTMP	:DECREMENT IT
114	003345	003310			BR	WRITE1	:RETRY

Line	Address	Offset	Label	Operation	Comment
115	003346			2\$: INC R5	:YUP - INCREMENT COUNTER
116	003346	115405		NO: INC NEXT1	:INCREMENT IT
117	003347	115400	001245	CLR TMPTRY	:FOR RESET
118	003351	114000	001504	MOV RECOV,RECTMP	:GET RECOVERY LEVELS
119	003353	104300	001503	MOV #CURBN,R4	:FOR ADD
120	003356	104204	001113	MOV #FCTFMT,R3	:FOR ADD
121	003360	104203	001240	CALL DADD	:POINT TO NEXT COPY
122	003362	021506		CMP FCTCPY,NEXT1	:DONE THIS SECTOR ?
123	003363	106300	001244	BNE LOOPP	:NO - WRITE NEXT FCT COPY
124	003366	053231		CMP FCTCPY,R5	:ERROR ON EVERY WRITE ?
125	003367	106305	001244	BEQ ERROR	:YUP - BIG TROUBLE
126	003371	013465		BIT #BSTGS,FLAG	:BEST GUESS ?
127	003372	102200	001217	BNE DLLRET	:YUP - JUST WANT TO WRITE FIRST BLOCK
128	003375	053446		BIT #NDLL,FLAG	:ALL DONE ???
129	003376	102200	001217	BNE DLLRT1	:YUP - EXIT
130	003401	053432		XFC UPDATE	:LET HOST KNOW STILL ALIVE
131	003402	060022		INC COUNT	:INCREMENT IT
132	003403	115400	001501	DUBINC CURXBN	:INCREMENT IT
133	003405			MOV CURXBN,CURBN	:GET LOW ORDER
134	003412	104300	001115	MOV CURXBN+1,CURBN+1	:GET HIGH ORDER
135	003415	104300	001116	CMP FCTNPD,COUNT	:AT THE LAST NON-PAD ENTRY
136	003420	106300	001260	BNE 12\$	:NOPE
137	003423	053426		DEC FCTCPY	:DECREMENT - NO PAD ON LAST COPY
138	003424	117400	001244	CMP FCTFMT,COUNT	:DONE ?
139	003426	106300	001240	BNE LOOP	:NOPE - DO NEXT SECTOR
140	003431	053144		BIS #FCTAVL,FLAG	:SET FCT AVAILABLE
141	003432	101200	000001	MOV #CR,R0	:POINT TO CHARACTERISTICS BLK
142	003435	104207	001037	MOV FRCPY(R0),R3	:GET F/RCT COPIES
143	003437	104673	000001	SWAB R3	:GET INTO JW BYTE
144	003441	110703		BIC #FCLR,R3	:CLEAR OUT REST OF GARBAGE
145	003442	103203	177760	MOV R3,FCTCPY	:RESTORE NUM OF COPIES
146	003444	104030	001244	MOV #F2,R1	:FOR OVERLAY #2
147	003446	104201	000003	CALL NEXT	:LBN FORMATTING
148	003450	022412		MOV #256.,R5	:COUNT FOR BLOCK INIT
149	003451	104205	000400	MOV #RDBUF,R4	:POINT TO BUFFER
150	003453	104204	004535	CLR R3	:INIT TO 0
151	003455	114003		MOV R3,(R4)+	:CLEAR ONE WORD
152	003456	100243		DEC R5	:DEC COUNTER
153	003457	117405		BNE LOOP3	:CONTINUE TILL DONE
154	003460	053456		BIS #NDLL,FLAG	:SET FLAG
155	003461	101200	004000	BR LOOPP2	:CONTINUE
156	003464	003203		MOV R1,R2	:GET XFC ERROR CODE
157	003465	104012		MOV #13.,R1	:FCT WRITE ERROR
158	003466	104201	000015	DLERT: CALL ERRMNT	:ERROR RETURN
159	003470	022552			

1					.SBTTL RCT UPDATE OVERLAY (F4)	
2	003471				DMOVL F4,START	
3						
4						
5						
6						
7						
8					THIS OVERLAY UPDATES THE RCT WITH THE SECTORS	
9					IN THE ERRBUF BUFFER	
10						
11	003047	104200	000011	001153	RCTUPD: MOV #F4,CURDVL	:GET OVERLAY
12	003052	104303	001112		MOV CURTRK,R3	:GET CURRENT TRACK
13	003054				PUSH R3	:SAVE FOR RESTORE
14	003055	104303	001125		MOV CYLNUM,R3	:GET LOW ORDRE CYLINDER
15	003057				PUSH R3	:SAVE FOR RESTORE
16	003060	104303	001126		MOV CYLNUM+1,R3	:GET HIGH ORDER
17	003062				PUSH R3	:SAVE FOR RESTORE
18	003063	104300	001133	001227	MOV LBNLBN,HOLD	:GET LOW ORDER COUNT OF LBN'S
19	003066	104300	001134	001230	MOV LBNLBN+1,HOLD+	:GET HIGH ORDER
20	003071	104203	001475		MOV #TOTRCT,R3	:FOR SUBTRACT
21	003073	104204	001227		MOV #HOLD,R4	:DITTO
22	003075	021524			CALL DSUB	:GET STARTING RCT LBN
23	003076	104300	001224	001474	MOV ERRBUF,UPDPNT	:POINT TO ERROR BUFFER
24	003101	104302	001474		ROVER: MOV UPDPNT,R2	:GET POINTER TO BAD LIST
25	003103	104120	000731		MOV (R2),DDUMMY	:GET LOW ORDER
26	003105	104620	000001	000732	MOV 1(R2),DDUMMY+1	:GET HIGH ORDER
27	003110	102200	100000	000732	BIT #BIT15,DDUMMY+1	:IS IT AN RBN ??
28	003113	013116			BEQ ROVER1	:NO - REGULAR HASH
29	003114	104201	177777		MOV #-1,R1	:HASH FOR RBN
30	003116	103200	170000	000732	ROVER1: BIC #HD.CLR,DDUMMY+1	:CLEAR THE HEADER
31	003121	104204	000731		MOV #DDUMMY,R4	:FOR HASH
32	003123	023357			CALL UHASH	:FIND THE RCT ENTRY FOR CURRENT ERR BLOCK
33	003124	104143			MOV (R4),R3	:GET BLOCK NUMBER
34	003125	105203	000002		ADD #2,R3	:ADD TO GET PAST FIRST 2 BLOCKS
35	003127	100143			MOV R3,(R4)	:STORE BACK
36	003130	104030	001477		MOV R3,RCTCNT	:FOR LATER PING-PONG
37	003132	104203	001227		MOV #HOLD,R3	:FOR ADD
38	003134	021506			CALL DADD	:TO GET LBN OF RCT BLOCK
39	003135	104040	001253		MOV R4,BUFPNT	:STORE POINTER TO BLOCK NUMBER
40	003137	104201	000055		MOV #H1,R1	:RCT READ OVERLAY
41	003141	022444			CALL PAGE	:DO IT
42	003142	104205	006621		MOV #RCTBUF,R5	:POINT TO BUFFER
43	003144	104303	000736		MOV OFFSET,R3	:GET OFFSET
44	003146	105035			ADD R3,R5	:POINT TO HIT ENTRY
45	003147	104302	001474		MOV UPDPNT,R2	:RESTORE POINTER
46	003151	104623	000001		MOV 1(R2),R3	:GET THE HEADER
47	003153	103203	007777		BIC #LO,R3	:CLEAR ALL BUT HEADER
48	003155	106203	110000		CMP #HD.BAD,R3	:IS IT A BAD RBN ?
49	003157	053226			BNE NOTR	:NOPE - CHECK FOR PRIMARY
50	003160	104650	000000	000733	MOV 0(R5),TEMP2	:GET LOW ORDER CURRENT RESIDENT
51	003163	104650	000001	000734	MOV 1(R5),TEMP2+1	:GET HIGH ORDER
52	003166	103203	170000		BIC #HD.CLR,R3	:CLEAR HEADER
53	003170	101203	040000		BIS #RC.UNU,R3	:MARK AS UNUSUABLE
54	003172	103203	007777		BIC #LO,R3	:CLEAR LOW ORDER
55	003174	100653	000001		MOV R3,1(R5)	:STORE IT BACK
56	003176	114003			CLR R3	:CLEAR FOR STORE
57	003177	100153			MOV R3,(R5)	:CLEAR LOW ORDER

58	003200	102200	020000	000734	BIT	#BIT13,TEMP2+1	:ANY THING DISPLACED ???
59	003203	013322			BEO	BOTTOM	:NO - NO NEED TO PING-PONG
60	003204	023421			CALL	RCTWT	:WRITE UT BLOCK
61	003205	023612			CALL	PNGPG	:FIND IT A NEW HOME
62	003206	104204	006621		MOV	#RCTBUF,R4	:POINT TO BUFFER
63	003210	105054			ADD	R5,R4	:ADD OFFSET
64	003211	104202	000733		MOV	#TEMP2,R2	:POINT TO OLD RESIDENT
65	003213	104123			MOV	(R2),R3	:GET LOW ORDER
66	003214	100143			MOV	R3,(R4)	:PUT IT IN
67	003215	104623	000001		MOV	1(R2),R3	:GET HIGH ORDER
68	003217	103203	170000		BIC	#HD.CLR,R3	:CLEAR HEADER
69	003221	101203	030000		BIS	#RC.SND,R3	:MARK AS SECONDARY
70	003223	100643	000001		MOV	R3,1(R4)	:STORE IT
71	003225	003322			BR	BOTTOM	:GO TO BOTTOM OF LOOP
72	003226	106203	050000		NOTR: CMP	#HD.PRIV,R3	:PRIMARY REVECTOR ??
73	003230	053302			BNE	SECNDY	:NO - TREAT AS SECONDARY
74	003231	104653	000001		MOV	1(R5),R3	:GET RCT HEADER
75	003233	103203	007777		BIC	#LO,R3	:CLEAR ALL BUT HEADER
76	003235	106203	000000		CMP	#RC.FRE,R3	:IS IT FREE ??
77	003237	053255			BNE	SWAP	:NO - SWAP ENTRIES
78	003240	104123			MOV	(R2),R3	:GET LOW BLOCK NUMBER
79	003241	100153			MOV	R3,(R5)	:STORE IN RCT
80	003242	104623	000001		MOV	1(R2),R3	:GET HIGH ORDER
81	003244	107303	001335		SUB	ST.LBN,R3	:SUBTRACT STARTING LBN BITS
82	003246	103203	170000		BIC	#HD.CLR,R3	:CLEAR HEADER
83	003250	101203	020000		BIS	#RC.PRIV,R3	:SIGNAL PRIMARY REVECTOR IN RCT
84	003252	100653	000001		MOV	R3,1(R5)	:STORE IN RCT
85	003254	003322			BR	BOTTOM	:GO TO BOTTOM OF LOOP
86	003255	104650	000000	000733	SWAP: MOV	0(R5),TEMP2	:GET LOW ORDER CURRENT RESIDENT
87	003260	104650	000001	000734	MOV	1(R5),TEMP2+1	:GET HIGH ORDER CURRENT RESIDENT
88	003263	104123			MOV	(R2),R3	:GET LOW ORDER NEW RESIDENT
89	003264	100153			MOV	R3,(R5)	:PUT IN RCT
90	003265	104623	000001		MOV	1(R2),R3	:GET HIGH ORDER NEW RESIDENT
91	003267	107303	001335		SUB	ST.LBN,R3	:SUBTRACT STARTING LBN BITS
92	003271	103203	170000		BIC	#HD.CLR,R3	:CLEAR THE HEADER
93	003273	101203	020000		BIS	#RC.PRIV,R3	:SET AS PRIMARY
94	003275	100653	000001		MOV	R3,1(R5)	:PUT IN RCT
95	003277	023421			CALL	RCTWT	:WRITE OUT PRIMARY BLOCK
96	003300	104202	000733		MOV	#TEMP2,R2	:POINT TO OLD RESIDENT
97	003302	023612			SECNDY: CALL	PNGPG	:FIND RCT ENTRY FOR SECONDARY
98	003303	104204	006621		MOV	#RCTBUF,R4	:POINT TO BUFFER
99	003305	105054			ADD	R5,R4	:ADD OFFSET
100	003306	104123			MOV	(R2),R3	:GET LOW ORDER NEW ENTRY
101	003307	100143			MOV	R3,(R4)	:PUT IN RCT
102	003310	104623	000001		MOV	1(R2),R3	:GET HIGH ORDER NEW ENTRY
103	003312	107303	001335		SUB	ST.LBN,R3	:SUBTRACT STARTING LBN BITS
104	003314	103203	170000		BIC	#HD.CLR,R3	:CLEAR HEADER
105	003316	101203	030000		BIS	#RC.SND,R3	:FLAG AS SECONDARY
106	003320	100643	000001		MOV	R3,1(R4)	:STORE IN RCT
107	003322	023421			BOTTOM: CALL	RCTWT	:WRITE OUT RCT BLOCK
108	003323	105200	000002	001474	ADD	#ERLEN,UPDPNT	:POINT TO NEXT ERROR SECOTR
109	003326	117400	001254		DEC	REVCNT	:DECREMENT IT
110	003330	053101			BNE	ROVER	:NOT DONE - DO NEXT SECTOR
111	003331				POP	R3	:GET HIGH ORDER CYL
112	003332	104030	001126		MOV	R3,CYLNUM+1	:RESTORE IT
113	003334	104030	001077		MOV	R3,ISEEK+2	:PUT IN SEEK COMMAND
114	003336				POP	R3	:GET LOW ORDER

115	003337	104030	001125	MOV	R3,CYLNUM	
116	003341	104030	001076	MOV	R3,ISEEK+1	:PUT IN SEEK COMMAND
117	003343			POP	R3	:GET TRACK NUMBER
118	003344	104030	001112	MOV	R3,CURTRK	:RESTORE IT
119	003346	102200	040000 001217	BIT	#FINI,FLAG	:DO THE SEEK ?
120	003351	053356		BNE	NOSEK	:NOPE
121	003352	104300	001463 001100	MOV	CURGRP,ISEEK+3	:RESTORE GROUP TO SEEK
122	003355	022251		CALL	SEEK	:GET BACK TO RIGHT CYLINDER
123	003356	000000		NOSEK:	RETURN	

```

1
2
3
4 003357 115001
5 003360 073410
6 003361 104207 001052
7 003363 022757
8 003364 104200 000200 000736 UHASH: TST R1 ;NEED TO COMPUTE PRIMARY RBN ?
9 003367 114000 000737 UHASH: BMI UHKIP1 ;NO - SKIP IT
10 003371 104300 001151 000731 UHASH: MOV #SCR,R0 ;POINT TO CUBUNIT CHARACTERISTICS
11 003374 104300 001152 000732 UHASH: CALL PRIMRB ;COMPUTE PRIMARY RBN
12 003377 104204 000731 UHASH: MOV #128,TEMP ;DIVIDE BY 128 TO GET BLOCK NUMBER
13 003401 104203 000736 UHASH: CLR TEMP+1 ;FOR STORE
14 003403 021570 UHASH: MOV REVRBN,DDUMMY ;GET PRIMARY RBN
15 UHASH: MOV REVRBN+1,DDUMMY+1 ;GET HIGH ORDER
16 003404 104131 UHASH: MOV #DDUMMY,R4 ;FOR DIVIDE
17 003405 105011 UHASH: MOV #TEMP,R3 ;DITTO
18 003406 100131 UHASH: CALL DDIV ;DDUMMY=RCT BLOCK NUMBER
19 003407 000000 UHASH: MOV (R3),R1 ;TEMP=OFFSET
20 003410 104140 001151 UHASH: ADD R1,R1 ;GET OFFSET
21 003412 104640 000001 001152 UHASH: MOV R1,(R3) ;MULTIPLY BY 2
22 003415 107300 001336 001152 UHASH: RETURN ;STORE BACK
23 003420 003364 UHASH: UHKIP1: MOV (R4),REVRBN ;FOR DIVIDE SETUP
UHASH: MOV 1(R4),REVRBN+1 ;DITTO
UHASH: SUB ST,RBN,REVRBN+1 ;SUBTRACT STARTING RBN BITS
UHASH: BR UHKIP ;DO DIVIDE

```

1									
2									
3									
4	003421	114005							
5	003422	104050	001245						
6	003424	104204	000731						
7	003426	104203	001052						
8	003430	104632	000011						
9	003432	103202	177400						
10	003434	104207	001430						
11	003436	100672	000004						
12	003440	104632	000001						
13	003442	103202	007777						
14	003444	100672	000001						
15	003446	114002							
16	003447	100672	000000						
17	003451	022714							
18	003452	104207	000721						
19	003454	104203	122400						
20	003456	104302	001112						
21	003460	101023							
22	003461	100673	000004						
23	003463	104202	006621						
24	003465	022627							
25	003466	100623	000400						
26	003470	100672	000001						
27	003472	104143							
28	003473	100673	000002						
29	003475	104643	000001						
30	003477	105303	001335						
31	003501	101203	000000						
32	003503	100673	000003						
33	003505	104203	000726						
34	003507	100673	000005						
35	003511	104303	001333						
36	003513	104304	001334						
37	003515	104302	000740						
38	003517	104207	000721						
39	003521	060012							
40	003522	060003							
41	003523	115001							
42	003524	013546							
43	003525	106300	001502	001504					
44	003530	013534							
45	003531	115400	001504						
46	003533	003511							
47	003534	104303	001505						
48	003536	073545							
49	003537	022612							
50	003540	114000	001504						
51	003542	117400	001505						
52	003544	003511							
53	003545								
54	003545	115405							
55	003546	115400	001245						
56	003550	114000	001504						
57	003552	104300	001503	001505					

WRITE AN RCT BLOCK

```

RCTWT: CLR R5
      MOV R5,NEXT1
      MOV #DDUMMY,R4
RCTWLP: MOV #SCR,R3
      MOV LBNTRK(R3),R2
      BIC #HIBYTE,R2
      MOV #CONBLK,R0
      MOV R2,V3(R0)
      MOV STCYL(R3),R2
      BIC #LO,R2
      MOV R2,V1+1(R0)
      CLR R2
      MOV R2,V1(R0)
      CALL CVTSK
      MOV #WRBLK,R0
      MOV #WRCMD,R3
      MOV CURTRK,R2
      BIS R2,R3
      MOV R3,RW.CMD(R0)
      MOV #RCTBUF,R2
      CALL CEDC
      MOV R3,RW.EDC(R2)
      MOV R2,RW.BUF(R0)
      MOV (R4),R3
      MOV R3,RW.LOW(R0)
      MOV 1(R4),R3
      ADD ST.LBN,R3
      BIS #HD.LBN,R3
      MOV R3,RW.HI(R0)
      MOV #HSLIM-1,R3
      MOV R3,RW.DUM(R0)
WRITE2: MOV HPREA,R3
      MOV DPREA,R4
      MOV UNIT,R2
      MOV #WRBLK,R0
      XFC SIP
      XFC WRITE
      TST R1
      BEQ RWGD
      CMP RETRY,TMPTRY
      BEQ 1$
      INC TMPTRY
      BR WRITE2
1$: MOV RECTMP,R3
      BMI 2$
      CALL ERRHND
      CLR TMPTRY
      DEC RECTMP
      BR WRITE2
2$: INC R5
RWGD: INC NEXT1
      CLR TMPTRY
      MOV RECOV,RECTMP

```

```

;CLEAR ERROR COUNTER
;RESET NEXT COUNTER
;POINT TO BLOCK
;POINT TO CHARACTERISTICS
;GET LBN/TRACK
;CLEAR REST OF WORD
;POINT TO CONVERT BLOCK
;FOR CONVERT
;STARTING CLYLINDER
;CLEAR REST OF WORD
;STORE
;CLEAR FOR STORE
;LOW ORDER ALWAYS 0
;CONVERT AND SEEK
;POINT TO COMMAND BLOCK
;GET WRITE COMMAND
;GET CURRENT TRACK
;SET TRACK FOR WRITE
;STORE IN COMMAND BLOCK
;POINT TO BUFFER
;COMPUTE EDC - RETURNED IN R3
;STORE IT
;STICK IN COMMAND BLOCK
;GET LOW ORDER HEADER
;STORE IN WRITE BLOCK
;GET HIGH ORDER
;ADD STARTING LBN BITS
;SET HEADER
;STORE IN WRITE BLOCK
;GET DUMMY SDI POINTER
;POINT IN COMMAND BLOCK
;GET HEADER PREAMBLE
;GET DATA PREAMBLE
;SET UNIT
;MAKE SURE POINTING AT BLOCK
;WAIT FOR SECTOR PULSE
;WRITE SECTOR
;ANY ERROR ?
;NOPE
;MAX ?
;YES - TRY SOME RECOVERY
;INC RETRY COUNT
;DO RETRY
;GET CURRENT ERROR RECOVERY LEVEL
;IF NEGATIVE THEN FRIED
;TRY RECOVERY
;FOR INIT
;DECREMENT IT
;RETRY
;YUP - INCREMENT COUNTER
;INCREMENT IT
;FOR RESET
;GET RECOVERY LEVELS

```

58	003555	104204	000731		MOV	#DDUMMY,R4	:FOR ADD
59	003557	104203	001242		MOV	#RCTFMT,R3	:FOR ADD
60	003561	021506			CALL	DADD	:POINT TO NEXT COPY
61	003562	106300	001244	001245	CMP	FCTCPY,NEXT1	:DONE THIS SECTOR ?
62	003565	053426			BNE	RCTWLP	:NO - WRITE NEXT FCT COPY
63	003566	106305	001244		CMP	FCTCPY,R5	:ERROR ON EVERY WRITE ?
64	003570	013606			BEQ	RCTERR	:YUP - BIG TROUBLE
65	003571	104303	001245	RCXLP:	MOV	NEXT1,R3	:ANY REPEATS ?
66	003573	013604			BEQ	RTDON	:NO
67	003574	104204	000731		MOV	#DDUMMY,R4	:TO GET IT BACK
68	003576	104203	001242		MOV	#RCTFMT,R3	:DITTO
69	003600	021524			CALL	DSUB	
70	003601	117400	001245		DEC	NEXT1	:SUB IT
71	003603	003571			BR	RCXLP	:REPEAT
72	003604	060022		RTDON:	XFC	UPDATE	:LET HOST KNOW STILL ALIVE
73	003605	000000			RETURN		
74	003606	104012		RCTERR:	MOV	R1,R2	:XFC ERROR CODE
75	003607	104201	000017		MOV	#15.,R1	:RCT WRITE ERROR
76	003611	022552			CALL	ERRMNT	:ERROR QUIT

```

1
2
3
4 003612 114000 001222
5 003613 114002
6 003615 104303 000736
7 003616 104035
8 003620 105025
9 003621 102205 000400
10 003622 053637
11 003625 104651 006622
12 003627 103201 007777
13 003631 106201 100000
14 003633 013675
15 003634 106201 000000
16 003636 013707
17 003637 104025
18 003640 107052
19 003641 073645
20 003642 105202 000002
21 003643 106202 000400
22 003645 053620
23
24
25
26
27
28
29
30 003650 115400 001477
31 003652 104303 001477
32 003654 104204 000731
33 003656 100143
34 003657 114003
35 003660 100643 000001
36 003662 104030 000736
37 003664 104203 001227
38 003666 021506
39 003667 104040 001253
40 003671 104201 000055
41 003673 022444
42 003674 003615
43 003675 104303 001222
44 003677 053711
45 003700 104200 000002 001477
46 003703 104200 000002 001222
47 003706 003652
48
49
50 003707
51 003710 000000
52 003711 104201 000020
53 003713 114002
54 003714 022552

:
: SEARCH FOR OPEN ENTRY IN RCT
:
: PINGPG: PUSH R2 ;SAVE R2
: CLR WRFLG ;CLEAR WRAP FLAG
: XNGBLK: CLR R2 ;FOR FLOP SET
: MOV OFFSET,R3 ;GET OFFSET
: XAGAIN: MOV R3,R5 ;MOV OFFSET INTO BUFF POINTER
: ADD R2,R5 ;ADD FLOP VALUE
: BIT #BIT8,R5 ;PAST ONE END (OR BOTH)
: BNE XFLIP ;YUP - FLIP OTHER DIRECTION
: MOV RCTBUF+1(R5),R1 ;GET HEADER CODE
: BIC #LO,R1 ;CLEAR LOW ORDER
: CMP #RC,NUL,R1 ;END OF RCT ?
: BEQ XEORCT ;YUP - WRAP TO FIRST BLOCK
: CMP #RC,FRE,R1 ;FREE ?
: BEQ XPRET ;YUP - ALL DONE
: XFLIP: MOV R2,R5 ;GET FLIP VALUE
: CLR R2
: SUB R5,R2 ;NEGATE IT
: BMI XNOINC ;IF NEGATIVE DON'T INC
: ADD #2,R2 ;ADD TO NEXT VALUE
: XNOINC: CMP #256,R2 ;DONE EVERY SLOT IN BLOCK ?
: BNE XAGAIN ;NOPE - TRY NEXT ONE

:
: IN THIS SECTION THE BLOCKS ARE PING-PONGED BUT
: THE SEARCH WITHIN BLOCKS IS LINEAR FROM HIGHEST BUFFER
: ADDRESS TO LOWEST
:
:
: XPNGRD: INC RCTCNT ;INC TO NEXT ONE
: MOV RCTCNT,R3 ;FOR STORE
: MOV #DDUMMY,R4 ;FOR ADD
: MOV R3,(R4) ;STORE BLOCK NUMBER
: CLR R3 ;FOR RESETS
: MOV R3,1(R4) ;CLEAR HIGH WORD
: MOV R3,OFFSET ;MAKE OFFSET AT BEGINNING
: MOV #HOLD,R3 ;POINT TO FIRST RCT LBN
: CALL DADD ;GET LBN OF THIS RCT BLOCK
: MOV R4,BUFFPNT ;STORE BLOCK NUMBER
: MOV #H1,R1 ;READ RCT OVERLAY
: CALL PAGE ;DO IT
: BR XNGBLK ;SEARCH THIS BLOCK
: XEORCT: MOV WRFLG,R3 ;GET WRAP FLAG
: BNE XPERR ;IF BEEN HERE ONCE THEN RCT FULL
: MOV #2,RCTCNT ;FOR FIRST RCT BLOCK
: MOV #2,WRFLG ;MAKE WRAP FLAG NON-ZERO
: BR XPNGRD ;READ IT AND CONTINUE

:
: XPRET: POP R2 ;RESTORE R2
: RETURN ;SUCCESSFUL RETURN
: XPERR: MOV #16,R1 ;RCT FULL
: CLR R2 ;NO SUBCODE
: CALL ERRMNT ;ERROR RETURN

```

Address	Instruction	Comments
003715	SBTTL RCT READ OVERLAY (H1)	
	DMOVL H1, START	
	READ A BLOCK OF THE RCT	
003047	PUSHA	
003055	MOV #H1, CUROVL	:FOR INIT
003060	MOV BUFPNT, R4	:GET POINTER TO BLOCK NUMBER
003062	CLR R5	:CLEAR ERROR COUNTER
003063	MOV #SCR, R3	:POINT TO CHARACTERISTICS
003065	MOV STCYL(R3), R2	:GET STARTING CYLINDER
003067	BIC #LO, R2	:CLEAR REST OF WORD
003071	MOV #CONBLK, R0	:POINT TO CONVERT BLOCK
003073	MOV R2, V1+1(R0)	:STORE FOR CONVERT
003075	CLR R2	:FOR STORE
003076	MOV R2, V1(R0)	:LOW ORDER ALWAYS 0
003100	MOV LBNTRK(R3), R2	:GET LBN/TRK
003102	BIC #HIBYTE, R2	:CLEAR HIGH BYTE
003104	MOV R2, V3(R0)	:STORE IN CONVERT BLOCK
003106	OCLOOP: CALL CVTSK	:CONVERT RCT BLOCK NUMBER AND SEEK
003107	MOV #RDBLK, R0	:PREPARE FOR READ SECTORS
003111	MOV #HSLIM-1, R3	:POINTER TO DUMMY SDI BLOCK
003113	MOV R3, RW.DUM(R0)	:STORE IN COMMAND BLOCK
003115	MOV (R4), R3	:LO ORDER BLOCK NUMBER
003116	MOV R3, RW.LOW(R0)	:STORE IN READ CMD BLOCK
003120	MOV 1(R4), R3	:HI ORDER BLOCK NUM AND CODE
003122	ADD ST.LBN, R3	:ADD STARTING LBN BITS
003124	MOV R3, RW.HI(R0)	:STORE IN READ CMD BLOCK
003126	MOV #RCTBUF, R3	:LOAD ADDRESS OF DATA BUFFER
003130	MOV R3, RW.BUF(R0)	:STORE IN COMMAND BUFFER
003132	MOV #RWCMD, R3	:LOAD SDI READ COMMAND
003134	MOV CURTRK, R1	:GET CURRENT HEAD NUMBER IN R1
003136	BIS R1, R3	:SET IT IN COMMAND
003137	MOV R3, RW.CMD(R0)	:STORE BACK
003141	MOV #RDBLK, R0	:MAKE SURE POINTING AT BLOCK
003143	MOV #RDCMD, R3	:MARK AS ONLY REQUEST
003145	MOV R3, (R0)	:STORE IN CMD BLOCK
003146	MOV UNIT, R2	:SDI INTERCONNECT
003150	BIS #BIT15, R0	:SET NO REVECTORING
003152	XFC SIP	:WAIT FOR PULSE
003153	XFC READ	:READ 1 SECTOR
003154	TST R1	:ANY ERRORS ?
003155	BNE 100\$	:YES - TRY RECOVERY
003156	MOV (R0), R3	:GET STATUS WORD
003157	BIT #ECCF, R3	:ECC ERROR ?
003161	BEQ 101\$	:NOPE - VERIFY EDC
003162	CALL ECCCK	:CORRECT ECC
003163	TST R1	:TEST FLAG
003164	BNE 100\$	:UNCORRECTABLE
003165	MOV #RCTBUF, R2	:POINT TO BUFFER
003167	CALL CEDC	:COMPUTE EDC
003170	CMP RW.EDC(R2), R3	:O.K. ?
003172	BEQ 102\$	:YUP - CONSIDER GOOD
003173	CMP RETRY, TMPTRY	:MAX ?
003176	BEQ 1\$	:YES - TRY SOME RECOVERY
003177	INC TMPTRY	:INC RETRY COUNT

58	003201	003141	-		BR	READ11	:DO RETRY
59	003202	104303	001505	1\$:	MOV	RECTMP,R3	:GET CURRENT ERROR RECOVERY LEVEL
60	003204	073213			BMI	2\$	:IF NEGATIVE THEN FRIED
61	003205	022612			CALL	ERRHND	:TRY RECOVERY
62	003206	114000	001504		CLR	TMPTRY	:FOR INIT
63	003210	117400	001505		DEC	RECTMP	:DECREMENT IT
64	003212	003141			BR	READ11	:RETRY
65	003213			2\$:			
66	003213	115405			INC	R5	:INCREMENT BAD COUNTER
67	003214	106305	001244		CMP	FCTCPY,R5	:ALL BAD ?
68	003216	013344			BEQ	ORFAL	:YUP - ALL OVER
69	003217	104203	001242		MOV	#RCTFMT,R3	:SIZE OF TABLE - R4 -> BLOCK NUMBER
70	003221	021506			CALL	DADD	:ADD TO POINT TO NEXT COPY
71	003222	114000	001504		CLR	TMPTRY	:RESET RETRY COUNT
72	003224	104300	001503	001505	MOV	RECOV,RECTMP	:DITTO RECOVERY LEVELS
73	003227	003106			BR	CCLOOP	:BRANCH BACK
74	003230			102\$:			
75	003230	114000	001504		CLR	TMPTRY	:FOR RESET
76	003232	104300	001503	001505	MOV	RECOV,RECTMP	:GET RECOVERY LEVELS
77	003235	115005			TST	R5	:ANY ERRORS ?
78	003236	013335			BEQ	RLDONE	:NO - EXIT
79	003237	104203	001240		MOV	#FCTFMT,R3	:SIZE OF TABLE
80	003241	021524			CALL	DSUB	:GET BACK TO PREVIOUS COPY
81	003242	022714			CALL	CVTSK	:CONVERT AND SEEK
82	003243	104207	000721		MOV	#WRBLK,R0	:POINT TO COMMAND BLOCK
83	003245	104203	122400		MOV	#WRCMD,R3	:GET WRITE COMMAND
84	003247	104302	001112		MOV	CURTRK,R2	:GET CURRENT TRACK
85	003251	101023			BIS	R2,R3	:SET TRACK FOR WRITE
86	003252	100673	000004		MOV	R3,RW.CMD(R0)	:STORE IN COMMAND BLOCK
87	003254	104203	006621		MOV	#RCTBUF,R3	:POINT TO BUFFER
88	003256	100673	000001		MOV	R3,RW.BUF(R0)	:STICK IN COMMAND BLOCK
89	003260	104143			MOV	(R4),R3	:GET LOW ORDER HEADER
90	003261	100673	000002		MOV	R3,RW.LOW(R0)	:STORE IN WRITE BLOCK
91	003263	104643	000001		MOV	1(R4),R3	:GET HIGH ORDER
92	003265	105303	001335		ADD	ST.LBN,R3	:ADD STARTING LBN BITS
93	003267	100673	000003		MOV	R3,RW.HI(R0)	:STORE IN WRITE BLOCK
94	003271	104203	000726		MOV	#HSLIM-1,R3	:GET DUMMY SDI POINTER
95	003273	100673	000005		MOV	R3,RW.DUM(R0)	:POINT IN COMMAND BLOCK
96	003275	104303	001333		MOV	HPREA,R3	:GET HEADER PREAMBLE
97	003277	104304	001334		MOV	DPREA,R4	:GET DATA PREAMBLE
98	003301	104302	000740		MOV	UNIT,R2	:SET UNIT
99	003303	104207	000721		MOV	#WRBLK,R0	:MAKE SURE POINTING AT BLOCK
100	003305	101207	100000		BIS	#BIT15,R0	:SET NO REVECTORING
101	003307	060012			XFC	SIP	:WAIT FOR SECTOR PULSE
102	003310	060003			XFC	WRITE	:WRITE SECTOR
103	003311	115001			TST	R1	:ANY ERROR ?
104	003312	013333			BEQ	2\$	:NO - SKIP RETRY
105	003313	106300	001502	001504	CMP	RETRY,TMPTRY	:MAX ?
106	003316	013322			BEQ	1\$	:YES - TRY SOME RECOVERY
107	003317	115400	001504		INC	TMPTRY	:INC RETRY COUNT
108	003321	003275			BR	WRIT12	:DO RETRY
109	003322	104303	001505	1\$:	MOV	RECTMP,R3	:GET CURRENT ERROR RECOVERY LEVEL
110	003324	073333			BMI	2\$	:IF NEGATIVE THEN FRIED
111	003325	022612			CALL	ERRHND	:TRY RECOVERY
112	003326	114000	001504		CLR	TMPTRY	:FOR INIT
113	003330	117400	001505		DEC	RECTMP	:DECREMENT IT
114	003332	003275			BR	WRIT12	:RETRY

115 003333
116 003333 117405
117 003334 003230
118 003335
119 003343 000000
120 003344 104012
121 003345 104201 000016
122 003347 022552

2\$:
DEC R5
BR OCDONE
RLDONE: POPA
RETURN
ORFTAL: MOV R1,R2
MOV #14.,R1
CALL ERMNT

:DEREMENT COUNTER
:SEE IF ANY MORE TO DO

:ALL DONE
:XFC ERROR CODE
:RCT READ ERROR
:ERROR RETURN

```
1  
2  
3  
4  
5  
6  
7 003350  
8  
9  
10  
11 003047 003212
```

```
      .SBTTL  FCT->RCT CONVERSION OVERLAY (F5)  
      CONVERT FCT INTO RCT  
  
      DMOVLY  F5,START  
  
      JMP     START2      ; SKIP SUBROUTINES
```



```

1
2
3
4 003112 114000 001222
5 003114 114002
6 003115 104303 000736
7 003117 104035
8 003120 105025
9 003121 102205 000400
10 003123 053136
11 003124 104651 006622
12 003126 103201 007777
13 003130 106201 100000
14 003132 013173
15 003133 106201 000000
16 003135 013205
17 003136 104025
18 003137 114002
19 003140 107052
20 003141 073144
21 003142 105202 000002
22 003144 106202 000400
23 003146 053117
24
25
26
27
28
29 003147 115400 001477
30 003151 104303 001477
31 003153 104204 000731
32 003155 100143
33 003156 114000 000732
34 003160 114000 000736
35 003162 104203 001227
36 003164 021506
37 003165 104040 001253
38 003167 104201 000055
39 003171 022444
40 003172 003114
41 003173 104303 001222
42 003175 053206
43 003176 104200 000002 001477
44 003201 104200 000002 001222
45 003204 003151
46
47
48 003205 000000
49
50 003206 104201 000020
51 003210 114002
52 003211 022552

:
: SEARCH FOR OPEN ENTRY IN RCT
:
: PNGPNG: CLR WRFLG ;CLEAR WRAP FLAG
: PNGBLK: CLR R2 ;FOR FLOP SET
: MOV OFFSET,R3 ;GET OFFSET
: PAGAIN: MOV R3,R5 ;MOV OFFSET INTO BUFF PCINTER
: ADD R2,R5 ;ADD FLOP VALUE
: BIT #BIT8,R5 ;PAST ONE END (OR BOTH)
: BNE FLIP ;YUP - FLIP OTHER DIRECTION
: MOV RCTBUF+1(R5),R1 ;GET HEADER CODE
: BIC #LO,R1 ;CLEAR LOW ORDER STUFF
: CMP #RC.NUL,R1 ;END OF RCT ?
: BEQ EORCT ;YUP - WRAP TO FIRST BLOCK
: CMP #RC.FRE,R1 ;FREE ?
: BEQ PRET ;YUP - ALL DONE
: FLIP: MOV R2,R5 ;GET FLIP VALUE
: CLR R2
: SUB R5,R2 ;NEGATE IT
: BMI NOINC ;IF NEGATIVE DON'T INC
: ADD #2,R2 ;ADD TO NEXT VALUE
: NOINC: CMP #256,R2 ;DONE EVERY SLOT IN BLOCK ?
: BNE PAGAIN ;NOPE - TRY NEXT ONE

:
: IN THIS SECTION THE BLOCKS ARE PING-PONGED BUT
: THE SEARCH WITHIN BLOCKS IS LINEAR FROM HIGHEST BUFFER
: ADDRESS TO LOWEST
:
:
: PNGRD: INC RCTCNT ;INC TO NEXT ONE
: MOV RCTCNT,R3 ;FOR STORE
: MOV #DDUMMY,R4 ;FOR ADD
: MOV R3,(R4) ;STORE BLOCK NUMBER
: CLR DDUMMY+1 ;FOR RESETS
: CLR OFFSET ;MAKE IT AT ZERO
: MOV #HOLD,R3 ;POINT TO FIRST RCT LBN
: CALL DADD ;GET LBN OF THIS RCT BLOCK
: MOV R4,BUFFPNT ;STORE POINTER TO BLOCK NUMBER
: MOV #H1,R1 ;RCT READ OVERLAY
: CALL PAGE ;DO IT
: BR PNGBLK ;SEARCH THIS BLOCK
: EORCT: MOV WRFLG,R3 ;GET WRAP FLAG
: BNE PERR ;IF BEEN HERE ONCE THEN RCT FULL
: MOV #2,RCTCNT ;FOR FIRST RCT BLOCK
: MOV #2,WRFLG ;MAKE WRAP FLAG NON-ZERO
: BR PNGRD ;READ IT AND CONTINUE

:
: PRET: RETURN ;SUCCESSFUL RETURN
:
: PERR: MOV #16,R1 ;R5=OFFSET
: CLR R2 ;RCT FULL
: CALL ERRMNT ;NO SUBCODE
: ;ERROR RETURN

```

1									
2									
3									
4									
5									
6									
7	003212				START2:	MOV	#F5,CURVOL		:GET OVERLAY NUMBER
8	003212	104200	000014	001153	FCTRCT:	MOV	CURTRK,R3		:GET CURRENT TRACK
9	003215	104303	001112			PUSH	R3		:SAVE IT
10	003217					MOV	CYLNUM,R3		:GET LOW ORDER CYLINDR
11	003220	104303	001125			PUSH	R3		:SAVE FOR RESTORE
12	003222					MOV	CYLNUM+1,R3		:GET HIGH ORDER
13	003223	104303	001126			PUSH	R3		:SAVE FOR RESTORE
14	003225					MOV	#FCTCNT,R4		:FOR SUB
15	003226	104204	001255			MOV	#ONE,R3		:DITTO
16	003230	104203	001256			CALL	DSUB		:SUB TO GET CURRENT FT BLOCK NUM
17	003232	021524				MOV	(R4),R3		:GET IT
18	003233	104143				PUSH	R3		:FOR LATER RESTORE
19	003234					CLR	FCTCNT		:CLEAR FOR INIT
20	003235	114000	001256			MOV	#RDBUF,BUFPNT		:POINT TO BUFFER
21	003237	104200	004535	001253		MOV	#F6,R1		:READ A BLOCK OF THE FCT
22	003242	104201	000017			CALL	PAGE		:EXECUTE IT
23	003244	022444				MOV	#RDBUF,R4		:MAKE SURE POINT TO IT
24	003245	104204	004535			MOV	C512(R4),MNCNT		:GET COUNT OF USED ENTRIES
25	003247	104640	000016	001262		MOV	#128,SECCNT		:ENTRIES IN A FCT SECTOR
26	003252	104200	000200	001454		MOV	LBNLBN,HOLD		:GET LOW ORDER COUNT OF LBN'S
27	003255	104300	001133	001227		MOV	LBNLBN+1,HOLD+1		:GET HIGH ORDER
28	003260	104300	001134	001230		MOV	#TOTRCT,R3		:FOR SUBTRACT
29	003263	104203	001475			MOV	#HOLD,R4		:DITTO
30	003265	104204	001227			CALL	DSUB		:GET STARTING RCT LBN
31	003267	021524				MOV	#F7,R1		:RCT INIT OVERLAY
32	003270	104201	000022			CALL	PAGE		:INIT RCT
33	003272	022444				MOV	MNCNT,R3		:GET COUNT
34	003273	104303	001262			BEQ	FCTSP		:QUIT IF NO ENTRIES
35	003275	013762				MOV	#RDBUF,BUFPNT		:POINT TO BUFFER
36	003276	104200	004535	001253	FBEGIN:	MOV	#F6,R1		:FCT READ OVERLAY
37	003301	104201	000017			CALL	PAGE		:DO IT
38	003303	022444				MOV	#RDBUF,FCTPTR		:MAKE SURE POINT TO IT
39	003304	104200	004535	001255		MOV	FCTPTR,R4		:FOR USE
40	003307	104304	001255			MOV	(R4),CURPBN		:GET LOW ORDER PBN
41	003311	104140	001110		FBEG2:	MOV	1(R4),CURPBN+1		:GET HIGH ORDER
42	003313	104640	000001	001111		BIC	#HD.CLR,CURPBN+1		:CLEAR THE FCT HEADER CODE
43	003316	103200	170000	001111		MOV	#CURPBN,R3		:FOR COMAPRE
44	003321	104203	001110			MOV	#SCR+LBNHOST,R4		:DITTO
45	003323	104204	001064			CALL	DCMP		:IN RCT ??
46	003325	021626				BPL	XYZ1		:YUP
47	003326	033720				MOV	#G5,R1		:OVERLAY TO CONVERT FORM PBN TO OTHER BN
48	003327	104201	000044			CALL	PAGE		:EXECUTE IT
49	003331	022444				MOV	CURBN+1,R3		:GET HIGH ORDER CONVERTED BLOCK NUM
50	003332	104303	001114			BIC	#LO,R3		:CLEAR ALL BUT HEADER
51	003334	103203	007777			CMP	#HD.LBN,R3		:IS IT AN LBN ?
52	003336	106203	000000			BNE	NOLBN		:NO - SKIP DOWN
53	003340	053510				MOV	#CURBN,R4		:POINT TO BLOCK NUMBER
54	003341	104204	001113			CALL	HASH		:COMPUTE RCT ENTRY
55	003343	023050				MOV	(R4),R3		:GET RCT BLOCK
56	003344	104143				ADD	#2,R3		:ADD TO GET BY FIRST 2 BLOCKS
57	003345	105203	000002						

58	003347	100143		MOV	R3,(R4)	:STORE BACK
59	003350	104030	001477	MOV	R3,RCTCNT	:SAVE FOR LATER PNGPNG
60	003352	104203	001227	MOV	#HOLD,R3	:FOR ADD
61	003354	021506		CALL	DADD	:TO GET LBN OF RCT BLOCK
62	003355	104040	001253	MOV	R4,BUFPT	:STORE POINTER TO BLOCK NUMBER
63	003357	104201	000055	MOV	#H1,R1	:RCT READ OVERLAY
64	003361	022444		CALL	PAGE	:EXECUTE IT
65	003362	104205	006621	MOV	#RCTBUF,R5	:POINT TO BUFFER
66	003364	104303	000736	MOV	OFFSET,R3	:GET OFFSET
67	003366	105035		ADD	R3,R5	:POINT TO ENTRY
68	003367	104653	000001	MOV	1(R5),R3	:GET HIGH ORDER
69	003371	103203	007777	BIC	#LO,R3	:CLEAR ALL BUT HEADER
70	003373	106203	000000	CMP	#RC.FRE,R3	:IS IT FREE ?
71	003375	013666		BEQ	FILLIT	:YES - FILL IT
72	003376	106203	040000	CMP	#RC.UNU,R3	:UNUSABLE RBN ?
73	003400	013645		BEQ	BADRBN	:YES - MUST BE SECONDARY
74	003401	104150	000733	MOV	(R5),TEMP2	:ELSE SWITCH
75	003403	104650	000001	MOV	1(R5),TEMP2+1	:HIGH ORDER
76	003406	104303	001113	MOV	CURBN,R3	:GET NEW RESIDENT LOW ORDER
77	003410	100153		MOV	R3,(R5)	:STORE IN RCT
78	003411	104303	001114	MOV	CURBN+1,R3	:GET HIGH ORDER
79	003413	107303	001335	SUB	ST.LBN,R3	:SUBTRACT STARTING LBN BITS
80	003415	103203	170000	BIC	#HD.CLR,R3	:CLEAR THE HEADER
81	003417	101203	020000	BIS	#RC.PRIV,R3	:MARK AS PRIMARY
82	003421	100653	000001	MOV	R3,1(R5)	:STORE IT
83	003423	102200	000400	BIT	#DLL,FLAG	:DID WE CREATE THE FCT ?
84	003426	013466		BEQ	FCTSKP	:NO - THEN DON'T CHANGE IT
85	003427	104302	001255	MOV	FCTPTR,R2	:GET POINTER TO CURRENT FCT BLOCK POS
86	003431	104623	000001	MOV	1(R2),R3	:GET HIGH ORDER FCT ENTRY
87	003433	101203	100000	BIS	#PRMY,R3	:MAKE IT SECONDARY
88	003435	100623	000001	MOV	R3,1(R2)	:STORE IT BACK
89	003437	106202	004535	CMP	#RDBUF,R2	:IS THIS THE FIRST ENTRY IN THE BLOCK
90	003441	053460		BNE	FCTSK1	:NOPE WE'RE SAFE
91	003442	104300	000731	MOV	DDUMMY,CURXBN	:SAVE RCT BLOCK NUMBER
92	003445	104300	000732	MOV	DDUMMY+1,CURXBN+1	:DITTO
93	003450	024015		CALL	FIXFCT	:YUP - GOT SOME GYRATIONS TO DO
94	003451	104300	001115	MOV	CURXBN,DDUMMY	:RESTORE RCT BLOCK NUMBER
95	003454	104300	001116	MOV	CURXBN+1,DDUMMY+1	:DITTO
96	003457	003466		BR	FCTSKP	:THEN CONTINUE ON
97	003460	107202	000001	FCTSK1: SUB	#1,R2	:POINT BACK ONE
98	003462	104123		MOV	(R2),R3	:GET HIGH ORDER
99	003463	103203	100000	BIC	#PRMY,R3	:CLEAR PRIMARY IF SET
100	003465	100123		MOV	R3,(R2)	:STORE IT
101	003466	024056		FCTSKP: CALL	RCTWRT	:WRITE OUT CHANGED BLOCK
102	003467	023112		CALL	PNGPNG	:FIND IT A NEW HOME
103	003470	104204	006621	MOV	#RCTBUF,R4	:POINT TO BUFFER
104	003472	105054		ADD	R5,R4	:ADD OFFSET
105	003473	104202	000733	MOV	#TEMP2,R2	:POINT TO OLD RESIDENT
106	003475	104123		MOV	(R2),R3	:GET LOW ORDER
107	003476	100143		MOV	R3,(R4)	:PUT IT IN
108	003477	104623	000001	MOV	1(R2),R3	:GET HIGH ORDER
109	003501	103203	170000	BIC	#HD.CLR,R3	:CLEAR HEADER
110	003503	101203	030000	BIS	#RC.SND,R3	:MARK AS SECONDARY
111	003505	100643	000001	MOV	R3,1(R4)	:STORE IT
112	003507	003717		BR	XYZ	:SKIP TO END
113	003510	106203	060000	NOLBN: CMP	#HD.RBN,R3	:BAD RBN ?
114	003512	053717		BNE	XYZ	:NO - THEN DON'T CARE ABOUT IT

115	003513	104303	001114		MOV	CURBN+1,R3	:GET HEADER	
116	003515	103203	170000		BIC	#HD.CLR,R3	:CLEAR IT	
117	003517	104030	001114		MOV	R3,CURBN+1	:STORE IT BACK	
118	003521	104201	177777		MOV	#-1,R1	:SIGNAL RCT BLOCK	
119	003523	104204	001113		MOV	#CURBN,R4	:POINT TO BLOCK NUMBER	
120	003525	023050			CALL	HASH	:GET RCT BLOCK AND OFFSET	
121	003526	104143			MOV	(R4),R3	:GET RCT BLOCK	
122	003527	105203	000002		ADD	#2,R3	:ADD TO GET BY 2 BLOCKS	
123	003531	100143			MOV	R3,(R4)	:STORE BACK	
124	003532	104030	001477		MOV	R3,RCTCN	:SAVE FOR LATER PNGPNG	
125	003534	104203	001227		MOV	#HOLD,R3	:FOR ADD	
126	003536	021506			CALL	DADD	:TO GET LBN OF RCT BLOCK	
127	003537	104040	001253		MOV	R4,BUFPT	:STORE POINTER TO BLOCK NUMBER	
128	003541	104201	000055		MOV	#H1,R1	:RCT READ OVERLAY	
129	003543	022444			CALL	PAGE	:DO IT	
130	003544	104205	006621		MOV	#RCTBUF,R5	:POINT TO BLOCK	
131	003546	104304	000736		MOV	OFFSET,R4	:GET OFFSET	
132	003550	105045			ADD	R4,R5	:POINT TO ENTRY	
133	003551	104653	000001		MOV	1(R5),R3	:GET HIGH ORDER	
134	003553	103203	007777		BIC	#LO,R3	:CLEAR ALL BUT HEADER	
135	003555	106203	000000		CMF	#RC.FRE,R3	:IS IT FREE ?	
136	003557	053567			BNE	RRPL	:NO - RELOCATE CURRENT RESIDENT	
137	003560	103203	170000		BIC	#HD.CLR,R3	:CLEAR THE HEADER	
138	003562	101203	040000		BIS	#RC.UNU,R3	:MARK AS UNUSABLE	
139	003564	100653	000001		MOV	R3,1(R5)	:STORE IT BACK	
140	003566	003717			BR	XYZ	:BRANCH TO THE END	
141	003567	104650	000000	000733	RRPL:	MOV	C(R5),TEMP2	:GET LOW ORDER CURRENT RESIDENT
142	003572	104650	000001	000734		MOV	1(R5),TEMP2+1	:GET HIGH ORDER
143	003575	103203	170000		BIC	#HD.CLR,R3	:CLEAR HEADER	
144	003577	101203	040000		BIS	#RC.UNU,R3	:MARK AS UNUSABLE	
145	003601	103203	007777		BIC	#LO,R3	:CLEAR LOW ORDER	
146	003603	100653	000001		MOV	R3,1(R5)	:STORE IT BACK	
147	003605	114003			CLR	R3	:CLEAR FOR STORE	
148	003606	100153			MOV	R3,(R5)	:CLEAR LOW ORDER	
149	003607	024056			CALL	RCTWRT	:WRITE UT BLOCK	
150	003610	023112			CALL	PNGPNG	:FIND IT A NEW HOME	
151	003611	104204	006621		MOV	#RCTBUF,R4	:POINT TO BUFFER	
152	003613	105054			ADD	R5,R4	:POINT TO ENTRY	
153	003614	104202	000733		MOV	#TEMP2,R2	:POINT TO OLD RESIDENT	
154	003616	104123			MOV	(R2),R3	:GET LOW ORDER	
155	003617	100143			MOV	R3,(R4)	:PUT IT IN	
156	003620	104623	000001		MOV	1(R2),R3	:GET HIGH ORDER	
157	003622	103203	170000		BIC	#HD.CLR,R3	:CLEAR HEADER	
158	003624	101203	030000		BIS	#RC.SND,R3	:MARK AS SECONDARY	
159	003626	100643	000001		MOV	R3,1(R4)	:STORE IT	
160	003630	102200	000400	001217	BIT	#DLL.FLAG	:DID WE CREATE THE FCT ?	
161	003633	013644			BEQ	FCTSLP	:NO - THEN DON'T CHANGE IT	
162	003634	104302	001255		MOV	FCTPTR,R2	:GET FCT PPOINTER	
163	003636	107202	000001		SUB	#1,R2	:POINT BACK ONE	
164	003640	104123			MOV	(R2),R3	:GET HIGH ORDER	
165	003641	103203	100000		BIC	#PRMY,R3	:CLEAR PRIMARY IF SET	
166	003643	100123			MOV	R3,(R2)	:STORE IT	
167	003644	003717			BR	XYZ	:GO TO END	
168	003645	023112			BADRBN:	CALL	PNGPNG	:FIND A NEW SLOT
169	003646	104204	006621		MOV	#RCTBUF,R4	:POINT TO BUFFER	
170	003650	105054			ADD	R5,R4	:POINT TO ENTRY	
171	003651	104202	001113		MOV	#CURBN,R2	:POINT TO OLD RESIDENT	

172	003653	104123		MOV	(R2),R3	:GET LOW ORDER
173	003654	100143		MOV	R3,(R4)	:PUT IT IN
174	003655	104623	000001	MOV	1(R2),R3	:GET HIGH ORDER
175	003657	103203	170000	BIC	#HD.CLR,R3	:CLEAR HEADER
176	003661	101203	030000	BIS	#RC.SND,R3	:MARK AS SECONDARY
177	003663	100643	000001	MOV	R3,1(R4)	:STORE IT
178	003665	003717		BR	XYZ	:GO TO END
179	003666	104303	001113	FILLIT: MOV	CURBN,R3	:GET LOW ORDER BN
180	003670	100153		MOV	R3,(R5)	:PUT IN RCT
181	003671	104303	001114	MOV	CURBN+1,R3	:GET HIGH ORDER AND HEADER
182	003673	107303	001335	SUB	ST.LBN,R3	:SUBTRACT STARTING LBN BITS
183	003675	103203	170000	BIC	#HD.CLR,R3	:CLEAR HEADER
184	003677	101203	020000	BIS	#RC.PRIV,R3	:MARK AS PRIMARY
185	003701	100653	000001	MOV	R3,1(R5)	:STORE IN RCT
186	003703	102200	000400	001217	BIT	#DLL.FLAG
187	003706	013717		BEQ	XYZ	:DID WE CREATE THE FCT ?
188	003707	104303	001255	MOV	FCTPTR,R3	:NO - THEN DON'T CHANGE IT
189	003711	104634	000001	MOV	1(R3),R4	:GET POINTER TO FCT ENTRY
190	003713	101204	100000	BIS	#PRIV,R4	:GET HIGH ORDER
191	003715	100634	000001	MOV	R4,1(R3)	:SET AS PRIMARY
192	003717	024056		XYZ: CALL	RCTWRT	:STORE BACK
193	003720	105200	000002	001255	XYZ1: ADD	:WRITE OUT BUFFER
194	003723	117400	001262	DEC	#2,FCTPTR	:POINT TO NEXT ENTRY
195	003725	013750		BEQ	MNCNT	:DEREMENT IT
196	003726	117400	001454	DEC	FRDONE	:IF ZERO THEN DONE
197	003730	053744		DEC	SECCN	:DECREMENT IT
198	003731	102200	000400	001217	BNE	:IF STILL IN BLOCK - CONTINUE
199	003734	013276		BIT	#DLL.FLAG	:DID WE CREATE THE FCT ?
200	003735	104200	004535	001253	BEQ	:NO - THEN DON'T CHANGE IT
201	003740	104201	000030	MOV	FBEGIN	:POINT TO BUFFER
202	003742	022444		MOV	#RDBUF,BUFPNT	:FCT WRITE OVERLAY
203	003743	003276		CALL	#F9,R1	:EXECUTE IT
204	003744	104304	001255	BR	PAGE	:AND GET A NEW ONE
205	003746	060022		FRSKP: MOV	FBEGIN	:ELSE GET CURRENT POINTER
206	003747	003311		XFC	FCTPTR,R4	:LET HOST KNOW STILL ALIVE
207	003750	102200	000400	001217	UPDATE	:AND DO NEXT ENTRY
208	003753	013762		FRDONE: BIT	FBEG2	:DID WE CREATE THE FCT ?
209	003754	104200	004535	001253	#DLL.FLAG	:NO - THEN DON'T CHANGE IT
210	003757	104201	000030	MOV	FCTSP	:POINT TO BUFFER
211	003761	022444		MOV	#RDBUF,BUFPNT	:WRITE LAST FCT BLOCK
212	003762			CALL	#F9,R1	:DO IT
213	003763	104030	001256	FCTSP: POP	PAGE	:GET CURRENT FCT BLOCK NUM
214	003765	104200	005152	001253	R3	:RESTORE IT
215	003770	104201	000017	MOV	R3,FCTCNT	:RE-READ IT IN CASE OF HEADER CHANGES
216	003772	022444		MOV	#PBNBUF,BUFPNT	:FCT READ OVERLAY
217	003773			MOV	#F6,R1	:DO IT
218	003774	104030	001126	CALL	PAGE	:GET HIGH ORDER CYLINDER
219	003776	104030	001077	POP	R3	:STORE IT
220	004000			MOV	R3,CYLNUM+1	:STORE IN SEEK COMMAND
221	004001	104030	001125	POP	R3,ISEEK+2	:GET LOW ORDER
222	004003	104030	001076	MOV	R3	:RESTORE IT
223	004005			MOV	R3,CYLNUM	:STORE IN SEEK COMMAND
224	004006	104030	001112	POP	R3,ISEEK+1	:GET CURRENT TRACK
225	004010	104300	001463	001100	R3	:RESTORE IT
226	004013	022251		MOV	R3,CURTRK	:RESTORE GROUP
227	004014	000000		CALL	CURGRP,ISEEK+3	:RESTORE TO PREVIOUS CYLINDER
				RETURN	SEEK	

229					
230					
231					
232					
233					
234	004015	104204	001256		
235	004017	104203	001470		
236	004021	021524			
237	004022	104200	007275	001253	
238	004025	104201	000017		
239	004027	022444			
240	004030	104203	007275		
241	004032	105203	000377		
242	004034	104632	000001		
243	004036	103202	100000		
244	004040	100632	000001		
245	004042	104200	007275	001253	
246	004045	104201	000030		
247	004047	022444			
248	004050				
249	004055	000000			

...	THIS ROUTINE FIXES UP THE PREVIOUS BLOCK	
...	OF THE FCT. IT CLEARS THE PRIMARY FLAG OF THE	
...	LAST ENTRY.	
FIXFCT:	MOV #FCTCNT,R4	:FOR SUBTRACT
	MOV #TWOC,R3	:DOUBLE WORD OF 2
	CALL DSUB	:SUBTRACT TO GET BACK TO RIGHT NUM
	MOV #IMAGE,BUFPNT	:USE IMAGE BUFFER
	MOV #F6,R1	:FCT READ OVERLAY
	CALL PAGE	:READ THE BLOCK
	MOV #IMAGE,R3	:POINT TO BUFFER
	ADD #255,R3	:POINT TO LAST ENTRY
	MOV 1(R3),R2	:GET HIGH ORDER
	BIC #PRMY,R2	:CLEAR FLAG
	MOV R2,1(R3)	:STORE IT BACK
	MOV #IMAGE,BUFPNT	:STORE BUFFER POINTER
	MOV #F9,R1	:FCT WRITE OVERLAY
	CALL PAGE	:WRITE IT BACK OUT
	DJBINC FCTCNT	:GET FCTCNT BACK TO NORMAL
	RETURN	

			WRITE AN RCT BLOCK		
			DDUMMY	BLOCK NUMBER TO BE WRITTEN	
5	004056	114005	RCTWRT: CLR	R5	:CLEAR ERROR COUNTER
6	004057	104050	MOV	R5,NEXT1	:INIT NEXT COPY COUNTER
7	004061	104204	MOV	#DDUMMY,R4	:POINT TO RCT LBN
8	004063	104203	RCTRLP: MOV	#SCR,R3	:POINT TO CHARACTERISTICS
9	004065	104207	MOV	#CONBLK,R0	:POINT TO CONVERT BLOCK
10	004067	104632	MOV	LBNTRK(R3),R2	:GET LBN/TRACK
11	004071	103202	BIC	#HIBYTE,R2	:CLEAR REST OF WORD
12	004073	100672	MOV	R2,V3(R0)	:FOR CONVERT
13	004075	104632	MOV	STCYL(R3),R2	:STARTING CLYINDER
14	004077	103202	BIC	#LO,R2	:CLEAR REST OF WORD
15	004101	100672	MOV	R2,V1+1(R0)	:STORE
16	004103	114002	CLR	R2	:FOR STORE
17	004104	100672	MOV	R2,V1(R0)	:LOW ORDER ALWAYS 0
18	004106	022714	CALL	CVTSK	:CONVERT AND SEEK
19	004107	104207	MOV	#WRBLK,R0	:POINT TO COMMAND BLOCK
20	004111	104203	MOV	#WRCMD,R3	:GET WRITE COMMAND
21	004113	104302	MOV	CURTRK,R2	:GET CURRENT TRACK
22	004115	101023	BIS	R2,R3	:SET TRACK FOR WRITE
23	004116	100673	MOV	R3,RW.CMD(R0)	:STORE IN COMMAND BLOCK
24	004120	104202	MOV	#RCTBUF,R2	:POINT TO BUFFER
25	004122	022627	CALL	CFDC	:COMPUTE EDC - RETURNED IN R3
26	004123	100623	MOV	R3,RW.EDC(R2)	:STORE IT
27	004125	100672	MOV	R2,RW.BUF(R0)	:STICK IN COMMAND BLOCK
28	004127	104143	MOV	(R4),R3	:GET LOW ORDER HEADER
29	004130	100673	MOV	R3,RW.LOW(R0)	:STORE IN WRITE BLOCK
30	004132	104643	MOV	1(R4),R3	:GET HIGH ORDER
31	004134	105303	ADD	ST.LBN,R3	:ADD STARTING LBN BITS
32	004136	101203	BIS	#HD.LBN,R3	:SET HEADER
33	004140	100673	MOV	R3,RW.HI(R0)	:STORE IN WRITE BLOCK
34	004142	104203	MOV	#HSLIM-1,R3	:GET DUMMY SDI POINTER
35	004144	100673	MOV	R3,RW.DUM(R0)	:POINT IN COMMAND BLOCK
36	004146	104303	WRITE3: MOV	HPREA,R3	:GET HEADER PREAMBLE
37	004150	104304	MOV	DPREA,R4	:GET DATA PREAMBLE
38	004152	104302	MOV	UNIT,R2	:SET UNIT
39	004154	104207	MOV	#WRBLK,R0	:MAKE SURE POINTING AT BLOCK
40	004156	101207	BIS	#BIT15,R0	:SET NO REVECTORING
41	004160	060012	XFC	SIF	:WAIT FOR SECTOR PULSE
42	004161	060003	XFC	WRITE	:WRITE SECTOR
43	004162	115001	TST	R1	:ANY ERROR ?
44	004163	014205	BEQ	RWGOOD	:NOPE
45	004164	106300	CMP	RETRY,TMPTRY	:MAX ?
46	004167	014173	BEQ	1\$	:YES - TRY SOME RECOVERY
47	004170	115400	INC	TMPTRY	:INC RETRY COUNT
48	004172	004146	BR	WRITE3	:DO RETRY
49	004173	104303	1\$: MOV	RECTMP,R3	:GET CURRENT ERROR RECOVERY LEVEL
50	004175	074204	BMI	2\$	:IF NEGATIVE THEN FRIED
51	004176	022612	CALL	ERRHND	:TRY RECOVERY
52	004177	114000	CLR	TMPTRY	:FOR INIT
53	004201	117400	DEC	RECTMP	:DECREMENT IT
54	004203	004146	BR	WRITE3	:RETRY
55	004204		2\$:		
56	004204	115405	INC	R5	:YUP - INCREMENT COUNTER
57	004205	115400	RWGOOD: INC	NEXT1	:INCREMENT IT

58	004207	114000	001504		CLR	TMPTRY		:FOR RESET
59	004211	104300	001503	001505	MOV	RECOV,RECTMP		:GET RECOVERY LEVELS
60	004214	104204	000731		MOV	#DDUMMY,R4		:FOR ADD
61	004216	104203	001242		MOV	#RCTFMT,R3		:FOR ADD
62	004220	021506			CALL	DADD		:POINT TO NEXT COPY
63	004221	106300	001244	001245	CMP	FCTCPY,NEXT1		:DONE THIS SECTOR ?
64	004224	054063			BNE	RCTRLP		:NO - WRITE NEXT FCT COPY
65	004225	106305	001244		CMP	FCTCPY,R5		:ERROR ON EVERY WRITE ?
66	004227	014244			BEQ	RCWERR		:YUP - BIG TROUBLE
67	004230	104303	001245		RCFXLP: MOV	NEXT1,R3		:ANY REPEATS ?
68	004232	014243			BEQ	RWTDON		:NO
69	004233	104204	000731		MOV	#DDUMMY,R4		:TO GET IT BACK
70	004235	104203	001242		MOV	#RCTFMT,R3		:DITTO
71	004237	021524			CALL	DSUB		
72	004240	117400	001245		DEC	NEXT1		:SUB IT
73	004242	004230			BR	RCFXLP		:REPEAT
74	004243	000000			RWTDON: RETURN			
75	004244	104012			RCWERR: MOV	R1,R2		:XFC ERROR CODE
76	004245	104201	000017		MOV	#15,R1		:RCT WRITE ERROR
77	004247	022552			CALL	ERRMNT		:ERROR RETURN

				.SBTTL RCT INITIALIZE OVERLAY (F7)		
				DMOVLY F7,START		
				INITIALIZE RCT TO ALL UNUSED		
1						
2	004250					
3						
4						
5						
6	003047	104200	000014	001153	RCTINI: MOV	#F5,CUROVL
7	003052	104203	000400		MOV	#256.,R3
8	003054	114001			CLR	R1
9	003055	104204	004535		MOV	#RDBUF,R4
10	003057	100241			RCLP: MOV	R1,(R4)+
11	003060	117403			DEC	R3
12	003061	053057			BNE	RCLP
13	003062	104201	001305		MOV	#SERNUM,R1
14	003064	104204	004535		MOV	#RDBUF,R4
15	003066	105204	000000		ADD	#RSER,R4
16	003070	104205	000004		MOV	#4,R5
17	003072	104212		6\$:	MOV	(R1)+,R2
18	003073	100242			MOV	R2,(R4)+
19	003074	117405			DEC	R5
20	003075	053072			BNE	6\$
21	003076	104303	001112		MOV	CURTRK,R3
22	003100				PUSH	R3
23	003101	104303	001125		MOV	CYLNUM,R3
24	003103				PUSH	R3
25	003104	104303	001126		MOV	CYLNUM+1,R3
26	003106				PUSH	R3
27	003107	104300	001135	000736	MOV	RBNLBN,TEMP
28	003112	104300	001136	000737	MOV	RBNLBN+1,TEMP+1
29	003115	104204	000736		MOV	#TEMP,R4
30	003117	104200	000400	000731	MOV	#256.,DDUMMY
31	003122	114000	000732		CLR	DDUMMY+1
32	003124	104203	000731		MOV	#DDUMMY,R3
33	003126	021506			CALL	DADD
34	003127	104300	001133	001227	MOV	LBNLBN,HOLD
35	003132	104300	001134	001230	MOV	LBNLBN+1,HOLD+1
36	003135	104203	001475		MOV	#TOTRCT,R3
37	003137	104204	001227		MOV	#HOLD,R4
38	003141	021524			CALL	DSUB
39	003142	104300	001227	001113	MOV	HOLD,CURBN
40	003145	104300	001227	001115	MOV	HOLD,CURLBN
41	003150	104300	001230	001114	MOV	HOLD+1,CURBN+1
42	003153	104300	001230	001116	MOV	HOLD+1,CURLBN+1
43	003156	114000	001501		CLR	COUNT
44	003160	104203	001052		MOV	#SCR,R3
45	003162	104207	001430		MOV	#CONBLK,R0
46	003164	104632	000011		MOV	LBNTRK(R3),R2
47	003166	103202	177400		BIC	#HIBYTE,R2
48	003170	100672	000004		MOV	R2,V3(R0)
49	003172	104632	000001		MOV	STCYL(R3),R2
50	003174	103202	007777		BIC	#LO,R2
51	003176	100672	000001		MOV	R2,V1+1(R0)
52	003200	114002			CLR	R2
53	003201	100672	000000		MOV	R2,V1(R0)
54	003203	114005			RCINLP: CLR	R5
55	003204	104050	001245		MOV	R5,NEXT1
56	003206	104204	001113		RCLP2: MOV	#CURBN,R4
57	003210	022714			CALL	CVTSK

:FOR OVERLAY INIT
 :SECTOR WORD COUNT
 :FOR INIT OF RCT WORDS
 :BUFFER
 :STORE IN BUFFER
 :DECREMENT COUNTER
 :BRANCH BACK TILL DONE
 :POINT TO SERIAL NUMBER
 :POINT TO BUFFER
 :POINT TO SERIAL NUMBER
 :COUNTER
 :GET WORD
 :STORE WORD
 :DECREMENT COUNTER
 :CONTINUE TILL DONE
 :GET CURRENT TRACK
 :SAVE IT
 :GET LOW ORDER CYLINDR
 :SAVE FOR RESTORE
 :GET HIGH ORDER
 :SAVE FOR RESTORE
 :GET NUMBER OF RBN'S IN LBN AREA
 :HIGH ORDER
 :FOR ADD
 :2 BLOCKS(CONTROL) WORTH OF RBN'S
 :CLEAR HIGH ORDER
 :FOR ADD
 :ADD TO GET 'REAL' NUMBER OF RBN'S
 :GET LOW ORDER COUNT OF LBN'S
 :GET HIGH ORDER
 :FOR SUBTRACT
 :DITTO
 :GET STARTING RCT LBN
 :GET STARTING RCT BLOCK NUMBER
 :ALSO SAVE
 :GET HIGH ORDER
 :AND SAVE
 :CLEAR BLOCK COUNTER
 :POINT TO CHARACTERISTICS
 :POINT TO CONVERT BLOCK
 :GET LBN/TRACK
 :CLEAR HIGH BYTE
 :FOR CONVERT
 :STARTING CLYLINDER
 :CLEAR REST OF WORD
 :STORE
 :FOR STORE
 :LOW ORDER ALWAYS 0
 :CLEAR ERROR COUNTER
 :INIT COPY COUNT
 :FOR CONVERT
 :CONVERT AND SEEK

58	003211	104207	000721		MOV	#WRBLK,R0	:POINT TO COMMAND BLOCK
59	003213	104203	122400		MOV	#WRCMD,R3	:GET WRITE COMMAND
60	003215	104302	001112		MOV	CURTRK,R2	:GET CURRENT TRACK
61	003217	101023			BIS	R2,R3	:SET TRACK FOR WRITE
62	003220	100673	000004		MOV	R3,RW.CMD(R0)	:STORE IN COMMAND BLOCK
63	003222	104202	004535		MOV	#RDBUF,R2	:POINT TO BUFFER
64	003224	022627			CALL	CEDC	:COMPUTE EDC - RETURNED IN R3
65	003225	100623	000400		MOV	R3,RW.EDC(R2)	:STORE IT
66	003227	100672	000001		MOV	R2,RW.BUF(R0)	:STICK IN COMMAND BLOCK
67	003231	104303	001113		MOV	CURBN,R3	:GET LOW ORDER HEADER
68	003233	100673	000002		MOV	R3,RW.LOW(R0)	:STORE IN WRITE BLOCK
69	003235	104303	001114		MOV	CURBN+1,R3	:GET HIGH ORDER
70	003237	105303	001335		ADD	ST.LBN,R3	:ADD STARTING LBN BITS
71	003241	101203	000000		BIS	#HD.LBN,R3	:SET HEADER
72	003243	100673	000003		MOV	R3,RW.HI(R0)	:STORE IN WRITE BLOCK
73	003245	104203	000726		MOV	#HSLIM-1,R3	:GET DUMMY SDI POINTER
74	003247	100673	000005		MOV	R3,RW.DUM(R0)	:POINT IN COMMAND BLOCK
75	003251	104303	001333		WRITE4: MOV	HPREA,R3	:GET HEADER PREAMBLE
76	003253	104304	001334		MOV	DPREA,R4	:GET DATA PREAMBLE
77	003255	104302	000740		MOV	UNIT,R2	:SET UNIT
78	003257	104207	100721		MOV	#<WRBLK.BIT15>,R0	:MAKE SURE POINTING AT BLOCK
79	003261	060012			XFC	SIP	:WAIT FOR SECTOR PULSE
80	003262	060003			XFC	WRITE	:WRITE SECTOR
81	003263	115001			TST	R1	:ANY ERROR ?
82	003264	013306			BEQ	NOGOOD	:NOPE
83	003265	106300	001502	001504	CMP	RETRY,TMPTRY	:MAX ?
84	003270	013274			BEQ	1\$	:YES - TRY SOME RECOVERY
85	003271	115400	001504		INC	TMPTRY	:INC RETRY COUNT
86	003273	003251			BR	WRITE4	:DO RETRY
87	003274	104303	001505	1\$:	MOV	RECTMP,R3	:GET CURRENT ERROR RECOVERY LEVEL
88	003276	073305			BMI	2\$	:IF NEGATIVE THEN FRIED
89	003277	022612			CALL	ERR4ND	:TRY RECOVERY
90	003300	114000	001504		CLR	TMPTRY	:FOR INIT
91	003302	117400	001505		DEC	RECTMP	:DECREMENT IT
92	003304	003251			BR	WRITE4	:RETRY
93	003305			2\$:			
94	003305	115405			INC	R5	:YUP - INCREMENT COUNTER
95	003306	115400	001245		NOGOOD: INC	NEXT1	:INCREMENT IT
96	003310	114000	001504		CLR	TMPTRY	:FOR RESET
97	003312	104300	001503	001505	MOV	RECOV,RECTMP	:GET RECOVERY LEVELS
98	003315	104204	001113		MOV	#CURBN,R4	:FOR ADD
99	003317	104203	001242		MOV	#RCTFMT,R3	:FOR ADD
100	003321	021506			CALL	DADD	:POINT TO NEXT COPY
101	003322	106300	001244	001245	CMP	FCTCPY,NEXT1	:DONE THIS SECTOR ?
102	003325	053206			BNE	RCLP2	:NO - WRITE NEXT FCT COPY
103	003326	106305	001244		CMP	FCTCPY,R5	:ERROR ON EVERY WRITE ?
104	003330	013462			BEQ	RCINER	:YUP - BIG TROUBLE
105	003331	102200	000040	001217	BIT	#RCINIT,FLAG	:ALREADY FIXED IT UP
106	003334	053353			BNE	4\$	:YUP - NO NEED TO DO IT AGAIN
107	003335	104204	000736		MOV	#TEMP,R4	:FOR SUBTRACT (RBN'S NOT DONE)
108	003337	104200	000200	000731	MOV	#128,DDUMMY	:SUBTRACT ONE BLOCKS WORTH
109	003342	114000	000732		CLR	DDUMMY+1	:FOR CLEAR
110	003344	104203	000731		MOV	#DDUMMY,R3	:FOR SUBTRACT
111	003346	021524			CALL	DSUB	:SUBTRACT
112	003347	021626			CALL	DCMP	:IN LAST BLOCK ?
113	003350	073353			BMI	4\$	:NOPE
114	003351	013353			BEQ	4\$	:IF EQUAL - NO PARTIAL BLOCK

115	003357	023466			CALL	FIXBLK	;YES - CHANGE HEADERS TO NULL	
116	003358	115000	001501	4\$:	TST	COUNT	:ON FIRST BLOCK ?	
117	003359	053366			BNE	7\$	:NO - NO NEED TO FIX UP	
118	003356	114005			CLR	R5	:FOR BLOCK FIXUP	
119	003357	104204	004535		MOV	#RDBUF,R4	:POINT TO BUFFER	
120	003361	104201	000004		MOV	#4,R1	:COUNTER	
121	003363	100245		5\$:	MOV	R5,(R4)+	:CLEAR DATE AREA	
122	003364	117401			DEC	R1	:DECREMENT COUNTER	
123	003365	053363			BNE	5\$	:CONT TILL DONE	
124	003366	102200	000100	001220	7\$:	BIT	#RCINDN,FLAG1	:ALL DONE ??
125	003371	053420			BNE	RCLP6	:YUP - CUT OUT	
126	003372				DUBINC	CURLBN	:INCREMENT IT	
127	003377	104300	001115	001113	MOV	CURLBN,CURBN	:GET LOW ORDER	
128	003402	104300	001116	001114	MOV	CURLBN+1,CURBN+1	:GET HIGH ORDER	
129	003405	106300	001261	001501	CMP	RCTLBN,COUNT	:DONE RCT BLOCKS(NOT PAD)	
130	003410	013442			BEQ	RCFIX	:YUP - REINIT BLOCK	
131	003411	060022			XFC	UPDATE	:LET HOST KNOW STILL ALIVE	
132	003412	115400	001501		INC	COUNT	:INCREMENT BLOCK COUNTER	
133	003414	106300	001242	001501	CMP	RCTFMT,COUNT	:DONE ?	
134	003417	053203			BNE	RCINLP	:NOPE - DO NEXT SECTOR	
135	003420				RCLP6:	POP	:GET HIGH ORDER CYLINDER	
136	003421	104030	001126		MOV	R3,CYLNUM+1	:STORE IT	
137	003423	104030	001077		MOV	R3,ISEEK+2	:STORE IN SEEK COMMAND	
138	003425				POP	R3	:GET LOW ORDER	
139	003426	104030	001125		MOV	R3,CYLNUM	:RESTORE IT	
140	003430	104030	001076		MOV	R3,ISEEK+1	:STORE IN SEEK COMMAND	
141	003432				POP	R3	:GET CURRENT TRACK	
142	003433	104030	001112		MOV	R3,CURTRK	:RESTORE IT	
143	003435	104300	001463	001100	MOV	CURGRP,ISEEK+3	:RESTORE GROUP NUMBER	
144	003440	022251			CALL	SEEK	:RESTORE TO PREVIOUS CYLINDER	
145	003441	000000			RETURN			
146	003442	104202	000200		RCFIX:	MOV	#128.,R2	:INIT COUNT
147	003444	104204	004535		MOV	#RDBUF,R4	:INIT POINTER	
148	003446	114003			CLR	R3	:FOR STORE	
149	003447	114005			CLR	R5	:DITTO	
150	003450	101205	100000		BIS	#RC.NUL,R5	:SET NULL HEADER	
151	003452	100243			RCLP3:	MOV	R3,(R4)+	:STORE LOW ORDER
152	003453	100245			MOV	R5,(R4)+	:STORE HIGH ORDER	
153	003454	117402			DEC	R2	:DECREMENT COUNTER	
154	003455	053452			BNE	RCLP3	:LOOP UNTIL DONE	
155	003456	101200	000100	001220	BIS	#RCINDN,FLAG1	:DONE ALL NON-PAD - ONE MORE THEN FINISH	
156	003461	003411			BR	RCLP4	:CONTINUE TILL DONE ALL SECTORS	
157	003462	104012			RCINER:	MOV	R1,R2	:XFC ERROR CODE
158	003463	104201	000017		MOV	#15.,R1	:RCT INIT ERROR	
159	003465	022552			CALL	ERRMNT	:ERROR RETURN	

1					.SBTTL FCT READ OVERLAY (F6)	
2	003513				DMOVLY F6,START	
3						
4					READ A BLOCK OF THE FCT	
5					R5 -> BUFFER	
6						
7	003047	104200	000017	001153	MOV #F6,CUROVL	:OVERLAY NUMBER
8	003052	104300	001256	000731	MOV FCTCNT,DDUMMY	:GET CURRENT COUNT
9	003055	114000	000732		CLR DDUMMY+1	:FOR HIGH ORDER STORE
10	003057	114005			CLR R5	:CLEAR ERROR COUNTER
11	003060	104204	000731		MOV #DDUMMY,R4	:FOR CONVERT
12	003062	104303	001143		MOV LBNCYL,R3	:GET LBN CYLINDERS
13	003064	104207	001430		MOV #CONBLK,R0	:POINT TO CONVERT BLOCK
14	003066	100673	000000		MOV R3,V1(R0)	:STORE IT FOR CONVERT
15	003070	104303	001144		MOV LBNCYL+1,R3	:HIGH ORDER
16	003072	100673	000001		MOV R3,V1+1(R0)	:STORE IT
17	003074	104303	001127		MOV SECTRK,R3	:GET SECTORS/TRACK
18	003076	100673	000004		MOV R3,V3(R0)	:STORE FOR CONVERT
19	003100	022714			CALL CVTSK	:CONVERT FCT BLOCK NUMBER AND SEEK
20	003101	104207	000721		MOV #RDBLK,R0	:PREPARE FOR READ SECTORS
21	003103	104203	000726		MOV #HSLIM-1,R3	:POINTER TO DUMMY SDI BLOCK
22	003105	100673	000005		MOV R3,RW.DUM(R0)	:STORE IN COMMAND BLOCK
23	003107	104303	000731		MOV DDUMMY,R3	:LO ORDER BLOCK NUMBER
24	003111	100673	000002		MOV R3,RW.LOW(R0)	:STORE IN READ CMD BLOCK
25	003113	104303	000732		MOV DDUMMY+1,R3	:GET HIGH ORDER
26	003115	105303	001337		ADD ST.XBN,R3	:ADD STARTING LBN BITS
27	003117	101203	120000		BIS #HD.XBN,R3	:HEADER CODE
28	003121	100673	000003		MOV R3,RW.HI(R0)	:STORE IN READ CMD BLOCK
29	003123	104303	001253		MOV BUFPNT,R3	:GET BUFFER POINTER
30	003125	100673	000001		MOV R3,RW.BUF(R0)	:STORE BUFFER ADDRESS IN COMMAND BUFFER
31	003127	104203	013400		MOV #RWCMD,R3	:LOAD SDI READ COMMAND
32	003131	104301	001112		MOV CURTRK,R1	:GET CURRENT HEAD NUMBER IN R1
33	003133	101013			BIS R1,R3	:SET IT IN COMMAND
34	003134	100673	000004		MOV R3,RW.CMD(R0)	:STORE BACK
35	003136	104207	000721		MOV #RDBLK,R0	:MAKE SURE POINTING AT BLOCK
36	003140	104203	100000		MOV #RDCMD,R3	:MARK AS ONLY REQUEST
37	003142	100173			MOV R3,(R0)	:STORE IN CMD BLOCK
38	003143	104302	000740		MOV UNIT,R2	:SDI INTERCONNECT
39	003145	101207	100000		BIS #BIT15,R0	:SET NO REVECTORING
40	003147	060012			XFC SIP	:WAIT FOR PULSE
41	003150	060002			XFC READ	:READ 1 SECTOR
42	003151	115001			TST R1	:ANY ERRORS ?
43	003152	053170			BNE 100\$	:YES - TRY RECOVERY
44	003153	104173			MOV (R0),R3	:GET STATUS WORD
45	003154	102203	010000		BIS #ECCF,R3	:ECC ERROR ?
46	003156	013162			BEQ 101\$	:NOPE - VERIFY EDC
47	003157	023027			CALL ECCCK	:CORRECT ECC
48	003160	115001			TST R1	:TEST FLAG
49	003161	053170			BNE 100\$	:UNCORRECTABLE
50	003162	104302	001253		MOV BUFPNT,R2	:POINT TO BUFFER
51	003164	022627			CALL CEDC	:COMPUTE EDC
52	003165	106623	000400		CMP RW.EDC(R2),R3	:O.K. ?
53	003167	013227			BEQ 102\$	:YUP - CONSIDER GOOD
54	003170	106300	001502	001504	CMP RETRY,IMPTRY	:MAX ?
55	003173	013177			BEQ 1\$	:YES - TRY SOME RECOVERY
56	003174	115400	001504		INC IMPTRY	:INC RETRY COUNT
57	003176	003136			BR READ7	:DO RETRY

58	003177	104303	001505	1\$:	MOV	RECTMP,R3	:GET CURRENT ERROR RECOVERY LEVEL
59	003201	073210			BMI	2\$	:IF NEGATIVE THEN FRIED
60	003202	022612			CALL	ERRHND	:TRY RECOVERY
61	003203	114000	001504		CLR	TMPTRY	:FOR INIT
62	003205	117400	001505		DEC	RECTMP	:DECREMENT IT
63	003207	003136			BR	RFAD7	:RETRY
64	003210			2\$:			
65	003210	115405			INC	R5	:INCREMENT BAD COUNTER
66	003211	106305	001244		CMP	FCTCPY,R5	:ALL BAD ?
67	003213	013344			BEQ	OFATAL	:YUP - ALL OVER
68	003214	104204	000731		MOV	#DDUMMY,R4	:PCINT TO COUNT
69	003216	104203	001240		MOV	#FCTFMT,R3	:SIZE OF TABLE
70	003220	021506			CALL	DADD	:ADD TO POINT TO NEXT COPY
71	003221	114000	001504		CLR	TMPTRY	:RESET RETRY LEVEL
72	003223	104300	001503	001505	MOV	RECOV,RECTMP	:DITTO RECOVERY LEVELS
73	003226	003060			BR	FOLOOP	:BRANCH BACK
74	003227			102\$:			
75	003227	114000	001504		CLR	TMPTRY	:FOR RESET
76	003231	104300	001503	001505	MOV	RECOV,RECTMP	:GET RECOVERY LEVELS
77	003234	115005			TST	R5	:ANY ERRORS ?
78	003235	013341			BEQ	OLDONE	:NO - EXIT
79	003236	104204	000731		MOV	#DDUMMY,R4	:POINT TO BLOCK COUNT
80	003240	104203	001240		MOV	#FCTFMT,R3	:SIZE OF TABLE
81	003242	021524			CALL	DSUB	:GET BACK TO PREVIOUS COPY
82	003243	022714			CALL	CVTSK	:CONVERT AND SEEK
83	003244	104207	000721		MOV	#WRBLK,R0	:POINT TO COMMAND BLOCK
84	003246	104203	122400		MOV	#WRCMD,R3	:GET WRITE COMMAND
85	003250	104302	001112		MOV	CURTRK,R2	:GET CURRENT TRACK
86	003252	101023			BIS	R2,R3	:SET TRACK FOR WRITE
87	003253	100673	000004		MOV	R3,RW.CMD(R0)	:STORE IN COMMAND BLOCK
88	003255	104303	001253		MOV	BUFPT,R3	:GET BUFFER ADDRESS
89	003257	100673	000001		MOV	R3,RW.BUF(R0)	:STICK IN COMMAND BLOCK
90	003261	104303	000731		MOV	DDUMMY,R3	:GET LOW ORDER HEADER
91	003263	100673	000002		MOV	R3,RW.LOW(R0)	:STORE IN WRITE BLOCK
92	003265	104303	000732		MOV	DDUMMY+1,R3	:GET HIGH ORDER
93	003267	105303	001337		ADD	ST.XBN,R3	:ADD STARTING XBN BITS
94	003271	101203	120000		BIS	#HD.XBN,R3	:HEADER CODE
95	003273	100673	000003		MOV	R3,RW.HI(R0)	:STORE IN WRITE BLOCK
96	003275	104203	000726		MOV	#HSLIM-1,R3	:GET DUMMY SDI POINTER
97	003277	100673	000005		MOV	R3,RW.DUM(R0)	:POINT IN COMMAND BLOCK
98	003301	104303	001333		MOV	HPREA,R3	:GET HEADER PREAMBLE
99	003303	104304	001334		MOV	DPREA,R4	:GET DATA PREAMBLE
100	003305	104302	000740		MOV	UNIT,R2	:SET UNIT
101	003307	104207	000721		MOV	#WRBLK,R0	:MAKE SURE POINTING AT BLOCK
102	003311	101207	100000		BIS	#BIT15,R0	:SET NO REVECTORING
103	003313	060012			XFC	SIP	:WAIT FOR SECTOR PULSE
104	003314	060003			XFC	WRITE	:WRITE SECTOR
105	003315	115001			TST	R1	:ANY ERROR ?
106	003316	013337			BEQ	2\$	:NO - SKIP RETRY
107	003317	106300	001502	001504	CMP	RETRY,TMPTRY	:MAX ?
108	003322	013326			BEQ	1\$	:YES - TRY SOME RECOVERY
109	003323	115400	001504		INC	TMPTRY	:INC RETRY COUNT
110	003325	003301			BR	WRITE8	:DO RETRY
111	003326	104303	001505	1\$:	MOV	RECTMP,R3	:GET CURRENT ERROR RECOVERY LEVEL
112	003330	073337			BMI	2\$	:IF NEGATIVE THEN FRIED
113	003331	022612			CALL	ERRHND	:TRY RECOVERY
114	003332	114000	001504		CLR	TMPTRY	:FOR INIT

115	003334	117400	001505		DEC	RECTMP	:DECREMENT IT
116	003336	003301			BR	WRITE8	:RETRY
117	003337			2\$:			
118	003337	117405			DEC	R5	:DEREMENT COUNTER
119	003340	003227			BR	FODONE	:SEE IF ANY MORE TO DO
120	003341	115400	001256	OLDONE:	INC	FCTCNT	:INCREMENT IT
121	003343	000000			RETURN		:ALL DONE
122	003344	115000	001256	OFATAL:	TST	FCTCNT	:SECTOR 0 ? (MEDIA INFO)
123	003346	013362			BEQ	OQUIT	:IF YES THEN IT'S ALL OVER
124	003347	102200	000020	001217	BIT	#GOBAD,FLAG	:CONTINUE AS BEST GUESS?
125	003352	013362			BEQ	OQUIT	:NOPE - GIVE UP
126	003353	101200	002004	001217	BIS	#FCTBAD+BSTGS,FLAG	:SET FCT BAD FLAG
127	003356	103200	000001	001217	BIC	#FCTAVL,FLAG	:NO MORE FCT
128	003361	000000			RETURN		:RETURN
129	003362	104012			MOV	R1,R2	:XFC ERROR CODE
130	003363	104201	000021	OQUIT:	MOV	#17,R1	:FCT READ ERROR
131	003365	022552			CALL	ERRMNT	:ERROR RETURN

1					.SBTTL GET FCT BLOCK FOR D/XBN FORMAT (G2) -	
2						
3						
4					GET'S ONE FCT BLOCK FROM HOST FOR D/BN FORMATTING	
5						
6						
7	003366				DMOVLY G2,START	
8						
9						
10						
11	003047	104200	000033	001153	MOV #G2,CUROVL	:SIGNAL OVERLAY 11
12	003052	104205	001263		MOV #DMBUF,R5	:POINT TO MAINT BUFFER
13	003054	104303	001321		MOV FCMSG,R3	:GET DUP CODE
14	003056	100153			MOV R3,(R5)	:STORE IT
15	003057	104303	001322		MOV FCMSG+1,R3	:GET 'F' IDENTIFIER
16	003061	100653	000001		MOV R3,1(R5)	:STORE IT IN MESSAGE
17	003063	104303	001256		MOV FCTCNT,R3	:GET BLOCK NUMBER DESIRED
18	003065	100653	000002		MOV R3,2(R5)	:STORE IT
19	003067	022536			CALL SNDMNT	:SFND REQUEST
20	003070	022544			CALL RCMNT	:RECEIVE ANSWER
21	003071	104653	000002		MOV 2(R5),R3	:GET BLOCK COUNT RECEIVED
22	003073	013113			BEQ DLERR	:ERROR IF ZERO
23	003074	104653	000003		MOV 3(R5),R3	:GET HOST ADDRESS
24	003076	104030	001425		MOV R3,OVLBLK+1	:STORE IN OVERLAY BLOCK
25	003100	114000	001426		CLR OVLBLK+2	:ZERO HIGH ORDER
26	003102	104200	000401	001424	MOV #257,OVLBLK	:GET LENGTH
27	003105	104204	001424		MOV #OVLBLK,R4	:FOR OVERLAY ROUTINE
28	003107	104203	005152		MOV #PBNUF,R3	:POINT TO BUFFER
29	003111	022526			CALL OVRLAY	:GET THE SECTOR
30	003112	000000			RETURN	
31	003113	104201	000023		DLERR: MOV #19,R1	:SIGNAL DLL ERROR
32	003115	104302	001256		MOV FCTCNT,R2	:BLOCK FAILED ON
33	003117	022552			CALL ERRMNT	:ERROR RETURN

1					.SBTTL GET FCT BLOCK FOR LBN FORMAT (G3)	
2						
3						
4					GET RIGHT FCT BLOCK FOR LBN FORMATTING	
5						
6						
7	003120				DMOVLY G3,START	
8						
9						
10						
11	003047	104200	000036	001153	MOV #G3,CUROVL	:FOR CURRENT OVERLAY
12	003052	114000	001256		CLR FCTCNT	:FOR FIRST FCT BLOCK
13	003054	104200	005152	001253	MOV #PBNBUF,BUFPNT	:POINT TO BUFFER
14	003057	104201	000017		MOV #F6,R1	:FCT READ OVERLAY
15	003061	022444			CALL PAGE	:READ IT IN
16	003062	102200	000001	001217	BIT #FCTAVL,FLAG	:FCT STILL HERE ?
17	003065	013243			BEQ NGD	:NOPE - CAN IT
18	003066	104207	005152		MOV #PBNBUF,R0	:POINT TO BUFFER
19	003070	104173			MOV (R0),R3	:GET FORMAT MEDIA WORD
20	003071	106203	126736		CMP #M512,R3	:IS IT 512 ?
21	003073	013101			BEQ 13\$	:YUP - O.K.
22	003074	106203	074161		CMP #M576,R3	:IS IT 576 ?
23	003076	013101			BEQ 13\$	:YUP - O.K.
24	003077	115003			TST R3	:IS IT FORMAT IN PROGRESS
25	003100	053243			BNE NGD	:NOPE - FCT NO GOOD
26	003101	104673	000025	13\$:	MOV FCTFLG(R0),R3	:GET FLAG WORD
27	003103	102203	100000		BIT #NOFCT,R3	:IS THERE REALLY AN FCT ??
28	003105	053243			BNE NGD	:NOPE - FIND OUT IF WE QUIT OR ROUGH IT
29	003106	104673	000016		MOV C512(R0),R3	:GET COUNT OF USED ENTRIES
30	003110	104030	001237		MOV R3,FCNT	:STORE IT
31	003112	053116			BNE 12\$	:IF NOT ZERO THEN ENTRIES EXIST
32	003113	101200	000002	001217	BIS #FCTEMT,FLAG	:SET EMPTY FLAG
33	003116	114003		12\$:	CLR R3	:FOR FCT INIT
34	003117	100173			MOV R3,(R0)	:SIGNAL FORMAT IN PROGRESS
35	003120	104203	005152		MOV #PBNBUF,R3	:POINT TO BUFFER
36	003122	105203	000002		ADD #FSER,R3	:POINT TO SERIAL NUMBER
37	003124	104204	001305		MOV #SERNUM,R4	:SERIAL NUMBER BLOCK
38	003126	104205	000004		MOV #4,R5	:COUNTER
39	003130	104232		8\$:	MOV (R3)+,R2	:GET WORD
40	003131	100242			MOV R2,(R4)+	:STORE WORD
41	003132	117405			DEC R5	:DECREMENT COUNTER
42	003133	053130			BNE 8\$	:CONT TILL DONE
43	003134	104200	005152	001253	MOV #PBNBUF,BUFPNT	:POINT TO BUFFER
44	003137	104201	000030		MOV #F9,R1	:FCT WRITE OVERLAY
45	003141	022444			CALL PAGE	:DO IT
46	003142	115000	001237		TST FCNT	:ANY ENTRIES ?
47	003144	013237			BEQ RDONE1	:NOPE - ALL DONE
48	003145	104200	000001	001256	MOV #1,FCTCNT	:FOR FCT COUNT INIT
49	003150	104200	005152	001253	MOV #PBNBUF,BUFPNT	:POINT TO BUFFER
50	003153	104201	000017		MOV #F6,R1	:FCT READ OVERLAY
51	003155	022444			CALL PAGE	:DO THE READ
52	003156	102200	000001	001217	BIT #FCTAVL,FLAG	:STILL HAVE FCT ?
53	003161	013243			BEQ NGD	:NOPE - CAN IT
54	003162	104204	005152		MOV #PBNBUF,R4	:POINT TO THE BUFFER
55	003164	105204	000376		ADD #254,R4	:POINT TO LAST ENTRY
56	003166	104203	001154		MOV #HGHPBN,R3	:HIGHEST PBN IN LBN AREA
57	003170	021626			CALL DCMPL	:IS IT RIGHT BLOCK ?

58	003171	033176			BPL	BLKFND	:YES - FIND RIGHT ENTRY
59	003172	107200	000200	001237	SUB	#128.,FCNT	:SUBTRACT ONE BLOCKS WORTH
60	003175	003150			BR	RLOOP	:BRANCH BACK
61	003176	104200	000200	001500	BLKFND: MOV	#128.,PCNT	:FOR INIT OF COUNT
62	003201	104204	005152		MOV	#PBNBUF,R4	:POINT TO PBN BUFFER
63	003203	104647	000001		RLOOP1: MOV	1(R4),R0	:GET HIGH ORDER
64	003205	104071			MOV	R0,R1	:SAVE IT TEMPORARILY
65	003206	103207	170000		BIC	#HD.CLR,R0	:CLEAR FOR COMPARE
66	003210	100647	000001		MOV	R0,1(R4)	:STORE IT BACK
67	003212	104203	001154		MOV	#HGHPBN,R3	:POINT TO HIGHEST PBN
68	003214	021626			CALL	DMP	:COMPARE
69	003215	033233			BPL	RDONE	:IF LESS THAN OR EQUAL THEN FOUND FIRST LBN
70	003216	100641	000001		MOV	R1,1(R4)	:STORE HEADER BACK
71	003220	117400	001500		DEC	PCNT	:DECREMENT COUNT
72	003222	117400	001237		DEC	FCNT	:DEC IT
73	003224	105204	000002		ADD	#2,R4	:POINT TO NEXT ENTRY
74	003226	106200	000001	001237	CMP	#1,FCNT	:COUNT AT 1 ?
75	003231	013233			BEQ	RDONE	:YUP - THEN LAST ENTRY IS IT
76	003232	003203			BR	RLOOP1	:TRY NEXT ENTRY
77	003233	100641	000001		RDONE: MOV	R1,1(R4)	:STORE HEADER BACK
78	003235	104040	001223		MOV	R4,BADPBN	:MAKE CURRENT BAD POINTER
79	003237	104300	001237	001311	RDONE1: MOV	FCNT,FCTREV	:FCT ENTRY COUNT FOR LATER USE
80	003242	000000			RETURN		:RETURN
81	003243	102200	000020	001217	NGD: BIT	#GOBAD,FLAG	:CONTINUE AS BEST GUESS ?
82	003246	013256			BEQ	RQUIT	:NOPE - GIVE UP
83	003247	101200	002004	001217	BIS	#FCTBAD+BSTGS,FLAG	:SET BAD FCT FLAG
84	003252	103200	000001	001217	BIC	#FCTAVL,FLAG	:NO MORE FCT
85	003255	000000			RETURN		
86	003256	104201	000022		RQUIT: MOV	#18.,R1	:ERROR CODE
87	003260	114002			CLR	R2	:NO SUBCODE
88	003261	022552			CALL	ERRMNT	:ERROR RETURN


```

58 003201 024222
59 003202 024510
60 003203 104207 001263
61 003205 104301 001327
62 003207 100171
63 003210 104301 001330
64 003212 100671 000001
65 003214 114002
66 003215 104303 001312
67 003217 100673 000002
68 003221 104303 001313
69 003223 100673 000004
70 003225 100672 000005
71 003227 104303 001314
72 003231 100673 000006
73 003233 100672 000007
74 003235 104303 001315
75 003237 100673 000010
76 003241 100672 000011
77 003243 104303 001473
78 003245 100673 000012
79 003247 102200 002000 001217
80 003252 013260
81 003253 104202 000002
82 003255 100672 000013
83 003257 003264
84 003260 104202 000001
85 003262 100672 000013
86 003264 022536
87 003265 022324
88 003266 114007
89 003267 060021
  
```

```

CLSKP3: CALL FCTCK
        CALL RCTCK
        MOV #DMBUF,R0
        MOV DONMSG,R1
        MOV R1,(R0)
        MOV DONMSG+1,R1
        MOV R1,1(R0)
        CLR R2
        MOV LBNBAD,R3
        MOV R3,LBD(R0)
        MOV RCTBAD,R3
        MOV R3,RBD(R0)
        MOV R2,RBD+1(R0)
        MOV DBBAD,R3
        MOV R3,DBD(R0)
        MOV R2,DBD+1(R0)
        MOV XBBAD,R3
        MOV R3,XBD(R0)
        MOV R2,XBD+1(R0)
        MOV RTYCNT,R3
        MOV R3,RTCNT(R0)
        BIT #BSTGS,FLAG
        BEQ GDFCT
        MOV #2,R2
        MOV R2,FCT(R0)
        BR ALDONE
GDFCT:  MOV #1,R2
        MOV R2,FCT(R0)
ALDONE: CALL SNDMNT
        CALL DISCON
        CLR R0
        XFC DONE
  
```

```

:VERIFY FCT
:VERIFY RCT
:POINT TO DMBUF
:GET DUP CODE
:STORE IT
:'D' FLAG
:STORE IT
:FOR CLEAR OF BUFFER
:GET COUNT OF REVECTORED LBN'S
:STORE IN BLOCK
:RCT BLOCKS BAD
:STORE IT
:CLEAR OTHER ERROR
:DBN BLOCKS BAD
:STORE IT
:CLEAR OTHER COUNT
:XBN LOCKS BAD
:STORE IT
:CLEAR OTHER ERROR
:RETRY COUNT
:STORE IT
:DID WE USE FCT ?
:YUP
:NOPE - FLAG AS 2
:STORE IT
:SKIP GOOD FCT FLAGGIN
:GOOD IS A 1
:STORE IT
:SEND IT
:DISCONNECT/SPINDOWN DRIVE
:MAKE SURE QUIT NICELY
:DONE
  
```

1						
2						
3						PROCESS REVECTOR BLOCK
4	003270					CLEWRT: PUSH
5	003276	104204	006204			MOV #REVBUFF,R4
6	003300	104300	001501	000736		MOV COUNT,TEMP
7	003303	104642	000002			CLHERE: MOV 2(R4),R2
8	003305	104643	000003			MOV 3(R4),R3
9	003307	101203	060000			BIS #HD,RBN,R3
10	003311	104205	005152			MOV #PBNBUF,R5
11	003313	104201	000200			MOV #RBNRPT,R1
12	003315	100252				WLOOP: MOV R2,(R5)+
13	003316	100253				MOV R3,(R5)+
14	003317	117401				DEC R1
15	003320	053315				BNE WLOOP
16	003321	104203	001052			MOV #SCR,R3
17	003323	104632	000011			MOV LBNTRK(R3),R2
18	003325	103202	177400			BIC #HIBYTE,R2
19	003327	104207	001430			MOV #CONBLK,R0
20	003331	100672	000004			MOV R2,V3(R0)
21	003333	104632	000001			MOV STCYL(R3),R2
22	003335	103202	007777			BIC #LO,R2
23	003337	100672	000001			MOV R2,V1+1(R0)
24	003341	114002				CLR R2
25	003342	100672	000000			MOV R2,V1(R0)
26	003344	023455				CALL CS
27	003345	104207	000721			MOV #WRBLK,R0
28	003347	104203	122400			MOV #WRCMD,R3
29	003351	104302	001112			MOV CURTRK,R2
30	003353	101023				BIS R2,R3
31	003354	100673	000004			MOV R3,RW.CMD(R0)
32	003356	104202	005152			MOV #PBNBUF,R2
33	003360	022627				CALL CEDC
34	003361	100623	000400			MOV R3,RW.EDC(R2)
35	003363	100672	000001			MOV R2,RW.BUF(R0)
36	003365	104143				MOV (R4),R3
37	003366	100673	000002			MOV R3,RW.LOW(R0)
38	003370	104643	000001			MOV 1(R4),R3
39	003372	105303	001335			ADD ST.LBN,R3
40	003374	100673	000003			MOV R3,RW.HI(R0)
41	003376	104203	000726			MOV #HSLIM-1,R3
42	003400	100673	000005			MOV R3,RW.DUM(R0)
43	003402	104040	000731			MOV R4,DDUMMY
44	003404	104303	001333			WRITE9: MOV HPREA,R3
45	003406	104304	001334			MOV DPREA,R4
46	003410	104302	000740			MOV UNIT,R2
47	003412	104207	000721			MOV #WRBLK,R0
48	003414	101207	100000			BIS #BIT15,R0
49	003416	060012				XFC SIP
50	003417	060003				XFC WRITE
51	003420	115001				TST R1
52	003421	013431				BEQ 1\$
53	003422	106200	000010	001504		CMR #MAXTRY,IMPTRY
54	003425	013431				BEQ 1\$
55	003426	115400	001504			INC IMPTRY
56	003430	003404				BR WRITE9
57	003431					1\$:

:POINT TO BUFFER
 :GET COUNT
 :GET LOW ORDER RBN
 :GET HIGH ORDER RBN
 :SET IN HDR CODE
 :POINT TO BUFFER
 :INIT COUNTER
 :STORE LOW ORDER
 :STORE HIGH ORDER
 :DECREMENT COUNTER
 :CONTINUE TILL DONE
 :POINT TO CHARACTERISTICS
 :GET LBN/TRACK
 :CLEAR HIGH BYTE
 :POINT TO CONVERT BLOCK
 :FOR CONVERT
 :STARTING CLYLINDER
 :CLEAR REST OF WORD
 :STORE
 :FOR STORE
 :LOW ALWAYS ZERO
 :CONVERT AND SEEK
 :POINT TO COMMAND BLOCK
 :GET WRITE COMMAND
 :GET CURRENT TRACK
 :SET TRACK FOR WRITE
 :STORE IN COMMAND BLOCK
 :POINT TO BUFFER
 :COMPUTE EDC - RETURNED IN R3
 :STORE IT
 :STICK IN COMMAND BLOCK
 :GET LOW ORDER HEADER
 :STORE IN WRITE BLOCK
 :GET HIGH ORDER
 :ADD STARTING LBN BITS
 :STORE IN WRITE BLOCK
 :GET DUMMY SDI POINTER
 :POINT IN COMMAND BLOCK
 :SAVE R4
 :GET HEADER PREAMBLE
 :GET DATA PREAMBLE
 :SET UNIT
 :MAKE SURE POINTING AT BLOCK
 :SET NO REVECTORING
 :WAIT FOR SECTOR PULSE
 :WRITE SECTOR
 :ANY ERROR ?
 :NO - SKIP RETRY
 :MAX ?
 :YES - GIVE UP
 :INC RETRY COUNT
 :DO RETRY

58	003431	114000	001504		CLR	TMPTRY		:CLEAR RETRY COUNT
59	003433	104304	000731		MOV	DDUMMY,R4		:RESTORE R4
60	003435	105204	000004		ADD	#REVLEN,R4		:POINT TO NEXT ENTRY
61	003437	115400	001501		INC	COUNT		:INC COUNTER
62	003441	106200	000100	001501	CMP	#64,COUNT		:DONE ?
63	003444	053303			BNE	CIHERE		:NO - REPEAT
64	003445	023522			CALL	RBWRT		:WRITE GOOD EDC'S TO RBN'S
65	003446				POPA			
66	003454	000000			RETURN			
67								
68								
69								
70								
71	003455	104207	001430		MOV	#CONBLK,R0		:POINT TO CONVERT BLOCK
72	003457	104143			MOV	(R4),R3		:GET LOW ORDR
73	003460	100673	000002		MOV	R3,V2(R0)		:STORE IT
74	003462	104643	000001		MOV	1(R4),R3		:HIGH ORDER
75	003464	103203	170000		BIC	#HD,CLR,R3		:CLEAR HEADER
76	003466	100673	000003		MOV	R3,V2+1(R0)		:STORE IT
77	003470	104201	001052		MOV	#SCR,R1		:POINT TO SUBUNIT CHARACTERISTICS
78	003472	060020			XFC	CVT		:CONVERT IT
79	003473	104670	000011	001112	MOV	TRK(R0),CURTRK		:GET TRACK NUMBER
80	003476	104670	000006	001076	MOV	CYL(R0),ISEEK+1		:LOW ORDER CYLINDER
81	003501	104670	000007	001077	MOV	CYL+1(R0),ISEEK+2		:HIGH ORDER CYLINDER
82	003504	104670	000010	001100	MOV	GRP(R0),ISEEK+3		:GROUP NUMBER
83	003507	022251			CALL	SEEK		:DO SEEK
84	003510	115001			TST	R1		:ANY ERROR
85	003511	053513			BNE	CKR		:YUP
86	003512	000000			RETURN			
87	003513	104201	000012		MOV	#10,R1		:SEEK ERROR
88	003515	104207	001430		MOV	#CONBLK,R0		:CONVERT BLOCK
89	003517	104672	000006		MOV	CYL(R0),R2		:CYLINDER FAILED ON
90	003521	022552			CALL	ERRMNT		:ERROR RETURN

1 003522 104204 006204
 2 003524 104203 005567
 3 003526 104302 001451
 4 003530 100632 000400
 5 003532 104203 001052
 6 003534 104632 000004
 7 003536 103202 177600
 8 003540 104207 001430
 9 003542 000672 000004
 10 003544 104632 000011
 11 003546 103202 177400
 12 003550 100672 000005
 13 003552 104632 000001
 14 003554 103202 007777
 15 003556 100672 000001
 16 003560 114002
 17 003561 100672 000000
 18 003563 104640 000002
 19 003566 104640 000003
 20 003571 104040 000731
 21 003573 104204 000733
 22 003575 022714
 23 003576 104207 000721
 24 003600 104203 122400
 25 003602 104302 001112
 26 003604 101023
 27 003605 100673 000004
 28 003607 104202 005567
 29 003611 100672 000001
 30 003613 104143
 31 003614 100673 000002
 32 003616 104643 000001
 33 003620 105303 001336
 34 003622 101203 060000
 35 003624 100673 000003
 36 003626 104203 000726
 37 003630 100673 000005
 38 003632 104303 001333
 39 003634 104304 001334
 40 003636 104302 000740
 41 003640 104207 000721
 42 003642 101207 100000
 43 003644 060012
 44 003645 060003
 45 003646 115001
 46 003647 013670
 47 003650 106300 001502
 48 003653 013657
 49 003654 115400 001504
 50 003656 003632
 51 003657 104303 001505
 52 003661 073670
 53 003662 022612
 54 003663 114000 001504
 55 003665 117400 001505
 56 003667 003632
 57 003670

RBNWRT: MOV #REVBUR, R4
 MOV #GDBLK, R3
 MOV EDC, R2
 MOV R2, RW.EDC(R3)
 MOV #SCR, R3
 MOV RBNTRK(R3), R2
 BIC #HI1BYTE, R2
 MOV #CONBLK, R0
 MOV R2, V3(R0)
 MOV LBNTRK(R3), R2
 BIC #HI1BYTE, R2
 MOV R2, V4(R0)
 MOV STCYL(R3), R2
 BIC #LO, R2
 MOV R2, V1+1(R0)
 CLR R2
 MOV R2, V1(R0)
 000733 RBNWHER: MOV 2(R4), TEMP2
 000734 MOV 3(R4), TEMP2+1
 MOV R4, DDJMMY
 MOV #TEMP2, R4
 CALL CVTSK
 MOV #WRBLK, R0
 MOV #WRCMD, R3
 MOV CURTRK, R2
 BIS R2, R3
 MOV R3, RW.CMD(R0)
 MOV #GDBLK, R2
 MOV R2, RW.BUF(R0)
 MOV (R4), R3
 MOV R3, RW.LOW(R0)
 MOV 1(R4), R3
 ADD ST.RBN, R3
 BIS #HD.RBN, R3
 MOV R3, RW.HI(R0)
 MOV #HSLIM-1, R3
 MOV R3, RW.DUM(R0)
 WRIT13: MOV HPREA, R3
 MOV DPREA, R4
 MOV UNIT, R2
 MOV #WRBLK, R0
 BIS #BIT15, R0
 XFC SIP
 XFC WRITE
 TST R1
 BEQ 2\$
 CMP RETRY, TMPTRY
 BEQ 1\$
 INC TMPTRY
 BR WRIT13
 1\$: MOV RECTMP, R3
 BMI 2\$
 CALL ERRHND
 CLR TMPTRY
 DEC RECTMP
 BR WRIT13
 2\$:

:POINT TO BUFFER
 :POINT TO GOOD BLOCK
 :GET GOOD EDC
 :STORE IT
 :POINT TO CHARACTERISTICS
 :GET RBN/TRACK
 :CLEAR HIGH GARBAGE
 :POINT TO CONVERT BLOCK
 :FOR CONVERT
 :GET LBN/TRACK
 :CLEAR HIGH BYTE
 :SET UP FOR RBN'S
 :STARTING CLYINDER
 :CLEAR REST OF WORD
 :STORE
 :FOR STORE
 :LOW ALWAYS ZERO
 :GET LOW ORDER RBN
 :GET HIGH ORDER
 :SAVE R4
 :FOR CONVERT
 :CONVERT AND SEEK
 :POINT TO COMMAND BLOCK
 :GET WRITE COMMAND
 :GET CURRENT TRACK
 :SET TRACK FOR WRITE
 :STORE IN COMMAND BLOCK
 :POINT TO BLOCK
 :STICK IN COMMAND BLOCK
 :GET LOW ORDER HEADER
 :STORE IN WRITE BLOCK
 :GET HIGH ORDER
 :ADD STARTING RBN BITS
 :GIVE RBN HEADER
 :STORE IN WRITE BLOCK
 :GET DUMMY SDI POINTER
 :POINT IN COMMAND BLOCK
 :GET HEADER PREAMBLE
 :GET DATA PREAMBLE
 :SET UNIT
 :MAKE SURE POINTING AT BLOCK
 :SET NO REVECTORING
 :WAIT FOR SECTOR PULSE
 :WRITE SECTOR
 :ANY ERROR ?
 :NO - SKIP RETRY
 :MAX ?
 :YES - TRY SOME RECOVERY
 :INC RETRY COUNT
 :DO RETRY
 :GET CURRENT ERROR RECOVERY LEVEL
 :IF NEGATIVE THEN FRIED
 :TRY RECOVERY
 :FOR INIT
 :DECREMENT IT
 :RETRY

58	003670	11400C	001504		CLR	TMPTRY	:RESET
59	003672	104300	001503	001505	MOV	RECOV,RECTMP	:DITTO
60	003675	104304	000731		MOV	DDUMMY,R4	:RESTORE R4
61	003677	105204	000004		ADD	#REVLN,R4	:POINT TO NEXT ENTRY
62	003701	115400	000736		INC	TEMP	:INC COUNTER
63	003703	106200	000100	000736	CMP	#64.,TEMP	:DONE ?
64	003706	053563			BNE	RNWH&R	:NO - REPEAT
65	003707	104207	001430		MOV	#CONBLK,R0	:FOR RESET
66	003711	114002			CLR	R2	:DITTO
67	003712	100672	000005		MOV	R2,V4(R0)	:RESET FOR NON-RBN'S
68	003714	000000			RETURN		

1									
2									
3									
4									
5	003715								
6	003723	114005							
7	003724	104203	001052						
8	003726	104632	000001						
9	003730	103202	007777						
10	003732	104207	001430						
11	003734	100672	000001						
12	003736	114002							
13	003737	100672	000000						
14	003741	104632	000011						
15	003743	103202	177400						
16	003745	100672	000004						
17	003747	104040	000731						
18	003751	114000	000732						
19	003753	104204	000731						
20	003755	104203	001227						
21	003757	021506							
22	003760	023455							
23	003761	104207	000721						
24	003763	104203	000726						
25	003765	100673	000005						
26	003767	104143							
27	003770	100673	000002						
28	003772	104643	000001						
29	003774	105303	001335						
30	003776	100673	000003						
31	004000	104203	006621						
32	004002	100673	000001						
33	004004	104203	013400						
34	004006	104301	001112						
35	004010	101013							
36	004011	100673	000004						
37	004013	104207	000721						
38	004015	104203	100000						
39	004017	100173							
40	004020	104302	000740						
41	004022	101207	100000						
42	004024	060012							
43	004025	060002							
44	004026	115001							
45	004027	054045							
46	004030	104173							
47	004031	102203	010000						
48	004033	014037							
49	004034	023027							
50	004035	115001							
51	004036	054045							
52	004037	104202	006621						
53	004041	022627							
54	004042	106623	000400						
55	004044	014102							
56	004045	106300	001502	001504	100\$:				
57	004050	014054							

READ A BLOCK OF THE RCT
 R4 ->BLOCK NUMBER OF RCT

RRC:

PUSHA

CLR R5
 MOV #SCR,R3
 MOV STCYL(R3),R2
 BIC #LO,R2
 MOV #CONBLK,R0
 MOV R2,V1+1(R0)
 CLR R2
 MOV R2,V1(R0)
 MOV LBNTRK(R3),R2
 BIC #HIBYTE,R2
 MOV R2,V3(R0)
 MOV R4,DDUMMY
 CLR DDUMMY+1
 MOV #DDUMMY,R4
 MOV #HOLD,R3

RCL:

CALL DADD
 CALL CS
 MOV #RDBLK,R0
 MOV #HSLIM-1,R3
 MOV R3,RW.DUM(R0)
 MOV (R4),R3
 MOV R3,RW.LOW(R0)
 MOV 1(R4),R3
 ADD ST.LBN,R3
 MOV R3,RW.HI(R0)
 MOV #RCTBUF,R3
 MOV R3,RW.BUF(R0)
 MOV #RWCMD,R3
 MOV CURTRK,R1
 BIS R1,R3

READ8:

MOV R3,RW.CMD(R0)
 MOV #RDBLK,R0
 MOV #RDCMD,R3
 MOV R3,(R0)
 MOV UNIT,R2
 BIS #BIT15,R0
 XFC SIP
 XFC READ
 TST R1
 BNE 100\$
 MOV (R0),R3
 BIT #ECCF,R3
 BEQ 101\$
 CALL ECCCK
 TST R1
 BNE 100\$

101\$:

MOV #RCTBUF,R2
 CALL CEDC
 CMP RW.EDC(R2),R3
 BEQ 102\$
 CMP RETRY,IMPTRY
 BEQ 1\$

;CLEAR ERROR COUNTER
 ;POINT TO CHARACTERISTICS
 ;GET LOW ORDER STARTING CYLINDER
 ;CLR REST OF WORD
 ;POINT TO CONVERT BLOCK
 ;STORE FOR CONVERT
 ;FOR STORE
 ;LOW ORDER ALWAYS ZERO
 ;GET LBN/TRK
 ;CLEAR HIGH BYTE
 ;STORE IN CONVERT BLOCK
 ;STORE BLOCK NUMBER
 ;FOR STORE
 ;POINT FOR ADD
 ;STARTING RCT LBN
 ;GET RCT LBN
 ;CONVERT FCT BLOCK NUMBER AND SEEK
 ;PREPARE FOR READ SECTORS
 ;POINTER TO DUMMY SDI BLOCK
 ;STORE IN COMMAND BLOCK
 ;LO ORDER BLOCK NUMBER
 ;STORE IN READ CMD BLOCK
 ;HI ORDER BLOCK NUM AND CODE
 ;ADD STARTING LBN BITS
 ;STORE IN READ CMD BLOCK
 ;LOAD ADDRESS OF DATA BUFFER
 ;STORE IN COMMAND BUFFER
 ;LOAD SDI READ COMMAND
 ;GET CURRENT HEAD NUMBER IN R1
 ;SET IT IN COMMAND
 ;STORE BACK
 ;MAKE SURE POINTING AT BLOCK
 ;MARK AS ONLY REQUEST
 ;STORE IN CMD BLOCK
 ;SDI INTERCONNECT
 ;SET NO REVECTORING
 ;WAIT FOR PULSE
 ;READ 1 SECTOR
 ;ANY ERRORS ?
 ;YES - TRY RECOVERY
 ;GET STATUS WORD
 ;ECC ERROR ?
 ;NOPE - VERIFY EDC
 ;CORRECT ECC
 ;TEST FLAG
 ;UNCORRECTABLE
 ;POINT TO BUFFER
 ;COMPUTE EDC
 ;O.K. ?
 ;YUP - CONSIDER GOOD
 ;MAX ?
 ;YES - TRY SOME RECOVERY

58	004051	115400	001504		INC	TMPTRY	:INC RETRY COUNT
59	004053	004013			BR	READ8	:DO RETRY
60	004054	104303	001505	1\$:	MOV	RECTMP,R3	:GET CURRENT ERROR RECOVERY LEVEL
61	004056	074065			BMI	2\$	:IF NEGATIVE THEN FRIED
62	004057	022612			CALL	ERRHND	:TRY RECOVERY
63	004060	114000	001504		CLR	TMPTRY	:FOR INIT
64	004062	117400	001505		DEC	RECTMP	:DECREMENT IT
65	004064	004013			BR	READ8	:RETRY
66	004055			2\$:			
67	004065	115400			INC	R5	:INCREMENT BAD COUNTER
68	004066	106305	001244		CMP	FCTCPY,R5	:ALL BAD ?
69	004070	014216			BEQ	RFTL	:YUP - ALL OVER
70	004071	104203	001242		MOV	#RCTFMT,R3	:SIZE OF TABLE - R4 -> BLOCK NUMBER
71	004073	021506			CALL	DADD	:ADD TO POINT TO NEXT COPY
72	004074	114000	001504		CLR	TMPTRY	:RESET RETRY LEVEL
73	004076	104300	001503	001505	MOV	RECOV,RECTMP	:DITTO RECOVERY LEVELS
74	004101	003760			BR	RCL	:BRANCH BACK
75	004102			102\$:			
76	004102	114000	001504		CLR	TMPTRY	:FOR RESET
77	004104	104300	001503	001505	MOV	RECOV,RECTMP	:GET RECOVERY LEVELS
78	004107	115005			TST	R5	:ANY ERRORS ?
79	004110	014207			BEQ	RLD	:NO - EXIT
80	004111	104203	001240		MOV	#FCTFMT,R3	:SIZE OF TABLE
81	004113	021524			CALL	DSUB	:GET BACK TO PREVIOUS COPY
82	004114	023455			CALL	CS	:CONVERT AND SEEK
83	004115	104207	000721		MOV	#WRBLK,R0	:POINT TO COMMAND BLOCK
84	004117	104203	122400		MOV	#WRCMD,R3	:GET WRITE COMMAND
85	004121	104302	001112		MOV	CURTRK,R2	:GET CURRENT TRACK
86	004123	101023			BIS	R2,R3	:SET TRACK FOR WRITE
87	004124	100673	000004		MOV	R3,RW.CMD(R0)	:STORE IN COMMAND BLOCK
88	004126	104203	006621		MOV	#RCTBUF,R3	:POINT TO BUFFER
89	004130	100673	000001		MOV	R3,RW.BUF(R0)	:STICK IN COMMAND BLOCK
90	004132	104143			MOV	(R4),R3	:GET LOW ORDER HEADER
91	004133	100673	000002		MOV	R3,RW.LOW(R0)	:STORE IN WRITE BLOCK
92	004135	104643	000001		MOV	1(R4),R3	:GET HIGH ORDER
93	004137	105303	001335		ADD	ST.LBN,R3	:ADD STARTING LBN BITS
94	004141	100673	000003		MOV	R3,RW.HI(R0)	:STORE IN WRITE BLOCK
95	004143	104203	000726		MOV	#HSLIM-1,R3	:GET DUMMY SDI POINTER
96	004145	100673	000005		MOV	R3,RW.DUM(R0)	:POINT IN COMMAND BLOCK
97	004147	104303	001333		MOV	HPREA,R3	:GET HEADER PREAMBLE
98	004151	104304	001334		MOV	DPREA,R4	:GET DATA PREAMBLE
99	004153	104302	000740		MOV	UNIT,R2	:SET UNIT
100	004155	104207	000721		MOV	#WRBLK,R0	:MAKE SURE POINTING AT BLOCK
101	004157	101207	100000		BIS	#BIT15,R0	:SET NO REVECTORING
102	004161	060012			XFC	SIP	:WAIT FOR SECTOR PULSE
103	004162	060003			XFC	WRITE	:WRITE SECTOR
104	004163	115001			TST	R1	:ANY ERROR ?
105	004164	014205			BEQ	2\$	:NO - SKIP RETRY
106	004165	106300	001502	001504	CMP	RETRY,TMPTRY	:MAX ?
107	004170	014174			BEQ	1\$	:YES - TRY SOME RECOVERY
108	004171	115400	001504		INC	TMPTRY	:INC RETRY COUNT
109	004173	004147			BR	WRIT10	:DO RETRY
110	004174	104303	001505	1\$:	MOV	RECTMP,R3	:GET CURRENT ERROR RECOVERY LEVEL
111	004176	074205			BMI	2\$	:IF NEGATIVE THEN FRIED
112	004177	022612			CALL	ERRHND	:TRY RECOVERY
113	004200	114000	001504		CLR	TMPTRY	:FOR INIT
114	004202	117400	001505		DEC	RECTMP	:DECREMENT IT

115	004204	004147		BR	WRIT10	:RETRY
116	004205		2\$:			
117	004205	117405		DEC	R5	:DEREMENT COUNTER
118	004206	004102		BR	RCD	:SEE IF ANY MORE TO DO
119	004207		RLD:	POPA		
120	004215	000000		RETURN		:ALL DONE
121	004216	104012	RFTL:	MOV	R1,R2	:XFC ERROR CODE
122	004217	104201		MOV	#14.,R1	:RCT READ ERROR
123	004221	022552		CALL	ERRMNT	:ERROR RETURN

1				VERIFY FCT	
2					
3					
4	004222	114000	001256	FCTCK- CLR	FCTCNT
5	004224	115400	001260	INC	FCTNPD
6	004226	104200	006204	MOV	#PRMBUF,BUFPNT
7	004231	104201	000017	MOV	#F6,R1
8	004233	022444		CALL	PAGE
9	004234	104205	006204	MOV	#PRMBUF,R5
10	004236	104203	126736	MOV	#M512,R3
11	004240	100153		MOV	R3,(R5)
12	004241	104050	001253	MOV	R5,BUFPNT
13	004243	104201	000030	MOV	#F9,R1
14	004245	022444		CALL	PAGE
15	004246	114000	001256	CLR	FCTCNT
16	004250	104300	001337	MOV	ST.XBN,CURXBN+1
17	004253	104300	001337	MOV	ST.XBN,CURBN+1
18	004256	114000	001115	CLR	CURXBN
19	004260	114000	001113	CLR	CURBN
20	004262	104203	001052	MOV	#SCR,R3
21	004264	104207	001430	MOV	#CONBLK,R0
22	004266	104632	000100	MOV	CYLB(R3),R2
23	004270	100672	000000	MOV	R2,V1(R0)
24	004272	104632	000001	MOV	CYLB+1(R3),R2
25	004274	100672	000001	MOV	R2,V1+1(R0)
26	004276	104303	001127	MOV	SECTRK,R3
27	004300	100673	000004	MOV	R3,V3(R0)
28	004302	114005		FCTCLF: CLR	R5
29	004303	104050	001245	MOV	R5,NEXT1
30	004305	104204	000736	FCTCL1: MOV	#TEMP,R4
31	004307	104300	001113	MOV	CURBN,TEMP
32	004312	104300	001114	MOV	CURBN+1,TEMP+1
33	004315	107300	001337	SUB	ST.XBN,TEMP+1
34	004320	023455		CALL	CS
35	004321	104207	000721	MOV	#RDBLK,R0
36	004323	104203	013400	MOV	#RWCMD,R3
37	004325	104302	001112	MOV	CURTRK,R2
38	004327	101023		BIS	R2,R3
39	004330	100673	000004	MOV	R3,RW.CMD(R0)
40	004332	104203	006204	MOV	#PRMBUF,R3
41	004334	100673	000001	MOV	R3,RW.BUF(R0)
42	004336	104303	001113	MOV	CURBN,R3
43	004340	100673	000002	MOV	R3,RW.LOW(R0)
44	004342	104303	001114	MOV	CURBN+1,R3
45	004344	101203	120000	BIS	#HD.XBN,R3
46	004346	100673	000003	MOV	R3,RW.HI(R0)
47	004350	104203	000726	MOV	#HSLIM-1,R3
48	004352	100673	000005	MOV	R3,RW.DUM(R0)
49	004354	104207	000721	READ9: MOV	#RDBLK,R0
50	004356	104203	100000	MOV	#RDCMD,R3
51	004360	100673	000000	MOV	R3,RW.STAT(R0)
52	004362	104302	000740	MOV	UNIT,R2
53	004364	101207	100000	BIS	#BIT15,R0
54	004366	060012		XFC	SIP
55	004367	060002		XFC	READ
56	004370	115001		TST	R1
57	004371	054407		BNE	100\$

;FOR INIT
 ;SO CHECK THE NULL BLOCK ALSO
 ;POINT TO BUFFER
 ;FCT READ OVERLAY
 ;READ FIRST BLOCK
 ;POINT TO BUFFER
 ;GET MODE INDICATOR
 ;SIGNAL DONE FORMAT
 ;STORE BUFFER POINTER
 ;FCT WRITE OVERLAY
 ;WRITE IT OUT
 ;FOR FCTCNT INIT
 ;ALSO INITIALIZE XBN COUNTER
 ;HIGH ORDER
 ;LOW ORDER ALWAYS 0
 ;AND BLOCK NUMBER
 ;POINT TO CHARACTERISTICS
 ;POINT TO CONVERT BLOCK
 ;GET LOW ORDER CYLINDER
 ;STORE IT
 ;GET HIGH ORDER
 ;STORE IT
 ;GET SECTORS/TRACK
 ;STORE IT
 ;CLEAR WRITE ERROR COUNT
 ;CLEAR REPEAT COUNT
 ;POINT TO BLOCK
 ;FOR CONVERSION
 ;DITTO
 ;SUBTRACT STARTING XBN BITS
 ;CONVERT AND SEEK
 ;POINT TO COMMAND BLOCK
 ;GET READ COMMAND
 ;GET CURRENT TRACK
 ;SET TRACK FOR WRITE
 ;STORE IN COMMAND BLOCK
 ;POINT TO BUFFER
 ;STICK IN COMMAND BLOCK
 ;GET LOW ORDER HEADER
 ;STORE IN WRITE BLOCK
 ;GET HIGH ORDER
 ;SET HEADER
 ;STORE IN WRITE BLOCK
 ;GET DUMMY SDI POINTER
 ;POINT IN COMMAND BLOCK
 ;MAKE SURE POINTING AT BLOCK
 ;GET READ COMMAND
 ;STORE IT
 ;SET UNIT
 ;SET NO REVECTORING
 ;WAIT FOR SECTOR PULSE
 ;READ SECTOR
 ;ANY ERROR ?
 ;YES - TRY RECOVERY

58	004372	104173				MOV	(R0),R3	;GET STATUS WORD
59	004373	102203	010000			BIT	#ECCF,R3	;ECC ERROR ?
60	004375	014401				BEQ	101\$	;NOPE - VERIFY EDC
61	004376	023027				CALL	ECCCK	;CORRECT ECC
62	004377	115001				TST	R1	;TEST FLAG
63	004400	054407				BNE	100\$	;UNCORRECTABLE
64	004401	104202	006204		101\$:	MOV	#PRMBUF,R2	;POINT TO BUFFER
65	004403	022627				CALL	CEDC	;COMPUTE EDC
66	004404	106623	000400			CMP	RW.EDC(R2),R3	;O.K. ?
67	004406	014427				BEQ	102\$	;YUP - CONSIDER GOOD
68	004407	106300	001502	001504	100\$:	CMP	RETRY,TMPTRY	;MAX ?
69	004412	014416				BEQ	1\$	;YES - TRY SOME RECOVERY
70	004413	115400	001504			INC	TMPTRY	;INC RETRY COUNT
71	004415	074354				BR	READ9	;DO RETRY
72	004416	104303	001505		1\$:	MOV	RECTMP,R3	;GET CURRENT ERROR RECOVERY LEVEL
73	004420	074430				BMI	YES	;IF NEGATIVE THEN FRIED
74	004421	022612				CALL	ERRHND	;TRY RECOVERY
75	004422	114000	001504			CLR	TMPTRY	;FOR INIT
76	004424	117400	001505			DEC	RECTMP	;DECREMENT IT
77	004426	004354				BR	READ9	;RETRY
78	004427				102\$:			
79	004427				2\$:			
80	004427	115405				INC	R5	;NO - INCREMENT COUNTER
81	004430	115400	001245		YES:	INC	NEXT1	;INCREMENT IT
82	004432	106205	000002			CMP	#2,R5	;FOUND 2 GOOD ONES ?
83	004434	014454				BEQ	FCTCKD	;YUP - GO TO NEXT BLOCK
84	004435	104204	001113			MOV	#CURBN,R4	;FOR ADD
85	004437	104203	001240			MOV	#FCTFMT,R3	;FOR ADD
86	004441	021506				CALL	DADD	;POINT TO NEXT COPY
87	004442	114000	001504			CLR	TMPTRY	;RESET RETRY LEVEL
88	004444	104300	001503	001505		MOV	RECOV,RECTMP	;DITTO RECOVERY LEVELS
89	004447	106300	001244	001245		CMP	FCTCPY,NEXT1	;DONE THIS SECTOR ?
90	004452	054305				BNE	FCTCL1	;NO - WRITE NEXT FCT COPY
91	004453	004503				BR	FCTCKE	;2 NOT GOOD - TROUBLE
92	004454	102200	002000	001217	FCTCKD:	BIT	#BSTGS,FLAG	;BEST GUESS ?
93	004457	054502				BNE	4\$	;YUP - ONLY CHECK FIRST BLOCK
94	004460	060022				XFC	UPDATE	;LET HOST KNOW STILL ALIVE
95	004461	115400	001256			INC	FCTCNT	;INCREMENT IT
96	004463					DUBINC	CURXBN	;INCREMENT IT
97	004470	104300	001115	001113		MOV	CURXBN,CURBN	;GET LOW ORDER
98	004473	104300	001116	001114		MOV	CURXBN+1,CURBN+1	;GET HIGH ORDER
99	004476	106300	001260	001256		CMP	FCTNPD,FCTCNT	;DONE ?
100	004501	054302				BNE	FCTCLP	;NOPE - DO NEXT SECTOR
101	004502	000000			4\$:	RETURN		
102	004503	104201	000011		FCTCKE:	MOV	#9,R1	;SIGNAL ERROR
103	004505	104302	001256			MOV	FCTCNT,R2	;BLOCK FAILED ON
104	004507	022552				CALL	ERRMNT	;ERROR RETURN

1				...	VERIFY RCT	
2				...		
3				...		
4	004510	115400	001261		RCTICK: INC	RCTLBN
5	004512	104300	001133	001227	MOV	LBNLBN,HOLD
6	004515	104300	001134	001230	MOV	LBNLBN+1,HOLD+1
7	004520	104203	001475		MOV	#TOTRCT,R3
8	004522	104204	001227		MOV	#HOLD,R4
9	004524	021524			CALL	DSUB
10	004525	104300	001227	001113	MOV	HOLD,CURBN
11	004530	104300	001227	001115	MOV	HOLD,CURLBN
12	004533	104300	001230	001114	MOV	HOLD+1,CURBN+1
13	004536	104300	001230	001116	MOV	HOLD+1,CURLBN+1
14	004541	104200	000001	001501	MOV	#1,COUNT
15	004544	114005			RCTCLP: CLR	R5
16	004545	104050	001245		MOV	R5,NEXT1
17	004547	104203	001052		RCTCL1: MOV	#SCR,R3
18	004551	104632	000011		MOV	LBNTRK(R3),R2
19	004553	103202	177400		BIC	#HIBYTE,R2
20	004555	104207	001430		MOV	#CONBLK,R0
21	004557	100672	000004		MOV	R2,V3(R0)
22	004561	104632	000001		MOV	STCYL(R3),R2
23	004563	103202	007777		BIC	#LO,R2
24	004565	100672	000001		MOV	R2,V1+1(R0)
25	004567	114002			CLR	R2
26	004570	100672	000000		MOV	R2,V1(R0)
27	004572	104204	001113		MOV	#CURBN,R4
28	004574	023455			CALL	CS
29	004575	104207	000721		MOV	#RDBLK,R0
30	004577	104203	013400		MOV	#RWCMD,R3
31	004601	104302	001112		MOV	CURTRK,R2
32	004603	101023			BIS	R2,R3
33	004604	100673	000004		MOV	R3,RW.CMD(R0)
34	004606	104203	006204		MOV	#PRMBUF,R3
35	004610	100673	000001		MOV	R3,RW.BUF(R0)
36	004612	104303	001113		MOV	CURBN,R3
37	004614	100673	000002		MOV	R3,RW.LOW(R0)
38	004616	104303	001114		MOV	CURBN+1,R3
39	004620	105303	001335		ALD	ST.LBN,R3
40	004622	101203	000000		BIS	#HD.LBN,R3
41	004624	100673	000003		MOV	R3,RW.HI(R0)
42	004626	104203	000726		MOV	#HSLIM-1,R3
43	004630	100673	000005		MOV	R3,RW.DIM(R0)
44	004632	104207	000721		READ10: MOV	#RDBLK,R0
45	004634	104203	100000		MOV	#RDCMD,R3
46	004636	100173			MOV	R3,(R0)
47	004637	104302	000740		MOV	UNIT,R2
48	004641	101207	100000		BIS	#BIT15,R0
49	004643	060012			XFC	SIP
50	004644	060002			XFC	READ
51	004645	115001			TST	R1
52	004646	054664			BNE	100\$
53	004647	104173			MOV	(R0),R3
54	004650	102203	010000		BIT	#ECCF,R3
55	004652	014656			BEQ	101\$
56	004653	023027			CALL	ECCCK
57	004654	115001			TST	R1

:CHECK NON-PAD PLUS NULL BLOCK
 :GET LOW ORDER COUNT OF LBN'S
 :GET HIGH ORDER
 :FOR SUBTRACT
 :DITTO
 :GET STARTING RCT LBN
 :GET STARTING RCT BLOCK NUMBER
 :ALSO SAVE
 :GET HIGH ORDER
 :AND SAVE

:CLEAR ERROR COUNTER
 :INIT COPY COUNT
 :POINT TO CHARACTERISTICS
 :GET LBN/TRACK
 :CLEAR HIGH BYTE
 :POINT TO CONVERT BLOCK
 :FOR CONVERT
 :STARTING CLYLINDER
 :CLEAR REST OF WORD
 :STORE
 :FOR STORE
 :LOW ORDER ALWAYS ZERO
 :FOR CONVERT
 :CONVERT AND SEEK
 :POINT TO COMMAND BLOCK
 :GET READ COMMAND
 :GET CURRENT TRACK
 :SET TRACK FOR WRITE
 :STORE IN COMMAND BLOCK
 :POINT TO BUFFER
 :STICK IN COMMAND BLOCK
 :GET LOW ORDER HEADER
 :STORE IN WRITE BLOCK
 :GET HIGH ORDER
 :ADD STARTING LBN BITS
 :SET HEADER
 :STORE IN WRITE BLOCK
 :GET DUMMY SDI POINTER
 :POINT IN COMMAND BLOCK
 :MAKE SURE POINTING AT BLOCK
 :GET STATUS WORD
 :STORE IT
 :SET UNIT
 :SET NO REVECTORING
 :WAIT FOR SECTOR PULSE
 :WRITE SECTOR
 :ANY ERROR ?
 :YES - TRY RECOVERY
 :GET STATUS WORD
 :ECC ERROR ?
 :NOPE - VERIFY EDC
 :CORRECT ECC
 :TEST FLAG

58	004655	054664			BNE	100\$	:UNCORRECTABLE
59	004656	104202	006204	101\$:	MOV	#PRMBUF,R2	:POINT TO BUFFER
60	004660	022627			CALL	CEDC	:COMPUTE EDC
61	004661	106623	000400		CMP	RW.EDC(R2),R3	:O.K. ?
62	004663	014704			BEQ	102\$	:YUP - CONSIDER GOOD
63	004664	106300	001502	001504	100\$:	CMP	RETRY,TMPTRY
64	004667	014673			BEQ	1\$	:MAX ?
65	004670	115400	001504		INC	TMPTRY	:YES - TRY SOME RECOVERY
66	004672	004632			BR	READ10	:INC RETRY COUNT
67	004673	104303	001505	1\$:	MOV	RECTMP,R3	:DO RETRY
68	004675	074705			BMI	RCTNGD	:GET CURRENT ERROR RECOVERY LEVEL
69	004676	022612			CALL	ERRHND	:IF NEGATIVE SKIP GOOD INCREMENT
70	004677	114000	001504		CLR	TMPTRY	:TRY RECOVERY
71	004701	117400	001505		DEC	RECTMP	:FOR INIT
72	004703	004632			BR	READ10	:DECREMENT IT
73	004704			102\$:			:RETRY
74	004704			2\$:			
75	004704	115405			INC	R5	:YUP - INCREMENT COUNTER
76	004705	115400	001245	RCTNGD:	INC	NEXT1	:INCREMENT IT
77	004707	114000	001504		CLR	TMPTRY	:FOR RESET
78	004711	104300	001503	001505	MOV	RECOV,RECTMP	:GET RECOVERY LEVELS
79	004714	106205	000002		CMP	#2,R5	:FOUND 2 GOOD ONES ?
80	004716	014736			BEQ	RCTCKD	:YUP - GO TO NEXT BLOCK
81	004717	104204	001113		MOV	#CURBN,R4	:FOR ADD
82	004721	104203	001242		MOV	#RCTFMT,R3	:FOR ADD
83	004723	021506			CALL	DADD	:POINT TO NEXT COPY
84	004724	114000	001504		CLR	TMPTRY	:RESET RETRY LEVEL
85	004726	104300	001503	001505	MOV	RECOV,RECTMP	:DITTO RECOVERY LEVELS
86	004731	106300	001244	001245	CMP	FCTCPY,NEXT1	:DONE THIS SECTOR ?
87	004734	054547			BNE	RCTCL1	:NO - READ NEXT FCT COPY
88	004735	004761			BR	RCTCKE	:2 NOT GOOD - TROUBLE
89	004736	060022		RCTCKD:	XFC	UPDATE	:LET HOST KNOW STILL ALIVE
90	004737	115400	001501		INC	COUNT	:INCRMENT IT
91	004741				DUBINC	CURLBN	:INCREMENT IT
92	004746	104300	001115	001113	MOV	CURLBN,CURBN	:GET LOW ORDER
93	004751	104300	001116	001114	MOV	CURLBN+1,CURBN+1	:GET HIGH ORDER
94	004754	106300	001261	001501	CMP	RCTLBN,COUNT	:DONE ?
95	004757	054544			BNE	RCTCLP	:NOPE - DO NEXT SECTOR
96	004760	000000			RETURN		
97	004761	104201	000013	RCTCKE:	MOV	#11,R1	:SET ERROR CODE
98	004763	104302	001501		MOV	COUNT,R2	:RCT BLOCK FAILED ON
99	004765	022552			CALL	ERRMNT	:ERROR RETURN

```

1      .SBTTL FCT WRITE OVERLAY (F9)
2 004766 DMOVLY F9,START
3
4      WRITE AN FCT BLOCK
5
6
7 003047 104200 000030 001153 FCTWRT: MOV #F9,CUROVL      ;FOR INIT
8 003052 114005          CLR R5                      ;CLEAR ERROR COUNTER
9 003053 104050 001245      MOV R5,NEXT1              ;INIT NEXT COPY COUNTER
10 003055 104204 001256     MOV #FCTCNT,R4            ;POINT TO FCT BLOCK NUMBER
11 003057 104203 001466     MOV #ONE,R3              ;FOR SUB
12 003061 021524          CALL DSUB                   ;SUB TO GET RIGHT ONE
13 003062 104305 001253     MOV BUFPNT,R5            ;GET BUFFER POINTER
14 003064 104303 001143     FCTRLP: MOV LBNCYL,R3     ;GET LBN CYLINDERS
15 003066 104207 001430     MOV #CONBLK,R0           ;POINT TO CONVERT BLOCK
16 003070 100673 000000     MOV R3,V1(R0)            ;STORE FOR CONVERT
17 003072 104303 001144     MOV LBNCYL+1,R3         ;GET HIGH ORDER
18 003074 100673 000000     MOV R3,V1+1(R0)          ;STORE IT
19 003076 104303 001127     MOV SECTRK,R3            ;GET SECTORS/TRACK
20 003100 100673 000004     MOV R3,V3(R0)            ;STORE IT
21 003102 022714          CALL CVTSK                  ;CONVERT AND SEEK
22 003103 104207 000721     MOV #WRBLK,R0            ;POINT TO COMMAND BLOCK
23 003105 104203 122400     MOV #WRCMD,R3            ;GET WRITE COMMAND
24 003107 104302 001112     MOV CURTRK,R2            ;GET CURRENT TRACK
25 003111 101023          BIS R2,R3                  ;SET TRACK FOR WRITE
26 003112 100673 000004     MOV R3,RW.CMD(R0)        ;STORE IN COMMAND BLOCK
27 003114 104052          MOV R5,R2                  ;POINT TO BUFFER
28 003115 022627          CALL CEDC                   ;COMPUTE EDC - RETURNED IN R3
29 003116 100623 000400     MOV R3,RW.EDC(R2)        ;STORE IT
30 003120 100672 000001     MOV R2,RW.BUF(R0)        ;STICK IN COMMAND BLOCK
31 003122 104143          MOV (R4),R3                 ;GET LOW ORDER HEADER
32 003123 100673 000002     MOV R3,RW.LOW(R0)        ;STORE IN WRITE BLOCK
33 003125 104643 000001     MOV 1(R4),R3             ;GET HIGH ORDER
34 003127 105303 001337     ADD ST.XBN,R3            ;ADD STARTING XBN BITS
35 003131 101203 120000     BIS #HD.XBN,R3           ;SET HEADER
36 003133 100673 000003     MOV R3,RW.HI(R0)         ;STORE IN WRITE BLOCK
37 003135 104203 000726     MOV #HSLIM-1,R3         ;GET DUMMY SDI POINTER
38 003137 100673 000005     MOV R3,RW.DUM(R0)        ;POINT IN COMMAND BLOCK
39 003141 104303 001333     WRITE5: MOV HPREA,R3      ;GET HEADER PREAMBLE
40 003143 104304 001334     MOV DPREA,R4            ;GET DATA PREAMBLE
41 003145 104302 000740     MOV UNIT,R2             ;SET UNIT
42 003147 104207 000721     MOV #WRBLK,R0            ;MAKE SURE POINTING AT BLOCK
43 003151 101207 100000     BIS #BIT15,R0           ;SET NO RVECTURING
44 003153 060012          XFC SIP                    ;WAIT FOR SECTOR PULSE
45 003154 060003          XFC WRITE                  ;WRITE SECTOR
46 003155 115001          TST R1                     ;ANY ERROR ?
47 003156 013200          BEQ FWGOOD                   ;NOPE
48 003157 106300 001502 001504 CMP RETRY,TMPTRY     ;MAX ?
49 003162 013166          BEQ 1$                      ;YES - TRY SOME RECOVERY
50 003163 115400 001504     INC TMPTRY                ;INC RETRY COUNT
51 003165 003141          BR WRITES                    ;DO RETRY
52 003166 104303 001505     1$: MOV RECTMP,R3        ;GET CURRENT ERROR RECOVERY LEVEL
53 003170 073177          BMI 2$                      ;IF NEGATIVE THEN FRIED
54 003171 022612          CALL ERRHND                 ;TRY RECOVERY
55 003172 114000 001504     CLR TMPTRY                ;FOR INIT
56 003174 117400 001505     DEC RECTMP               ;DECREMENT IT
57 003176 003141          BR WRITES                    ;RETRY

```

58	003177			25:	INC	R5	:YUP - INCREMENT COUNTER
59	003177	115405			INC	NEXT1	:INCREMENT IT
60	003200	115400	001245	FWGOOD:	CLR	TMPTRY	:FOR RESET
61	003202	114000	001504		MOV	RECOV,RECTMP	:GET RECOVERY LEVELS
62	003204	104300	001503	001505	MOV	#FCTCNT,R4	:FOR ADD
63	003207	104204	001256		MOV	#FCTFMT,R3	:FOR ADD
64	003211	104203	001240		CALL	DADD	:POINT TO NEXT COPY
65	003213	021506			CMP	FCTCPY,NEXT1	:DONE THIS SECTOR ?
66	003214	106300	001244	001245	BNE	FCTRLP	:NO - WRITE NEXT FCT COPY
67	003217	053064			CMP	FCTCPY,R5	:ERROR ON EVERY WRITE ?
68	003220	106305	001244		BEQ	FCWERR	:YUP - BIG TROUBLE
69	003222	013244			MOV	NEXT1,R3	:ANY REPEATS ?
70	003223	104303	001245	FCFXLP:	BEQ	FWTDON	:NO
71	003225	013236			MOV	#FCTCNT,R4	:TO GET IT BACK
72	003226	104204	001256		MOV	#FCTFMT,R3	:DITTO
73	003230	104203	001240		CALL	DSUB	
74	003232	021524			DEC	NEXT1	:SUB IT
75	003233	117400	001245		BR	FCFXLP	:REPEAT
76	003235	003223			FWTDON:	DURINC	:PUT BACK THE WAY IT WAS
77	003236				RETURN		
78	003243	000000			FCWERR:	MOV	:XFC ERROR CODE
79	003244	104012			MOV	#15.,R1	:RCT WRITE ERROR
80	003245	104201	000017		CALL	ERRMNT	:ERROR RETURN
81	003247	022552					

.SBTTL PBN->D,X,L,RBN CONVERSION OVERLAY (G5)
 DMOVLV G5,START

CONVERT FROM PBN TO D,X,L,RBN

003047 104303 001111
 003051 103203 170000
 003053 104030 001111
 003055 104300 001133 000731
 003060 104300 001134 000732
 003063 104204 000731
 003065 104203 001135
 003067 021506
 003070 104203 001110
 003072 021626
 003073 073142
 003074 104204 001110
 003076 104203 000731
 003100 021524
 003101 104203 001137
 003103 021626
 003104 073125
 003105 104204 001110
 003107 104203 001137
 003111 021524
 003112 104641 000001
 003114 105301 001340
 003116 103201 170000
 003120 101201 140000
 003122 104010 001111
 003124 003276
 003125 104204 001110
 003127 104641 000001
 003131 105301 001337
 003133 103201 170000
 003135 101201 120000
 003137 104010 001111
 003141 003276

PCON:

MOV CURPBN+1,R3
 BIC #HD.CLR,R3
 MOV R3,CURPBN+1
 MOV LBNLBN,DDUMMY
 MOV LBNLBN+1,DDUMMY+1
 MOV #DDUMMY,R4
 MOV #RBNLBN,R3
 CALL DADD
 MOV #CURPBN,R3
 CALL DCMPL
 BMI LRBN
 MOV #CURPBN,R4
 MOV #DDUMMY,R3
 CALL DSUB
 MOV #XBNSEC,R3
 CALL DCMPL
 BMI XBNFND
 MOV #CURPBN,R4
 MOV #XBNSEC,R3
 CALL DSUB
 MOV 1(R4),R1
 ADD ST.DBN,R1
 BIC #HD.CLR,R1
 BIS #HD.DBN,R1
 MOV R1,CURPBN+1
 BR PDONE
 XBNFND: MOV #CURPBN+1,R4
 MOV 1(R4),R1
 ADD ST.XBN,R1
 BIC #HD.CLR,R1
 BIS #HD.XBN,R1
 MOV R1,CURPBN+1
 BR PDONE

;GET PBN HEADER
 ;CLEAR THE HEADER
 ;STORE BACK
 ;GET TOTAL NUMBER OF LBN'S
 ;HIGH ORDER
 ;FOR ADD
 ;TOTAL NUMBER OF RBN'2
 ;ADD TO GET TOTAL BLOCKS IN LBN AREA
 ;CURRENT PBN
 ;IS IT LBN OR RBN ?
 ;YUP - GO COMPUTE IT
 ;ELSE X OR DBN - GET VALUE
 ;TOTAL LBN+RBN BLOCKS
 ;SUBTRACT
 ;TOTAL XBN SECTORS
 ;IS IT AN XBN ?
 ;YUP - GO FIXIT
 ;ELSE DBN - GET VALUE
 ;TOTAL XBN SECTORS
 ;SUBTRACT TO GET RELATIVE DBN
 ;GET HIGH ORDER
 ;ADD TO GET ABSOLUTE DBN
 ;CLEAR THE HEADER
 ;MARK AS DBN
 ;STORE BACK
 ;CLEAN UP AND RETURN
 ;POINT TO BLOCK NUMBER
 ;GET HIGH ORDER
 ;ADD TO GET ABSOLUTE XBN
 ;CLEAR HEADER
 ;MARK AS XBN
 ;STORE BACK
 ;CLEAN UP AND RETURN

LRBN:

MOV SECTRK,DDUMMY
 MOV SECTRK+1,DDUMMY+1
 MOV #DDUMMY,R3
 MOV #CURPBN,R4
 CALL DDIV
 MOV #SCR,R5
 MOV LBNTRK(R5),R3
 BIC #HIBYTE,R3
 MOV #DDUMMY,R1
 MOV (R1),R4
 CMP R4,R3
 BMI LBNFND
 MOV RBNTRK(R5),R3
 BIC #HIBYTE,R3
 MOV R3,TEMP
 CLR TEMP+1
 MOV #TEMP,R3

;GET LOW ORDER SECTRS/TRACK
 ;HIGH ORDER
 ;FOR DIVIDE
 ;DITTO
 ;DIVIDE TO GET TRACK NUMBER AND REMAINDER
 ;POINT TO CHARACTERISTICS
 ;GET LBN/TRK
 ;CLEAR HIGH BYTE
 ;POINT TO REMAINDER
 ;GET IT
 ;COMPARE
 ;IF MINUS THEN LBN
 ;GET RBN/TRACK
 ;CLEAR OUT GARBAGE
 ;STORE IT
 ;FOR STORE
 ;FOR MULTIPLY

003142 104300 001127 000731
 003145 104300 001130 000732
 003150 104203 000731
 003152 104204 001110
 003154 021570
 003155 104205 001052
 003157 104653 000011
 003161 103203 177400
 003163 104201 000731
 003165 104114
 003166 106043
 003167 073242
 003170 104653 000004
 003172 103203 177600
 003174 104030 000736
 003176 114000 000737
 003200 104203 000736

58	003202	104204	001110	MOV	#CURPBN,R4	:DITTO - NUM OF TRACKS
59	003204	021542		CALL	DMUL	:MULTIPLY BY TRACK NUMBER
60	003205	104204	000731	MOV	#DDUMMY,R4	:FOR SUBTRACT
61	003207	104653	000111	MOV	LBNTRK(R5),R3	:GET LBN/TRK
62	003211	103203	177400	BIC	#HIBYTE,R3	:CLEAR HIGH BYTE
63	003213	104030	000736	MOV	R3,TEMP	:STORE IT
64	003215	114000	000737	CLR	TEMP+1	:FOR CLEAR
65	003217	104203	000736	MOV	#TEMP,R3	:POINT FOR SUBTRACT
66	003221	021524		CALL	DSUB	:SUBTRACT TO GET RESIDUE RBN
67	003222	104204	001110	MOV	#CURPBN,R4	:TO GET RBN NUMBER
68	003224	104203	000731	MOV	#DDUMMY,R3	:DITTO
69	003226	021506		CALL	DADD	:GIVES RELATIVE RBN
70	003227	104641	000001	MOV	1(R4),R1	:GET HIGH ORDER
71	003231	105301	001336	ADD	ST.RBN,R1	:ADD TO GET ABSOLUTE RBN
72	003233	103201	170000	BIC	#HD.CLR,R1	:CLEAR TH EHADER
73	003235	101201	060000	BIS	#HD.RBN,R1	:SET AS A RBN
74	003237	104010	001111	MOV	R1,CURPBN+1	:STORE BACK
75	003241	003276		BR	PDONE	:CLEAN UP AND RETURN
76						
77						
78	003242	104204	001110	LBNFND: MOV	#CURPBN,R4	:MULT NUM OF TRACKS
79	003244	104653	000011	MOV	LBNTRK(R5),R3	:GET LBN/TRK
80	003246	103203	177400	BIC	#HIBYTE,R3	:CLEAR HIGH BYTE
81	003250	104030	000736	MOV	R3,TEMP	:STORE IT
82	003252	114000	000737	CLR	TEMP+1	:FOR CLEAR
83	003254	104203	000736	MOV	#TEMP,R3	:POINT FOR MULT
84	003256	021542		CALL	DMUL	:GET LBN'S
85	003257	104203	000731	MOV	#DDUMMY,R3	:PLUS RESIDUE
86	003261	021506		CALL	DADD	:GIVES LBN NUMBER
87	003262	104207	001052	MOV	#SCR,R0	:POINT TO CHARACTERISTICS
88	003264	104641	000001	MOV	1(R4),R1	:GET HIGH ORDER
89	003266	105301	001335	ADD	ST.LBN,R1	:ADD TO GET ABSOLUTE LBN
90	003270	103201	170000	BIC	#HD.CLR,R1	:CLEAR HEADER
91	003272	101201	000000	BIS	#HD.LBN,R1	:SET AS LBN
92	003274	104010	001111	MOV	R1,CURPBN+1	:STORE BACK
93	003276	104300	001110	001113 PDONE: MOV	CURPBN,CURBN	:GET LOW ORDER
94	003301	104300	001111	001114 MOV	CURPBN+1,CURBN+1	:HIGH ORDER
95	003304	000000		RETURN		
96	003305			DMEND		
97		000001		.END		

ACC	001024	CLEWRT	003270	DCLR	001026	ERDN	- 000010	FIXLP	003503
ACCESS	002235	CLHERE	003303	DCMP	001626	ERECOV	001102	FKIP1	004121
AGAIN	003544	CLSKP2	003165	DCMP1	001651	ERFLAG	001221	FKIP10	004145
ALDONE	003264	CLSKP3	003201	DCMP2	001644	ERLEN	= 000002	FKIP2	004102
ALLOVR	002603	CLSKP4	003200	DCMP3	001657	ERPNT	001252	FKIP9	004460
ALLOV1	002611	CMDBUF	= 006621	DCMP4	001637	ERR	001226	FKP1	004136
AOUT	002362	CMPCAT	= 000006	DDIV	001570	ERRBUF	001224	FLAG	001217
ATTN	= 000002	CNT	001247	DDUMMY	000731	ERRCNT	001453	FLAG1	001220
A*TN1	002353	CNTCYL	001460	DEAD	= 000020	ERRHND	002612	FLGDON	004427
BADEDC	001452	CONBLK	001430	DESC	003050	ERRLN	= 000003	FLGSET	004373
BADPEJN	001223	CONINT	003067	DFCT	004423	ERRLST	002624	FLGSKP	004405
BADREN	003645	COUNT	001501	DINIT	- 000011	ERRMNT	002552	FLIP	003136
BD	= 020000	CR	001037	DIS	001021	ERROR	003465	FLKIP1	004434
BDHD	= 000040	CR.ACC	000765	DISCON	002324	ERRSYM	- 000002	FLKIP2	004415
BDIRCT	004136	CR.CLR	000771	DLERR	003113	ERRT	002101	FLKP1	004451
BIT0	= 000001	CR.DIS	000755	DLERT	003470	ER1	003753	FODONE	003227
BIT1	= 000002	CR.ERV	001005	DLL	000400	FBDHD	= 010000	FOLOOP	003060
BIT10	= 002000	CR.GCR	000745	DLLRET	003446	FBEGIN	003276	FORERR	003431
BIT11	= 004000	CR.GSR	000751	DLLRT1	003432	FBEG2	003311	FORMAT	= 000001
BIT12	= 010000	CR.GST	000741	DMBUF	001263	FCFXLP	003223	FRCPY	= 000001
BIT13	= 020000	CR.ONL	001011	DMBUFL	= 000016	FCLR	= 177760	FRDONE	003750
BIT14	= 040000	CR.RCL	001001	DMFLG	000004	FMSG	001321	FRSKP	003744
BIT15	= 100000	CR.RUN	000761	DMOST	= 000001	FCNT	001237	FSER	= 000002
BIT2	= 000004	CR.SEK	000775	DMSG1	003054	FCPG	003657	FT.BUF	= 000000
BIT3	= 000010	CS	003455	DMSG1L	= 000007	FCT	= 000013	FT.HI	= 000002
BIT4	= 000020	CSKIP	003522	DMUL	001542	FCTAVL	= 000001	FT.LOW	= 000001
BIT5	= 000040	CSKIP1	003534	DMUNIT	= 000003	FCTBAD	= 000004	FULL	= 040000
BIT6	= 000100	CSKIP2	003533	DOLBN	003061	FCTCK	004222	FWGOOD	003200
BIT7	= 000200	CSKIP3	003773	DONDLL	003451	FCTCKD	004454	FWRD	001445
BIT8	= 000400	CSKIP6	003743	DONE	= 000021	FCTCKE	004503	FWTDON	003236
BIT9	= 001000	CSKIP7	004000	DONE1	003206	FCTCLP	004302	F1	= 000000
BLKFND	003176	CURBN	001113	DONMSG	001327	FCTCL1	004305	F2	= 000003
BMAX	= 007775	CURGRP	001463	DPBN	003307	FCTCNT	001256	F3	= 000006
BOTTOM	003322	CURLBN	001115	DPREA	001334	FCTCPY	001244	F4	= 000011
BRBN	004034	CUROVL	001153	DSUB	001524	FCTEMT	= 000002	F5	= 000014
BREAK	= 000000	CURPBN	001110	DUPOVL	= 007774	FCTFLG	= 000025	F6	= 000017
BSTGS	= 002000	CURRBN	001106	DWRD	001450	FCTFMT	001240	F7	= 000022
BUFMSK	= 007777	CURTRK	001112	DXBN	003063	FCTNPD	001260	F8	= 000025
BUFPNT	001253	CURXBN	001115	DXCH	003500	FCTPTR	001255	F9	= 000030
BUF1	= 004535	CUTOF	001235	DXCHEC	003477	FCTRCT	003212	GCR	001016
BUF2	= 005152	CVT	= 000020	DXERR	003406	FCTREV	001311	GDBLK	= 005567
BUF3	= 005567	CVTERR	002750	DXFCPG	003407	FCTRLP	003064	GDECC	003046
BUF4	= 006204	CVTSK	002714	DXFCP1	003370	FCTSKP	003466	GDFCT	003260
BUF5	= 006621	CYL	= 000006	DXFORM	003114	FCTSK1	003460	GDONE	004337
BUF6	= 007275	CYLBN	= 000000	DXTRK	003052	FCTSLP	003644	GETUNT	004203
BUMER	003064	CYLMMSG	001316	EAGAIN	002636	FCTSP	003762	GOBAD	= 000020
CBUF	001444	CYLNUM	001125	ECC	= 000015	FCTSZ	= 000010	GOVER	004212
CDONE	004044	C512	= 000016	ECCCK	003027	FCTWRT	003047	GRP	= 000010
CEDC	002627	DADD	001506	ECCF	= 010000	FCWERR	003244	GRPCNT	001464
CHAR	001772	DADD1	001515	ECHO	= 000010	FDAT	= 000012	GRPCYL	= 000002
CHRDNE	= 010000	DAT	= 000011	EDC	001451	FDLL	003272	GSKIP	004315
CKR	003513	DATA	= 000005	EIMAGE	001231	FERR	003376	GSKIP1	004232
CLEAR	002243	DATE	001301	EMAX	001225	FILLIT	003666	GSKIP2	004250
CLEDON	003174	DBBAD	001314	EMSG	001324	FINI	= 040000	GSKIP3	004267
CLELP	003106	DBD	= 000006	ENTRY	000714	FIXBLK	003466	GSR	001017
CLELP2	003076	DBN	= 000010	EORCT	003173	FIXFCT	004015	GST	001015
CLESKP	003144	DBNCYL	= 000022	ERCV	000002	FIXIT	004046	GSTATS	001662

SYMBOL TABLE

```
G1      = 000060
G2      = 000033
G3      = 000036
G4      = 000041
G5      = 000044
G6      = 000047
G7      = 000047
G8      = 000052
HASH    = 003050
HD.BAD  = 110000
HD.CLR  = 170000
HD.CUR  = 001462
HD.DBN  = 140000
HD.LBN  = 000000
HD.PRIV = 050000
HD.RBN  = 060000
HD.REV  = 030000
HD.XBN  = 120000
HEAD    = 000005
HERE    = 003653
HGHPBN  = 001154
HIBYTE  = 177400
HI1BYT  = 177600
HI2BYT  = 177700
HKIP    = 003055
HKIP1   = 003101
HOLD    = 001227
HOLDBN  = 001117
HOLDPN  = 001123
HOLRBN  = 001121
HPREA   = 001333
HSLIM   = 000727
HSTHI   = 000002
HSTLO   = 000001
H1       = 000055
IMAGE   = 007275
IMLEN   = 000003
IMSTAR  = 001332
INDSEC  = 000013
INI      = 001246
INIRCT  = 020000
INITDD  = 002406
INITIT  = 002340
INITL   = 004430
INITPT  = 002371
INITP1  = 002376
INIT5   = 002347
INST    = 000001
IRECAL  = 001101
ISEEK   = 001075
ISKIP    = 003643
LAGAIN  = 003506
LAST    = 100000
LBD     = 000002
LBNBAD  = 001312
LBNCYL  = 001143
LBNFND  = 003242
LBNHOS  = 000012
```

LBALBN	001133
LBNPCY	001145
LBNTRK=	000011
LCHEC	003442
LDONE	004360
LEN =	000000
LEERR	003553
LER1	003720
LER2	003727
LER3	003715
LFERR	003441
LFIXIT	004361
LFORM	003047
LHERE	003615
LKIP10	003352
LKIP12	003152
LKIP13	003452
LKIP14	003546
LKIP15	003601
LKIP16	003645
LKIP17	003617
LKIP18	003627
LKIP19	003573
LKIP2	003105
LKIP3	003301
LKIP4	003311
LKIP5	003244
LKIP6	003471
LKIP7	003514
LKIP8	003365
LKIP9	003335
LLOVER	003402
LMORE	004375
LNOERR	003570
LO =	007777
LOAD	002160
LOADER	002231
LOAD1	002163
LOAD2	002205
LOAD3	002211
LOAD4	002226
LOAD5	002165
LOBYTE=	000377
LOK	004220
LONG =	000001
LONGTO=	000001
LOOP	003144
LOOPP	003231
LOOPP2	003203
LOOP1	002020
LOOP2	002044
LOOP3	003456
LOVER	004260
LOVER1	004355
LOVER2	004321
LOVER3	004312
LOVER4	004300

LPBN	003204
LRBN	003142
LRDC	004447
LSKIP	003464
LSKIP1	003476
LSKIP2	003475
LSKIP3	004223
LSKIP4	003214
LSKIP5	003372
LSKIP6	003707
LSKIP7	004230
LSKIP8	003633
LSKIP9	004210
LSND	004165
LTO	001250
LTRK	003047
MAINTR=	000016
MAINTW=	000017
MANU -	000200
MARBAD	003265
MASK -	000000
MAXTRY-	000010
MNCNT	001262
MORE	004062
MSG1	002036
MULPC	000733
M512	126736
M576	074161
N	001455
NDLL -	004000
NEXT	002412
NEXT1	001245
NEXT5	002417
NGD	003243
NN1	001457
NO	003347
NOCRY	002650
NODLL	003114
NOERR	003626
NOFCT =	100000
NOGOOD	003306
NOINC	003144
NOLBN	003510
NOSEK	003356
NOTHER	004352
NOTR	003226
NUM	001443
NUMDBN	003423
NUMLBN	003104
NUMRBN	003144
NUMXBN	003453
N1	001456
OCDONE	003230
OCLOOP	003106
OERR	002426
OERR2	002435
OFATAL	003344

```

OFFS = 000011
OFFSET 000736
OK 003770
OLDONE 003341
ONE 001466
ONEFIL= 000001
ONLIN 002332
ONLINE 001104
OOPS 004443
OQUIT 003362
ORF TAL 003344
OVCNT - 000021
OVER 003540
OVER1 004041
OVER2 004010
OVE.F1= 003047
OVE.F2= 003047
OVE.F3= 003047
OVE.F4= 003047
OVE.F5= 003047
OVE.F6= 003047
OVE.F7= 003047
OVE.F8= 003047
OVE.F9= 003047
OVE.G1= 003047
OVE.G2= 003047
OVE.G3= 003047
OVE.G4= 003047
OVE.G5= 003047
OVE.G7= 003047
OVE.G8= 003047
OVE.H1= 003047
OVE.MN= 000714
OVLBLK 001424
OVLEN = 000003
OVL TBL 001341
OVL.F1= 001103 G
OVL.F2= 001416 G
OVL.F3= 000423 G
OVL.F4= 000647 G
OVL.F5= 001202 G
OVL.F6= 000320 G
OVL.F7= 000445 G
OVL.F8= 000625 G
OVL.F9= 000202 G
OVL.G1= 001376 G
OVL.G2= 000052 G
OVL.G3= 000214 G
OVL.G4= 001720 G
OVL.G5= 000237 G
OVL.G7= 000336 G
OVL.G8= 000224 G
OVL.H1= 000302 G
OVL.MN= 002215 G
OVL...= 016575
OVLAY 002526
OVS.F1 010466 G

```

```

OVS.F2= 013570 G
OVS.F3= 020746 G
OVS.F4= 022014 G
OVS.F5= 024336 G
OVS.F6= 030054 G
OVS.F7= 026742 G
OVS.F8= 016624 G
OVS.F9= 035330 G
OVS.G1= 005472 G
OVS.G2= 030714 G
OVS.G3= 031040 G
OVS.G4= 031470 G
OVS.G5= 035734 G
OVS.G7= 012674 G
OVS.G8= 020276 G
OVS.H1= 023532 G
OVS.MN= 001040 G
OV... - 017215
PAERR 002510
PAERR1 002517
PAGAIN 003117
PAGE 002444
PAGER 002472
PALP1 002454
PALP2 002461
PALP3 002507
PALP4 002502
PARITY= 000200
PARIT1= 000400
PBNBUF= 005152
PCNT 001500
PCON 003047
PDONE 003276
PERR 003206
PNGBLK 003114
PNGPG 003612
PNGPNG 003112
PNGRD 003151
PRET 003205
PRIM = 001000
PRIMRB 002757
PRMBUF= 006204
PRMY = 100000
PROD = 000000
RBD = 000004
RBNBUF 006621
RBNLBN 001135
RBNPCY 001147
RBNRPT= 000200
RBNTRK= 000004
RBNWRT 003522
RB.BUF- 000000
RB.CMD= 000003
RB.HI - 000002
RB.IM - 000004
RB.LOW= 000001
RCD 004102

```

RCFIX	003442	READ9	004354	SECTRK	001127	STPNIC	002013	TRK	= 000011
RCFXLP	004230	RECAL	002154	SEEK	002251	STRBN	- 000003	TRKCNT	001465
RCINDN=	000100	R-CIR =	040000	SEEKER	003435	STSC	= 000012	TRKCYL	001141
RCINER	003462	RECOV	001503	SEEK0	002253	STSKP	002007	TRKGRP=	000003
RCINIT=	000040	RECTMP	001505	SEEK1	002262	STSK1	001702	TWO	003635
RCINLP	003203	REDO	004134	SEEK2	002310	STWLK	001764	TWOB	= 000006
RCL	003760	RETRY	001502	SEEK3	002321	STXBN	- 000002	TWOC	001470
RCLP	003057	REVBUFF=	006204	SEEK4	002302	ST.DB	= 001000	TWRD	001447
RCLP2	003206	REVCNT	001254	SEEK5	002307	ST.DBN	001340	UHASH	003357
RCLP3	003452	REVECT=	000100	SEEK6	002301	ST.DF	= 000020	UHKIP	003364
RCLP4	003411	REVLN=	000004	SEEK7	002276	ST.DR	= 000040	UHKIP1	003410
RCLP6	003420	REVRBN	001151	SEND	= 000004	ST.ERB=	000002	UID	= 000000
RCTBAD	001313	REVSEC=	000007	SER	= 000005	ST.ERR=	000374	UNIT	000740
RCTBUF=	006621	RFCT	004417	SERNUM	001305	ST.FD	= 002000	UNSEC	= 000175
RCTCK	004510	RFTL	004216	SETOVL	003103	ST.IN	= 000004	UN.ERI	000717
RCTCKD	004736	RINC1	003542	SFTRPT	004334	ST.LBN	001335	UN.ERR	000715
RCTCKE	004761	RLD	004207	SHORTO=	000000	ST.PS	= 000002	UN.ERT	000716
RCTCLP	004544	RLDONE	003335	SIP	= 000012	ST.RBN	001330	UN.SEK	000720
RCTCL1	004547	RLOOP	003150	SIXTN	- 000017	ST.RU	= 000001	UPDATE=	000022
RCTCNT	001477	RLOOP1	003203	SKERR	003402	ST.SR	= 000020	UPDPNT	001474
RCTERR	003606	RNWHER	003563	SKIP1	003200	ST.WE	= 000010	UREAD	= 000013
RCTFMT	001242	ROVER	003101	SKIP12	003365	ST.WP	= 170000	UWRITE=	000014
RCTINI	003047	ROVER1	003116	SKIP19	003161	ST.XBN	001337	VLD	= 000004
RCTLBN	001261	RPRIM	= 000004	SKIP3	003254	SWAP	003255	VLD1	= 000010
RCTNGD	004705	RPT	003223	SKIP4	003164	SWRD	001446	V1	= 000000
RCTRLP	004063	RPT1	004005	SKIP5	003202	TALIP1	002117	V2	= 000002
RCTSZ	= 000014	RQUIT	003256	SKIP6	003310	TALK	002016	V3	= 000004
RCTTOT	001235	RRC	003715	SKIP7	003257	TALKDN	002060	V4	= 000005
RCTUPD	003047	RRERR	003611	SKPCNT	001233	TALKIP	002115	WLOOP	003315
RCTWLP	003426	RRPL	003567	SLEEK	003210	TALKP	002076	WP	= 000001
RCTWRT	004056	RSER	= 000000	SLEEK2	003225	TALKRT	002070	WRBLK	000721
RCTWT	003421	RTCNT	= 000012	SLOOP	003122	TATTN1	002147	WRCMD	= 122400
RCV	= 000005	RTDON	003604	SND	004153	TBLK	001234	WRFLG	001222
RCVMNT	002544	RTRY	= 000001	SNDCNT	001472	TCLEAR	002103	WRITE	= 000003
RCVRDY=	000001	RTY	= 100000	SNDMNT	002536	TEMP	000736	WRITE1	003310
RCWERR	004244	RTYCN	001473	SND1	= 000003	TEMP2	000733	WRITE2	003511
RCXLP	003571	RTYDN	= 000002	SRCK	001745	TERR	002143	WRITE3	004146
RC.FRE=	000000	RUN	001023	ST	001030	THREB	= 000011	WRITE4	003251
RC.MUL	= 100000	RWCMD	= 013400	STACK	001215	TILOP	004046	WRITE5	003141
RC.PRIV=	020000	RWGD	003546	STARIT	001232	TILOP1	004053	WRITE8	003301
RC.SND=	030000	RWGOOD	004205	START	003047	TIMER	002656	WRITE9	003404
RC.LNU-	040000	RWRDY	= 100000	START2	003212	TIMLOP	004026	WRIT10	004147
RDBLK	000721	RWTDON	004243	START3	003063	TIMLP	002661	WRIT12	003275
RDBUF	- 004535	RW.BUF=	000001	STATFR	002710	TIMVAL=	100000	WRIT13	003632
RDCMD	- 100000	RW.CMD=	000004	STATRE	002674	TKIP1	003073	XAGAIN	003620
RDLEN	- 000005	RW.DAT=	000000	STATRT	002707	TKIP10	003163	XBBAD	001315
RDONE	003233	RW.DUM=	000005	STATST	001664	TKIP11	003230	XBD	= 000010
RDONE1	003237	RW.EDC=	000400	STATUS=	000007	TKIP2	003074	XBNCYL=	000021
READ	= 000002	RW.ER1=	000000	STATVL	002671	TKIP3	003146	XBNFND	003125
READ1	003573	RW.ER2=	000401	STCKSV	001216	TKIP4	003156	XBNIT	003171
READ10	004632	RW.HI	= 000003	STCLR	= 170377	TKIP5	003236	XBNSEC	001137
READ11	003141	RW.LOW=	000002	STCYL	= 000001	TKIP6	003302	XDONE	003416
READ2	003722	RW.STA=	000000	STDBN	= 000003	TKIP7	003133	XEORCT	003675
READ3	003535	SCR	001052	STDIA	001757	TKIP8	003254	XFLIP	003637
READ4	003666	SECCNT	001454	STFORM	001752	TKIP9	003264	XNGBLK	003615
READ7	003136	SECNDY	003302	STLBN	000002	TMPTRY	001504	XNOINC	003645
READ8	004013	SECTCY	001131	STO	001251	TOTRCT	001475	XPBN	003341

CKR	67-85	67-87#																
CLEAR	14-22	15-128#	34-91	35-182	40-58	42-260												
CLEDON	66-50	66-55#																
CLLPL	66-24#	66-46																
CLLPL2	66-20#	66-49																
CLESKP	66-27	66-43#	66-54															
CLEWRT	66-51	66-57	67-4#															
CLHERE	67-7#	67-63																
CLSKP2	66-42	66-51#																
CLSKP3	66-14	66-56	66-58#															
CLSKP4	66-40	66-57#																
CMDBUF	7-358#	35-69	36-7	42-21	43-7													
CMDDAT	7-57#	35-89	42-41	42-102														
CNT	9-165#	20-8																
CNTCYL	10-35#	34-35*	34-53*	34-107*	34-126*	40-15*	40-17*	40-18*	40-106	40-127	40-129							
CONBLK	10-6#	21-5	21-21	50-13	53-10	55-13	60-9	61-45	63-13	67-19	67-71	67-88	68-8	68-65				
	69-10	70-21	71-20	72-15														
CONINT	26-25#	33-33																
COUNT	10-52#	32-7*	32-26*	32-27*	32-30*	32-80*	35-7*	37-36*	37-55	39-7*	50-8*	50-59*	50-132*	50-136*				
	50-139*	61-43*	61-116*	61-129*	61-132*	61-133*	66-16*	66-38*	66-41	66-52*	66-55*	67-6	67-61*	67-62*				
	71-14*	71-90*	71-94*	71-98														
CR	9-48	9-77#	23-6	27-6	28-4	29-27	29-46	29-61	31-4	50-142								
CR.ACC	9-52#	15-121																
CR.CLR	9-53#	15-129																
CR.DIS	9-50#	15-172																
CR.ERV	9-56#	19-49																
CR.GCR	9-48#	14-64																
CR.GSR	9-49#	14-66																
CR.GST	9-47#	14-14	32-9															
CR.ONL	9-57#	15-180																
CR.RCL	9-55#	15-80																
CR.RUN	9-51#	15-87																
CR.SEK	9-54#	15-139																
CS	67-26	67-71#	69-22	69-82	70-34	71-28												
CSKIP	35-57	35-62#																
CSKIP1	35-66#	35-108																
CSKIP2	35-61	35-65#																
CSKIP3	35-159	35-162#																
CSKIP6	35-143	35-147#																
CSKIP7	35-114	35-118	35-164#															
CURBN	9-96#	34-56*	34-58*	34-100*	34-101*	37-24	37-40	37-46	37-51*	37-51*	38-15	38-36	40-90*	40-91*				
	44-21	44-49	44-54	44-76	44-83	44-89	44-94*	44-94*	46-12*	46-14*	49-5	49-6	49-23	49-24				
	50-10*	50-12*	50-45	50-75	50-76	50-89	50-91	50-120	50-134*	50-135*	59-50	59-54	59-76	59-78				
	59-115	59-117*	59-119	59-171	59-179	59-181	61-39*	61-41*	61-56	61-67	61-69	61-98	61-127*	61-128*				
	70-17*	70-19*	70-31	70-32	70-42	70-44	70-84	70-97*	70-98*	71-10*	71-12*	71-27	71-36	71-38				
	71-81	71-92*	71-93*	73-93*	73-94*													
CURGRP	10-38#	34-62*	34-63*	34-66	34-104*	37-67*	37-80	40-29*	40-30*	40-33	40-103*	42-261	44-188*	44-201				
	51-121	59-225	61-143															
CURLBN	9-98#	61-40*	61-42*	61-126*	61-126*	61-127	61-128	71-11*	71-13*	71-91*	71-91*	71-92	71-93					
CUROVL	9-122#	17-10	17-29*	34-10*	37-11*	40-8*	44-6*	46-9*	50-7*	51-11*	55-7*	59-8*	61-6*	63-7*				
	64-11*	65-11*	66-9*	72-7*														
CURPBN	9-93#	37-30*	37-34*	37-54*	37-54*	38-19*	38-20*	38-40*	38-41*	44-13*	44-14*	44-29*	44-33*	44-103*				
	44-103*	44-138	44-171	59-41*	59-42*	59-43*	59-44	73-6	73-8*	73-14	73-17	73-23	73-30*	73-32				
	73-37*	73-44	73-58	73-67	73-74*	73-78	73-92*	73-93	73-94									
CURRBN	9-91#	40-99*	40-100*	44-134	44-151	44-158	44-167*	44-167*	46-17*	46-18*	66-17*	66-18*	66-34	66-36				
	66-43*	66-43*																
CURTRK	9-95#	21-12*	34-75*	34-81	34-96*	35-131	35-172	36-29	40-42*	40-48	40-81*	42-85	42-250	43-29				

CROSS REFERENCE TABLE (CREF V01-05)

DMFLG	7-266#	33-5												
DMOST	7-264#													
DMSG1	24-6#	24-9	24-13											
DMSG1L	24-9#	24-14												
DMUL	12-26#	22-23	26-36	26-48	26-60	26-75	26-95	26-115	26-124	26-195	26-223	27-14	29-8	35-24
	35-41	37-84	37-92	44-205	44-213	47-10	48-10	49-22	73-59	73-84				
DMUNIT	7-265#	32-34	32-57	32-59										
DOLBN	26-10	26-13#												
DONDLI.	50-50	50-149#												
DONE	7-68#	19-42	66-89											
DONE1	37-60	37-62#												
DONMSG	9-191#	66-61	66-63											
DPBN	37-19	38-8#												
DPREA	9-195#	29-103*	29-104*	34-78	35-169	40-45	42-247	50-97	53-36	55-97	60-37	61-76	63-99	67-45
	68-39	69-98	72-40											
DSUB	12-2#	12-79	25-9	26-81	26-154	28-26	29-20	34-99	34-125	35-29	35-46	40-21	40-89	40-98
	40-108	40-126	44-223	47-19	48-19	51-22	53-69	55-80	59-17	59-31	59-236	60-71	61-38	61-111
	63-81	69-81	71-9	72-12	72-74	73-19	73-25	73-66						
DUPOVL	7-331#	25-5												
DWRD	10-21#	26-202	42-220	44-119										
DXBN	26-12	26-14#												
DXCH	35-54#	35-186												
DXCHEC	34-95	35-53#												
DXERR	34-131	34-134#												
DXFCP1	37-58	39-4#												
DXFCPG	34-40	34-46	35-4#											
DXFORM	34-12	34-20	34-29#											
DXTRK	37-12#													
EAGAIN	20-9#	20-16												
ECC	7-64#	23-10												
ECCCK	23-6#	35-144	42-97	55-48	63-47	69-49	70-61	71-56						
ECCF	7-321#	35-142	42-95	55-46	63-45	69-47	70-59	71-54						
ECHO	7-59#													
EDC	10-24#	26-213	44-16	68-3										
EIMAGE	9-150#	26-228*	36-33	36-36	37-94	43-33	43-36	44-215						
FMAX	9-145#	29-23*	40-62											
EMSG	9-189#	19-32	19-34											
ENTRY	9-4#													
EORCT	58-14	58-41#												
ER1	35-140	35-146	35-152#											
ERV	7-78#	29-28												
FRDN	7-242#	19-24	19-26											
ERECOV	9-56	9-84#	19-48*											
ERFLAG	9-141#	35-55*	35-157*	35-167*	42-7*	42-194*	42-217*							
ERLEN	7-314#	42-169	51-108											
ERPNT	9-168#	29-13*	40-69*	42-168	42-195	42-205*								
ERR	9-147#	35-54*	35-99*	35-109*	35-163*	35-165*	42-6*	42-52*	42-61*	42-212*	42-215*			
ERRBUF	9-144#	29-12*	40-69	51-23										
ERRCNT	10-30#	34-15	34-17*	34-23	34-25*	35-158*	40-137	42-149*	42-207*					
ERRHND	19-46#	50-111	53-49	55-61	55-111	60-51	61-89	63-60	63-113	68-53	69-62	69-112	70-74	71-69
	72-54													
ERRLN	7-319#													
ERRLIST	19-47	19-52#												
ERRMNT	14-74	15-49	15-72	15-116	15-222	16-24	17-46	19-24#	21-23	26-16	32-85	34-134	40-148	50-159
	53-76	54-54	55-122	58-52	60-77	61-159	63-131	64-33	65-88	67-90	69-123	70-104	71-99	72-81
ERROR	50-126	50-157#												
ERRSYM	7-77#	23-7	31-5	31-7*										

[illegible]

STATS	9-47	9-61#	15-163	33-29													
H1	14-12#	15-73	54-40	55-7	58-38	59-63	59-128										
HASH	7-211#	51-40	59-120														
HD.BAD	57-4#	59-55	37-42	42-184	44-85	44-153	51-48										
HD..LR	7-148#	35-155	26-91	32-33	32-61	35-154	37-33	37-41	37-47	40-18	42-116	42-160	42-175	42-183			
	7-150#	26-71	44-50	44-77	44-84	44-90	44-152	44-159	47-7	48-7	51-30	51-52	51-68	51-82			
	42-201	44-32	59-43	59-80	59-109	59-116	59-137	59-143	59-157	59-175	59-183	65-65	66-31	67-75			
	51-92	51-104	73-35	73-72	73-90												
	73-7	73-28	34-52*	35-117	37-48												
HD.CUR	10-37#	34-49*	36-16	73-29													
HD.DBN	7-151#	34-49	43-14	44-91	53-31	59-52	60-32	61-71	71-40	73-91							
HD.LBN	7-145#	42-69	42-155	42-172	44-51	51-72											
HD.PR.V	7-149#	42-122	42-113	42-131	42-239	43-16	44-58	44-160	59-113	67-9	68-34	73-73					
HD.RBN	7-146#	42-71	42-176	42-191	42-202	44-78	66-32										
HD.REV	7-147#	42-161	36-14	50-92	63-27	63-94	70-45	72-35	73-36								
HD.XBN	7-152#	34-52															
HEAD	7-97#	29-105															
HERE	35-112#	35-161	35-166														
HGHPBN	9-123#	28-29*	28-30*	65-56	65-67												
H1BYT	7-128#	22-19	26-55	26-102	40-94	44-116	48-15	49-18	68-7	73-54							
H12BYT	7-127#	29-37															
H1BYTE	7-125#	14-18	22-13	23-8	26-27	26-31	26-43	26-104	26-143	27-9	29-29	29-104	29-107	34-34			
	34-61	34-72	35-19	37-72	40-27	40-39	40-85	44-9	44-193	47-15	49-12	53-9	55-18	60-11			
	61-47	67-18	68-11	69-15	71-19	73-48	73-62	73-80									
HKIP	57-8#	57-23															
HP IP1	57-5	57-20#															
HOLD	9-148#	51-18*	51-19*	51-21	51-37	54-37	58-35	59-27*	59-28*	59-30	59-60	59-125	61-34*	61-35*			
	61-37	61-39	61-40	61-41	61-42	69-20	71-5*	71-6*	71-8	71-10	71-11	71-12	71-13				
HOLDBN	9-101#	34-55*	34-57*	34-97	34-100	34-101	40-82	40-90	40-91	46-13*	46-15*						
HOLDPN	9-105#	44-13	44-14	44-221	49-28*	49-29*											
HOL RBN	9-103#	40-92	40-99	40-100	46-19*	46-20*											
MPREA	9-194#	29-108*	34-79	35-170	40-46	42-248	50-96	53-35	55-96	60-36	61-75	63-78	67-44	68-38			
	69-97	72-39															
MSLIM	9-29#	35-80	35-122	42-32	42-76	50-94	53-33	55-23	55-94	60-34	61-73	63-21	63-96	67-41			
	68-36	69-74	69-95	70-47	71-42	72-37											
HSTHI	7-195#	18-8															
HSTLO	7-194#	18-7	25-11														
IMAGE	7-361#	26-224	29-10	34-83	35-111	35-174	36-8	36-37	36-39	37-14	37-75	40-50	42-63	42-152			
	42-235	42-252	43-8	43-37	43-39	44-12	44-196	59-237	59-240	59-245							
IMLEN	7-323#	29-4	35-164	36-42	37-63	37-89	42-157	42-214	42-241	43-42	44-93	44-168	44-184	44-210			
IMSTAR	9-193#	34-80	35-171	37-75*	37-76	37-96*	40-47	42-249	44-196*	44-197	44-217*						
INDSEC	7-258#																
INI	9-164#	20-7															
INIRCT	7-230#	40-71	40-73	44-99													
INIT5	15-191#	15-202															
INITDD	15-214	15-216	15-220#														
INITIT	15-187#	33-25															
INITL	26-6	33-24#															
INITP1	15-213#	15-218															
INITPT	15-159	15-210#	34-90	35-181	40-57	42-259											
INST	7-294#	50-36*															
IRECAL	9-55	9-83#															
ISFEK	9-54	9-79#	21-13*	21-14*	21-15*	34-64*	34-65*	34-66*	40-31*	40-32*	40-33*	42-261*	51-113*	51-116*			
	51-121*	59-219*	59-222*	59-225*	61-137*	61-140*	61-143*	67-80*	67-81*	67-82*							
SKIP	27-19	27-22#															
AGAIN	42-22#	42-58															
LAST	7-325#	37-98	44-219														

BL	7-280#	66-67#												
BMBAD	9-181#	40-136	40-139#	44-100#	66-66									
BMBYL	9-117#	26-131#	26-134#	40-14	40-15	40-16	40-17	47-5	47-6	48-5	48-6	63-12	63-15	72-14
	72-17													
BMBND	73-52	73-78#												
BMBOS	7-91#	26-78	26-79	59-45										
BMBBN	9-113#	26-76#	26-77#	28-20	28-21	38-11	38-31	38-32	51-18	51-19	59-27	59-28	61-34	61-35
	71-5	71-6	73-9	73-10										
BMBPY	9-118#	26-49#	26-50#	26-74	47-9									
BMBTRK	7-87#	22-12	26-42	26-103	40-84	44-8	47-14	49-11	53-8	55-17	60-10	61-46	67-17	68-10
	69-14	71-18	73-47	73-61	73-79									
CMEC	40-70	42-6#	42-265											
CDONE	42-62	42-218	42-266#											
CFN	7-193#	18-9												
LER1	42-99	42-105	42-107#											
LER2	42-108	42-111#												
LER3	42-93	42-106#												
LERR	42-39	42-44#												
LFERR	40-145	40-148#												
LFIXIT	42-17	43-6#												
LFORM	40-8#													
LHERE	42-64#	42-210	42-216											
LKIP10	44-102	44-105	44-107#											
LKIP12	44-37	44-39#												
LKIP13	44-149	44-15#												
LKIP14	44-170	44-180	44-182#											
LKIP15	44-195	44-199#												
LKIP16	44-198	44-218#												
LKIP17	44-206#	44-209												
LKIP18	44-207	44-210#												
LKIP19	44-189	44-196#												
LKIP2	44-19#	44-108												
LKIP3	44-24	44-26	44-30	44-34	44-89#									
LKIP4	44-71	44-81	44-88	44-93#										
LKIP5	44-46	44-72#												
LKIP6	44-137	44-142	44-146	44-158#										
LKIP7	44-157	44-163	44-167#											
LKIP8	44-118#	44-183												
LKIP9	44-96	44-98	44-101#											
LLOVER	44-125#	44-129												
LMORE	43-12#	43-45												
LNUERR	42-24	42-43	42-48	42-53#										
LO	7-129#	26-150	32-40	35-116	36-13	42-68	42-112	42-154	42-171	42-238	43-13	51-47	51-54	51-75
	53-13	54-13	55-12	58-12	59-51	59-69	59-134	59-145	60-14	61-50	66-25	67-22	68-14	69-9
	71-23													
LOAD	14-47	15-85#												
LOAD1	15-87#													
LOAD2	15-94	15-99#												
LOAD3	15-101#	15-109												
LOAD4	15-106	15-111#												
LOAD5	15-81	15-88#	15-98											
LOADER	15-97	15-110	15-114#											
LOBYTE	7-126#	31-6												
LOK	42-104	42-209#												
LONG	7-9#	34-115	40-116											
LONGTO	7-76#	29-47												
LOOP	50-23	50-40#	50-140											

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

XFCRT	54-15	54-13#				
XFLIP	54-11	54-18#				
XNGBLK	54-6#	54-16				
XNOINC	54-21	54-23#				
XPBA	37-21	38-31#				
XPERR	54-44	54-52#				
XFNGRD	54-31#	54-47				
XPRET	54-17	54-50#				
XSKIP1	40-10	40-13#				
XSKIP2	40-45#	40-60				
XSKIP3	40-43#	40-102				
XSKIP4	40-65	40-70#				
XSKIP5	40-72	40-78	40-81#			
XSKIP6	40-75	40-79#				
XSLEEK	40-25#	40-128	40-130			
XSLEK2	40-31#	40-105				
XVZ	59-112	59-114	59-140	59-167	59-178	59-187 59-192#
XYZ1	59-47	59-193#				
VES	70-73	70-81#				

