.REM &

IDENTIFICATION

PRODUCT CODE: AC-S839A-MC

PRODUCT NAME: CZUDFA0 UDA FMTR DATA FILE

PRODUCT DATE: 10-JULY-81

MAINTAINER: DIAGNOSTIC ENGINEERING

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
UDA50 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 KDUBA0.PAK) AND THE HOST PORTION OF THE FORMATTER (CURRENTLY ZDUBA0.BIN) MUST EXIST ON THE BOOT DEVICE. THE FORMATTER IS STARTED BY ENTERING:

```
RUN ZDUBA0
```

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 PARAMATERS 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

X IS THE CURRENT DEFAULT (INITIALLY Y). IF THIS QUESTION IS ANSWERED 'Y' THEN A REFORMAT MODE FORMAT WILL BE PREFORMED 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 PREFORMED 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 PARAMETERS, 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 FCT 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 'START'
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:
        EXPLAINATION
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 EXPLAINATION 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.

SEQ 0010

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
;
```

```
                        ;
                        ;       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
```

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

```
                                .SBTTL  EQUATES
                        ;
                        ;       EQUATES
                        ;
                        ;
                        ;       CONDITIONAL FLAGS
                        ;
                        ;
000001          LONG    -       1.              ;COMPLETE MESSAGE ON CYL (1)
                                                ;OR MOD 16 CYL (0)
000000          PROD    -       0               ;PRODUCTION (1)
                                                ;NOT PRODUCTION (0)
                        ;
                        ;       OFFSETS FOR FORMAT TRACK TABLE
                        ;
000000          FT.BUF  -       0.              ;BUFFER POINTER OFFSET
000001          FT.LOW  -       1.              ;LOW ORDER HEADER OFFSET
000002          FT.HI   -       2.              ;HI ORDER HEADER OFFSET
                        ;
                        ;
                        ;


                        ;       OFFSETS FOR READ/WRITE BUFFERS

000000          RW.STAT =       0.              ; STATUS (12-15), NEXT BUFR PTR (0-14)
000000          RW.ER1  -       0.              ; ALSO USED AS ECC ERROR INDICATOR
000005          RW.DUM  =       5.              ; POINTER TO DUMMY SDI CONTROL BLOCK
000001          RW.BUF  -       1.              ; POINTER TO DATA BUFFER
000002          RW.LOW          2.              ; 1ST HEADER WORD (LO ORDER LBN)
000003          RW.HI   =       3.              ; 2ND HEADER WORD (HI ORDER LBN)
000004          RW.CMD  =       4.              ; SDI RT CMD (8-15), HEAD ADDR (0-7)
000000          RW.DAT  =       0.              ; 1ST WORD OF 256 WORD DATA BUFFER
000400          RW.EDC  =       256.            ; EDC
000401          RW.ER2          257.            ; 1ST ECC RESIDUE


                        ;
                        ;       OFFSETS FOR CHECK PASS READ COMMANDS
                        ;

000000          RB.BUF  =       0               ;BUFFER OFFSET
000001          RB.LOW  -       1.              ;LOW ORDER BLOCK NUMBER
000002          RB.HI   -       2.              ;HIGH ORDER BLOCK NUMBER
000003          RB.CMD  -       3.              ;READ COMMAND AND TRACK NUMBER
000004          RB.IM   -       4.              ;IMAGE COUNTER

                        ;       XFC DEFINITION EQUATES
                        ;
000000          BREAK   -       0.              ;BREAKPOINT XFC CODE
000001          FORMAT          1.              ;FORMAT TRACK XFC CODE
000002          READ    -       2.              ;READ N SECTORS XFC CODE
000003          WRITE   =       3.              ;WRITE N SECTORS XFC CODE
000004          SEND            4.              ;SEND SDI COMMAND XFC CODE
000005          RCV             5.              ;RECEIVE SDI MESSAGE XFC CODE
000006          CMPDAT          6.              ;COMPARE DATA PATTERN XFC CODE
```

```
000007                  STATUS  =       7.              ;RETURN DRIVE STATUS XFC CODE
000010                  ECHO    =       8.              ;ECHO DATA TO DRIVE XFC CODE
000011                  DINIT   =       9.              ;DRIVE INITIALIZE XFC CODE
000012                  SIP     =       10.             ;WAIT FOR SECTOR/INDEX PULSE XFC CODE
000013                  UREAD   =       11.             ;READ UNIBUS MEMORY XFC CODE
000014                  UWRITE  =       12.             ;WRITE UNIBUS MEMORY XFC CODE
000015                  ECC     =       13.             ;DO ECC ON BUFFER XFC CODE
000016                  MAINTR  =       14.             ;SEND MAINT READ DATA XFC CODE
000017                  MAINTW  =       15.             ;RECEIVE MAINT WRITE DATA XFC CODE
000020                  CVT     =       16.             ;CONVERT TO PHYSICAL ADDRESS XFC CODE
000021                  DONE    =       17.             ;TERMINATE DM PROGRAM XFC CODE
000022                  UPDATE  =       18.             ;UPDATE DUP PROGRESS INDICATOR XFC
                        ;
                        ;       GET CHARACTERISTICS OFFSETS
                        ;
000000                  SHORTO  =       0.              ;SHORT TIME OUT
000001                  FRCPY   =       1.              ;NUMBER OF F/RCT COPIES
000001                  RTRY    =       1.              ;NUMBER OF RETRIES
000001                  LONGTO  =       1.              ;LONG TIMEOUT
000002                  ERRSYM  =       2.              ;NUMBER OF ALLOWABLE ECC ERRORS
000002                  FRCV    =       2.              ;ERROR RECOVERY LEVELS SUPPORTED
000007                  REVSEC  =       7.              ;REVS/SECOND
000011                  OFFS    =       9.              ;GROUP OFFSET
000000                  CYLBN   =       0.              ;CYLINDERS IN LBN AREA
000002                  STLBN   =       2.              ;HIGH ORDER STARTING LBN
000003                  STRBN   =       3.              ;HIGH ORDER STARTING RBN
000002                  STXBN   =       2.              ;HIGH ORDER STARTING XBN
000003                  STDBN   =       3.              ;HIGH ORDER STARTING DBN
000001                  STCYL   =       1.              ;HIGH ORDER STARTING CYLINDER
000011                  LBNTRK  =       9.              ;NUMBER OF LBNS PER TRACK (512)
000004                  RBNTRK  =       4.              ;NUMBER OF RBNS PER TRACK
000021                  XBNCYL  =       17.             ;NUMBER OF CYLINDERS IN XBN AREA
000022                  DBNCYL  =       18.             ;NUMBER OF CYLINDERS IN DBN AREA
000012                  LBNHOST =       10.             ;NUMBER OF LBN'S IN HOST AREA
000002                  GRPCYL  =       2.              ;GROUPS/CYLINDER
000003                  TRKGRP  =       3.              ;TRACKS/GROUP
000010                  FCTSZ   =       8.              ;FCT SIZE IN SECTORS
000014                  RCTSZ   =       12.             ;RCT SIZE IN LBN'S
000005                  DATA    =       5.              ;DATA PREAMBLE SIZE
000005                  HEAD    =       5.              ;HEADER PREAMBLE SIZE
                        ;
                        ;
                        ;       BIT DEFINITIONS
                        ;
                        ;
000001                  BIT0    =       000001
000002                  BIT1    =       000002
000004                  BIT2    =       000004
000010                  BIT3    =       000010
000020                  BIT4    =       000020
000040                  BIT5    =       000040
000100                  BIT6    =       000100
000200                  BIT7    =       000200
000400                  BIT8    =       000400
001000                  BIT9    =       001000
002000                  BIT10   =       002000
004000                  BIT11   =       004000
```

```
                                ;
                                ;
                                ;       RANDOM DEFINITIONS
                                ;
                                ;
000000 177400   HIBYTE  =       177400          ;HIGH BYTE MASK
000000 000377   LOBYTE  =       000377          ;LOW BYTE MASK
000000 177700   HI2BYTE =       177700          ;HIGH BYTE PLUS 2 BITS
000000 177600   HI1BYTE =       177600          ;HIGH BYTE PLUS 1 BIT
000000 007777   LO      =       007777          ;ALL BUT HEADER CODE
000000 177760   FCLR    =       177760          ;CLEAR FOR FRCPY
000000 170377   STCLR   =       170377          ;CLEAR FOR STARTING BITS
000000 007777   BUFMSK  =       007777          ;BUFFER CLEAR MASK
000000 000017   SIXTN    =       BIT0+BIT1+BIT2+BIT3     ;FOR CHECK OF MOD 16 CYLINDER
000000 000004   VLD     =       BIT2            ;STATUS VALID BIT(1=VALID)
000000 000010   VLD1    =       BIT3            ;STATUS VALID BIT(1=VALID)
000000 000200   PARITY  =       BIT7            ;STATUS PARITY BIT(1=PARITY ERROR)
000000 000400   PARIT1  =       BIT8            ;REAL TIME ERROR(1-ERROR)
                                ;
                                ;
000000 000175   UNSEC   =       000175          ;UNSUCCESSFUL COMPLETION
                                ;
                                ;       HEADER CODES
                                ;
000000 000000   HD.LBN  =       000000          ;GOOD LBN
000000 060000   HD.RBN  =       060000          ;GOOD RBN, PERHAPS UNUSED
000000 030000   HD.REV  =       030000          ;REVECTORED LBN
000000 110000   HD.BAD  =       110000          ;BAD BLOCK
000000 050000   HD.PRV  =       050000          ;PRIMARY REVECTORED BLOCK
000000 170000   HD.CLR  =       170000          ;CLEAR HDR CODE
000000 140000   HD.DBN  =       140000          ;GOOD DBN
000000 120000   HD.XBN  =       120000          ;GOOD XBN
000000 100000   PRMY    =       BIT15           ;PRIMARY BIT IN FCT
000000 010000   FBDHD   =       BIT12           ;BAD HEADER CODE IN FCT
                                ;
                                ;       RCT HEADER CODES
                                ;
000000 000000   RC.FRE  =       000000          ;FREE REPLACEMENT BLOCK
000000 020000   RC.PRV  =       020000          ;PRIMARY REVECTOR
000000 030000   RC.SND  =       030000          ;SECONDARY REVECTOR
000000 040000   RC.UNU  =       040000          ;BAD REPLACEMENT BLOCK
000000 100000   RC.NUL  =       100000          ;NULL(FILL) BLOCK
                                ;
                                ;       DRIVE STATUS BITS
                                ;
000000 100000   RWRDY   =       BIT15           ;READ/WRITE READY BIT POSITION
000000 000002   ATTN    =       BIT1            ;ATTENTION
000000 000001   RCVRDY  =       BIT0            ;RECEIVER READY
                                ;
                                ;
                                ;
```

```
                        ;       GET STATUS BIT MASKS
                        ;
000001          ST.RU   =       BIT0                    ;RUN/STOP SWITCH 1=IN
000002          ST.PS   =       BIT1                    ;PORT SWITCH 1=IN
000040          ST.DR   =       BIT5                    ;DIAGNOSTIC REQUESTED 1=YES
170000          ST.WP           BIT12+BIT13+BIT14+BIT15 ;WRITE PROTECT SWITCH SU:0,1 1=IN
000020          ST.SR   =       BIT4                    ;SPINDLE READY 1=YES
001000          ST.DB   =       BIT9                    ;DIAG CYL ACCESS ENABLED 1=YES
002000          ST.FO   =       BIT10                   ;FORMAT CYL ACCESS ENABLED 1=YES
000004          ST.IN           BIT2                    ;DRIVE INITIALIZED 1=YES
000010          ST.WE   =       BIT3                    ;WRITE ERROR (WRITE LOCKED)
000020          ST.DF   =       BIT4                    ;DIAG FAILED - CANNOT DRIVE CLEAR
000374          ST.ERR  =       000374                  ;COMBINED CLEARABLE ERRORS BITS SET
000002          ST.ERB  =       2.                      ;ERROR BYTE OFFSET (3RD WORD)
                        ;
                        ;
                        ;       OVERLAY TABLE OFFSETS
                        ;
                        ;
000021          OVCNT   =       17.                     ;NUMBER OF OVERLAYS
000003          OVLEN           3                       ;LENGTH OF 1 OVERLAY BLOCK
000000          LEN     =       0                       ;WORD COUNT OF OVERLAY
000001          HSTLO   =       1                       ;LOW ORDER UNIBUS ADDRESS
000002          HSTHI   =       2                       ;HI ORDER UNIBUS ADDRESS
000000          F1      =       0                       ;OFFSET INTO TABLE
000003          F2      =       3                       ;SECOND OVERLAY OFFSET INTO TABLE
000006          F3      =       6.                      ;THIRD OVERLAY OFFSET INTO TABLE
000011          F4      =       9.                      ;FOURTH OVERLAY OFFSET INTO TABLE
000014          F5      =       12.                     ;FIFTH OVERLAY OFFSET INTO TABLE
000017          F6      =       15.                     ;SIXTH OVERLAY OFFSET INTO TABLE
000022          F7      =       18.                     ;SEVENTH OVERLAY OFFSET INTO TABLE
000025          F8      =       21.                     ;EIGTH OVERLAY
000030          F9      =       24.                     ;NINTH OVERLAY
000033          G2      =       27.                     ;ELEVENTH OVERLAY
000036          G3      =       30.                     ;TWELVTH OVERLAY
000041          G4      =       33.                     ;THIRTEENTH OVERLAY
000044          G5      =       36.                     ;FOURTEENTH OVERLAY
000047          G7      =       39.                     ;SIXTEENTH OVERLAY
000052          G8      =       42.                     ;SEVENTEENTH OVERLAY
000055          H1      =       45.                     ;NINETEENTH OVERLAY
000060          G1      =       48.                     ;TENTH OVERLAY
                        ;
                        ;
                        ;       FLAG EQUATES
                        ;
                        ;
000001          FCTAVL          BIT0                    ;FCT AVAILABLE
000010          DBN             BIT3                    ;FORMAT DBN AREA
000100          REVECT          BIT6                    ;REVECTOR FLAG
001000          PRIM            BIT9                    ;PRIMARY FOUND FLAG
000002          FCTEMT  =       BIT1                    ;FCT EMPTY FLAG
000020          GOBAD   =       BIT4                    ;DO BEST GUESS IF FCT BAD
000040          RCINIT          BIT5                    ;RCT LAST BLOCK FIXED UP
000004          FCTBAD          BIT2                    ;FCT FOUND BAD (FOR STATS)
000200          MANU            BIT7                    ;MANUFACTURING FORMAT
000400          DLL     =       BIT8                    ;DOWN-LINE LOAD FLAG
002000          BSTGS           BIT10                   ;BEST GUESS FORMAT
```

```
        004000          NDLL    =       BIT11                   ;ONLY WRITE FCT SCRATCH
        020000          INIRCT  =       BIT13                   ;INIT RCT FLAG
        040000          FINI    =       BIT14                   ;FORMAT FINISHED FLAG
        010000          CHRDNE  =       BIT12                   ;CHARACTERISTICS DONE FLAG
        100000          RTY     =       BIT15                   ;RETRY FLAG
                        ;
                        ;       FLAG1   EQUATES
                        ;
                        ;
        000001          WP      =       BIT0                    ;WRITE PROTECT FLAG
        000002          RTYDN   =       BIT1                    ;RETRY DONE ON THIS SECTOR
        000004          RPRIM   =       BIT2                    ;FLAG FOR PRIMARY GOOD EDC
        000010          ERDN    =       BIT3                    ;FLAG FOR ERROR EXIT TRY
        000020          DEAD    =       BIT4                    ;HOST GONE FLAG
        000040          BDHD    =       BIT5                    ;BAD HEADER ON CHECK PASS READ
        000100          RCINDN  =       BIT6                    ;RCT INIT DONE (WITH ONE FULL PAD BLK)
                        ;
                        ;
                        ;       PHYSICAL CONVERSION XFC BLOCK EQUATES
                        ;
        000000          V1      =       0                       ;CYLINDER PARAMETER
        000002          V2      =       2                       ;BLOCK NUMBER PARAMETER
        000004          V3      =       4.                      ;BLOCKS PER TRACK PARAMETER
        000005          V4      =       5.                      ;ONLY FOR RBN'S
        000006          CYL     =       6.                      ;CYLINDER RETURNED
        000010          GRP     =       8.                      ;GROUP RETURNED
        000011          TRK     =       9.                      ;TRACK RETURNED
        000012          STSC    =       10.                     ;STARTING SECTOR RETURNED
        000013          INDSEC  =       11.                     ;SECTOR FROM INDEX
                        ;
                        ;
                        ;       DMBUF OFFSETS
                        ;
                        ;
        000001          DMOST   =       1.                      ;STARTING OVERLAY ADDRESS
        000003          DMUNIT  =       3.                      ;DESIRED UNIT NUMBER
        000004          DMFLG   =       4.                      ;FLAG WORD
        000016          DMBUFL  =       14.                     ;BUFFER LENGTH
                        ;       FLAG FORMAT
                        ;
                        ;       BIT0 -  0       DO NOT USE FCT
                        ;               1       USE FCT
                        ;       BIT1 -  0       CONTINUE IF FCT BAD
                        ;               1       STOP IF FCT BAD
                        ;       BIT2 -  0       NO DLL OF FCT
                        ;               1       DLL FCT
                        ;
        000005          SER     =       5.                      ;SERIAL NUMBER
        000011          DAT     =       9.                      ;DATE IN VAX/VMS FORMAT

        000002          LBD     =       2.                      ;LBN'S BAD
        000003          SND1    =       3.                      ;SECONDARY REVECTORS
        000004          RBD     =       4.                      ;RCT BLOCKS BAD
        000006          DBD     =       6.                      ;DBN BLOCKS BAD
        000010          XBD     =       8.                      ;XBN BLOCKS BAD
        000012          RTCNT   =       10.                     ;RETRY COUNT
```

EQUATES

```
000013          FCT     =       11.             ;FCT FLAG
                ;
                ;
                ;
                ;       FCT BLOCK OFFSETS
                ;
                ;
000002          FSER    =       2               ;SERIAL NUMBER
000001          INST    =       1               ;FORMATTIN INSTANCE NUMBER
000016          C512    =       14.             ;COUNT OF USED 512 ENTRIES IN FCT
000012          FDAT    =       10.             ;MOST RECENT FORMAT DATE
000025          FCTFLG  =       21.             ;FCT FLAG FOR GOOD/BAD FCT
100000          NOFCT   =       BIT15           ;FLAG - 0 - FCT GOOD
                ;                               ;       1 - FCT KNOW BAD
                ;
                ;
                ;       RCT BLOCK OFFSETS
                ;
                ;
000000          RSER    =       0               ;SERIAL NUMBER OFFSET
                ;
                ;       MISC    DEFINITIONS
                ;
                ;
000006          TWOB    =       6.              ;LENGTH OF 2 IMAGE ENTRIES
000011          THREB   =       9.              ;LENGTH OF 3 IMAGE ENTRIES
000005          RDLEN   =       5.              ;LENGTH OF CHECK PASS READ BLOCK
000002          ERLEN   =       2.              ;LENGTH OF REVECTOR TABLE ENTRY
000004          REVLEN  =       4.              ;LENGTH OF SECONDARY TABLE
013400          RWCMD   =       013400          ;SDI READ COMMAND
122400          WRCMD   =       122400          ;SDI WRITE COMMAND
100000          RDCMD   =       100000          ;SIGNAL TO XFC NO MORE BLOCKS
000003          ERRLN   =       3.              ;LENGTH OF ERROR LIST ENTRY
040000          FULL    =       BIT14           ;FULL BBUFFER BIT
010000          ECCF    =       BIT12           ;ECC ERROR BIT
000200          RBNRPT  =       128.            ;NUMBER OF RBN COPIES IN REVECTOR
000003          IMLEN   =       3.              ;LENGTH OF IMAGE BLOCK
020000          BD      =       BIT13           ;BAD FLAG IN IMAGE BUFFER
100000          LAST    =       BIT15           ;LAST FLAG IN IMAGE BUFFER
040000          RECIR   =       BIT14           ;RECIRCULATE IN FORMAT IMAGE BUFFER
126736          M512    =       126736          ;FCT MODE INDICATOR FOR 512
074161          M576    =       07416'          ;FCT MODE INDICATOR FOR 576
100000          TIMVAL  =       32768.          ;TIMER LOOP VALUE
000010          MAXTRY  =       8.              ;FINAL SECONDARY WRITE RETRY LIMIT
007774          DUPOVL  =       7774            ;OVERLAY STARTING ADDRESS FROM DUP
                ;
                ;
                ;       STATUS OFFSETS
                ;
                ;
000000          MASK    =       0               ;SUBUNIT OFFSET MASK
000000          UID     =       0               ;UNIT NUMBER OFFSET
                ;
                ;
                ;       BUFFER DEFINITIONS
                ;       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                             ;               BUFFER ASSIGNMENTS
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                  REVBUF          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
```

```
                                        .SBTTL  DATA STRUCTURES
                                ;
000000                                  DMCODE  KKDUB,0,714,13,255.
000714  003047                  ENTRY:  JMP     START                   ;JUMP TO START LOCATION
                                ;
                                ;       DATA STRUCTURES
                                ;
                                ;       RETRY COUNTERS
                                ;
000715  000000                  UN.ERR: .WORD   0                       ;UNSUCCESSFUL CMD RETRY CNTR
000716  000000                  UN.ERT: .WORD   0                       ;TRANSMISSION ERROR RETRY CNTR
000717  000000                  UN.ERI: .WORD   0                       ;INITLIZATION ERROR RETRY CNTR
000720  000000                  UN.SEK: .WORD   0                       ;SEEK RETRY COUNT
                                ;
                                ;
                                ;       READ COMMAND BLOCK
                                ;
                                ;
000721                          WRBLK:
000721  100000                  RDBLK:  .WORD   100000                  ;STATUS POINTER
000722  000000                          .WORD   0                       ;POINTER TO DATA BUFFER
000723  000000                          .WORD   0                       ;FIRST WORD OF EXPECTED HDR
000724  000000                          .WORD   0                       ;SECOND WORD  "    "     "
000725  000000                          .WORD   0                       ;REAL-TIME SDI COMMAND
000726  000000                          .WORD   0                       ;POINTER TO SDI BLOCK
                                ;
                                ;       DUMMY DOUBLE WORDS AND DUMMY SDI COMMAND
                                ;
000727  000200                  HSLIM:  .WORD   200                     ;HEADER CMP LIMIT
000730  001045                          .WORD   SCR-5                   ;POINTER TO SUBUNIT CHAR
000731  000000                  DDUMMY: .WORD   0                       ;DUMMY DOUBLE WORD FOR ONE
000732  000000                          .WORD   0                       ;BYTE OPERAND CONVERSION
000733                          TEMP2:                                  ;ALSO USE AS TEMP
000733                          MULPC:  .BLKW   2                       ;MULTIPLICATION BUFFER
000735  000000                          .WORD   0                       ;RESERVED LOCATION (A+7)
000736                          OFFSET:                                 ;FOR EASIER REFERENCE
000736                          TEMP:   .BLKW   2                       ;USED FOR COMPUTATIONS
                                ;
                                ;
                                ;       CURRENT UDA PORT
                                ;
000740  000000                  UNIT:   .WORD   0                       ;SDI INTERCONNECT
                                ;
                                ;       MESSAGE TABLES
                                ;
000741                          CR.GST: MSG     GST,1,ST,7              ;GET STATUS
000745                          CR.GCR: MSG     GCR,1,CR,11.            ;GET CHARACTERISTICS
000751                          CR.GSR: MSG     GSR,2,SCR,19.           ;GET SUBUNIT CHARACTERISTICS
000755                          CR.DIS: MSG     DIS,2,ST,6              ;UNLOAD DRIVE
000761                          CR.RUN: MSG     RUN,1,ST,6              ;LOAD DRIVE
000765                          CR.ACC: MSG     ACC,3,ST,6              ;SET FORMAT ACCESS
000771                          CR.CLR: MSG     DCLR,2,ST,6             ;DRIVE CLEAR
000775                          CR.SEK: MSG     ISEEK,6,ST,6            ;SEEK
001001                          CR.RCL: MSG     IRECAL,1,ST,6           ;RECALIBRATE
001005                          CR.ERV: MSG     ERECOV,2,ST,6           ;ERROR RECOVERY COMMAND
001011                          CR.ONL: MSG     ONLINE,2,ST,6           ;ONLINE COMMAND
```

```
                                        ;
                                        ;       MESSAGES AND COMMANDS
                                        ;
001015  004400          GST:    .WORD   000011*256.     ;GET STATUS COMMAND
001016  103400          GCR:    .WORD   000207*256.     ;GET CHARACTERISTICS
001017  104000          GSR:    .WORD   00210*256.      ;GET SUBUNIT CHARACTERISTICS
001020  000000                  .WORD   0               ;SUBUNIT MASK
001021  102000          DIS:    .WORD   000204*256.     ;UNLOAD DRIVE
001022  000000                  .WORD   0               ;NO SPIN DOWN MODIFIER
001023  006000          RUN:    .WORD   000014*256.     ;INITIATE LOAD
001024  100400          ACC:    .WORD   000201*256.     ;ACCESS DIAG AND FMT CYL
001025  003006                  .WORD   3006            ;MASK BYTE/MODE BYTE
001026  002400          DCLR:   .WORD   000005*256.     ;DRIVE CLEAR
001027  000374                  .WORD   374             ;BITS TO CLEAR
001030                  ST:     .BLKW   7               ;STATUS MESSAGE BUFFER
001037                  CR:     .BLKW   11.             ;CHARACTERISTICS MESSAGE BUFF
001052                  SCR:    .BLKW   19.             ;SUBUNIT CHARACTERISTICS BUFF
001075  005000          ISEEK:  .WORD   000012*256.     ;INITIATE SEEK
001076  000000                  .WORD   0               ;       ''
001077  000000                  .WORD   0               ;       ''
001100  000000                  .WORD   0               ;
001101  107000          IRECAL: .WORD   000216*256.     ;INITIATE RECAL
001102  003000          ERECOV: .WORD   000006*256.     ;ERROR RECOVERY COMMAND
001103  000000                  .WORD   0               ;RECOVERY LEVEL
001104  105400          ONLINE: .WORD   000213*256.     ;ONLINE COMMAND
001105  000377                  .WORD   377             ;COMMAND TIMEOUT (SECS)
                                        ;
                                        ;       DISK LOCATION POINTERS
                                        ;
001106  000000          CURRBN: .WORD   0               ;CURRENT RBN
001107  000000                  .WORD   0               ;       ''
001110  000000          CURPBN: .WORD   0               ;CURRENT PBN
001111  000000                  .WORD   0               ;       ''
001112  000000          CURTRK: .WORD   0               ;CURRENT TRACK
001113  000000          CURBN:  .WORD   0               ;CURRENT BLOCK NUMBER
001114  000000                  .WORD   0
001115                  CURLBN:                         ;FOR RCT INIT
001115  000000          CURXBN: .WORD   0               ;CURRENT XBN NUMBER
001116  000000                  .WORD   0
001117  000000          HOLDBN: .WORD   0               ;BLOCK NUMBER OF FIRST BLOCK ON CYL
001120  000000                  .WORD   0               ;       ''
001121  000000          HOLRBN: .WORD   0               ;BLOCK NUM OF FIRST RBN ON CYLINDER
001122  000000                  .WORD   0
001123  000000          HOLDPN: .WORD   0               ;PBN OF FIRST SECOTR ON TRACK
001124  000000                  .WORD   0
001125  000000          CYLNUM: .WORD   0               ;CURRENT CYLINDER NUMBER
001126  000000                  .WORD   0               ;       ''
001127  000000          SECTRK: .WORD   0               ;SECTORS/TRACK
001130  000000                  .WORD   0               ;       ''
001131  000000          SECTCY: .WORD   0               ;SECTORS/CYLINDER
001132  000000                  .WORD   0               ;       ''
001133                  LBNLBN: .BLKW   2               ;LBN'S IN LBN AREA
001135                  RBNLBN: .BLKW   2               ;RBN'S IN LBN AREA
001137                  XBNSEC: .BLKW   2               ;SECTORS IN LBN AREA
001141                  TRKCYL: .BLKW   2               ;TRACKS/CYLINDER
001143                  LBNCYL: .BLKW   2               ;NUMBER OF LBN CYLINDERS
001145                  LBNPCY: .BLKW   2               ;LBN'S/CYLINDER
```

```
001147                  RBNPCY: .BLKW   2               ;RBN'S/CYLINDER
001151  000000          REVRBN: .WORD   0               ;REVECTORED RBN NUMBER
001152  000000                  .WORD   0               ;
001153  000000          CUROVL: .WORD   0               ;CURRENT OVERLAY
001154                  HGHPBN: .BLKW   2               ;HIGHEST PBN IN LBN AREA
                        ;
                        ;
                        ;
                        ;       STACK
                        ;
                        ;
                        ;
001156                          .BLKW   31.             ;STACK
001215  000000          STACK:  .WORD   0               ;TOP OF STACK
001216  000000          STCKSV: .WORD   0               ;STACK PTR TEMP SAVE
                        ;
                        ;
                        ;       MISC    DEFINITIONS
                        ;
                        ;
001217  000000          FLAG:   .WORD   0               ;FLAG WORD
001220  000000          FLAG1:  .WORD   0               ;FLAG WORD
001221  000000          ERFLAG: .WORD   0               ;RE-FORMAT FLAG
001222  000000          WRFLG:  .WORD   0               ;RCT WRAP FLAG
001223  000000          BADPBN: .WORD   0               ;POINTER TO PBNTAB ENTRY
001224  000000          ERRBUF: .WORD   0               ;POINTER TO BEGINNING OF REVECTOR BUFFER
001225  000000          EMAX:   .WORD   0               ;MAX NUMBER OF REVECTORS BEFORE
                                                        ;RCT UPDATE ROUTINE IS CALLED
001226  000000          ERR:    .WORD   0               ;NUMBER OF SECTORS IN ERROR
001227  000000          HOLD:   .WORD   0               ;DOUBLE WORD TEMP STORAGE
001230  000000                  .WORD   0
001231  000000          EIMAGE: .WORD   0               ;ADDRESS OF END IMAGE BLOCK
001232  000000          STARIT: .WORD   0               ;STARTING ADDRES OF THIS PASS
001233  000000          SKPCNT: .WORD   0               ;OFFSET FOR FIRST READ CHECK
001234  000000          TBLK:   .WORD   0               ;INTERLEAVE
                                                        ;6 - BI-LEAVE
                                                        ;9 - TRI-LEAVE
001235                  RCTTOT:                         ;ALSO RCT TOTAL HOLDING AREA
001235  005606          CUTOF:  .WORD   2950.           ;SECT/SECOND CUTOFF
001236  000000                  .WORD   0               ;DOUBLE WORD
001237  000000          FCNT:   .WORD   0               ;COUNT OF USED FCT ENTRIES FOR FORMATTING
001240                  FCTFMT: .BLKW   2               ;SIZE OF ONE FCT COPY
001242                  RCTFMT: .BLKW   2               ;SIZE OF ONE RCT COPY
001244  000000          FCTCPY: .WORD   0               ;NUMBER OF FCT COPIES
001245  000000          NEXT1:  .WORD   0               ;MUTLI-COPY COUNTER
001246  000105          INI:    .WORD   69.             ;INITIAL VALUE FOR EDC
001247  000400          CNT:    .WORD   256.            ;COUNT FOR EDC
001250  000100          LTO:    .WORD   100             ;LONG TIMEOUT
001251  002000          STO:    .WORD   1024.           ;SHORT TIMEOUT (IN MILLESECS)
001252  000000          ERPNT:  .WORD   0               ;REVECTOR LIST POINTER
001253  000000          BUFPNT: .WORD   0               ;BUFFER POINTER FOR FCT READ
001254  000000          REVCNT: .WORD   0               ;REVECTOR COUNT
001255  000000          FCTPTR: .WORD   0               ;POINT TO CURRENT LOCATION IN FCT LBOKC
001256  000001          FCTCNT: .WORD   1               ;CURRENT FCT BLOCK
001257  000000                  .WORD   0
001260  000000          FCTNPD: .WORD   0               ;NON-PAD FCT BLOCKS
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                   FCMSG:  .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                                  ;       OVERLAY POINTERS
203                                  ;       NOTE:
204                                  ;
205                                  ;               WHEN ADDING AN ENTRY TO THIS TABLE EQUATE
206                                  ;               OVCNT MUST BE INCREMENTED
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  000000                           .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 EIGTH 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
```

```
233 001372  035330                    .WORD   OVS.F9          ;LOW ORDER HOST ADDRESS
234 001373  000000                    .WORD   0               ;HIGH ORDER HOST ADDRESS
235 001374  000052                    .WORD   OVL.G2          ;LENGTH OF EIGTH OVERLAY
236 001375  030714                    .WORD   OVS.G2          ;LOW ORDER HOST ADDRESS
237 001376  000000                    .WORD   0               ;HIGH ORDER HOST ADDRESS
238 001377  000214                    .WORD   OVL.G3          ;LENGTH OF EIGTH OVERLAY
239 001400  031040                    .WORD   OVS.G3          ;LOW ORDER HOST ADDRESS
240 001401  000000                    .WORD   0               ;HIGH ORDER HOST ADDRESS
241 001402  001720                    .WOPD   OVL.G4          ;LENGTH OF EIGTH OVERLAY
242 001403  031470                    .WORD   OVS.G4          ;LOW ORDER HOST ADDRESS
243 001404  000000                    .WORD   0               ;HIGH ORDER HOST ADDRESS
244 001405  000237                    .WORD   OVL.G5          ;LENGTH OF FOURTEENTH OVERLAY
245 001406  035734                    .WORD   OVS.G5          ;LOW ORDER HOST ADDRESS
246 001407  000000                    .WORD   0               ;HIGH ORDER HOST ADDRESS
247 001410  000336                    .WORD   OVL.G7          ;LENGTH OF SIXTEENTH OVERLAY
248 001411  012674                    .WORD   OVS.G7          ;LOW ORDER HOST ADDRESS
249 001412  000000                    .WORD   0               ;HIGH ORDER HOST ADDRESS
250 001413  000224                    .WORD   OVL.G8          ;LENGTH OF SEVENTEENTH OVERLAY
251 001414  020276                    .WORD   OVS.G8          ;LOW ORDER HOST ADDRESS
252 001415  000000                    .WORD   0               ;HIGH ORDER HOST ADDRESS
253 001416  000302                    .WORD   OVL.H1          ;LENGTH OF SEVENTEENTH OVERLAY
254 001417  023532                    .WORD   OVS.H1          ;LOW ORDER HOST ADDRESS
255 001420  000000                    .WORD   0               ;HIGH ORDER HOST ADDRESS
256 001421  001376                    .WORD   OVL.G1          ;LENGTH OF SEVENTEENTH OVERLAY
257 001422  005472                    .WORD   OVS.G1          ;LOW ORDER HOST ADDRESS
258 001423  000000                    .WORD   0               ;HIGH ORDER HOST ADDRESS
259 001424                  OVLBLK:   .BLKW   4               ;FOR BUFFER OVERLAYS
```

```
                                ;
                                ;
                                ;       CONVERSION BUFFER
                                ;
                                ;
001430  001045                  CONBLK: .WORD   1045            ;STARTING CYL PLUS LBN CYL
001431  000000                          .WORD   0               ;HIGH ORDER
001432  000000                          .WORD   0               ;LOW ORDER XBN-X
001433  000000                          .WORD   0               ;SECTORS PER TRACK
001434  000040                          .WORD   40              ;HIGH - BOTH FILLED IN DYNAMICALLY
001435                                  .BLKW   6               ;FILLED IN BY XFC
                                ;
                                ;
                                ;       COMPARE DATA XFC CONTROL BLOCK
                                ;
001443  000377                  NUM:    .WORD   255.            ;NUMBER OF WORDS IN PATTERN
001444  004536                  CBUF:   .WORD   RDBUF+1         ;BUFFER TO COMPARE(NOT FIRST WORD)
001445  155555                  FWRD:   .WORD   155555          ;FIRST WORD OF PATTERM
001446  133333                  SWRD:   .WORD   133333          ;SECOND WORD OF PATTERN
001447  155555                  TWRD:   .WORD   155555          ;THIRD WORD OF PATTERN
001450  167356                  DWRD:   .WORD   167356          ;DIAGNOSTIC WORD(FIRST IN SECTOR)
                                ;
                                ;
001451  125677                  EDC:    .WORD   125677          ;EDC FOR ABOVE DATA PATTERN
001452  052100                  BADEDC: .WORD   52100           ;BAD EDC FOR RBN BLOCKS
                                ;
                                ;
                                ;       COUNTERS
                                ;
001453  000000                  ERRCNT: .WORD   0               ;FOR TESTING VERIFICATION
001454  000000                  SECCNT: .WORD   0               ;SECTOR COUNT
001455  000000                  N:      .WORD   0               ;NUMBER OF ORIGINAL CHECK PASS READ
001456  000000                  N1:     .WORD   0               ;NUMBER OF ERROR READS
001457  000000                  NN1:    .WORD   0               ;DITTO
001460  000000                  CNTCYL: .WORD   0               ;NUMBER OF CYLINDERS TO FORMAT
001461  000000                          .WORD   0
001462  000000                  HD.CUR: .WORD   0               ;CURRENT HEADER
001463  000000                  CURGRP: .WORD   0               ;CURRENT GROUP
001464  000000                  GRPCNT: .WORD   0               ;NUMBER OF GROUPS TO DO
001465  000000                  TRKCNT: .WORD   0               ;NUMBER OF TRACKS TO DO
001466  000001                  ONE:    .WORD   1               ;WORD CONSTANT OF 1
001467  000000                          .WORD   0               ;DOUBLE WORD
001470  000002                  TWOC:   .WORD   2               ;WORD CONSTANT OF 2
001471  000000                          .WORD   0               ;DOUBLE WORD
001472  000000                  SNDCNT: .WORD   0               ;COUNT OF SECONDARY REVECTORS
001473  000000                  RTYCNT: .WORD   0               ;COUNT OF SECTORS RETRYED
001474  000000                  UPDPNT: .WORD   0               ;POINTER FOR RCT UPDATE
001475  000000                  TOTRCT: .WORD   0               ;TOTAL LBN'S IN RCT'S
001476  000000                          .WORD   0               ;
001477  000000                  RCTCNT: .WORD   0               ;CURRENT RCT BLOCK
001500  000000                  PCNT:   .WORD   0               ;PBN BLOCK COUNTER
001501  000000                  COUNT:  .WORD   0               ;COUNT FOR XBN DLL
001502  000005                  RETRY:  .WORD   5               ;RETRIES FROM SDI
001503  000000                  RECOV:  .WORD   0               ;RECOVERY LEVELS SUPPORTE BY DRIVER
001504  000000                  TMPTRY: .WORD   0               ;TEMP FOR RETRY COUNT
001505  000000                  RECTMP: .WORD   0               ;TEMP FOR ERROR RECOVERY LEVEL
```

```
                                        .SBTTL  MATH SUBROUTINES
                                ;
                                ;
                                ;       SUBROUTINES
                                ;
                                ;
                                ;       DOUBLE ADD ROUTINE
                                ;
                                ;       INPUT PARAMETERS
                                ;
                                ;       R3      CONTAINS POINTER TO OPERAND 1
                                ;
                                ;       R4      CONTAINS POINTER TO OPERAND 2
                                ;
                                ;       OUTPUT PARAMETER
                                ;
                                ;       R4      CONTAINS THE RESULT
                                ;
001506  100465                  DADD:  MOV     R5,-(SP)                ;SAVE A SCRATCH REGISTER
001507  100461                         MOV     R1,-(SP)                ;SAVE ANOTHER
001510  104235                         MOV     (R3)+,R5                ;GET LOW ORDER OPERAND
001511  104131                         MOV     (R3),R1                 ;GET HIGH ORDER OPERAND
001512  105245                         ADD     (R4)+,R5                ;ADD LOW ORDER OPERAND
001513  002[15                         BCC     DADD1                   ;BRANCH IF NO CARRY
001514  115401                         INC     R1                      ;ADD ONE TO HIGH IF CARRY
001515  105141                  DADD1: ADD     (R4),R1                 ;ADD OP 2
001516  100141                         MOV     R1,(R4)                 ;SAVE HIGH ORDER
001517  100445                         MOV     R5,-(R4)                ;SAVE LOW ORDER
001520  104261                         MOV     (SP)+,R1                ;RESTORE R1
001521  104265                         MOV     (SP)+,R5                ;RESTORE R5
001522  117403                         DEC     R3                      ;RESTORE R3
001523  000000                         RETURN
                                ;
```

```
                                                ;++
                                                ; DOUBLE PRECISION FIXED POINT SUBTRACT ROUTINE
                                                ;
                                                ; INPUTS:
                                                ;       R3    POINTER TO OPERAND 1 (SUBTRAHEND)
                                                ;       R4 -  POINTER TO OPERAND 2 (MINUEND)
                                                ;
                                                ; OUTPUT:
                                                ;       R4 -  POINTER TO RESULT  WHERE (R4) = (R4) - (R3)
                                                ; --
001524                                  DSUB:
001524                                          PUSH    R1,R5                   ; SAVE REGISTERS
001526  104245                                  MOV     (R4)+,R5                ; GET LO ORDER MINUEND
001527  104141                                  MOV     (R4),R1                 ; GET HI ORDER MINUEND
001530  107135                                  SUB     (R3),R5                 ; SUBTRACT LOW ORDER OPERANDS
001531  041533                                  BCC     10$                     ; POSITIVE RESULT
001532  117401                                  DEC     R1                      ; BORROW FROM HI ORDER OPERAND
001533  107631  000001                  10$:    SUB     1(R3),R1                ; SUBTRACT HI ORDER OPERANDS
001535  100141                                  MOV     R1,(R4)                 ; STORE HI ORDER RESULT
001536  100445                                  MOV     R5,-(R4)                ; STORE LO ORDER RESULT
001537                                          POP     R5,R1                   ; RESTORE REGISTERS
001541  000000                                  RETURN
001542                                  DMUL:
                                                ;++
                                                ; DOUBLE PRECISION FIXED POINT MULTIPLY ROUTINE
                                                ; INPUTS:
                                                ;       R3 = POINTER TO MULTIPLIER (SINGLE PRECISION)
                                                ;       R4 - POINTER TO MULTIPLICANT (DOUBLE PRECISION)
                                                ;
                                                ; OUTPUT:
                                                ;       R4   POINTER TO RESULT WHERE (R4) - (R4) * (R3)
                                                ; --
001542                                          PUSH    R0,R3                   ; SAVE R0 & R3
001544  104137                                  MOV     (R3),R0                 ; GET MULTIPLIER
001545  051552                                  BNE     5$                      ; MULTIPLIER NOT = 0
001546  100147                                  MOV     R0,(R4)                 ; LOAD LO ORDER RESULT
001547  100647  000001                          MOV     R0,1(R4)                ; LOAD HI ORDER RESULT
001551  001565                                  BR      20$                     ; RETURN
001552  104140  000733          5$:     MOV     (R4),MULPC              ; COPY MULTIPLICANT FOR DADD
001554  104640  000001  000734                  MOV     1(R4),MULPC+1
001557  104203  000733                          MOV     #MULPC,R3
001561  117407                          10$:    DEC     R0                      ; ADJUST MULTIPLIER FOR *1
001562  011565                                  BEQ     20$                     ; MULTIPLIER = 0, EXIT
001563  021506                                  CALL    DADD                    ; PERFORM ITERATIVE ADDS
001564  001561                                  BR      10$
001565                          20$:    POP     R3,R0                   ; RESTORE R0 & R3
001567  000000                                  RETURN
001570                                  DDIV:
                                                ;++
                                                ; DOUBLE PRECISION FIXED POINT DIVIDE
                                                ; INPUTS:
                                                ;       R3 = POINTER TO DIVISOR (SINGLE PRECISION)
```

```
                                        ;           HIGH ORDER WORD MUST BE ZERO
                                        ;       R4 - POINTER TO DIVIDENT (DOUBLE PRECISION)
                                        ;
                                        ; OUTPUT:
                                        ;       R3   POINTER TO REMAINDER
                                        ;       R4 = POINTER TO QUOTIENT
                                        ;
                                        ; NOTE - THE CASES WHERE EITHER THE DIVISOR OR DIVIDENT ARE ZERO,
                                        ;        ARE NOT CONSIDERED IN THIS ROUTINE.
                                        ; --

001570                                  PUSH    R0,R1,R2,R5             ; SAVE REGISTERS
001574  114007                          CLR     R0                      ; CLR LO ORDER QUOTIENT REG
001575  114001                          CLR     R1                      ; CLR HI ORDER QUOTIENT REG
001576  104132                          MOV     (R3),R2                 ; GET DIVISOR
001577  104645  000001          10$:    MOV     1(R4),R5                ; GET HI ORDER DIVIDENT
001601  051606                          BNE     20$                     ; DIVISOR NOT = 0
001602  104145                          MOV     (R4),R5                 ; GET LO ORDER DIVIDENT
001603  106052                  15$:    CMP     R5,R2                   ; IS DIVIDENT < DIVISOR ?
001604  041606                          BCC     20$                     ; NO, CONTINUE
001605  001614                          BR      30$                     ; YES, STOP
001606  021524                  20$:    CALL    DSUB                    ; SYNTHESIZE DIVIDE
001607  105207  000001                  ADD     #1,R0                   ; INCR LO ORDER QUOTIENT
001611  041577                          BCC     10$                     ; DID NOT OVERFLOW
001612  115401                          INC     R1                      ; ADJUST HI ORDER QUOTIENT
001613  001577                          BR      10$
001614  104145                  30$:    MOV     (R4),R5                 ; GET REMAINDER
001615  100147                          MOV     R0,(R4)                 ; LOAD LO ORDER QUOTIENT
001616  100641  000001                  MOV     R1,1(R4)                ; LOAD HI ORDER QUOTIENT
001620  100135                          MOV     R5,(R3)                 ; LOAD REMAINDER
00162?                                  POP     R5,R2,R1,R0             ; RESTORE REGISTERS
001625  000000                          RETURN

```

```
                                        ;
                                        ;       DOUBLE COMPARE
                                        ;
                                        ;       INPUT PARAMETERS
                                        ;
                                        ;       R3      CONTAINS A POINTER TO THE FIRST OPERAND
                                        ;
                                        ;       R4      CONATINS A POINTER TO THE SECOND OPERAND
                                        ;
                                        ;       OUTPUT PARAMETERS
                                        ;
                                        ;       THE FLAGS ARE SET AS IF A SINGLE PRECISSION ''CMP'' HAD OCCURED
                                        ;
001626  10046'                  DCMP:   MOV     R0,-(SP)                ;SAVE R0 FOR USE AS SCRATCH
001627  10046                           MOV     R1,-(SP)                ;SAVE R1 FOR USE AS SCRATCH
001630  104171                          MOV     (R4),R1                 ;GET LOW ORDER DEST OPERAND
001631  104147  000001                 MOV     1(R4),R0                ;GET HIGH ORDER DEST OPERAND
001633  106037  000001                 CMP     1(R3),R0                ;DO ACTUAL HIGH ORDER TEST
001635  001651                          BEQ     DCMP1                   ;GO DO ADDITIONAL TESTING
001636  041644                          BCC     DCMP2                   ;SRC HI, CLEAN UP AND RTN
001637  104261                  DCMP4:  MOV     (SP)+,R1                ;RESTORE R1
001640  104267                          MOV     (SP)+,R0                ;RESTORE R0
001641  106204  000000                 CMP     #0,R4                   ;SET CONDITION CODES - SRC LSS
001643  000000                          RETURN                          ;AND RETURN
001644  104261                  DCMP2:  MOV     (SP)+,R1                ;RESTORE R1
001645  104267                          MOV     (SP)+,R0                ;RESTORE R0
001646  106204  077777                 CMP     #077777,R4              ;SET CONDITION CODES - DST LSS
001650  000000                          RETURN                          ;AND RETURN
001651  106131                  DCMP1:  CMP     (R3),R1                 ;TEST LOW ORDER
001652  051657                          BNE     DCMP3                   ;BRANCH IF NOT EQUAL
001653  104261                          MOV     (SP)+,R1                ;RESTORE R1
001654  104267                          MOV     (SP)+,R0                ;RESTORE R0
001655  106044                          CMP     R4,R4                   ;SET CONDITION CODES - EQUAL
001656  000000                          RETURN                          ;AND RETURN
001657  106131                  DCMP3:  CMP     (R3),R1                 ;COMPARE AGAIN
001660  041644                          BCC     DCMP2                   ;BRANCH ON SRC HI
001661  001637                          BR      DCMP4                   ;BRANCH ON SRC LOW
```

```
                                        .SBTTL  SDI SUBROUTINES
                                ;
                                ;       GET STATUS
                                ;
                                ;       OUTPUT PARAMETERS
                                ;
                                ;       CLEARS DRIVE STATUS AND GETS CHARACTERISTICS
                                ;       IF NOT ALREADY RECEIVED
                                ;
                                ;
001662                          GSTATS: PUSH    R3                      ;PUSH R3
001663                                  PUSH    R5                      ;PUSH R5
001664  104203  000741          STATST: MOV     #CR.GST,R3              ;POINT TO GET STATUS TABLE
001666  022016                          CALL    TALK                    ;GET STATUS
001667  104207  001030                  MOV     #ST,R0                  ;POINT TO SUBUNIT CHARACTERISTICS
001671  104673  000002                  MOV     ST.ERB(R0),R3           ;GET ERROR BYTE
001673  103203  177420                  BIC     #ST.DF+HIBYTE,R3        ;CLEAR HIGH BYTE AND DF BIT
001675  115003                          TST     R3                      ;ANY NEED TO ISSUE DRIVE CLEAR ?
001676  011702                          BEQ     STSK1                   ;NOPE - SKIP IT
001677  104030  001027                  MOV     R3,DCLR+1               ;STORE MASK IN DRIVE CLR COMMAND
001701  022243                          CALL    CLEAR                   ;DO A DRIVE CLEAR
001702  104205  001030          STSK1:  MOV     #ST,R5                  ;POINT TO STATUS BLOCK
001704  105205  000001                  ADD     #1,R5                   ;POINT TO 2ND WORD
001706  104253                          MOV     (R5)+,R3                ;GET FIRST WORD OF STATUS
001707  104202  000001                  MOV     #1,R2                   ;ERROR SUBCODE IN CASE
001711  102203  000040                  BIT     #ST.DR,R3               ;IS DRIVE IN DIAGNOSTIC MODE
001713  052013                          BNE     STPNIC                  ;YES, WE LOSE
001714  115402                          INC     R2                      ;ERROR SUBCODE 2
001715  102203  000001                  BIT     #ST.RU,R3               ;IS RUN STOP SWITCH OUT
001717  012013                          BEQ     STPNIC                  ;YES, LOSE AGAIN
001720  104202  000004                  MOV     #4,R2                   ;SUBCODE
001722  102203  000002                  BIT     #ST.PS,R3               ;PORT SWITCH OUT ?
001724  012013                          BEQ     STPNIC                  ;YES - DIE PAINFULLY
001725  104032                          MOV     R3,R2                   ;GET STATUS MODE BYTE
001726  110702                          SWAB    R2                      ;SWITCH WRITE PROTECT TO LOW BYTE
001727  102302  001020                  BIT     GSR+1,R2                ;WRITE PROTECTED ?
001731  011745                          BEQ     SRCK                    ;IF NOT CHECK IF SPINNING
001732  104202  000003                  MOV     #3,R2                   ;IN CASE IT'S FATAL
001734  102200  000001  001220          BIT     #WP,FLAG1               ;BEEN HERE ONCE ?
001737  052013                          BNE     STPNIC                  ;YUP - GIVE UP
001740  101200  000001  001220          BIS     #WP,FLAG1               ;SET BEEN HERE FLAG
001743  022235                          CALL    ACCESS                  ;TRY TO RESET IT
001744  001664                          BR      STATST                  ;AND SEE IF IT WORKED
001745  102203  000020          SRCK:   BIT     #ST.SR,R3               ;IS PACK SPINNING?
001747  051752                          BNE     STFORM                  ;YES, TEST FOR FORMAT ENABLE
001750  022160                          CALL    LOAD                    ;NO, SPIN PACK
001751  001664                          JMP     STATST                  ;SEE IF ANYTHING CHANGED
001752  102203  002000          STFORM: BIT     #ST.FO,R3               ;IS FORMATTING ENABLED?
001754  051757                          BNE     STDIAG                  ;YES, TEST FOR DIAG ACCESS
001755  022235                          CALL    ACCESS                  ;NO, SET UP DIAG/FORM ACCESS
001756  001664                          JMP     STATST                  ;SEE IF ANYTHING CHANGED
001757  102203  001000          STDIAG: BIT     #ST.DB,R3               ;IS DIAG CYL ACCESS ALLOWED
001761  051764                          BNE     STWLK                   ;YES, CHECK FOR ERRORS
001762  022235                          CALL    ACCESS                  ;NO, SET UP DIAG/FORM ACCESS
001763  001664                          JMP     STATST                  ;SEE IF ANYTHING CHANGED
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    ;CHARCTERISTICS 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                         ST KP:  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  1042C1  000001         ST"NIC: MOV     #1,R1           ;INDICATE STATUS FAILURE
    74 002015  022552                         CALL    ERRMNT          ;SEND ERROR MSG AND QUIT
```

```
                                        ;
                                        ;       TALK ROUTINE
                                        ;
                                        ;       INPUT PARAMETERS
                                        ;
                                        ;       R2      CONTAINS UNIT NUMBER CODE
                                        ;
                                        ;       R3      CONTAINS A POINTER TO A COMMAND/RESPONSE ADDRESS/SIZE TABLE
                                        ;
                                        ;       OUTPUT PARAMETERS
                                        ;
                                        ;
002016                          TALK:   PUSH    R3
002017                                  PUSH    R4
002020  104663  000001        LOOP1:  MOV     1(SP),R3        ;RESTORE R3 FOR RETRIES
002022  104237                          MOV     (R3)+,R0        ;GET COMMAND ADDRESS
002023  104231                          MOV     (R3)+,R1        ;GET COMMAND SIZE
002024  104302  000740                  MOV     UNIT,R2         ;MAKE SURE HAVE INTERCONNECT
002026  060004                          XFC     SEND            ;SEND GET STATUS COMMAND
002027  115001                          TST     R1              ;SUCESSFUL?
002030  012036                          BEQ     MSG1            ;YES, BRANCH
002031  115400  000716                  INC     UN.ERT          ;INCREMENT ERROR COUNT
002033  104201  000002                  MOV     #2,R1           ;ERROR NUMBER IN CASE
002035  002103                          BR      TCLEAR          ;DO RECOVERY
002036  102200  100000  001217  MSG1:   BIT     #RTY,FLAG       ;IN A RETRY ?
002041  052044                          BNE     LOOP2           ;YES - DON'T CLEAR COUNTER
002042  114000  000716                  CLR     UN.ERT          ;FOR RESET
002044  104231                  LOOP2:  MOV     (R3)+,R1        ;POINT TO RCV BUFFER
002045  104137                          MOV     (R3),R0         ;SET SIZE OF REPLY
002046  104302  000740                  MOV     UNIT,R2         ;MAKE SURE HAVE INTERCONNECT
002050  060005                          XFC     RCV             ;RCV REPLY TO GET STATUS
002051  115001                          TST     R1              ;SUCESSFUL?
002052  012060                          BEQ     TALKDN          ;YES, CHECK STATUS
002053  115400  000715                  INC     UN.ERR          ;INCREMENT ERROR COUNT
002055  104201  000004                  MOV     #4,R1           ;ERROR CODE IN CASE
002057  002103                          BR      TCLEAR          ;DO RECOVERY
002060  106207  000175          TALKDN: CMP     #UNSEC,R0       ;WAS CMD UNSUCESSFUL?
002062  052070                          BNE     TALKRT          ;YES, DONE
002063  115400  000715                  INC     UN.ERR          ;INCREMENT ERROR COUNT
002065  104201  000003                  MOV     #3,R1           ;ERROR CODE IN CASE
002067  002103                          BR      TCLEAR          ;NO, TRY AGAIN
002070  102200  100000  001217  TALKRT: BIT     #RTY,FLAG       ;IN A RETRY ?
002073  052076                          BNE     TALKP           ;YUP - SKIP CLEAR
002074  114000  000715                  CLR     UN.ERR          ;CLEAR FOR REST
002076                          TALKP:  POP     R4              ;RESTORE R4
002077                                  POP     R3              ;RESTORE R3
002100  000000                          RETURN
002101  114002                  ERRT:   CLR     R2              ;CLEAR SUBCODE
002102  022552                          CALL    ERRMNT          ;ERROR EXIT
002103  102200  100000  001217  TCLEAR: BIT     #RTY,FLAG       ;IN A RETRY ?
002106  052115                          BNE     TALKIP          ;YUP - SKIP FLAG SET AND STACK SAVE
002107  101200  100000  001217          BIS     #RTY,FLAG       ;SET FLAG
002112  104060  001216                  MOV     SP,STCKSV       ;SAVE STACK POINTER
002114  002117                          BR      TALIP1          ;SKIP RETRY HANDLING
002115  104306  001216          TALKIP: MOV     STCKSV,SP       ;RESTORE STACK POINTER
002117  106300  001502  000716  TALIP1: CMP     RETRY,UN.ERT    ;DONE RETRIES ?
002122  072101                          BMI     ERRT            ;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          TERR:    MOV     #20.,R1                 ;SET ERROR CODE
 71 002145  114002                           CLR     R2                      ;CLEAR SUBCODE
 72 002146  022552                           CALL    ERRMNT                  ;DIE PEACEFULLY
 73 002147  021662                  TATTN1:  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                                  ;       RECAL ROUTINE
 78                                  ;
 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            ;DONF 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
```

```
                                        CLR     R2                      ;CLEAR SUBCODE
                                        CALL    ERRMNT                  ;DIE
                                ;
                                ;       ACCESS ROUTINE
                                ;
                                ACCESS: MOV     R3,-(SP)                ;SAVE R3
                                        MOV     #CR.ACC,R3              ;POINT TO ACCESS TABLE
                                        CALL    TALK                    ;SEND ACCESS CMD
                                        MOV     (SP)+,R3                ;RESTORE R3
                                        RETURN                          ;RETURN TO CALLER
                                ;
                                ;       CLEAR ROUTINE
                                ;
                                CLEAR:  MOV     R3,-(SP)                ;SAVE R3
                                        MOV     #CR.CLR,R3              ;POINT TO CLEAR TABLE
                                        CALL    TALK                    ;SEND CLEAR CMD
                                        MOV     (SP)+,R3                ;RESTORE R3
                                        RETURN                          ;RETURN TO CALLER
                                ;
                                ;       SEEK ROUTINE
                                ;
                                SEEK:   MOV     R3,-(SP)                ;SAVE R3
                                        MOV     R0,-(SP)                ;SAVE R0
                                SEEK0:  MOV     UNIT,R2                 ;MAKE SURE HAVE UNIT
                                        MOV     #CR.SEK,R3              ;POINT TO SEEK TABLE
                                        CALL    TALK                    ;SEND SEEK COMMAND
                                        MOV     STO,R3                  ;SHORT TIMEOUT
                                SEEK1:  CALL    STATVL                  ;CHECK FOR STATUS VALIDITY
                                        BNE     SEEK5                   ;IF NOT ZERO - DIE
                                        BIT     #ATTN,R1                ;ANY PROBLEMS
                                        BNE     SEEK2                   ;YES, BRANCH
                                        BIT     #RWRDY,R1               ;NO, DONE?
                                        BNE     SEEK6                   ;ALL DONE
                                        DEC     R3                      ;DECREMENT COUNTER
                                        BEQ     SEEK3                   ;IF ZERO THEN DEAD
                                        MOV     #160.,R0                ;1MS DELAY
                                SEEK7:  DEC     R0                      ;DECREMENT COUNTER
                                        BNE     SEEK7                   ;DELAY LOOP
                                        BR      SEEK1                   ;TRY AGAIN
                                SEEK6:  CLR     R1                      ;CLEAR ERROR FLAG
                                SEEK4:  CLR     UN.SEK                  ;FOR RESET
                                        MOV     (SP)+,R0                ;YES, RESTORE R0
                                        MOV     (SP)+,R3                ;RESTORE R3
                                        RETURN                          ;RETURN TO CALLER
                                SEEK5:  CALL    INITPT                  ;INIT THE DRIVE
                                SEEK2:  INC     UN.SEK                  ;INCREMENT RETRY COUNTER
                                        CMP     RETRY,UN.SEK            ;HAVE WE DONE ALL RETRIES?
                                        BEQ     SEEK3                   ;YES, PANIC
                                        CALL    GSTATS                  ;PANIC AND CALL GET STATUS
                                        CALL    RECAL                   ;RECAL DRIVE
                                        BR      SEEK0                   ;AND TRY AGAIN
                                SEEK3:  MOV     #-3,R1                  ;SET ERROR CODE
                                        BR      SEEK4                   ;RESTORE REGS AND RETURN
                                ;
                                ;       UNLOAD ROUTINE
                                ;
                                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                             ;
208                             ;       INIT GIVEN PORT
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
```

```
                                                .SBTTL  OVERLAY PROCESSING ROUTINES
                                        ;
                                        ;       OVERLAY PROCESSING ROUTINES
                                        ;       R1 = OFFSET INTO TABLE
                                        ;
                                        ;
                                        ;       NEXT CALLS OVERLAY FOR NEXT CODE OVERLAY
                                        ;
002412  104204  001341        NEXT:   MOV     #OVLTBL,R4              ;GET POINTER TO OVERLAY TABLE
002414  105014                        ADD     R1,R4                   ;INDEX INTO TABLE
002415  104203  003047                MOV     #START,R3               ;UDA ADDRESS TO LOAD AT
002417  022526                NEXT5:  CALL    OVRLAY                  ;CALL ROUTINE TO DO OVERLAY
002420  115007                        TST     R0                      ;CHECKSUM O.K. ??
002421  052426                        BNE     OERR                    ;YES - RETRY IF POSSIBLE
002422  114000  000715                CLR     UN.ERR                  ;CLEAR ERROR COUNT
002424                                POP     R1                      ;POP CURRENT RETURN ADDRESS
002425  003047                        BR      START                   ;GO TO OVERLAY
002426  106300  001502  000715  OERR:   CMP     RETRY,UN.ERR            ;DONE ALL RETRIES ?
002431  012435                        BEQ     OERR2                   ;YUP
002432  115400  000715                INC     UN.ERR                  ;INC ERROR AND
002434  002417                        BR      NEXT5                   ;TRY AGAIN
002435  104012                OERR2:  MOV     R1,R2                   ;GET ERROR CODE FROM XFC
002436  104201  000005                MOV     #5,R1                   ;SET UNIBUS READ ERROR
002440  101200  000020  001220        BIS     #DEAD,FLAG1             ;INDICATE HOST GONE
002443  022552                        CALL    ERRMNT                  ;ERROR RETURN
```

```
                                        ;
                                        ;       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
                                        ;
                                        ;
002444  104303  001153          PAGE:   MOV     CUROVL,R3       ;GET ''CALLING'' OVERLAY
002446                                  PUSH    R3              ;PUSH FOR LATER RETURN
002447  104204  001341                  MOV     #OVLTBL,R4      ;POINT TO OVERLAY TABLE
002451  105014                          ADD     R1,R4           ;POINT TO ENTRY FOR NEW OVERLAY
002452  104203  003047                  MOV     #START,R3       ;POINT TO UDA LOAD ADDRESS
002454  022526                  PALP1:  CALL    OVRLAY          ;BRING IN NEW OVERLAY
002455  115007                          TST     R0              ;EDC O.K. ?
002456  012461                          BEQ     PALP2           ;YUP
002457  022510                          CALL    PAERR           ;CALL ERROR HANDLER
002460  002454                          BR      PALP1           ;TRY AGAIN
002461  114000  000715          PALP2:  CLR     UN.ERR          ;CLEAR ERROR COUNT
002463  104203  002472                  MOV     #PAGER,R3       ;ADDRESS FOR OVERLAY TO RETURN TO
002465  100463                          MOV     R3,-(SP)        ;PUSH ON STACK
002466  104203  003047                  MOV     #START,R3       ;STARTING ADDRESS OF NEW OVERLAY
002470  100463                          MOV     R3,-(SP)        ;PUSH ON STACK FOR ''CALL''
002471  000000                          RETURN                  ;''CALL'' OVERLAY
                                                                ;NEXT ADDRESS ON STACK IS RETURN
                                                                ;ADDRESS TO PAGE
002472                          PAGER:  POP     R1              ;POP OLD OVERLAY NUMBER
002473  104010  001153                  MOV     R1,CUROVL       ;MAKE IT CURRENT
002475  104204  001341                  MOV     #OVLTBL,R4      ;POINT TO OVERLAY TABLE
002477  105014                          ADD     R1,R4           ;POINT TO OLD OVERLAY BLOCK
002500  104203  003047                  MOV     #START,R3       ;POINT TO UDA LOAD ADDRESS
002502  022526                  PALP4:  CALL    OVRLAY          ;BRING IT IN
002503  115007                          TST     R0              ;EDC O.K. ?
002504  012507                          BEQ     PALP3           ;YUP
002505  022510                          CALL    PAERR           ;ERROR HANDLER
002506  002502                          BR      PALP4           ;TRY AGAIN
002507  000000                  PALP3:  RETURN                  ;RETURN TO ADDRESS PAGE CALLED FROM
002510  106300  001502  000715  PAERR:  CMP     RETRY,UN.ERR    ;DONE ALL RETRIES ?
002513  012517                          BEQ     PAERR1          ;YUP
002514  115400  000715                  INC     UN.ERR          ;INC ERROR COUNT
002516  000000                          RETURN                  ;RETURN AND TRY AGAIN
002517  101200  000020  001220  PAERR1: BIS     #DEAD,FLAG1     ;SET HOST GONE
002522  114002                          CLR     R2              ;NO SUBCODE
002523  104201  000005                  MOV     #5,R1           ;UNIBUS READ ERROR
002525  022552                          CALL    ERRMNT          ;DIE
```

```
                                ;
                                ;
                                ;       OVERLAY ROUTINE
                                ;       ISSUES UNIBUS READ TO GET OVERLAY
                                ;       R3 -> UDA ADDRESS TO LOAD
                                ;
002526  104647  000001  OVRLAY: MOV     HSTLO(R4),R0            ;LOW ORDER UNIBUS ADDRESS
002530  104641  000002,        MOV     HSTHI(R4),R1            ;HI ORDER UNIBUS ADDRESS
002532  104642  000000          MOV     LEN(R4),R2              ;WORD COUNT TO OVERLAY
002534  060013                  XFC     UREAD                   ;ISSUE UNIBUS READ
002535  000000                  RETURN                          ;RETURN TO CALLING ROUTINE
```

```
                                        .SBTTL  MISCELANEOUS COMMON ROUTINES
                                ;
                                ;       SEND MAINTENANCE READ COMMAND
                                ;
002536  104207  001263          SNDMNT: MOV     #DMBUF,R0               ;POINT TO BUFFER
002540  104201  000016                  MOV     #DMBUFL,R1              ;LENGTH
002542  060016                          XFC     MAINTR                  ;ISSUE COMMAND
002543  000000                          RETURN
                                ;
                                ;
                                ;       RECEIVE MAINTENANCE WRITE DATA
                                ;
                                ;
002544  104207  001263          RCVMNT: MOV     #DMBUF,R0               ;POINT TO BUFFER
002546  104201  000016                  MOV     #DMBUFL,R1                      ;SIZE OF BUFFER
002550  060017                          XFC     MAINTW                  ;RECEIVE MAINT WRITE DATA
002551  000000                          RETURN
                                ;
                                ;
                                ;       ERROR RETURN TO HOST
                                ;
                                ;
                                ;
002552  102200  000010  001220  ERRMNT: BIT     #ERDN,FLAG1                     ;TRIED TO QUIT YET ?
002555  052603                          BNE     ALLOVR                          ;YUP - DISCON FAILED
002556  101200  000010  001220          BIS     #ERDN,FLAG1                     ;SET FLAG
002561  104207  001263                  MOV     #DMBUF,R0                       ;POINT TO MAINT BUFFER
002563  100671  000002                  MOV     R1,2(R0)                        ;PUT ERROR NUMBER IN MSG
002565  100672  000003                  MOV     R2,3(R0)                        ;PUT IN ERROR SUBCODE
002567  114002                          CLR     R2                              ;CLEAR *** TEMP ***
002570  100672  000004                  MOV     R2,4(R0)                        ;ONLY ONE WORD RIGHT NOW ***
002572  104301  001324                  MOV     EMSG,R1                         ;GET DUP CODE
002574  100171                          MOV     R1,(R0)                         ;STORE IT
002575  104301  001325                  MOV     EMSG+1,R1                       ;GET 'E ' IDENTIFIER
002577  100671  000001                  MOV     R1,1(R0)                        ;STORE IT
002601  022536                          CALL    SNDMNT                          ;SEND TO HOST
002602  022324                          CALL    DISCON                          ;DISCONNECT/SPINDOWN DRIVE
002603  114007                  ALLOVR: CLR     R0                              ;IN CASE O.K
002604  102200  000020  001220          BIT     #DEAD,FLAG1                     ;DIE OR JUST QUIT ?
002607  012611                          BEQ     ALLOV1                          ;JUST QUIT
002610  115407                          INC     R0                              ;MAKE NON ZERO
002611  060021                  ALLOV1: XFC     DONE                            ;EXIT DM MODE
                                ;
                                ;       ERROR RETRY SEQUENCER
                                ;
002612  115000  001505          ERRHND: TST     RECTMP                          ;RECOVERY LEVEL 0 ?
002614  012624                          BEQ     ERRLST                          ;YES - DO LAST DITCH EFFORT
002615  104300  001505  001103          MOV     RECTMP,ERECOV+1                 ;STORE LEVEL IN COMMAND
002620  104203  001005                  MOV     #CR.ERV,R3                      ;POIT TO COMMAND
002622  022016                          CALL    TALK                            ;DO ERROR RECOVERY
002623  000000                          RETURN                                  ;RETURN
002624  022154                  ERRLST: CALL    RECAL                           ;DO A RECALIBRATE
002625  022251                          CALL    SEEK                            ;RESEEK
002626  000000                          RETURN
```

```
                                ;
                                ;
                                ;       COMPUTE EDC
                                ;       R2 -> BUFFER
                                ;
002627                          CEDC:   PUSH    R2,R4,R5
002632  104303  001246                  MOV     INI,R3          ;INITIAL VALUE (69)
002634  104305  001247                  MOV     CNT,R5          ;SECTOR SIZE (258)
002636  104224                  EAGAIN: MOV     (R2)+,R4        ;MOVE A WORD INTO R4
002637                                  XOR     R4,R3           ;XOR INTO EDC
002643  105203  000000                  ADD     #0,R3           ;CLEAR CARRY
002645  110203                          ROL     R3              ;CIRCLE LEFT
002646  042650                          BCC     NOCRY           ;IF NO CARRY SKIP
002647  115403                          INC     R3              ;INCREMENT IF CARRY
002650  117405                  NOCRY:  DEC     R5              ;DECREMENT COUNTER
002651  052636                          BNE     EAGAIN          ;IF NOT 0 DO NEXT WORD
002652                                  POP     R5,R4,R2
002655  000000                          RETURN                  ;R3 = EDC
                                ;
                                ;
                                ;
                                ;       ONE SECOND DELAY ROUTINE
                                ;
                                ;
002656                          TIMER:  PUSH    R3              ;SAVE R3
002657  104203  100000                  MOV     #TIMVAL,R3      ;LOOP COUNTER (32768)
002661  104011                  TIMLP:  MOV     R1,R1           ;DELAY
002662  104011                          MOV     R1,R1           ;DELAY
002663  104011                          MOV     R1,R1           ;DELAY
002664  104011                          MOV     R1,R1           ;DELAY
002665  117403                          DEC     R3              ;DECRMENT COUNTER
002666  052661                          BNE     TIMLP           ;LOOP TILL DONE
002667                                  POP     R3              ;RESTORE R3
002670  000000                          RETURN                  ;RETUR TO CALLER
                                ;
                                ;
                                ;
                                ;       VALIDATE XFC STATUS RESPONSE
                                ;
                                ;
002671                          STATVL: PUSH    R5              ;SAVE R5
002672  104205  077777                  MOV     #77777,R5       ;COUNTER (1 SECOND)
002674  060007                  STATRE: XFC     STATUS          ;GET DRIVE STATUS
002675  117405                          DEC     R5              ;INCREMENT COUNTER
002676  012710                          BEQ     STATFR          ;IF ZERO - NO GOOD
002677  102201  000004                  BIT     #VLD,R1         ;LOW VALID BIT GOOD ?
002701  012674                          BEQ     STATRE          ;NO - RETRY
002702  102201  000400                  BIT     #PARITI,R1      ;PARITY ERROR ?
002704  052674                          BNE     STATRE          ;YES - RETRY
002705                                  POP     R5              ;RESTORE R5
002706  106011                          CMP     R1,R1           ;SET COND CODE EQ ZERO
002707  000000                  STATRT: RETURN                  ;RETURN
002710                          STATFR: POP     R5              ;RESTORE R5
002711  106201  177777                  CMP     #177777,R1      ;MAKE SURE NOT EQUAL
002713  002707                          BR      STATRT          ;RETURN
```

```
                                        ;
                                        ;       CONVERT BLOCK NUMBER TO PHYSICAL ADDRESS AND SEEK
                                        ;       R4 -> BLOCK NUMBER
                                        ;
002714  104207  001430          CVTSK:  MOV     #CONBLK,R0              ;POINT AT CONVERT BLOCK
002716  104641  000001                  MOV     1(R4),R1                ;GET HIGH ORDER
002720  100671  000003                  MOV     R1,V2+1(R0)             ;STORE IT
002722  104143                          MOV     (R4),R3                 ;GET LOW ORDER
002723  100673  000002                  MOV     R3,V2(R0)               ;STORE IT
002725  104201  001052                  MOV     #SCR,R1                 ;POINT TO SUBUNIT CHARACTERISTICS
002727  060020                          XFC     CVT                     ;CONVERT IT
002730  104670  000011  001112          MOV     TRK(R0),CURTRK          ;GET TRACK NUMBER
002733  104670  000006  001076          MOV     CYL(R0),ISEEK+1         ;LOW ORDER XYLINDER
002736  104670  000007  001077          MOV     CYL+1(R0),ISEEK+2       ;HIGH ORDER CYLINDR
002741  104670  000010  001100          MOV     GRP(R0),ISEEK+3         ;GROUP NUMBER
002744  022251                          CALL    SEEK                    ;DO SEEK
002745  115001                          TST     R1                      ;ANY ERROR ?
002746  052750                          BNE     CVTERR                  ;YUP ERROR
002747  000000                          RETURN
002750  104201  000012          CVTERR: MOV     #10.,R1                 ;SEEK ERROR
002752  104207  001430                  MOV     #CONBLK,R0              ;CONVERT BLOCK
002754  104672  000006                  MOV     CYL(R0),R2              ;CYLINDER FAILED ON
002756  022552                          CALL    ERRMNT                  ;ERROR RETURN
```

```
                                        ;
                                        ;       COMPUTE PRIMARY RBN FOR GIVEN LBN
                                        ;       R0 ->   CHARACTERISTICS BLOCK
                                        ;       RBN = (QUO(CURBN-STLBN)/LBNTRK)*RBNTRK
                                        ;
002757                                  PRIMRB: PUSH    R2
002760  104140  000736                          MOV     (R4),TEMP               ;GET LOW ORDER LBN
002762  104642  000001                          MOV     1(R4),R2                ;GET HIGH ORDER
002764  104204  000736                          MOV     #TEMP,R4                ;FOR SUBTRACT
002766  107302  001335                          SUB     ST.LBN,R2               ;DO SUBTRACT
002770  100642  000001                          MOV     R2,1(R4)                ;STORE BACK
002772  104673  000011                          MOV     LBNTRK(R0),R3           ;GET LBN/TRACK
002774  103203  177400                          BIC     #HIBYTE,R3              ;CLEAR HIGH BYTE
002776  104030  000731                          MOV     R3,DDUMMY               ;STORE FOR COMPUTATION
003000  114000  000732                          CLR     DDUMMY+1                ;CLEAR FOR STORE
003002  104203  000731                          MOV     #DDUMMY,R3              ;FOR DIVIDE
003004  021570                                  CALL    DDIV                    ;
003005  104673  000004                          MOV     RBNTRK(R0),R3           ;GET RBN/TRACK
003007  103203  177600                          BIC     #HI1BYTE,R3             ;CLEAR GARBAGE
003011  104030  000731                          MOV     R3,DDUMMY               ;FOR COMPUTATION
003013  114000  000732                          CLR     DDUMMY+1                ;CLEAR HIGH ORDER
003015  104203  000731                          MOV     #DDUMMY,R3              ;FOR MULTIPLY
003017  021542                                  CALL    DMUL
003020  104140  001151                          MOV     (R4),REVRBN             ;GET LOW ORDER
003022  104640  000001  001152                  MOV     1(R4),REVRBN+1          ;STORE HIGH ORDER
003025                                          POP     R2
003026  000000                                  RETURN
```

```
                                ;
                                ;
                                ;       COMPUTE ECC SYMBOLS IN ERROR
                                ;
                                ;
003027  104202  001037  FCCCK:  MOV     #CR,R2          ;POINT TO CHARACTERISTICS
003031  104623  000002          MOV     ERRSYM(R2),R3   ;GET THRESHOLD
003033  103203  177400          BIC     #HIBYTE,R3      ;CLAR HIGH GARBAGE
003035  104207  000721          MOV     #RDBLK,R0       ;POINT TO COMMAND BLOCK
003037  060015                  XFC     ECC             ;PERFORM ECC CORRECTION
003040  115001                  TST     R1              ;SUCCESSFUL ?
003041  053044                  BNE     105$            ;NOPE
003042  106073                  CMP     R0,R3           ;WITHIN BOUNDS ?
003043  073046                  BMI     GDECC           ;YES CONSIDER GOOD
003044  104201  177777  105$:   MOV     #-1,R1          ;ELSE SIGNAL BAD
003046  000200          GDECC:  RETURN                  ;RETURN
```

```
003047  003063                          START:  JMP     START3          ;SKIP LOCAL DATA STORAGE
                                        ;
                                        ;
                                        ;
003050                                  DESC:   .BLKW   4               ;HOST BUFFER DESCRIPTOR
003054  010000                          DMSG1:  .WORD   10000           ;TYPE AND SEQUENCE NUMBER
003055     123     040                          .ASCII  'S '            ;START UP
003056     125     120     040                  .ASCIZ  'UP YOURS'      ;MESSAGE ZERO
003057     131     117     125
003061     122     123     000
        000007                          DMSG1L  =       .-DMSG1         ;LENGTH OF MESSAGE ZERO
                                        ;
                                        ;
003063  104206  001215                  START3: MOV     #STACK,SP       ;SET UP STACK
003065  104207  003054                          MOV     #DMSG1,R0       ;MESSAGE
003067  104201  000007                          MOV     #DMSG1L,R1      ;LENGTH
003071  060016                                  XFC     MAINTR          ;SEND MESSAGE
003072  104207  001263                          MOV     #DMBUF,R0       ;ADDRESS OF BUFFER
003074  104201  000016                          MOV     #DMBUFL,R1      ;LENGTH OF BUFFER
003076  060017                                  XFC     MAINTW          ;GET ANSWER
003077  023103                                  CALL    SETOVL          ;SET UP OVERLAY ADDRESSES
003100  104201  000060                          MOV     #G1,R1          ;NEXT OVERLAY
003102  022412                                  CALL    NEXT            ;BRING IT IN
```

```
                                        ;
                                        ;       SET UP OVERLAY TABLE
                                        ;
003103  104205  000021          SETOVL: MOV     #OVCNT,R5               ;GET COUNT OF OVERLAYS
003105  104204  007774                  MOV     #DUPOVL,R4              ;POINT TO OVERLAY ADDRESS (2 WORDS)
003107  104200  005472  000731          MOV     #OVS.G1,DDUMMY          ;RELATIVE START OF FIRSTOVERLAY
003112  114000  000732                  CLR     DDUMMY+1                ;CLEAR HIGH ORDER
003114  104203  000731                  MOV     #DDUMMY,R3              ;FOR SUB
003116  021524                          CALL    DSUB                    ;GET OFFSET (MUST ADD TO RELATIVE START ADDRESS OF E
003117  104043                          MOV     R4,R3                   ;CHANGE POINTER FOR ADDS
003120  104204  001342                  MOV     #OVLTBL+HSTLO,R4        ;POINT TO LOW HOST ADD OF FIRST ENTRY
003122  021506                  SLOOP:  CALL    DADD                    ;ADD OFFSET
003123  105204  000003                  ADD     #OVLEN,R4               ;POINT TO LOW HOST ADD. OF NEXT ENTRY
003125  117405                          DEC     R5                      ;DECREMENT COUNTER
003126  053122                          BNE     SLOOP                   ;IF NON-ZERO THEN CONTINUE
003127  000000                          RETURN                          ;ELSE DONE
```

```
                                                .SBTTL  INITIALIZATION OVERLAY
                                        ;
                                        ;       INITIALIZATION OVERLAY
                                        ;
003130                                  DMOVLY  G1,START
003047  024430                          CALL    INITL                   ;INITIALIZE DISK
003050  115001                          TST     R1                      ;ANY ERRORS ?
003051  053064                          BNE     BUMER                   ;YUP - QUIT
003052  102200  000001  001217          BIT     #FCTAVL,FLAG            ;USE RESIDENT FCT ?
003055  053061                          BNE     DOLBN                   ;YES - ONLY DO LBN
003056  104201  000000                  MOV     #F1,R1                  ;ELSE DO D/XBN FIRST
003060  003063                          BR      DXBN                    ;SKIP LBN FLAGGING
003061  104201  000003          DOLBN:  MOV     #F2,R1                  ;SIGNAL LBN FORMAT
003063  022412                  DXBN:   CALL    NEXT                    ;BRING IN NEXT OVERLAY
003064  104201  000006          BUMER:  MOV     #6,R1                   ;SIGNAL INIT
003066  022552                          CALL    ERRMNT                  ;ERROR RETURN
                                        ;
                                        ;
                                        ;       COMPUTE DISK CONSTANTS FROM CHARACTERISTICS
                                        ;
                                        ;
                                        ;
                                        ;       COMPUTE TRACKS/CYLINDER
                                        ;
003067  104207  001052          CONINT: MOV     #SCR,R0                 ;POINT TO SUB CHARACTERISTICS
003071  104673  000003                  MOV     TRKGRP(R0),R3           ;LOAD TRACKS/GROUP
003073  103203  177400                  BIC     #HIBYTE,R3              ;CLEAR HIGH BYTE
003075  104030  000731                  MOV     R3,DDUMMY               ;STORE IN DUMMY AREA
003077  114000  000732                  CLR     DDUMMY+1                ;CLEAR FOR STORE
003101  104673  000002                  MOV     GRPCYL(R0),R3           ;GET GROUPS/CYLINDER
003103  103203  177400                  BIC     #HIBYTE,R3              ;CLEAR HIGH BYTE
003105  104030  000736                  MOV     R3,TEMP                 ;STORE IN TEMP AREA
003107  114000  000737                  CLR     TEMP+1                  ;CLEAR HIGH ORDER
003111  104203  000731                  MOV     #DDUMMY,R3              ;SETR UP FOR MULT
003113  104204  000736                  MOV     #TEMP,R4                ;DITTO
003115  021542                          CALL    DMUL                    ;COMPUTE IT
003116  104140  001141                  MOV     (R4),TRKCYL             ;LOAD FOR STORE
003120  104640  000001  001142          MOV     1(R4),TRKCYL+1          ;LOAD FOR STORE
                                        ;
                                        ;       COMPUTE LBN'S/CYLINDER
                                        ;
003123  104673  000011                  MOV     LBNTRK(R0),R3           ;GET LBN'S/TRACK
003125  103203  177400                  BIC     #HIBYTE,R3              ;CLEAR HIGH BYTE
003127  104030  000736                  MOV     R3,TEMP                 ;FOR MULT
003131  114000  000737                  CLR     TEMP+1                  ;FOR STORE
003133  104204  000736                  MOV     #TEMP,R4                ;FOR MULTIPLY
003135  104203  001141                  MOV     #TRKCYL,R3              ;DITTO
003137  021542                          CALL    DMUL                    ;GET LBN'S/CYL
003140  104140  001145                  MOV     (R4),LBNPCY             ;GET LOW ORDER
003142  104640  000001  001146          MOV     1(R4),LBNPCY+1          ;GET HIGH ORDER
                                        ;
                                        ;       COMPUTE RBN'S/CYLINDER
                                        ;
003145  104673  000004                  MOV     RBNTRK(R0),R3           ;GET RBN'S/TRACK
003147  103203  177600                  BIC     #HI1BYTE,R3             ;CLEAR OUT GARBAGE
003151  104030  000736                  MOV     R3,TEMP                 ;STORE FOR MULT
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     #HIBYTE,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
```

```
                                                CALL    DMUL                    ;COMPUTE IT
                                                MOV     (R4),SECTCY             ;LOAD FOR STORE
                                                MOV     1(R4),SECTCY+1          ;LOAD FOR STORE
                                        ;
                                        ;       COMPUTE SECTORS IN XBN AREA (R4 -> SECTORS/CYL)
                                        ;
                                                MOV     XBNCYL(R0),TEMP         ;LOAD NUMBER OF XBN CYLINDERS
                                                CLR     TEMP+1                  ;CLEAR FOR STORE
                                                MOV     #TEMP,R3                ;SET UP FOR MULT
                                                CALL    DMUL                    ;GET XBN SECTORS
                                                MOV     (R4),XBNSEC             ;LOAD FOR STORE
                                                MOV     1(R4),XBNSEC+1          ;LOAD FOR STORE
                                        ;
                                        ;       COMPUTE LAST CYLINDER NUMBER ON SUBUNIT
                                        ;
                                                MOV     #SCR,R0                 ;POINT TO CHARACTERISTICS
                                                MOV     CYLBN(R0),LBNCYL        ;GET LOW ORDER LBN CYLINDERS
                                                MOV     CYLBN(R0),TEMP          ;ALSO FOR MATH
                                                MOV     CYLBN+1(R0),R4          ;GET HIGH ORDER
                                                MOV     R4,LBNCYL+1             ;STORE HI ORDER
                                                MOV     R4,TEMP+1               ;ALSO FOR MATH
                                                MOV     #TEMP,R4                ;SET UP FOR ADD
                                                MOV     XBNCYL(R0),DDUMMY       ;LOAD XBN CYLINDERS
                                                CLR     DDUMMY+1                ;CLEAR HIGH ORDER
                                                MOV     #DDUMMY,R3              ;SET UP FOR ADD
                                                CALL    DADD                    ;GET LBN+XBN CYLINDERS
                                                MOV     DBNCYL(R0),R3           ;LOAD DBN CYLINDERS
                                                SWAB    R3                      ;GET INTO LO BYTE
                                                BIC     #HIBYTE,R3              ;CLEAR GARBAGE
                                                MOV     R3,DDUMMY               ;FOR DIVIDE
                                                CLR     DDUMMY+1                ;CLEAR HIGH ORDER
                                                MOV     #DDUMMY,R3              ;SET UP FRO ADD
                                                CALL    DADD                    ;GET LBN+XBN+DBN CYLINDERS
                                                MOV     1(R4),R2                ;GET HIGH ORDER
                                                MOV     STCYL(R0),R3            ;GET CYLINDER BITS
                                                BIC     #LO,R3                  ;CLEAR OUT REST OF WORD
                                                BIS     R3,R2                   ;OR IN CYLINDER HIGH BITS
                                                MOV     R2,1(R4)                ;STORE BACK
                                                MOV     #ONE,R3                 ;DECREMENT TO GET NUM OF LAST CYL
                                                CALL    DSUB                    ;DO IT
                                                MOV     (R4),CYLNUM             ;MAKE IT CURRENT CYLINDER
                                                MOV     1(R4),CYLNUM+1          ;LOAD HI ORDER
                                        ;
                                        ;       COMPUTE NON-PAD RCT
                                        ;
                                                MOV     RBNLBN,TEMP             ;GET LOW ORDER LBN'S IN HOST AREA
                                                MOV     RBNLBN+1,TEMP+1         ;GET HIGH ORDER
                                                MOV     #127.,DDUMMY            ;ADD 127 FOR DIV FUNCTION
                                                CLR     DDUMMY+1                ;FOR CLEAR
                                                MOV     #TEMP,R4                ;FOR ADD
                                                MOV     #DDUMMY,R3              ;DITTO
                                                CALL    DADD                    ;ADD
                                                MOV     #128.,DDUMMY            ;FOR DIVIDE (128 RBN/RCT BLOCK)
                                                CLR     DDUMMY+1                ;FOR STORE
                                                MOV     #DDUMMY,R3              ;POINT TO IT
                                                CALL    DDIV                    ;DO DIVIDE
                                                MOV     (R4),RCTLBN             ;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  00073'          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                                 ;       INITIALIZE GOOD DATA BLOCK
200                                 ;
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          OVER:   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                                 ;       SET UP END OF IMAGE POINTER
217                                 ;
218
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
```

```
                                ;
                                ;
                                ;       COMPUTE INTERLEAVE FACTOR
                                ;
                                ;
003602  104207  001037                MOV     #CR,R0                  ;POINT TO CHARACTERISTICS BLOCK
003604  104673  000007                MOV     REVSEC(R0),R3           ;GET REVS/SECOND
003606  110703                       SWAB    R3                      ;GET INTO LOW BYTE
003607  103203  177400                BIC     #HIBYTE,R3              ;CLEAR HIGH BYTE
003611  104030  000736                MOV     R3,TEMP                 ;FOR MULTIPLY
003613  114000  000737                CLR     TEMP+1                  ;CLEAR FOR STORE
003615  104204  000736                MOV     #TEMP,R4                ;SET UP FOR MULTIPLY
003617  104203  001127                MOV     #SECTRK,R3              ;SECTORS/TRACK
003621  021542                       CALL    DMUL                    ;GET SECTORS/SECOND
003622  106300  001235  000736        CMP     CUTOF,TEMP              ;WITHIN LIMITS ?
003625  033635                       BPL     TWO                     ;DO BI-LEAVE
003626  104200  000011  001234        MOV     #THREB,TBLK             ;ELSE DO TRI-LEAVE
003631  104200  000044  001233        MOV     #44,SKPCNT              ;INIT CHECK PASS OFFSET
003634  003643                       BR      ISKIP                   ;SKIP BI-LEAVE SETUP
003635  104200  000006  001234  TWO:  MOV     #TWOB,TBLK              ;DO BI-LEAVE
003640  104200  000033  001233        MOV     #33,SKPCNT              ;INIT CHECK PASS OFFSET
003643                          ISKIP:
```

```
                                        ;
                                        ;       FILL IN FCT INFO
                                        ;
003643  104207  001037                  MOV     #CR,R0                  ;POINT TO CHARACTERISTICS BLK
003645  104673  000001                  MOV     FRCPY(R0),R3            ;GET F/RCT COPIES
003647  110703                          SWAB    R3                      ;GET INTO LOW BYTE
003650  103203  177760                  BIC     #FCLR,R3                ;CLEAR OUT REST OF GARBAGE
003652  104030  001244                  MOV     R3,FCTCPY               ;STORE
003654  104207  001052                  MOV     #SCR,R0                 ;POINT TO SUBUNIT CHARACTERISTICS
003656  104670  000010  001240          MOV     FCTSZ(R0),FCTFMT        ;GET FCT SIZE IN SECTORS
003661  114000  001241                  CLR     FCTFMT+1                ;CLEAR HIHG ORDER
003663  104670  000014  001242          MOV     RCTSZ(R0),RCTFMT        ;GET RCT SIZE
003666  114000  001243                  CLR     RCTFMT+1                ;CLEAR HIHG ORDER
                                        ;
                                        ;
                                        ;
                                        ;       COMPUTE HIGHEST PBN IN LBN AREA
                                        ;
                                        ;
003670  104300  001133  000736          MOV     LBNLBN,TEMP             ;GET LOW ORDER
003673  104300  001134  000737          MOV     LBNLBN+1,TEMP+1         ;GET HIGH ORDER
003676  104204  000736                  MOV     #TEMP,R4                ;FOR SUB
003700  104203  001135                  MOV     #RBNLBN,R3              ;POINT TO RBN'S IN LBN AREA
003702  021506                          CALL    DADD                    ;ADD TO GET HIGHEST PBN
003703  104203  001466                  MOV     #ONE,R3                 ;DITTO
003705  021524                          CALL    DSUB                    ;TO GET ACTUAL PBN NUMBER
003706  104642  000001                  MOV     1(R4),R2                ;GET HIGH ORDER
003710  107302  001335                  SUB     ST.LBN,R2               ;SUBTRACT TO GET RELATIVE LAST BLOCK
003712  104020  001155                  MOV     R2,HGHPBN+1             ;STORE IT
003714  104140  001154                  MOV     (R4),HGHPBN             ;GET HIGH ORDER
```

```
                                        ;
                                        ;       COMPUTE REVECTOR BUFFER ADDRESS AND MAX REVECTOR COUNT
                                        ;
003716  104200  000003  000736          MOV     #IMLEN,TEMP             ;GET LENGTH OF FORMAT IMAGE BLOCK
003721  114000  000737                  CLR     TEMP+1                  ;CLEAR FOR STORE
003723  104204  000736                  MOV     #TEMP,R4                ;FOR MULT
003725  104203  001127                  MOV     #SECTRK,R3              ;SECTORS/TRACK
003727  021542                          CALL    DMUL                    ;GET LENGTH OF FORMAT BUFFER TABLE
003730  104303  000736                  MOV     TEMP,R3                 ;GET LENGTH
003732  104204  007275                  MOV     #IMAGE,R4               ;GET IMAGE BUFFER START ADDRESS
003734  105034                          ADD     R3,R4                   ;GET START ADDRESS OF REVECTOR BUFFER
003735  104040  001224                  MOV     R4,ERRBUF               ;STORE IT
003737  104040  001252                  MOV     R4,ERPNT                ;INIT POINTER
003741  104200  007775  000736          MOV     #BMAX,TEMP              ;GET MAX BUFFER ADDRESS
003744  114000  000737                  CLR     TEMP+1                  ;FOR CLEAR
003746  104040  000731                  MOV     R4,DDUMMY               ;STORE BEGINNING ADDRESS
003750  114000  000732                  CLR     DDUMMY+1                ;CLEAR HIGH WORD
003752  104204  000736                  MOV     #TEMP,R4                ;POINT TO END ADDRESS
003754  104203  000731                  MOV     #DDUMMY,R3              ;POINT TO BEGINNING ADDRESS
003756  021524                          CALL    DSUB                    ;SUBTRACT TO GET LENGTH
003757  104304  000736                  MOV     TEMP,R4                 ;GET LENGTH
003761  110604                          ROR     R4                      ;DIVIDE BY 2 (LENGTH OF 1 ENTRY)
003762  104040  001225                  MOV     R4,EMAX                 ;STORE AS MAX NUMBER
                                        ;
                                        ;       STORE RETRY AND RECOVERY LEVELS
                                        ;
003764  104207  001037                  MOV     #CR,R0                  ;POINT TO CHARACTERISTICS
003766  104673  000002                  MOV     ERCV(R0),R3             ;GET RECOVERY LEVELS
003770  103203  177400                  BIC     #HIBYTE,R3              ;CLEAR HIGH ORDER
003772  104030  001503                  MOV     R3,RECOV                ;STORE IT
003774  104030  001505                  MOV     R3,RECTMP               ;INIT COUNTER
003776  104673  000001                  MOV     RTRY(R0),R3             ;GET RETRY NUMBER
004000  110603                          ROR     R3                      ;SHIFT BY FOUR TO GET INTO LOW ORDER NIBBLE
004001  110603                          ROR     R3
004002  110603                          ROR     R3
004003  110603                          ROR     R3
004004  103203  177700                  BIC     #HI2BYTE,R3             ;CLEAR HIGH ORDER JUNK
004006  115403                          INC     R3                      ;ONE MORE BECAUSE OF WRONG INC
004007  104030  001502                  MOV     R3,RETRY                ;STORE IT
004011  114000  001504                  CLR     TMPTRY                  ;FOR STORE
                                        ;
                                        ;       SET UP LONG TIMEOUT
                                        ;
004013  104207  001037                  MOV     #CR,R0                  ;POINT TO COMMON CHARACTERISTICS
004015  104673  000001                  MOV     LONGTO(R0),R3           ;GET LONG TIMEOUT IN LOG2
004017  103203  177760                  BIC     #FCLR,R3                ;CLEAR ALL BUT TIMEOUT
004021  117403                          DEC     R3                      ;ALREADY HAVE SHIFTED ONCE
004022  104201  000001                  MOV     #1,R1                   ;INIT COUNTER
004024  105201  000000                  ADD     #0,R1                   ;CLEAR CARRY
004026  110201          TIMLOP: ROL     R1                      ;SHIFT
004027  117403                  DEC     R3                      ;DECREMENT SHIFT COUNT
004030  054026                  BNE     TIMLOP                  ;CONTINUE TILL DONE
004031  104010  001250                  MOV     R1,LTO                  ;STORE IT FOR US LATER
                                        ;
                                        ;
```

```
                                        ;       SET UP SHORT TIMEOUT
                                        ;
                                        ;
004033  104207  001037                          MOV     #CR,R0                  ;POINT TO COMM CHAR
004035  104673  000000                          MOV     SHORTO(R0),R3           ;GET SHORT TIMEOUT WORD
004037  103203  177760                          BIC     #FCLR,R3                ;CLEAR WORD
004041  117403                                  DEC     R3                      ;FOR LOG CALCULATION
004042  104201  000001                          MOV     #1,R1                   ;INIT COUNTER
004044  105201  000000                          ADD     #0,R1                   ;CLEAR CARRY
004046  110201                          TILOP:  ROL     R1                      ;ROTATE (MULT BY 2)
004047  117403                                  DEC     R3                      ;DECREMENT COUNTER
004050  054046                                  BNE     TILOP                   ;KEEP GOING TILL DONE
004051  104203  000012                          MOV     #10.,R3                 ;SHIFT COUNT FOR MILSEC CONVERSION
004053  110201                          TILOP1: ROL     R1                      ;GET NUMBER IN MILSECS
004054  117403                                  DEC     R3                      ;GO TILL DONE
004055  054053                                  BNE     TILOP1                  ;AGAIN
004056  104010  001251                          MOV     R1,STO                  ;STORE IT
                                        ;
                                        ;
                                        ;       GET SERIAL NUMER AND DATE FROM DMBUF
                                        ;
                                        ;
004060  104205  000004                          MOV     #4,R5                   ;COUNT OF WORDS IN SERIAL NUMBRE
004062  104207  001263                          MOV     #DMBUF,R0               ;POINT TO BUFFER
004064  105207  000005                          ADD     #SER,R0                 ;POINT TO SERIAL NUMBER
004066  104203  001305                          MOV     #SERNUM,R3              ;POINT TO BUFFER
004070  104272                          1$:     MOV     (R0)+,R2                ;GET WORD
004071  100232                                  MOV     R2,(R3)+                ;STORE WORD
004072  117405                                  DEC     R5                      ;DECREMENT COUNT
004073  054070                                  BNE     1$                      ;CONTINURE TILL DONE
004074  104205  000004                          MOV     #4,R5                   ;RESET COUNT FOR DATE
004076  104207  001263                          MOV     #DMBUF,R0               ;POINT TO DMBUF
004100  105207  000011                          ADD     #DAT,R0                 ;POINT TO DATE
004102  104203  001301                          MOV     #DATE,R3                ;POINT TO DATE BUFFER
004104  104272                          2$:     MOV     (R0)+,R2                ;GET WORD
004105  100232                                  MOV     R2,(R3)+                ;STORE WORD
004106  117405                                  DEC     R5                      ;DECREMENT COUTNER
004107  054104                                  BNE     2$                      ;CONTINUE TILL DONE
                                        ;
                                        ;
                                        ;       SET UP HEADER AND DATA PREAMBLE LENGTHS
                                        ;
                                        ;

004110  104207  001052                          MOV     #SCR,R0                 ;POINT TO CHARACTERISTICS BLK
004112  104670  000005  001334                  MOV     DATA(R0),DPREA          ;DATA PREAMBLE LENGTH
004115  103200  177400  001334                  BIC     #HIBYTE,DPREA           ;CLEAR OUT HIGH BYTE GARBAGE
004120  104673  000005                          MOV     HEAD(R0),R3             ;HEADER PREAMBLE LENGTH
004122  110703                                  SWAB    R3                      ;GET INTO LOW BYTE
004123  103203  177400                          BIC     #HIBYTE,R3              ;CLEAR OUT HIGH BYTE GARBAGE
004125  104030  001333                          MOV     R3,HPREA                ;STORE IT
004127  114001                                  CLR     R1                      ;FOR NO ERROR
```

```
                                ;
                                ;       SET UP STARTING ADDRESS BITS
                                ;
004130  104203  001052          MOV     #SCR,R3                 ;POINT TO SUBUNIT CHAR BLOCK
                                ;
                                ;       LBN
                                ;
004132  104637  000002          MOV     STLBN(R3),R0            ;GET THE WORD
004134  103207  170377          BIC     #STCLR,R0               ;CLEAR THE REST
004136  104070  001335          MOV     R0,ST.LBN               ;STORE IT
                                ;
                                ;       RBN
                                ;
004140  104637  000003          MOV     STRBN(R3),R0            ;GET RBN WORD
004142  103207  170377          BIC     #STCLR,R0               ;CLEAR THE REST
004144  104070  001336          MOV     R0,ST.RBN               ;STORE IT
                                ;
                                ;       XBN
                                ;
004146  104637  000002          MOV     STXBN(R3),R0            ;GET THE WORD
004150  110607                  ROR     R0
004151  110607                  ROR     R0
004152  110607                  ROR     R0
004153  110607                  ROR     R0                      ;GET INTO THE RIGHT NIBBLE
004154  103207  170377          BIC     #STCLR,R0               ;CLEAR THE REST
004156  104070  001337          MOV     R0,ST.XBN               ;STORE IT
                                ;
                                ;       DBN
                                ;
004160  104637  000003          MOV     STDBN(R3),R0            ;GET THE WORD
004162  110607                  ROR     R0
004163  110607                  ROR     R0
004164  110607                  ROR     R0
004165  110607                  ROR     R0                      ;GET INTO THE RIGHT NIBBLE
004166  103207  170377          BIC     #STCLR,R0               ;CLEAR THE REST
004170  104070  001340          MOV     R0,ST.DBN               ;STORE IT
```

```
                                ;
                                ;       CLEAR ECC THRESHOLD
                                ;
004172  104203  001037          MOV     #CR,R3                  ;POINT TO CHARACTERISTICS
004174  104632  000002          MOV     ERRSYM(R3),R2           ;GET THE WORD
004176  103202  000377          BIC     #LOBYTE,R2              ;CLEAR THE THRESHOLD
004200  100632  000002          MOV     R2,ERRSYM(R3)           ;STORE IT BACK
004202  000000                  RETURN
```

```
                                        ;
                                        ;
                                        ;       FIND SDI INTERCONNECT FOR DESIRED UNIT
                                        ;
                                        ;
004203  104205  001263          GETUNT: MOV     #DMBUF,R5               ;POINT TO STARTUP INFO
004205  114000  001501                  CLR     COUNT                   ;FOR INTERCONNECT INIT
004207  104400  000001  000740          MOV     #1,UNIT                 ;INIT INTERCONNECT
004212  104203  000741          GOVER:  MOV     #CR.GST,R3              ;FOR GET STATUS
004214  104237                          MOV     (R3)+,R0                ;GETCOMMAND ADDRESS
004215  104231                          MOV     (R3)+,R1                ;GET COMMAND SIZE
004216  104302  000740                  MOV     UNIT,R2                 ;GET INTERCONNECT
004220  060004                          XFC     SEND                    ;ISSUE GET STATUS COMMAND
004221  115001                          TST     R1                      ;SUCCESSFUL ?
004222  014232                          BEQ     GSKIP1                  ;YUP - SKIP RETRY
004223  115400  000717                  INC     UN.ERI                  ;INC COUNT
004225  106300  001502  000717          CMP     RETRY,UN.ERI            ;DONE ALL RETIES ?
004230  074352                          BMI     NOTHER                  ;YUP
004231  004212                          BR      GOVER
004232  114000  000717          GSKIP1: CLR     UN.ERI                  ;FOR ERROR CLEAR
004234  104231                          MOV     (R3)+,R1                ;GET RECEIVE BUFFRE
004235  104137                          MOV     (R3),R0                 ;GET BUFFER LENGTH
004236  060005                          XFC     RCV                     ;RECEIVE SDI RESPONSE
004237  115001                          TST     R1                      ;SUCCESSFUL ?
004240  014250                          BEQ     GSKIP2                  ;YUP - SKIP RETRY
004241  115400  001501                  INC     COUNT                   ;INC ERROR COUNT
004243  106300  001502  001501          CMP     RETRY,COUNT             ;DONE ALL RETRIES ?
004246  074352                          BMI     NOTHER                  ;YUP
004247  004212                          BR      GOVER                   ;TRY AGAIN
004250  114000  001501          GSKIP2: CLR     COUNT                   ;FOR COUNT CLEAR
004252  104204  001030                  MOV     #ST,R4                  ;POINT TO STATUS BUFFER
004254  104641  000000                  MOV     UID(R4),R1              ;GET UNIT NUMBER FROM STATUS
004256  103201  170000                  BIC     #HD.CLR,R1              ;CLEAR GARBAGE
004260  106651  000003                  CMP     DMUNIT(R5),R1           ;IS IT THE ONE ?
004262  074352                          BMI     NOTHER                  ;NOT EVEN A POSSIBLE SUBUNIT
004263  054267                          BNE     GSKIP3                  ;NO - SEE IF A SUBUNIT
004264  104202  000001                  MOV     #1,R2                   ;SET SUBUNIT MASK
004266  004337                          BR      GDONE                   ;AND EXIT
004267  104643  000000          GSKIP3: MOV     MASK(R4),R3             ;GET SUBUNIT MASK
004271  103203  007777                  BIC     #LO,R3                  ;CLEAR GARBAGE
004273  110703                          SWAB    R3                      ;GET INTO LOW BYTE
004274  110603                          ROR     R3                      ;TO GET IN LOW NIBBLE
004275  110603                          ROR     R3                      ;TO GET IN LOW NIBBLE
004276  110603                          ROR     R3                      ;TO GET IN LOW NIBBLE
004277  110603                          ROR     R3                      ;TO GET IN LOW NIBBLE
004300  114007                          CLR     R0                      ;CLEAR COUNTER
004301  102203  000002                  BIT     #BIT1,R3                ;ANYTHING THERE ?
004303  014315                          BEQ     GSKIP                   ;NOPE - ONLY 1
004304  115407                          INC     R0                      ;YUP - INCREMENT COUNTER
004305  102203  000004                  BIT     #BIT2,R3                ;ANYTHING HERE ?
004307  014315                          BEQ     GSKIP                   ;NOPE - ONLY 2
004310  115407                          INC     R0                      ;INC OUNTER
004311  102203  000010                  BIT     #BIT3,R3                ;ANYTHING HERE ?
004313  014315                          BEQ     GSKIP                   ;NOPE
004314  115407                          INC     R0                      ;YUP
004315  105071          GSKIP:  ADD     R0,R1                   ;ADD TO UNIT NUMBER
004316  107651  000003                  SUB     DMUNIT(R5),R1           ;GET RELATIVE OFFSET FROM GIVEN UNIT NUMBER
```

```
                                        BMI     NOTHER                  ;IF NEGATIVE THEN NOT IN THE RANGE
                                        MOV     DMUNIT(R5),R3           ;GET DESIRED
                                        MOV     UID(R4),R1              ;GET ORIGINAL
                                        BIC     #H..CLR,R1              ;CLEAR SUBUNIT MASK
                                        SUB     R1,R3                   ;FIGURE OUT WHICH SUBUNIT
                                        MOV     #1,R2                   ;INIT SUBUNIT MASK
                                        ADD     #0,R2                   ;TO CLEAR CARRY
                                SFTRPT: ROL     R2                      ;SHIFT LEFT
                                        DEC     R3                      ;DECREMTN COUNTER
                                        BNE     SFTRPT                  ;REPEAT SHIFT
                                GDONE:  ADD     #0,R2                   ;CLEAR CARRY
                                        ROL     R2                      ;HAVE TO ROTATE
                                        ROL     R2                      ;4 TIMES TO DO A
                                        ROL     R2                      ;SWAP NIBBLE
                                        ROL     R2                      ;FOR SUBUNIT MASK
                                        MOV     R2,GSR+1                ;STORE MASK IN SUBUNIT CHAR COMMAND
                                        BIS     R2,ACC+1                ;SET IN FOR CHANGE MODE COMMAND
                                        RETURN                          ;RETURN
                                NOTHER: MOV     UNIT,R2                 ;GET CURRENT INTERCONNECT
                                        ADD     R2,R2                   ;NEXT PORT
                                        MOV     R2,UNIT                 ;SAVE BACK
                                        CLR     UN.ERI                  ;FOR RESTORE
                                        CLR     COUNT                   ;CLEAR ERROR COUNT
                                        CMP     #8.,R2                  ;ALL DONE ?
                                        BPL     GOVER                   ;NOPE - TRY THIS INTERCONNECT
                                        MOV     2(R5),R2                ;UNIT SEARCHING FOR
                                        MOV     #7,R1                   ;SIGNAL NON-EXISTANT UNIT
                                        CALL    ERRMNT                  ;ERROR RETURN
```

```
                                        ;
                                        ;       SET UP FLAGS
                                        ;
004373  104203  001263        FLGSET: MOV     #DMBUF,R3               ;POINT TO BUFFER
004375  104634  000004                MOV     DMFLG(R3),R4            ;GET FLAG WORD
004377  102204  000002                BIT     #BIT1,R4                ;STOP IF BAD FLAG ?
004401  054405                       BNE     FLGSKP                  ;YES - SKIP SET
004402  101200  000020  001217       BIS     #GOBAD,FLAG             ;SET IT
004405  102204  000001        FLGSKP: BIT     #BIT0,R4                ;USE RESIDENT FCT ?
004407  054417                       BNE     RFCT                    ;YUP - SET IT UP
004410  102204  000004                BIT     #BIT2,R4                ;DOWN-LINE LOAD FCT ?
004412  054423                       BNE     DFCT                    ;YUP - SET IT UP
004413  101200  002000  001217       BIS     #BSTGS,FLAG             ;DO BEST GUESS
004416  004427                       BR      FLGDON                  ;ELS USE BEST GUESS
004417  101200  000001  001217 RFCT: BIS     #FCTAVL,FLAG            ;SET FLAG
004422  004427                       BR      FLGDON                  ;EXOT
004423  101200  000400  001217 DFCT: BIS     #DLL,FLAG               ;SET FOR DLL
004426  004427                       BR      FLGDON
004427  000000                FLGDON: RETURN                          ;DONE
                                        ;
                                        ;
                                        ;       INITIALIZATION ROUTINE
                                        ;
004430  104302  000740        INITL:  MOV     UNIT,R2                         ;SELECT UNIT
004432  022340                        CALL    INITIT                          ;INIT DRIVE
004433  024203                        CALL    GETUNT                          ;GET THE SDI INTERCONNECT
004434  024373                        CALL    FLGSET                          ;SET UP FLAGS
004435  022332                        CALL    ONLIN                           ;BRING IT ONLINE
004436  021662                        CALL    GSTATS                          ;GET STATUS
004437  115001                        TST     R1                              ;ANY ERRORS ?
004440  054443                        BNE     OOPS                            ;YUP - QUIT
004441  022154                        CALL    RECAL                           ;RECALIBRATE
004442  023067                        CALL    CONINT                          ;COMPUTE DISK CONSTANTS
004443  000000                OOPS:   RETURN                                  ,AND STOP
```

```
                                        .SBTTL  DBN/XBN FORMAT OVERLAY (F1)
                                ;
                                ;
                                ;       FORMAT DBN AND XBN AREAS
                                ;
                                ;
                                ;
004444                                  DMOVLY  F1,START
                                
003047  104200  000000  001153          MOV     #F1,CUROVL              ;OVERLAY #1
003052  101200  000010  001217          BIS     #DBN,FLAG               ;SET DBN FORMAT
003055  023114                          CALL    DXFORM                  ;FORMAT DBN AREA
003056  104303  001311                  MOV     FCTREV,R3               ;STARTING FCT ENTRY COUNT
003060  107303  001237                  SUB     FCNT,R3                 ;TOTAL BAD IN DBN AREA
003062  105303  001453                  ADD     ERRCNT,R3               ;GET FINAL TOTAL
003064  104030  001314                  MOV     R3,DBBAD                ;STORE IT FOR STATS
003066  114000  001453                  CLR     ERRCNT                  ;FOR CLEAR
003070  104300  001237  001311          MOV     FCNT,FCTREV             ;FOR BAD BLOCK COUNT
003073  103200  000010  001217          BIG     #DBN,FLAG               ;DO XBN AREA
003076  023114                          CALL    DXFORM                  ;FORMAT XBN AREA
003077  104303  001311                  MOV     FCTREV,R3               ;STARTING FCT ENTRY COUNT
003101  107303  001237                  SUB     FCNT,R3                 ;TOTAL BAD IN XBN AREA
003103  105303  001453                  ADD     ERRCNT,R3               ;GET FINAL TOTAL
003105  104030  001315                  MOV     R3,XBBAD                ;STORE IT FOR STATS
003107  114000  001453                  CLR     ERRCNT                  ;FOR CLEAR
003111  104201  000006                  MOV     #F3,R1                  ;FCT DLL OVERLAY
003113  022412                          CALL    NEXT                    ;BRING IN NEXT OVERLAY
                                
003114  104207  001052          DXFORM: MOV     #SCR,R0                 ;POINT TO CHARACTERISTICS BLOCK
003116  102200  000010  001217          BIT     #DBN,FLAG               ;DO DBN AREA ?
003121  013171                          BEQ     XBNIT                   ;NO - DO XBN AREA
003122  104673  000022                  MOV     DBNCYL(R0),R3           ;GET NUMBER OF CYLS TO FM
003124  110703                          SWAB    R3                      ;GET INTO LOW BYTE
003125  103203  177400                  BIC     #HIBYTE,R3              ;CLEAR HI BYTE
003127  104030  001460                  MOV     R3,CNTCYL               ;SET UP COUNTER
003131  102200  002000  001217          BIT     #BSTGS,FLAG             ;DOING BEST GUESS ???
003134  053164                          BNE     SKIP4                   ;YES - SKIP FCT SET UP
003135  104200  000200  001500          MOV     #128.,PCNT              ;FOR PBN COUNT INIT
003140  114000  001256                  CLR     FCTCNT                  ;FOR INIT FCT READ
003142  023407                          CALL    DXFCPG                  ;READ IT IN
003143  104200  005152  001223          MOV     #PBNBUF,BADPBN          ;ADDR OF BAD PBN LIST
003146  104300  005170  001237          MOV     PBNBUF+C512,FCNT        ;GET COUNT OF USED ENTRIES
003151  104300  005170  001311          MOV     PBNBUF+C512,FCTREV      ;STORE IT FOR STAT COMPUTATION
003154  013161                          BEQ     SKIP19                  ;IF ZERO - THEN NO ENTRIES
003155  115400  001256                  INC     FCTCNT                  ;START WITH SECOND FCT BLOCK
003157  023407                          CALL    DXFCPG                  ;BRING IT IN
003160  003164                          BR      SKIP4                   ;SKIP NO ENTRY STUFF
003161  101200  000002  001217  SKIP19: BIS     #FCTEMT,FLAG            ;SET EMPTY FLAG
003164  104200  140000  001462  SKIP4:  MOV     #HD.DBN,HD.CUR          ;GET DBN HEADER CODE
003167  023423                          CALL    NUMDBN                  ;GET DBN OF FIRST BLOCK ON LAST CYLINDER
003170  003200                          BR      SKIP1                   ;SKIP XBN SETUP
003171  104200  120000  001462  XBNIT:  MOV     #HD.XBN,HD.CUR          ;GET XBN HEADER CODE
003174  104670  000021  001460          MOV     XBNCYL(R0),CNTCYL       ;GET CYLINDERS IN XBN AREA
003177  023453                          CALL    NUMXBN                  ;GET XBN OF FIRST BLOCK ON LAST XBN CYL
003200  104140  001117          SKIP1:  MOV     (R4),HOLDBN             ;LO ORDER FIRST BLOCK NUM TO DO
003202  104240  001113                  MOV     (R4)+,CURBN             ;AND MAKE IT CURRENT NUMBER
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    SLEEK:   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  SLEEK2: 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    SKIP3:   MOV     #G7,R1              ;FORMAT SETUP OVERLAY
 77 003256  022444                     CALL    PAGE                ;DO IT
 78 003257  104304  001334    SKIP7:   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     CMP     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    SKIP6:   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
```

```
                                        ;
                                        ;PAGE IN NEW FCT BLOCK
                                        ;
003407                          DXFCPG: PUSH    R0
003410  104201  000033                  MOV     #G2,R1                  ;DLL OVERLAY
003412  022444                          CALL    PAGE                    ;EXECUTE OVERLAY
003413  104200  000200  001501          MOV     #128.,COUNT             ;FOR INIT
003416  104200  005152  001223          MOV     #PBNBUF,BADPBN          ;FOR POINTER RESET
003421                                  POP     R0                      ;RESTORE R0
003422  000000                          RETURN                          ;RETURN
                                        ;
                                        ;
                                        ;       COMPUTE NUMBER OF FIRST DBN ON LAST DBN TRACK
                                        ;       R0 -> CHARACTERISTICS BLOCK
                                        ;
                                        ;
003423  104673  000022          NUMDBN: MOV     DBNCYL(R0),R3           ;GET NUMBER OF CYLINDERS IN DBN AREA
003425  110703                          SWAB    R3                      ;GET INTO LOW BYTE
003426  103203  177400                  BIC     #HIBYTE,R3              ;CLEAR OUT OTHER INFO
003430  104030  000736                  MOV     R3,TEMP                 ;MOVE TO TEMP AREA
003432  114000  000737                  CLR     TEMP+1                  ;CLEAR FOR STORE
003434  104204  000736                  MOV     #TEMP,R4                ;POINT R4 AT TEMP AREA
003436  104203  001131                  MOV     #SECTCY,R3              ;POINT TO NUM OF SECTORS/CYLINDER
003440  021542                          CALL    DMUL                    ;MULTIPLY TO GET SECTORS BEFORE LAST CYL
003441  104641  000001                  MOV     1(R4),R1                ;GET HIGH ORDER
003443  105301  001340                  ADD     ST.DBN,R1               ;ADD HIGH ORDER STARTING DBN
003445  100641  000001                  MOV     R1,1(R4)                ;STORE BACK
003447  104203  001127                  MOV     #SECTRK,R3              ;WANT FIRST DN OF LAST TRACK
003451  021524                          CALL    DSUB                    ;SUB TO GET IT
003452  000000                          RETURN
                                        ;
                                        ;
                                        ;       COMPUTE NUMBER OF FIRST XBN ON LAST XBN TRACK
                                        ;       R0 -> CHARACTERISTICS BLOCK
                                        ;
                                        ;
003453  104670  000021  000736  NUMXBN: MOV     XBNCYL(R0),TEMP         ;GET NUMBER OF CYLINDERS IN XBN AREA
003456  114000  000737                  CLR     TEMP+1                  ;CLEAR FOR STORE
003460  104204  000736                  MOV     #TEMP,R4                ;POINT TO TEMP AREA
003462  104203  001131                  MOV     #SECTCY,R3              ;POINT TO SECTORS/CYLINDER
003464  021542                          CALL    DMUL                    ;MULTIPLY TO GET SECTORS IN XBN AREA
003465  104641  000001                  MOV     1(R4),R1                ;GET HIGH ORDER
003467  105301  001337                  ADD     ST.XBN,R1               ;ADD HIGH ORDER STARTING XBN
003471  100641  000001                  MOV     R1,1(R4)                ;STORE BACK
003473  104203  001127                  MOV     #SECTRK,R3              ;WANT XBN OF LAST TRACK
003475  021524                          CALL    DSUB                    ;SUB TO GET IT
003476  000000                          RETURN
                                        ;
                                        ;
                                        ;       CHECK PASS
                                        ;
                                        ;
003477                          DXCHEC: PUSH    R0                      ;SAVE PTR TO CHARACTERISTICS BLK
003500  114000  001226          DXCH:   CLR     ERR                     ;FOR ERROR COUNT RESET
003502  114000  001221                  CLR     ERFLAG                  ;CLEAR RE-FORMAT FLAG
003504  102200  000600  001217          BIT     #MANU+DLL,FLAG          ;FCT AVAILABLE ?
003507  013522                          BEQ     CSKIP                   ;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 RODER 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 COMMAD 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     CMPDAT                  ;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     RTYCNT                  ;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     SECCNT                  ;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 COMD 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    CEDC                    ;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.ERI        ;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                            BR      DXCH                ;RE-CYCLE CHECK PASS
187 004044                        CDONE:    POP     R0                  ;RESTORE CHARACTERISTICS PTR
188 004045  000000                            RETURN
```

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

```
                                        .SBTTL  DBN/XBN TRACK FORMAT OVERLAY (G7)
                                ;
                                ;
                                ;
                                ;       SET UP TRACK FORMAT
                                ;
                                ;
                                ;
                                ;
004151                                  DMOVLY  G7,START
003047  104200  000047  001153          MOV     #G7,CUROVL              ;GET OVERLAY INDICATOR
003052  104300  001127  001454  DXTRK:  MOV     SECTRK,SECCNT           ;MOVE SECTOR COUNT INTO R3
003055  104302  001223                  MOV     BADPBN,R2               ;MOVE PTR TO BAD LIST INTO R2
003057  104205  007275                  MOV     #IMAGE,R5               ;POINT TO FORMAT TRACK IMAGE
003061  102200  000600  001217          BIT     #MANU+DLL,FLAG          ;SEE IF  FCT AVAILABLE
003064  013074                          BEQ     TKIP2                   ;NO - SKIP PBN COMPUTATION
003065  102200  000010  001217          BIT     #DBN,FLAG               ;DO DBN AREA ??
003070  013073                          BEQ     TKIP1                   ;NO - DO XBN AREA
003071  023307                          CALL    DPBN                    ;COMPUTE PBN FOR STARTING DBN
003072  003074                          BR      TKIP2                   ;SKIP XBN COMPUTATION
003073  023341                  TKIP1:  CALL    XPBN                    ;COMPUTE PBN FOR STARTING XBN
003074  104203  005567          TKIP2.  MOV     #GDBLK,R3               ;POINT R3 AT GOOD DATA BLOCK
003076  100253                          MOV     R3,(R5)+                ;AND STORE PTR IN IMAGE BLOCK
003077  104303  001113                  MOV     CURBN,R3                ;GET LOW ORDER BLOCK NUMBER
003101  100253                          MOV     R3,(R5)+                ;AND STORE IN IMAGE BLOCK
003102  102200  002002  001217          BIT     #BSTGS+FCTEMT,FLAG      ;IS FCT AVAILABLE AND NON-EMPTY ?
003105  053146                          BNE     TKIP3                   ;NO - CONSIDER GOOD
003106  104302  001223                  MOV     BADPBN,R2               ;GET POINTER TO PBN
003110  104121                          MOV     (R2),R1                 ;GET LOW ORDER PBN
003111  106010  001110                  CMP     R1,CURPBN               ;ARE THEY EQUAL ?
003113  053146                          BNE     TKIP3                   ;NO - SKIP REST OF COMPARE
003114  104621  000001                  MOV     1(R2),R1                ;GET HIGH ORDER BAD
003116  103201  170000                  BIC     #HD.CLR,R1              ;CLEAR HEADER FOR COMPARE
003120  106010  001111                  CMP     R1,CURPBN+2             ;EQUAL ?
003122  053146                          BNE     TKIP3                   ;NO - MARK AS GOOD
003123  117400  001501                  DEC     COUNT                   ;DECREMENT IT
003125  117400  001237                  DEC     FCNT                    ;DEC IT
003127  053133                          BNE     TKIP7                   ;IF NOT EMPTY THEN CONTINUE
003130  101200  000002  001217          BIS     #FCTEMT,FLAG            ;SET FCT EMPTY FLAG
003133  104303  001114          TKIP7:  MOV     CURBN+1,R3              ;HI ORDER BLOCK NUM AND HDR CODE
003135  103203  170000                  BIC     #HD.CLR,R3              ;CLEAR THE HEADER CODE
003137  101203  110000                  BIS     #HD.BAD,R3              ;SET TO BAD HEADER CODE
003141  100253                          MOV     R3,(R5)+                ;AND STORE IN IMAGE BLOCK
003142  105200  000002  001223          ADD     #2,BADPBN               ;MOVE PTR TO NEXT BAD BLOCK
003145  003156                          BR      TKIP4                   ;SKIP GOOD MARKING
003146  104303  001114          TKIP3:  MOV     CURBN+1,R3              ;HI ORDER BLOCK NUM AND HDR CODE
003150  103203  170000                  BIC     #HD.CLR,R3              ;CLEAR HEADER CODE
003152  104301  001462                  MOV     HD.CUR,R1               ;GET CURRENT HEADER CODE(XBN OR DBN)
003154  101013                          BIS     R1,R3                   ;SET TO GOOD HEADER CODE
003155  100253                          MOV     R3,(R5)+                ;AND STORE IN IMAGE BLOCK
003156                          TKIP4:  DUBINC  CURBN                   ;INCREMENT IT
003163  102200  000600  001217  TKIP10: BIT     #MANU+DLL,FLAG          ;IS FCT AVAILABLE ?
003166  013202                          BEQ     SKIP5                   ;NO - SKIP PBN INCREMENT
003167                                  DUBINC  CURPBN                  ;INCREMENT IT
003174  104303  001501                  MOV     COUNT,R3                ;GET PBN COUNT
003176  053202                          BNE     SKIP5                   ;IF NOT DONE SKIP
003177  115400  001256                  INC     FCTCNT                  ;GET NEXT BLOCK NUMBER
```

```
    58 003201  023370                          CALL    DXFCP1          ;ELSE PAGE IN NEW FCT BLOCK
    59 003202  117400  001454          SKIP5:  DEC     SECCNT          ;DECREMENT IT
    60 003204  013206                          BEQ     DONE1           ;YUP - RETURN
    61 003205  003074                          BR      TKIP2           ;NO - DO NEXT SECTOR
    62 003206                          DONE1:
    63 003206  107205  000003                   SUB     #IMLEN,R5       ;POINT TO FLAG OF LAST ENTRY
    64 003210  104153                          MOV     (R5),R3         ;GET FLAG WORD
    65 003211  101203  040000                   BIS     #RECIR,R3       ;SET RECIRCULATION FLAG
    66 003213  100153                          MOV     R3,(R5)         ;STORE IT BACK
    67 003214  115000  001463                   TST     CURGRP          ;IS GROUP ZERO ???
    68 003216  013230                          BEQ     TKIP11          ;YES - NO OFFSET
    69 003217  104203  001052                   MOV     #SCR,R3         ;POINT TO CHARACTERISTICS
    70 003221  104632  000011                   MOV     OFFS(R3),R2     ;GET GROUP OFFSET
    71 003223  110702                          SWAB    R2              ;GET INTO LOWBYTE
    72 003224  103202  177400                   BIC     #HIBYTE,R2      ;CLEAR HIGH GARBAGE
    73 003226  115002                          TST     R2              ;ANY OFFSET ?
    74 003227  053236                          BNE     TKIP5           ;YUP - HANDLE IT
    75 003230  104200  007275  001332  TKIP11: MOV     #IMAGE,IMSTAR   ;ELSE START AT BEGINNING
    76 003233  104304  001332                   MOV     IMSTAR,R4       ;GET START
    77 003235  003302                          BR      TKIP6           ;SKIP OFFSET COMPUTATION
    78 003236  104020  000736          TKIP5:  MOV     R2,TEMP         ;STORE IT
    79 003240  114000  000737                   CLR     TEMP+1          ;FOR STORE
    80 003242  104300  001463  000731'         MOV     CURGRP,DDUMMY   ;GET CURRENT GROUP
    81 003245  114000  000732                   CLR     DDUMMY+1        ;CLEAR HIGH WORD
    82 003247  104203  000736                   MOV     #TEMP,R3        ;FOR MUL
    83 003251  104204  000731                   MOV     #DDUMMY,R4      ;DITTO
    84 003253  021542                          CALL    DMUL            ;MULTIPLY TO GET OFFSET FOR THIS GROUP
    85 003254  106300  001127  000731  TKIP8:  CMP     SECTRK,DDUMMY   ;IS TOTAL OFFSET MORE THAN NUMBER OF SECTORS ?
    86 003257  033264                          BPL     TKIP9           ;NO - ALL IS FINE
    87 003260  107300  001127  000731          SUB     SECTRK,DDUMMY   ;YES - SUBTRACT TILL IT IS
    88 003263  003254                          BR      TKIP8           ;CHECK AGAIN
    89 003264  104200  000003  000736  TKIP9:  MOV     #IMLEN,TEMP     ;GET LENGTH OF IMAGE BLOCK
    90 003267  114000  000737                   CLR     TEMP+1          ;FOR STORE
    91 003271  104203  000736                   MOV     #TEMP,R3        ;FOR MULT
    92 003273  021542                          CALL    DMUL            ;GET LENGTH TO OFFSET
    93 003274  104143                          MOV     (R4),R3         ;GET RESULT
    94 003275  104304  001231                   MOV     EIMAGE,R4       ;GET ADDRESS OF END OF IMAGE
    95 003277  107034                          SUB     R3,R4           ;SUBTRACT TO GET STARTING LOCATION
    96 003300  104040  001332                   MOV     R4,IMSTAR       ;STORE IT
    97 003302  104143                  TKIP6:  MOV     (R4),R3         ;GET BUFF POINTER
    98 003303  101203  100000                   BIS     #LAST,R3        ;SIGNAL AS LAST
    99 003305  100143                          MOV     R3,(R4)         ;STORE IT BACK
   100 003306  000000                          RETURN
```

```
                                        ;
                                        ;
                                        ;
                                        ;       COMPUTE PBN FROM GIVEN DBN
                                        ;       R0 -> CHARACTERISTICS BLOCK
                                        ;
                                        ;
003307  104300  001137  000736  DPBN:   MOV     XBNSEC,TEMP             ;GET NUMBER OF SECTORS IN XBN AREA
003312  104300  001140  000737          MOV     XBNSEC+1,TEMP+1         ;GET HI ORDER
003315  104204  000736                  MOV     #TEMP,R4                ;POINT R4 AT TEMP AREA
003317  104203  001133                  MOV     #LBNLBN,R3              ;POINT AT NUM OF LBN'S IN LBN AREA
003321  021506                          CALL    DADD                    ;ADD
003322  104203  001135                  MOV     #RBNLBN,R3              ;POINT TO NUM OF RBN'S IN LBN AREA
003324  021506                          CALL    DADD                    ;ADD TO GET SECTORS IN LBN + XBN AREA
003325  104203  001113                  MOV     #CURBN,R3               ;POINT TO CURRENT BLOCK NUMBER(DBN)
003327  021506                          CALL    DADD                    ;GET RELATIVE PBN
003330  104641  000001                  MOV     1(R4),R1                ;GET HIGH ORDER
003332  107301  001340                  SUB     ST.DBN,R1               ;SUBTRACT HIGH ORDER STARTING DBN
003334  104140  001110                  MOV     (R4),CURPBN             ;GET LO ORDER PBN
003336  104010  001111                  MOV     R1,CURPBN+1             ;STORE HIGH ORDER
003340  000000                          RETURN
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;       COMPUTE PBN FROM GIVEN XBN
                                        ;       R0 -> CHARACTERISTICS BLOCK
                                        ;
                                        ;
                                        ;
003341  104300  001133  000736  XPBN:   MOV     LBNLBN,TEMP             ;GET NUMBER OF LBN'S IN LBN AREA
003344  104300  001134  000737          MOV     LBNLBN+1,TEMP+1         ;GET HIGH ORDER
003347  104204  000736                  MOV     #TEMP,R4                ;POINT R4 TO TEMP AREA
003351  104203  001135                  MOV     #RBNLBN,R3              ;POINT R3 AT RBN'S IN LBN AREA
003353  021506                          CALL    DADD                    ;ADD TO GET TOTAL SECTORS IN LBN AREA
003354  104203  001113                  MOV     #CURBN,R3               ;POINT R3 AT CURRENT BLOCK NUMBER
003356  021506                          CALL    DADD                    ;ADD TO GET RELATIVE PBN
003357  104641  000001                  MOV     1(R4),R1                ;GET HIGH ORDER
003361  107301  001337                  SUB     ST.XBN,R1               ;SUBTRACT HIGH ORDER STARTING XBN
003363  104140  001110                  MOV     (R4),CURPBN             ;GET LO ORDER OF PBN
003365  104010  001111                  MOV     R1,CURPBN+1             ;SAVE HIGH ORDER
003367  000000                          RETURN
```

```
                                        ;
                                        ;PAGE IN NEW FCT BLOCK
                                        ;
003370                          DXFCP1: PUSH    R0
003371  104201  000033                  MOV     #G2,R1                  ;DLL OVERLAY
003373  022444                          CALL    PAGE                    ;EXECUTE OVERLAY
003374  104200  000200  001501          MOV     #128.,COUNT             ;FOR INIT
003377  104200  005152  001223          MOV     #PBNBUF,BADPBN          ;FOR POINTER RESET
003402                                  POP     R0                      ;RESTORE R0
003403  000000                          RETURN                          ;RETURN
```

```
                                        .SBTTL  LBN FORMATTING OVERLAY (F2)
                                ;
                                ;       LBN FORMATTING
                                ;
003404                                  DMOVLY  F2,START

003047  104200  000003  001153  LFORM:  MOV     #F2,CUROVL              ;OVERLAY #2
003052  102200  000001  001217          BIT     #FCTAVL,FLAG            ;FCT AVAILAABLE ?
003055  013061                          BEQ     XSKIP1                  ;NO - SKIP SET UP
003056  104201  000036                  MOV     #G3,R1                  ;OVERLAY TO GET RIGHT FCT BLOCK
003060  022444                          CALL    PAGE                    ;EXECUTE IT
003061  104207  001052          XSKIP1: MOV     #SCR,R0                 ;POINT TO CHARACTERISTICS BLOCK
003063  104300  001143  001125          MOV     LBNCYL,CYLNUM           ;GET LO ORDER CYLINDER COUNT
003066  104300  001143  001460          MOV     LBNCYL,CNTCYL           ;MAKE LO ORDER COUNT
003071  104300  001144  001126          MOV     LBNCYL+1,CYLNUM+1       ;GET HIGH ORDER
003074  104300  001144  001461          MOV     LBNCYL+1,CNTCYL+1       ;STORE IT
003077  103200  170000  001461          BIC     #HD.CLR,CNTCYL+1                ;CLEAR STARTING CYLINDER BITS
003102  104204  001125                  MOV     #CYLNUM,R4              ;SUBTRACT TO GET CYLINDER NUMBER
003104  104203  001466                  MOV     #ONE,R3                 ;1 - BECAUSE START AT 0
003106  021524                          CALL    DSUB                    ;DO SUBTRACT
003107  104300  001475  001235          MOV     TOTRCT,RCTTOT           ;GET TOTAL RCT LBN'S
003112  104201  000052                  MOV     #G8,R1                  ;POINT TO OVERLAY
003114  022444                          CALL    PAGE                    ;COMPUTE VARIOUS CONSTANTS
003115  104207  001052          XSLEEK: MOV     #SCR,R0                 ;POINT TO CHARACTERISTICS
003117  104673  000002                  MOV     GRPCYL(R0),R3           ;GET GROUPS/CYLINDER
003121  103203  177400                  BIC     #HIBYTE,R3              ;CLEAR OUT GARBAGE
003123  104030  001464                  MOV     R3,GRPCNT               ;USE AS COUNTER
003125  104030  001463                  MOV     R3,CURGRP               ;GROUP NUMBER
003127  117400  001463                  DEC     CURGRP                  ;DECREMENT TO GET ACTUAL NUMBER
003131  104300  001 25  001076  XSLEK2: MOV     CYLNUM,ISEEK+1          ;GET CURRENT CYLINDER NUMBER
003134  104300  001126  001077          MOV     CYLNUM+1,ISEEK+2        ;GET HIGH ORDER
003137  104300  001463  001100          MOV     CURGRP,ISEEK+3          ;LOAD GROUP NUMBER
003142  02?251                          CALL    SEEK                    ;DO THE SEEK
003143  115001                          TST     R1                      ;ANY ERROR ?
003144  073435                          BMI     SEEKER                  ;YUP - CUT OUT
003145  104207  001052                  MOV     #SCR,R0                 ;POINT TO CHARACTERISTICS
003147  104673  000003                  MOV     TRKGRP(R0),R3           ;GET TRACKS/GROUP
003151  103203  177400                  BIC     #HIBYTE,R3              ;CLEAR OUT GARBAGE
003153  104030  001465                  MOV     R3,TRKCNT               ;MAKE COUNTER
003155  117403                          DEC     R3                      ;WANT LAST TRACK NUMBER
003156  104030  001112                  MOV     R3,CURTRK               ;MAKE CURRENT TRACK 0
003160  104201  000025          XSKIP3: MOV     #F8,R1                  ;TRACK SET UP OVERLAY
003162  022444                          CALL    PAGE                    ;SET UP TRACK FORMAT
003163  104304  001334          XSKIP2: MOV     DPREA,R4                ;GET DATA PREAMBLE LENGTH
003165  104303  001333                  MOV     HPREA,R3                ;GET HEADER PREAMBLE LENGTH
003167  104307  001332                  MOV     IMSTAR,R0               ;POINT TO TRACK IMAGE START POINT
003171  104301  001112                  MOV     CURTRK,R1               ;TRACK TO FORMAT
003173  104302  000740                  MOV     UNIT,R2                 ;SDI INTERCONNECT
003175  104205  007275                  MOV     #IMAGE,R5               ;RECIRCULATION ADDRESS
003177  060001                          XFC     FORMAT                  ;DO FORMAT
003200  115001                          TST     R1                      ;ANY ERROR ?
003201  013214                          BEQ     LSKIP4                  ;NO - DO CHECK PASS
003202  115400  000717                  INC     UN.ERI                  ;INCREMENT IT
003204  106300  001502  000717          CMP     RETRY,UN.ERI            ;DONE ALL RETRIES ?
003207  073431                          BMI     FORERR                  ;YUP - ERROR
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  000013                  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     #HI1BYTE,R3     ;CLERA 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     TRKCNT          ;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
```

```
003361  104303  001125                  MOV     CYLNUM,R3       ;GET LOW ORDER CYLINDER NUMBER
003363  100673  000003                  MOV     R3,3(R0)        ;PUT IN MESSAGE
003365  104303  001126                  MOV     CYLNUM+1,R3     ;GET HIGH ORDER
003367  100673  000002                  MOV     R3,2(R0)        ;STORE IT
003371  022536                          CALL    SNDMNT          ;LET HOST KNOW WE'RE ALIVE
003372  104204  001125          LSKIP5: MOV     #CYLNUM,R4      ;GET CURRENT CYLINDER NUMBER
003374  104203  001466                  MOV     #ONE,R3         ;FOR DECREMENT
003376  021524                          CALL    DSUB            ;DECREMENT FOR NEW CYLINDER NUM
003377  104304  001460                  MOV     CNTCYL,R4       ;LOW ORDER ZERO ?
003401  053115                          BNE     XSLEEK          ;NO - CONTINUE
003402  104304  001461                  MOV     CNTCYL+1,R4     ;HIGH ORDER ZERO ?
003404  053115                          BNE     XSLEEK          ;NO - CONTINUE
003405  104303  001254                  MOV     REVCNT,R3       ;ANY LEFTOVER REVECTORS ?
003407  013416                          BEQ     XDONE           ;NOPE
003410  101200  040000  001217          BIS     #FINI,FLAG      ;SIGNAL NOT TO SEEK
003413  104201  000011                  MOV     #F4,R1          ;SIGNAL RCT UPDATE OVERLAY
003415  022444                          CALL    PAGE            ;UPDATE IT
003416  104303  001312          XDONE:  MOV     LBNBAD,R3       ;GET BAD BLOCKS FROM FCT
003420  104304  001453                  MOV     ERRCNT,R4       ;GET CHECK PASS BAD
003422  105043                          ADD     R4,R3           ;ADD TO GET TOTAL
003423  104030  001312                  MOV     R3,LBNBAD       ;STORE BACK
003425  104201  000041                  MOV     #G4,R1          ;RCT CLEANUP
003427  022412                          CALL    NEXT            ;GET NEXT OVERLAY
003430  000000                          RETURN
003431  104012                  FORERR: MOV     R1,R2           ;XFC ERROR CODE
003432  104201  000014                  MOV     #12.,R1         ;SIGNAL LBN FORMAT ERROR
003434  003441                          BR      LFERR
003435  104302  001125          SEEKER: MOV     CYLNUM,R2       ;CYLINDER FAILED ON
003437  104201  000012                  MOV     #10.,R1         ;SEEK ERROR
003441  022552                  LFERR:  CALL    ERRMNT          ;ERROR RETURN
```

```
                                        ;
                                        ;
                                        ;       CHECK PASS
                                        ;
                                        ;
003442  114000  001226          LCHEC:  CLR     ERR             ;FOR ERROR COUNT RESET
003444  114000  001221                  CLR     ERFLAG          ;CLEAR REFORMAT FLAG
003446  102200  002000  001217          BIT     #BSTGS,FLAG     ;BEST GUESS ?
003451  053464                          BNE     LSKIP           ;YES - DO EXTENSIVE READ
003452  104200  000001  001455          MOV     #1,N            ;SET UP FOR STORE
003455  104200  000005  001456          MOV     #5,N1           ;SET UP
003460  104300  001456  001457          MOV     N1,NN1          ;SAVE FOR LATER RESET
003463  003475                          BR      LSKIP2          ;SKIP EXTENSIVE READ SETUP
003464  104200  000003  001455  LSKIP:  MOV     #3,N            ;EXTENSIVE REGULAR READ
003467  104200  000024  001456          MOV     #20.,N1         ;EXTENSIVE ERROR READS
003472  104300  001456  001457          MOV     N1,NN1          ;SAVE FOR LATER RESET
003475  024361                  LSKIP2: CALL    LFIXIT          ;EXECUTE IT
003476  104302  000740          LSKIP1: MOV     UNIT,R2         ;GET SDI INTERCONNECT
003500  060012                          XFC     SIP             ;WAIT FOR PULSE
003501  104300  001127  001454          MOV     SECTRK,SECCNT   ;LOAD SECTORS/TRACK
003504  104205  006621                  MOV     #CMDBUF,R5      ;POINT TO COMMAND BUFFER
003506  104207  000721          LAGAIN: MOV     #RDBLK,R0       ;POINT TO READ COMMAND BLOCK
003510  104653  000000                  MOV     RB.BUF(R5),R3   ;BUFFER PTR ZERO ?
003512  013570                          BEQ     LNOERR          ;SECTOR BAD - SKIP CHECKS
003513  100673  000001                  MOV     R3,RW.BUF(R0)   ;ELSE STORE IN COMMAND BLOCK
003515  104653  000001                  MOV     RB.LOW(R5),R3   ;LOAD LOW ORDER SECTOR NUMBER
003517  100673  000002                  MOV     R3,RW.LOW(R0)   ;STORE IN COMMAND BLOCK
003521  104653  000002                  MOV     RB.HI(R5),R3    ;LOAD HIGH ORDER BLOCK NUMBER
003523  100673  000003                  MOV     R3,RW.HI(R0)    ;STORE IN COMMAND BLOCK
003525  104653  000003                  MOV     RB.CMD(R5),R3   ;GET READ COMMAND AND TRACK NUMBER
003527  100673  000004                  MOV     R3,RW.CMD(R0)   ;STORE IN COMMADN BLOCK
003531  104203  000726                  MOV     #HSLIM-1,R3     ;POINTER TO DUMMY SDI BLOCK
003533  100673  000005                  MOV     R3,RW.DUM(R0)   ;STORE IN READ BLOCK
003535  104207  100721          RFAD3:  MOV     #<BIT15!RDBLK>,R0 ;MAKE SURE POINTING AT BLOCK
003537  104203  100000                  MOV     #RDCMD,R3       ;RESET STATUS POINTER
003541  100673  000000                  MOV     R3,RW.STAT(R0)  ;STORE IT BACK
003543  060002                          XFC     READ            ;READ 1 SECTOR
003544  115001                          TST     R1              ;ANY ERRORS ?
003545  053553                          BNE     LERR            ;YES - UH OH
003546  104207  001443                  MOV     #NUM,R0         ;POINT TO COMPARE BLOCK
003550  060006                          XFC     CMPDAT          ;DO DATA COMPARE
003551  115001                          TST     R1              ;ANY ERROR IN COMPARE ?
003552  013570                          BEQ     LNOERR          ;NOPE - CONTINUE LOOP
003553                          LERR:
003553  104653  000004                  MOV     RB.IM(R5),R3    ;GET IMAGE POINTER
003555  104134                          MOV     (R3),R4         ;GET BUFFER POINTER WORD
003556  102204  020000                  BIT     #BD,R4          ;ALREADY BEEN HERE ??
003560  053570                          BNE     LNOERR          ;YUP - DON'T COUNT AGAIN
003561  101204  020000                  BIS     #BD,R4          ;MARK AS BAD
003563  100134                          MOV     R4,(R3)         ;STORE BACK
003564  115400  001473                  INC     RTYCNT          ;INC IT
003566  115400  001226                  INC     ERR             ;INCREMENT ERROR COUNT
003570  105205  000005          LNOERR: ADD     #RDLEN,R5       ;POINT TO NEXT READ CMD BLOCK
003572  114000  001504                  CLR     TMPTRY          ;FOR RESET
003574  104300  001503  001505          MOV     RECOV,RECTMP    ;GET RECOVERY LEVELS
003577  103200  000002  001220          BIC     #RTYDN,FLAG1    ;CLEAR RETRY DONE FLAG
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  053476                          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                  LHERE:  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 COMD 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          READ4:  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     CMPDAT          ;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.PRV,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    PRIMRB                  ;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              ;POIN  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$:     MOV     FT.HI(R3),R2            ;GET HEADER WORD
154 004046  103202  007777                  BIC     #LO,R2                  ;CLEAR ALL BUT HEADER
155 004050  106202  050000                  CMP     #HD.PRV,R2              ;IS IT THE PRIMARY ?
156 004052  014056                          BEQ     6$                      ;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$:     MOV     FT.HI(R3),R2            ;RESET TO HI ORDER
160 004060  103202  170000                  BIC     #HD.CLR,R2              ;CLEAR HEADER
161 004062  101202  030000                  BIS     #HD.REV,R2              ;MARK AS SECONDARY
162 004064  100632  000002                  MOV     R2,FT.HI(R3)            ;STORE BACK
163 004066  104202  005567                  MOV     #GDBLK,R2               ;POINT TO GOOD BLOCK
164 004070  104635  000000                  MOV     FT.BUF(R3),R5           ;GET BUFFER POINTER AND FLAGS
165 004072  103200  007777  000014          BIC     #BUFMSK,F5              ;CLEAR ONLY BUFFER POINTER
166 004075  101025                          BIS     R2,R5                   ;OR IN NEW BUFFER POINTER
167 004076  100635  000000                  MOV     R5,FT.BUF(R3)           ;MOVE IN BUFFER POINTER AND FLAGS
168 004100  104305  001252                  MOV     ERPNT,R5                ;GET REVECTOR POINTER
169 004102  107205  000002          8$:     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.PRV,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     SNDCNT              ;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          BDIRCT:   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 RCT ?
187 004151  054165                            BNE     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     BDIRCT              ;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     SNDCNT              ;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     SNDCNT              ;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
```

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

```
                                        .SBTTL  LBN FORMAT IMAGE SETUP OVERLAY (F8)
004464                                  DMOVLY  F8,START
                                ;
                                ;       SET UP TRACK FORMAT
                                ;
003047  104200  000025  001153  LTRK:   MOV     #F8.CUROVL              ;GET OVERLAY OFFSET
003052  104207  001052          .       MOV     #SCR,R0                 ;POINT TO CHARACTERISTICS BLOCK
003054  104673  000011                  MOV     LBNTRK(R0),R3           ;GET LBN'S/TRACK
003056  103203  177400                  BIC     #HIBYTE,R3              ;CLEAR HIGH BYTE
003060  104030  001454                  MOV     R3,SECCNT               ;USE AS COUNTER
003062  104302  001223                  MOV     BADPBN,R2               ;POINT TO CURRENT BAD PBN ENTRY
003064  104205  007275                  MOV     #IMAGE,R5               ;POINT TO TRACK IMAGE BUFFER
003066  104300  001123  001110          MOV     HOLDPN,CURPBN           ;GET LOW RODER PBN
003071  104300  001124  001111          MOV     HOLDPN+1,CURPBN+1       ;GET HIGH ORDER
003074  104203  005567                  MOV     #GDBLK,R3               ;POINT TO DATA BLOCK
003076  104304  001451                  MOV     EDC,R4                  ;GET GOOD EDC
003100  100634  000400                  MOV     R4,RW.EDC(R3)           ;STORE IN BUFFER
003102  103200  001000  001217          BIC     #PRIM,FLAG              ;CLEAR PRIMARY FLAG
003105  104203  005567          LKIP2:  MOV     #GDBLK,R3               ;POINT TO GOOD DATA BLOCK
003107  100653  000000                  MOV     R3,FT.BUF(R5)           ;STORE IN IMAGE BLOCK
003111  104303  001113                  MOV     CURBN,R3                ;GET LOW ORDER BLOCK NUMBER
003113  100653  000001                  MOV     R3,FT.LOW(R5)           ;STORE IN IMAGE BLOCK
003115  102200  000001  001217          BIT     #FCTAVL,FLAG            ;IS FCT AVAILABLE ?
003120  013301                          BEQ     LKIP3                   ;NO - ASSUME BLOCK GOOD
003121  102200  000002  001217          BIT     #FCTEMT,FLAG            ;FCT EMPTY ?
003124  053301                          BNE     LKIP3                   ;YUP - BLOCKIS GOOD
003125  104302  001223                  MOV     BADPBN,R2               ;GET FCT POINTER
003127  104121                          MOV     (R2),R1                 ;GET LOW ORDER BAD PBN
003130  106010  001110                  CMP     R1,CURPBN               ;ARE THEY EQUAL ?
003132  053301                          BNE     LKIP3                   ;NO - SKIP REST OF COMPARE
003133  104621  000001                  MOV     1(R2),R1                ;GET HIGH ORDER
003135  103201  170000                  BIC     #HD.CLR,R1              ;CLEAR THE HEADER
003137  106010  001111                  CMP     R1,CURPBN+1             ;ARETHEY EQUAL ?
003141  053301                          BNE     LKIP3                   ;NO - MUST BE GOOD
003142  117400  001500                  DEC     PCNT                    ;DECREMENT IT
003144  117400  001237                  DEC     FCNT                    ;DECREMENT FCT COUNT
003146  053152                          BNE     LKIP12                  ;IF NON - ZERO SKIP FLAG SET
003147  101200  000002  001217          BIS     #FCTEMT,FLAG            ;SET EMPTY FLAG
003152                          LKIP12:
003152  102200  000100  001217          BIT     #REVECT,FLAG            ;IN RCT ?
003155  053161                          BNE     6$                      ;NO - SKIP RCT STUFF
003156  115400  001313                  INC     RCTBAD                  ;INCREMENT IT
003160  003265                          BR      MARBAD                  ;GO MARK BAD
003161  104623  000001          6$:     MOV     1(R2),R3                ;GET BAD PBN HDR
003163  102203  100000                  BIT     #PRMY,R3                ;IS IT SECONDARY ?
003165  013244                          BEQ     LKIP5                   ;YES - GO DO IT
003166  101200  001000  001217          BIS     #PRIM,FLAG              ;SET FLAG FOR PRIMARY FOUND
003171  101200  000004  001220          BIS     #RPRIM,FLAG1            ;SET GOOD EDC NEEDED
003174  104303  001114                  MOV     CURBN+1,R3              ;GET HIGH ORDER HEADER
003176  103203  170000                  BIC     #HD.CLR,R3              ;CLEAR HEADER CODE
003200  101203  050000                  BIS     #HD.PRV,R3              ;MARK AS PRIMARY REVECTOR
003202  100653  000002                  MOV     R3,FT.HI(R5)            ;STORE IN IMAGE BLOCK
003204  104207  001052                  MOV     #SCR,R0                 ;MAKE SURE POINT TO CHAR BLOCK
003206  104204  001113                  MOV     #CURBN,R4               ;POINT TO BLOCK NUMBER
003210  022757                          CALL    PRIMRB                  ;GET PRIMARY RBN NUMBER
003211  104307  001151                  MOV     REVRBN,R0               ;GET NUMBER OF REVECTORED RBN
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  000400                    MOV     R3,RW.EDC(R2)           ;STORE IT
 71 003243  003311                            BR      LKIP4                   ;SKIP SECONDARY REVECTOR
 72 003244                            LKIP5:
 73 003244  115400  001472                    INC     SNDCNT                  ;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            LKIP3:  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            LKIP4:  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  LKIP9:    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            LKIP10: DEC     SECCNT                  ;DECREMENT IT
108 003354  053105                            BNE     LKIP2                   ;NO - DO NEXT LBN
109                                  ;
110                                  ;
111                                  ;       RBN FORMATTING
112                                  ;
113                                  ;
114 003355  104207  001052                    MOV     #SCR,R0                 ;POINT TO CHARACTERISTICS
```

```
003355 104673 000004                  MOV    RBNTRK(R0),R3      ;GET RBN'S/TRACK
003361 103203 177600                  BIC    #HI1BYTE,R3        ;CLEAR HIGH GARBAGE
003363 104030 001454                  MOV    R3,SECCNT          ;USE AS COUNTER
003365 104207 006621         LKIP8.   MOV    #RBNBUF,R0         ;POINT TO RBN BUFFER
003367 104301 001450                  MOV    DWRD,R1            ;DIAGNOSTIC WORD
003371 100271                         MOV    R1,(R0)+           ;STORE IT
003372 104204 000125                  MOV    #85.,R4            ;SET COUNTER
003374 104301 001445                  MOV    FWRD,R1            ;FIRST WORD OF PATTERN
003376 104302 001446                  MOV    SWRD,R2            ;SECOND WORD OF PATTERN
003400 104303 001447                  MOV    TWRD,R3            ;THIRD WORD OF PATTERN
003402 100271                LLOVER:  MOV    R1,(R0)+           ;STORE IT
003403 100272                         MOV    R2,(R0)+           ;STORE IT
003404 100273                         MOV    R3,(R0)+           ;STORE IT
003405 117404                         DEC    R4                 ;DECREMENT COUNTER
003406 053402                         BNE    LLOVER             ;REPEAT TILL DONE
003407 104302 001452                  MOV    BADEDC,R2          ;EDC FOR PATTERN (FORCED ERROR IND)
003411 100272                         MOV    R2,(R0)+           ;STORE IT
003412 104203 006621                  MOV    #RBNBUF,R3         ;POINT TO BUFFER
003414 100653 000000                  MOV    R3,FT.BUF(R5)      ;STORE IN IMAGE
003416 104303 001106                  MOV    CURRBN,R3          ;GET LOW ORDER RBN
003420 100653 000001                  MOV    R3,FT.LOW(R5)      ;STORE IN IMAGE
003422 102200 002002 001217           BIT    #BSTGS+FCTEMT,FLAG ;IS FCT AVAILABLE ?
003425 053471                         BNE    LKIP6              ;NO - CONSIDER GOOD
003426 104203 001110                  MOV    #CURPBN,R3         ;CURRETN PBN
003430 104121                         MOV    (R2),R1            ;GET LOW ORDER BAD
003431 104137                         MOV    (R3),R0            ;GET LOW ORDER CURRENT
003432 106071                         CMP    R0,R1              ;ARE THEY EQUAL ?
003433 053471                         BNE    LKIP6              ;NO - SKIP REST OF COMPARE
003434 104621 000001                  MOV    1(R2),R1           ;GET HIGH ORDER BAD
003436 104637 000001                  MOV    1(R3),R0           ;GET HIGH ORDER CURRENT
003440 106071                         CMP    R0,R1              ;ARETHEY EQUAL ?
003441 053471                         BNE    LKIP6              ;NO - MUST BE GOOD
003442 117400 001500                  DEC    PCNT               ;DECREMENT IT
003444 117400 001237                  DEC    FCNT               ;DECREMENT IT
003446 053452                         BNE    LKIP13             ;IF NON - ZERO SKIP FLAG SET
003447 101200 000002 001217           BIS    #FCTEMT,FLAG       ;SET EMPTY FLAG
003452 104303 001107         LKIP13:  MOV    CURRBN+1,R3        ;GET HIGH ORDFR RBN
003454 103203 170000                  BIC    #HD.CLR,R3         ;CLERA THE HEADER
003456 101203 110000                  BIS    #HD.BAD,R3         ;MARK AS BAD
003460 100653 000002                  MOV    R3,FT.HI(R5)       ;STORE IN IMAGE
003462 105200 000002 001223           ADD    #2,BADPBN          ;POINT TO NEXT BAD
003465 101200 001000 001217           BIS    #PRIM,FLAG         ;MAKE SURE NO PRIMARIES
003470 003514                         BR     LKIP7              ;SKIP GOOD MARKING
003471 104303 001107         LKIP6:   MOV    CURRBN+1,R3        ;GET HIGH ORDER RBN
003473 103203 170000                  BIC    #HD.CLR,R3         ;CLEAR HEADER
003475 101203 060000                  BIS    #HD.RBN,R3         ;MARK AS GOOD RBN
003477 100653 000002                  MOV    R3,FT.HI(R5)       ;STORE IN IMAGE
003501 102200 000004 001220           BIT    #RPRIM,FLAG1       ;NEED GOOD EDC ?
003504 013514                         BEQ    LKIP7              ;NOPE
003505 103200 000004 001220           BIC    #RPRIM,FLAG1       ;CLEAR FLAG
003510 104203 005567                  MOV    #GDBLK,R3          ;GET GOOD EDC BLOCK
003512 100653 000000                  MOV    R3,FT.BUF(R5)      ;STORE IT IN BUFFER POINTER
003514                       LKIP7:   DUBINC CURRBN             ;DO THE INCREMENT
003521 105205 000003                  ADD    #IMLEN,R5          ;POINT TO NEXT IMAGE BLOCK
003523 102200 002002 001217           BIT    #BSTGS+FCTEMT,FLAG ;DOING BEXT GUESS ?
003526 053546                         BNE    LKIP14             ;YUP - SKIP FCT STUFF
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   ;ELSE START AT VERY BEGINNING
197 003576  104304  001332                   MOV     IMSTAR,R4       ;POINT TO START FOR LATER
198 003600  003645                          BR      LKIP16          ;BRANCH AROUND OFFSET FIXUP
199 003601  104020  000736          LKIP15:  MOV     R2,TEMP         ;STORE IT
200 003603  114000  000737                   CLR     TEMP+1          ;FOR STORE
201 003605  104300  001463  000731           MOV     CURGRP,DDUMMY   ;GET CURRENT GROUP
202 003610  114000  000732                   CLR     DDUMMY+1        ;CLEAR HIGH ORDER
203 003612  104203  000736                   MOV     #TEMP,R3        ;FOR MUL
204 003614  104204  000731                   MOV     #DDUMMY,R4      ;DITTO
205 003616  021542                          CALL    DMUL            ;MULTIPLY TO GET OFFSET FOR THIS GROUP
206 003617  106300  001127  000731  LKIP17:  CMP     SECTRK,DDUMMY   ;IS TOTAL OFFSET MORE THAN NUMBER OF SECTORS ?
207 003622  033627                          BPL     LKIP18          ;NO - ALL IS FINE
208 003623  107300  001127  000731           SUB     SECTRK,DDUMMY   ;YES - SUBTRACT TILL IT IS
209 003626  003617                          BR      LKIP17          ;CHECK AGAIN
210 003627  104200  000003  000736  LKIP18:  MOV     #IMLEN,TEMP     ;MUST MULTIPLY BY ENTRY  ENGTH
211 003632  114000  000737                   CLR     TEMP+1          ;FOR CLEAR
212 003634  104203  000736                   MOV     #TEMP,R3        ;FOR MULT
213 003636  021542                          CALL    DMUL            ;DO MULTIPLY
214 003637  104143                          MOV     (R4),R3         ;GET RESULT
215 003640  104304  001231                   MOV     EIMAGE,R4       ;GET ADDRESS OF END OFIMAGE
216 003642  107034                          SUB     R3,R4           ;SUBTRACT TO GET STARTING LOCATION
217 003643  104040  001332                   MOV     R4,IMSTAR       ;STORE IN IMSTAR
218 003645  104143          LKIP16:  MOV     (R4),R3         ;GET BUFF POINTER
219 003646  101203  100000                   BIS     #LAST,R3        ;SIGNAL AS LAST
220 003650  100143                           MOV     R3,(R4)         ;STORE IT BACK
221 003651  104204  001123                   MOV     #HOLDPN,R4      ;FOR DECREMENT
222 003653  104203  001127                   MOV     #SECTRK,R3      ;DITTO
223 003655  021524                          CALL    DSUB            ;SUBTRAC TO GET NEXT TRACK
224 003656  000000                          RETURN
```

```
                        ;
                        ;       PAGE IN NEXT FCT BLOCK
                        ;
003657                  FCPG:   PUSH    R5                      ;SAVE R5
003660  104200  005152  001253         MOV     #PBNBUF,BUFPNT          ;POINT TO BUFFER
003663  104201  000017         MOV     #F6,R1                  ;OVERLAY F6 DOES IT
003665  022444                 CALL    PAGE                    ;EXECUTE IT
003666  104200  000200  001500         MOV     #128.,PCNT              ;RESET COUNT
003671                         POP     R5                      ;RESTORE R5
003672  000000                 RETURN                          ;RETURN
```

```
                                        .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
                                ;
003673                          ;       DMOVLY  G8,START
                                ;
                                ;
003047  104200  000052  001153          MOV     #G8,CUROVL                      ;FOR RECORDING
003052  104207  001052                  MOV     #SCR,R0                 ;POINT TO CHARACTERISTICS BLOCK
003054  023104                          CALL    NUMLBN                  ;GET NUMBER OF FIRST LBN ON LAST LBN CYL
003055  104140  001113                  MOV     (R4),CURBN              ;GET LOW ORDER
003057  104140  001117                  MOV     (R4),HOLDBN             ;SAVE FOR LATER
003061  104640  000001  001114          MOV     1(R4),CURBN+1           ;GET HIGH ORDER
003064  104640  000001  001120          MOV     1(R4),HOLDBN+1          ;SAVE FOR LATER
003067  023144                          CALL    NUMRBN                  ;GET NUM OF FIRST RBN ON LAST LBN CYLINDER
003070  104140  001106                  MOV     (R4),CURRBN             ;GET LOW ORDER
003072  104640  000001  001107          MOV     1(R4),CURRBN+1          ;GET HIGH ORDER
003075  104140  001121                  MOV     (R4),HOLRBN             ;SAVE LOW FOR LATER
003077  104640  000001  001122          MOV     1(R4),HOLRBN+1          ;SAVE HIGH FOR LATER
003102  023204                          CALL    LPBN                    ;GET PBN OF FIRST SECTOR ON LAST TRACK
003103  000000                          RETURN
```

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

```
                                        ;
                                        ;       COMPUTE STARTING RBN ON LAST TRACK IN LBN AREA
                                        ;       R0 ->   CHARACTERISTICS BLOCK
                                        ;
003144  104300  001143  000736  NUMRBN: MOV     LBNCYL,TEMP             ;GET LOW ORDER NUMBER OF LBN CYLINDER
003147  104300  001144  000737          MOV     LBNCYL+1,TEMP+1         ;GET HIGH ORDER
003152  103200  170000  000737          BIC     #HD.CLR,TEMP+1          ;CLEAR STARTING CYLINDER BITS
003155  104204  000736                  MOV     #TEMP,R4                ;DITTO
003157  104203  001147                  MOV     #RBNPCY,R3              ;GET RBN'S/CYLINDER
003161  021542                          CALL    DMUL                    ;GET FIRST RBN ON LAST CYLINDER
003162  104641  000001                  MOV     1(R4),R1                ;GET HIGH ORDER
003164  105301  001336                  ADD     ST.RBN,R1               ;ADD TO GET ABSOLUTE LBN
003166  100641  000001                  MOV     R1,1(R4)                ;STORE BACK
003170  104673  000004                  MOV     RBNTRK(R0),R3           ;GET RBN/TRACK
003172  103203  177600                  BIC     #HI1BYTE,R3             ;CLEAR OUT GARBAGE
003174  104030  000731                  MOV     R3,DDUMMY               ;STORE IT
003176  114000  000732                  CLR     DDUMMY+1                ;FOR STORE
003200  104203  000731                  MOV     #DDUMMY,R3              ;WANT LAST TRACK
003202  021524                          CALL    DSUB                    ;GET IT
003203  000000                          RETURN
```

```
    1                                   ;
    2                                   ;       COMPUTE PBN FROM GIVEN LBN
    3                                   ;       R0 ->   CHARACTERISTICS BLOCK
    4                                   ;
    5 003204  104300  001113  000736  LPBN:   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     #HIBYTE,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),HOLDPN             ;GET LOW ORDER
   29 003266  104640  000001  001124          MOV     1(R4),HOLDPN+1          ;STORE HIGH ORDER
   30 003271  000000                          RETURN
```

```
                                        .SBTTL  FCT DOWN-LINE LOAD OVERLAY (F3)
                                ;
                                ;       DOWNLINE LOADER FOR FCT
                                ;
003272                          FDLL:   DMOVLY  F3,START                ;OVERLAY #3
                                ;
003047  104200  000006  001153          MOV     #F3,CUROVL              ;OVERLAY #3
003052  104200  000001  001501          MOV     #1,COUNT                ;INIT COUNT TO 1
003055  104300  001337  001116          MOV     ST.XBN,CURXBN+1         ;ALSO INITIALIZE XBN COUNTER
003060  104300  001337  001114          MOV     ST.XBN,CURBN+1          ;HIGH ORDER
003063  114000  001115                  CLR     CURXBN                  ;LOW ORDER IS ZERO
003065  114000  001113                  CLR     CURBN                   ;DITTO
003067  104207  001430                  MOV     #CONBLK,R0              ;POINT TO CONVERT BLOCK
003071  104203  001052                  MOV     #SCR,R3                 ;POINT TO CHARACTERISTICS
003073  104632  000000                  MOV     CYLBN(R3),R2            ;GET LBN CYLINDERS
003075  100672  000000                  MOV     R2,V1(R0)               ;STORE IN CONVERT BLOCK
003077  104632  000001                  MOV     CYLBN+1(R3),R2          ;GET HIGH ORDER
003101  100672  000001                  MOV     R2,V1+1(R0)             ;STORE IT
003103  104303  001127                  MOV     SECTRK,R3               ;GET SECTORS/TRACK
003105  100673  000004                  MOV     R3,V3(R0)               ;STORE IN CONVERT BLOCK
003107  102200  002000  001217          BIT     #BSTGS,FLAG             ;IN BEST GUESS MODE ?
003112  053114                          BNE     NODLL                   ;YUP - FIX UP FIRST BLOCK
003113  003144                          BR      LOOP                    ;START LOOP
003114  104203  004535          NODLL:  MOV     #RDBUF,R3               ;POINT TO BUFFER
003116  114002                          CLR     R2                      ;SET MEDIA MODE TO 0 (IN FORMAT)
003117  100132                          MOV     R2,(R3)                 ;STORE IT
003120  104204  001305                  MOV     #SERNUM,R4              ;POINT TO SERIAL NUMBER
003122  105203  000002                  ADD     #FSER,R3                ;POINT TO ENTRY IN FCT BLOCK
003124  104205  000004                  MOV     #4,R5                   ;INIT COUNTER
003126  104242                  9$:     MOV     (R4)+,R2                ;GET WORD
003127  100232                          MOV     R2,(R3)+                ;STORE WORD
003130  117405                          DEC     R5                      ;DECRMENT COUNTER
003131  053126                          BNE     9$                      ;CONTINUE TILL DONE
003132  114002                          CLR     R2                      ;FOR INSTANCE NUMBER
003133  104203  004535                  MOV     #RDBUF,R3               ;RESET POINTER
003135  100632  000001                  MOV     R2,INST(R3)             ;STORE INSTANCE NUMBER IN BLOCK
003137  101202  100000                  BIS     #NOFCT,R2               ;SET NO FCT AVAILABLE BIT
003141  100632  000025                  MOV     R2,FCTFLG(R3)           ;STORE IT IN FCT INFO BLOCK
003143  003203                          BR      LOOPP2                  ;SKIP DLL STUFF
003144  104205  001263          LOOP:   MOV     #DMBUF,R5               ;POINT TO MAINT BUFFER
003146  104303  001321                  MOV     FCMSG,R3                ;GET DUP CODE
003150  100153                          MOV     R3,(R5)                 ;STORE IT IN MESSAGE
003151  104303  001322                  MOV     FCMSG+1,R3              ;GET 'F ' IDENTIFIER
003153  100653  000001                  MOV     R3,1(R5)                ;STORE IT
003155  104303  001113                  MOV     CURBN,R3                ;GET BLOCK NUMBER DESIRED
003157  100653  000002                  MOV     R3,2(R5)                ;STORE IT
003161  022536                          CALL    SNDMNT                  ;SEND REQUEST
003162  022544                          CALL    RCVMNT                  ;RECEIVE ANSWER
003163  104653  000002                  MOV     2(R5),R3                ;GET BLOCK COUNT RECEIVED
003165  013451                          BEQ     DONDLL                  ;DONE DLL OF ACTUAL FCT
003166  104650  000003  001425          MOV     3(R5),OVLBLK+1          ;GET HOST ADDRESS
003171  114000  001426                  CLR     OVLBLK+2                ;ZERO HIGH ORDER
003173  104200  000401  001424          MOV     #257.,OVLBLK            ;NUMBER OF WORDS TO TRANSFER
003176  104204  001424                  MOV     #OVLBLK,R4              ;FOR OVERLAY ROUTINE
003200  104203  004535                  MOV     #RDBUF,R3               ;POINT TO BUFFER
003202  022526                          CALL    OVRLAY                  ;GET THE SECTOR
003203  114005                  LOOPP2: CLR     R5                      ;CLEAR WRITE ERROR COUNT
```

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

```
115 003346                                  2$:
116 003346  115405                             INC     R5              ;YUP - INCREMENT COUNTER
117 003347  115400  001245                  NO: INC     NEXT1           ;INCREMENT IT
118 003351  114000  001504                      CLR     TMPTRY          ;FOR RESET
119 003353  104300  001503  001505              MOV     RECOV,RECTMP    ;GET RECOVERY LEVELS
120 003356  104204  001113                      MOV     #CURBN,R4       ;FOR ADD
121 003360  104203  001240                      MOV     #FCTFMT,R3      ;FOR ADD
122 003362  021506                              CALL    DADD            ;POINT TO NEXT COPY
123 003363  106300  001244  001245              CMP     FCTCPY,NEXT1    ;DONE THIS SECTOR ?
124 003366  053231                              BNE     LOOPP           ;NO - WRITE NEXT FCT COPY
125 003367  106305  001244                      CMP     FCTCPY,R5       ;ERROR ON EVERY WRITE ?
126 003371  013465                              BEQ     ERROR           ;YUP - BIG TROUBLE
127 003372  102200  002000  001217              BIT     #BSTGS,FLAG     ;BEST GUESS ?
128 003375  053446                              BNE     DLLRET          ;YUP - JUST WANT TO WRITE FIRST BLOCK
129 003376  102200  004000  001217              BIT     #NDLL,FLAG      ;ALL DONE ???
130 003401  053432                              BNE     DLLRT1          ;YUP - EXIT
131 003402  060022                              XFC     UPDATE          ;LET HOST KNOW STILL ALIVE
132 003403  115400  001501                      INC     COUNT           ;INCREMENT IT
133 003405                                      DUBINC  CURXBN          ;INCREMENT IT
134 003412  104300  001115  001113              MOV     CURXBN,CURBN    ;GET LOW ORDER
135 003415  104300  001116  001114              MOV     CURXBN+1,CURBN+1 ;GET HIGH ORDER
136 003420  106300  001260  001501              CMP     FCTNPD,COUNT    ;AT THE LAST NON-PAD ENTRY
137 003423  053426                              BNE     12$             ;NOPE
138 003424  117400  001244                      DEC     FCTCPY          ;DECREMENT - NO PAD ON LAST COPY
139 003426  106300  001240  001501  12$:       CMP     FCTFMT,COUNT    ;DONE ?
140 003431  053144                              BNE     LOOP            ;NOPE - DO NEXT SECTOR
141 003432  101200  000001  001217  DLLRT1:     BIS     #FCTAVL,FLAG    ;SET FCT AVAILABLE
142 003435  104207  001037                      MOV     #CR,R0          ;POINT TO CHARACTERISTICS BLK
143 003437  104673  000001                      MOV     FRCPY(R0),R3    ;GET F/RCT COPIES
144 003441  110703                              SWAB    R3              ;GET INTO  JW BYTE
145 003442  103203  177760                      BIC     #FCLR,R3        ;CLEAR OUT REST OF GARBAGE
146 003444  104030  001244                      MOV     R3,FCTCPY       ;RESTORE NUM OF COPIES
147 003446  104201  000003  DLLRET:             MOV     #F2,R1          ;FOR OVERLAY #2
148 003450  022412                              CALL    NEXT            ;LBN FORMATTING
149 003451  104205  000400  DONDLL:             MOV     #256.,R5        ;COUNT FOR BLOCK INIT
150 003453  104204  004535                      MOV     #RDBUF,R4       ;POINT TO BUFFER
151 003455  114003                              CLR     R3              ;INIT TO 0
152 003456  100243  LOOP3:                     MOV     R3,(R4)+        ;CLEAR ONE WORD
153 003457  117405                              DEC     R5              ;DEC COUNTER
154 003460  053456                              BNE     LOOP3           ;CONTINUE TILL DONE
155 003461  101200  004000  001217              BIS     #NDLL,FLAG      ;SET FLAG
156 003464  003203                              BR      LOOPP2          ;CONTINUE
157 003465  104012  ERROR:                     MOV     R1,R2           ;GET XFC ERROR CODE
158 003466  104201  000015                      MOV     #13.,R1         ;FCT WRITE ERROR
159 003470  022552  DLERT:                     CALL    ERRMNT          ;ERROR RETURN
```

```
                                        .SBTTL  RCT UPDATE OVERLAY (F4)
003471                                  DMOVLY  F4,START
                                ;
                                ;
                                ;
                                ;
                                ;       THIS OVERLAY UPDATES THE RCT WITH THE SECTORS
                                ;       IN THE ERRBUF BUFFER
                                ;
                                ;
003047  104200  000011  001153  RCTUPD: MOV     #F4,CUROVL              ;GET OVERLAY
003052  104303  001112                  MOV     CURTRK,R3               ;GET CURRENT TRACK
003054                                  PUSH    R3                      ;SAVE FOR RESTORE
003055  104303  001125                  MOV     CYLNUM,R3               ;GET LOW ORDRE CYLINDER
003057                                  PUSH    R3                      ;SAVE FOR RESTORE
003060  104303  001126                  MOV     CYLNUM+1,R3             ;GET HIGH ORDER
003062                                  PUSH    R3                      ;SAVE FOR RESTORE
003063  104300  001133  001227          MOV     LBNLBN,HOLD             ;GET LOW ORDER COUNT OF LBN'S
003066  104300  001134  001230          MOV     LBNLBN+1,HOLD+2         ;GET HIGH ORDER
003071  104203  001475                  MOV     #TOTRCT,R3              ;FOR SUBTRACT
003073  104204  001227                  MOV     #HOLD,R4                ;DITTO
003075  021524                          CALL    DSUB                    ;GET STARTING RCT LBN
003076  104300  001224  001474          MOV     ERRBUF,UPDPNT           ;POINT TO ERROR BUFFER
003101  104302  001474          ROVER:  MOV     UPDPNT,R2               ;GET POINTER TO BAD LIST
003103  104120  000731                  MOV     (R2),DDUMMY             ;GET LOW ORDER
003105  104620  000001  000732          MOV     1(R2),DDUMMY+1          ;GET HIGH ORDER
003110  102200  100000  000732          BIT     #BIT15,DDUMMY+1         ;IS IT AN RBN ??
003113  013116                          BEQ     ROVER1                  ;NO - REGULAR HASH
003114  104201  177777                  MOV     #-1,R1                  ;HASH FOR RBN
003116  103200  170000  000732  ROVER1: BIC     #HD.CLR,DDUMMY+1        ;CLEAR THE HEADER
003121  104204  000731                  MOV     #DDUMMY,R4              ;FOR HASH
003123  023357                          CALL    UHASH                   ;FIND THE RCT ENTRY FOR CURRENT ERR BLOCK
003124  104143                          MOV     (R4),R3                 ;GET BLOCK NUMBER
003125  105203  000002                  ADD     #2,R3                   ;ADD TO GET PAST FIRST 2 BLOCKS
003127  100143                          MOV     R3,(R4)                 ;STORE BACK
003130  104030  001477                  MOV     R3,RCTCNT               ;FOR LATER PING-PONG
003132  104203  001227                  MOV     #HOLD,R3                ;FOR ADD
003134  021506                          CALL    DADD                    ;TO GET LBN OF RCT BLOCK
003135  104040  001253                  MOV     R4,BUFPNT               ;STORE POINTER TO BLOCK NUMBER
003137  104201  000055                  MOV     #H1,R1                  ;RCT READ OVERLAY
003141  022444                          CALL    PAGE                    ;DO IT
003142  104205  006621                  MOV     #RCTBUF,R5              ;POINT TO BUFFER
003144  104303  000736                  MOV     OFFSET,R3               ;GET OFFSET
003146  105035                          ADD     R3,R5                   ;POINT TO HIT ENTRY
003147  104302  001474                  MOV     UPDPNT,R2               ;RESTORE POINTER
003151  104623  000001                  MOV     1(R2),R3                ;GET THE HEADER
003153  103203  007777                  BIC     #LO,R3                  ;CLEAR ALL BUT HEADER
003155  106203  110000                  CMP     #HD.BAD,R3              ;IS IT A BAD RBN ?
003157  053226                          BNE     NOTR                    ;NOPE - CHECK FOR PRIMARY
003160  104650  000000  000733          MOV     0(R5),TEMP2             ;GET LOW ORDER CURRENT RESIDENT
003163  104650  000001  000734          MOV     1(R5),TEMP2+1           ;GET HIGH ORDER
003166  103203  170000                  BIC     #HD.CLR,R3              ;CLEAR HEADER
003170  101203  040000                  BIS     #RC.UNU,R3              ;MARK AS UNUSUABLE
003172  103203  007777                  BIC     #LO,R3                  ;CLEAR LOW ORDER
003174  100653  000001                  MOV     R3,1(R5)                ;STORE IT BACK
003176  114003                          CLR     R3                      ;CLEAR FOR STORE
003177  100153                          MOV     R3,(R5)                 ;CLEAR LOW ORDER
```

```
 58 003200  102200  020000  000734          BIT     #BIT13,TEMP2+1          ;ANY THING DISPLACED ???
 59 003203  013322                           BEQ     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.PRV,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.PRV,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 RESDENT
 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.PRV,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                   ;WTIR 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
```

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

```
                                        ;
                                        ;       WRITE AN RCT BLOCK
                                        ;
003421  114005                  RCTWT:  CLR     R5                      ;CLEAR ERROR OCUNTER
003422  104050  001245                  MOV     R5,NEXT1                ;RESET NEXT COUNTER
003424  104204  000731                  MOV     #DDUMMY,R4              ;POINT TO BLOCK
003426  104203  001052          RCTWLP: MOV     #SCR,R3                 ;POINT TO CHARACTERISTICS
003430  104632  000011                  MOV     LBNTRK(R3),R2           ;GET LBN/TRACK
003432  103202  177400                  BIC     #HIBYTE,R2              ;CLEAR REST OF WORD
003434  104207  001430                  MOV     #CONBLK,R0              ;POINT TO CONVERT BLOCK
003436  100672  000004                  MOV     R2,V3(R0)               ;FOR CONVERT
003440  104632  000001                  MOV     STCYL(R3),R2            ;STARTING CLYLINDER
003442  103202  007777                  BIC     #LO,R2                  ;CLEAR REST OF WORD
003444  100672  000001                  MOV     R2,V1+1(R0)             ;STORE
003446  114002                          CLR     R2                      ;CLEAR FOR STORE
003447  100672  000000                  MOV     R2,V1(R0)               ;LOW ORDER ALWAYS 0
003451  022714                          CALL    CVTSK                   ;CONVERT AND SEEK
003452  104207  000721                  MOV     #WRBLK,R0               ;POINT TO COMMAND BLOCK
003454  104203  122400                  MOV     #WRCMD,R3               ;GET WRITE COMMAND
003456  104302  001112                  MOV     CURTRK,R2               ;GET CURRENT TRACK
003460  101023                          BIS     R2,R3                   ;SET TRACK FOR WRITE
003461  100673  000004                  MOV     R3,RW.CMD(R0)           ;STORE IN COMMAND BLOCK
003463  104202  006621                  MOV     #RCTBUF,R2              ;POINT TO BUFFER
003465  022627                          CALL    CEDC                    ;COMPUTE EDC - RETURNED IN R3
003466  100623  000400                  MOV     R3,RW.EDC(R2)           ;STORE IT
003470  100672  000001                  MOV     R2,RW.BUF(R0)           ;STICK IN COMMAND BLOCK
003472  104143                          MOV     (R4),R3                 ;GET LOW ORDER HEADER
003473  100673  000002                  MOV     R3,RW.LOW(R0)           ;STORE IN WRITE BLOCK
003475  104643  000001                  MOV     1(R4),R3                ;GET HIGH ORDER
003477  105303  001335                  ADD     ST.LBN,R3               ;ADD STARTING LBN BITS
003501  101203  000000                  BIS     #HD.LBN,R3              ;SET HEADER
003503  100673  000003                  MOV     R3,RW.HI(R0)            ;STORE IN WRITE BLOCK
003505  104203  000726                  MOV     #HSLIM-1,R3             ;GET DUMMY SDI POINTER
003507  100673  000005                  MOV     R3,RW.DUM(R0)           ;POINT IN COMMAND BLOCK
003511  104303  001333          WRITE2: MOV     HPREA,R3                ;GET HEADER PREAMBLE
003513  104304  001334                  MOV     DPREA,R4                ;GET DATA PREAMBLE
003515  104302  000740                  MOV     UNIT,R2                 ;SET UNIT
003517  104207  000721                  MOV     #WRBLK,R0               ;MAKE SURE POINTING AT BLOCK
003521  060012                          XFC     SIP                     ;WAIT FOR SECTOR PULSE
003522  060003                          XFC     WRITE                   ;WRITE SECTOR
003523  115001                          TST     R1                      ;ANY ERROR ?
003524  013546                          BEQ     RWGD                    ;NOPE
003525  106300  001502  001504          CMP     RETRY,TMPTRY            ;MAX ?
003530  013534                          BEQ     1$                      ;YES - TRY SOME RECOVERY
003531  115400  001504                  INC     TMPTRY                  ;INC RETRY COUNT
003533  003511                          BR      WRITE2                  ;DO RETRY
003534  104303  001505          1$:     MOV     RECTMP,R3               ;GET CURRENT ERROR RECOVERY LEVEL
003536  073545                          BMI     2$                      ;IF NEGATIVE THEN FRIED
003537  022612                          CALL    ERRHND                  ;TRY RECOVERY
003540  114000  001504                  CLR     TMPTRY                  ;FOR INIT
003542  117400  001505                  DEC     RECTMP                  ;DECREMENT IT
003544  003511                          BR      WRITE2                  ;RETRY
003545                          2$:
003545  115405                          INC     R5                      ;YUP - INCREMENT COUNTER
003546  115400  001245          RWGD:   INC     NEXT1                   ;INCREMENT IT
003550  114000  001504                  CLR     TMPTRY                  ;FOR RESET
003552  104300  001503  001505          MOV     RECOV,RECTMP            ;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 T'IS 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 REAPEATS ?
 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
```

```
                                        ;
                                        ;       SEARCH FOR OPEN ENTRY IN RCT
                                        ;
003612                          PNGPG:  PUSH    R2                      ;SAVE R2
003613  114000  001222                  CLR     WRFLG                   ;CLEAR WRAP FLAG
003615  114002                  XNGBLK: CLR     R2                      ;FOR FLOP SET
003616  104303  000736                  MOV     OFFSET,R3               ;GET OFFSET
003620  104035                  XAGAIN: MOV     R3,R5                   ;MOV OFFSET INTO BUFF POINTER
003621  105025                          ADD     R2,R5                   ;ADD FLOP VALUE
003622  102205  000400                  BIT     #BIT8,R5                ;PAST ONE END (OR BOTH)
003624  053637                          BNE     XFLIP                   ;YUP - FLIP OTHER DIRECTION
003625  104651  006622                  MOV     RCTBUF+1(R5),R1         ;GET HEADER CODE
003627  103201  007777                  BIC     #LO,R1                  ;CLEAR LOW ORDER
003631  106201  100000                  CMP     #RC.NUL,R1              ;END OF RCT ?
003633  013675                          BEQ     XEORCT                  ;YUP - WRAP TO FIRST BLOCK
003634  106201  000000                  CMP     #RC.FRE,R1              ;FREE ?
003636  013707                          BEQ     XPRET                   ;YUP - ALL DONE
003637  104025                  XFLIP:  MOV     R2,R5                   ;GET FLIP VALUE
003640  114002                          CLR     R2
003641  107052                          SUB     R5,R2                   ;NEGATE IT
003642  073645                          BMI     XNOINC                  ;IF NEGATIVE DON'T INC
003643  105202  000002                  ADD     #2,R2                   ;ADD TO NEXT VALUE
003645  106202  000400          XNOINC: CMP     #256.,R2                ;DONE EVERY SLOT IN BLOCK ?
003647  053620                          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
                                        ;
003650  115400  001477                  INC     RCTCNT                  ;INC TO NEXT ONE
003652  104303  001477          XPNGRD: MOV     RCTCNT,R3               ;FOR STORE
003654  104204  000731                  MOV     #DDUMMY,R4              ;FOR ADD
003656  100143                          MOV     R3,(R4)                 ;STORE BLOCK NUMBER
003657  114003                          CLR     R3                      ;FOR RESETS
003660  100643  000001                  MOV     R3,1(R4)                ;CLEAR HIGH WORD
003662  104030  000736                  MOV     R3,OFFSET               ;MAKE OFFSET AT BEGINNING
003664  104203  001227                  MOV     #HOLD,R3                ;POINT TO FIRST RCT LBN
003666  021506                          CALL    DADD                    ;GET LBN OF THIS RCT BLOCK
003667  104040  001253                  MOV     R4,BUFPNT               ;STORE BLOCK NUMBER
003671  104201  000055                  MOV     #H1,R1                  ;READ RCT OVERLAY
003673  022444                          CALL    PAGE                    ;DO IT
003674  003615                          BR      XNGBLK                  ;SEARCH THIS BLOCK
003675  104303  001222          XEORCT: MOV     WRFLG,R3                ;GET WRAP FLAG
003677  053711                          BNE     XPERR                   ;IF BEEN HERE ONCE THEN RCT FULL
003700  104200  000002  001477          MOV     #2,RCTCNT               ;FOR FIRST RCT BLOCK
003703  104200  000002  001222          MOV     #2,WRFLG                ;MAKE WRAP FLAG NON-ZERO
003706  003652                          BR      XPNGRD                  ;READ IT AND CONTINUE
                                        ;
                                        ;
003707                          XPRET:  POP     R2                      ;RESTORE R2
003710  000000                          RETURN                          ;SUCCESSFUL RETURN
003711  104201  000020          XPERR:  MOV     #16.,R1                 ;RCT FULL
003713  114002                          CLR     R2                      ;NO SUBCODE
003714  022552                          CALL    ERRMNT                  ;ERROR RETURN
```

```
                                        .SBTTL  RCT READ OVERLAY (H1)
003715                                  DMOVLY  H1,START
                                ;
                                ;       READ A BLOCK OF THE RCT
                                ;
003047                                  PUSHA
003055  104200  000055  001153          MOV     #H1,CUROVL              ;FOR INIT
003060  104304  001253                  MOV     BUFPNT,R4               ;GET POINTER TO BLOCK NUMBER
003062  114005                          CLR     R5                      ;CLEAR ERROR COUNTER
003063  104203  001052                  MOV     #SCR,R3                 ;POINT TO CHARACTERISTICS
003065  104632  000001                  MOV     STCYL(R3),R2            ;GET STARTING CYLINDER
003067  103202  007777                  BIC     #LO,R2                  ;CLEAR REST OF WORD
003071  104207  001430                  MOV     #CONBLK,R0              ;POINT TO CONVERT BLOCK
003073  100672  000001                  MOV     R2,V1+1(R0)             ;STORE FOR CONVERT
003075  114002                          CLR     R2                      ;FOR STORE
003076  100672  000000                  MOV     R2,V1(R0)               ;LOW ORDER ALWAYS 0
003100  104632  000011                  MOV     LBNTRK(R3),R2           ;GET LBN/TRK
003102  103202  177400                  BIC     #HIBYTE,R2              ;CLEAR HIGH BYTE
003104  100672  000004                  MOV     R2,V3(R0)               ;STORE IN CONVERT BLOCK
003106                          OCLOOP:
003106  022714                          CALL    CVTSK                   ;CONVERT RCT BLOCK NUMBER AND SEEK
003107  104207  000721                  MOV     #RDBLK,R0               ;PREPARE FOR READ SECTORS
003111  104203  000726                  MOV     #HSLIM-1,R3             ;POINTER TO DUMMY SDI BLOCK
003113  100673  000005                  MOV     R3,RW.DUM(R0)           ;STORE IN COMMAND BLOCK
003115  104143                          MOV     (R4),R3                 ;LO ORDER BLOCK NUMBER
003116  100673  000002                  MOV     R3,RW.LOW(R0)           ;STORE IN READ COMD BLOCK
003120  104643  000001                  MOV     1(R4),R3                ;HI ORDER BLOCK NUM AND CODE
003122  105303  001335                  ADD     ST.LBN,R3               ;ADD STARTING LBN BITS
003124  100673  000003                  MOV     R3,RW.HI(R0)            ;STORE IN READ CMD BLOCK
003126  104203  006621                  MOV     #RCTBUF,R3              ;LOAD ADDRESS OF DATA BUFFER
003130  100673  000001                  MOV     R3,RW.BUF(R0)           ;STORE IN COMMAND BUFFER
003132  104203  013400                  MOV     #RWCMD,R3               ;LOAD SDI READ COMMAND
003134  104301  001112                  MOV     CURTRK,R1               ;GET CURRENT HEAD NUMBER IN R1
003136  101013                          BIS     R1,R3                   ;SET IT IN COMMAND
003137  100673  000004                  MOV     R3,RW.CMD(R0)           ;STORE BACK
003141  104207  000721          READ11: MOV     #RDBLK,R0               ;MAKE SURE POINTING AT BLOCK
003143  104203  100000                  MOV     #RDCMD,R3               ;MARK AS ONLY REQUEST
003145  100173                          MOV     R3,(R0)                 ;STORE IN CMD BLOCK
003146  104302  000740                  MOV     UNIT,R2                 ;SDI INTERCONNECT
003150  101207  100000                  BIS     #BIT15,R0               ;SET NO REVECTORING
003152  060012                          XFC     SIP                     ;WAIT FOR PULSE
003153  060002                          XFC     READ                    ;READ 1 SECTOR
003154  115001                          TST     R1                      ;ANY ERRORS ?
003155  053173                          BNE     100$                    ;YES - TRY RECOVERY
003156  104173                          MOV     (R0),R3                 ;GET STATUS WORD
003157  102203  010000                  BIT     #ECCF,R3                ;ECC ERROR ?
003161  013165                          BEQ     101$                    ;NOPE - VERIFY EDC
003162  023027                          CALL    ECCCK                   ;CORRECT ECC
003163  115001                          TST     R1                      ;TEST FLAG
003164  053173                          BNE     100$                    ;UNCORRECTABLE
003165  104202  006621          101$:   MOV     #RCTBUF,R2              ;POINT TO BUFFER
003167  022627                          CALL    CEDC                    ;COMPUTE EDC
003170  106623  000400                  CMP     RW.EDC(R2),R3           ;O.K. ?
003172  013230                          BEQ     102$                    ;YUP - CONSIDER GOOD
003173  106300  001502  001504  100$:   CMP     RETRY,TMPTRY            ;MAX ?
003176  013202                          BEQ     1$                      ;YES - TRY SOME RECOVERY
003200  115400  001504                  INC     TMPTRY                  ;INC RETRY COUNT
```

```
003201  003141                          BR      READ11          ;DO RETRY
003202  104303  001505          1$:     MOV     RECTMP,R3       ;GET CURRENT ERROR RECOVERY LEVEL
003204  073213                          BMI     2$              ;IF NEGATIVE THEN FRIED
003205  022612                          CALL    ERRHND          ;TRY RECOVERY
003206  114000  001504                  CLR     TMPTRY          ;FOR INIT
003210  117400  001505                  DEC     RECTMP          ;DECREMENT IT
003212  003141                          BR      READ11          ;RETRY
003213                          2$:
003213  115405                          INC     R5              ;INCREMENT BAD COUNTER
003214  106305  001244                  CMP     FCTCPY,R5       ;ALL BAD ?
003216  013344                          BEQ     ORFTAL          ;YUP - ALL OVER
003217  104203  001242                  MOV     #RCTFMT,R3      ;SIZE OF TABLE - R4 -> BLOCK NUMBER
003221  021506                          CALL    DADD            ;ADD TO POINT TO NEXT COPY
003222  114000  001504                  CLR     TMPTRY          ;RESET RETRY COUNT
003224  104300  001503  001505          MOV     RECOV,RECTMP    ;DITTO RECOVERY LEVELS
003227  003106                          BR      OCLOOP          ;BRANCH BACK
003230                          102$:
003230  114000  001504          OCDONE: CLR     TMPTRY          ;FOR RESET
003232  104300  001503  001505          MOV     RECOV,RECTMP    ;GET RECOVERY LEVELS
003235  115005                          TST     R5              ;ANY ERRORS ?
003236  013335                          BEQ     RLDONE          ;NO - EXIT
003237  104203  001240                  MOV     #FCTFMT,R3      ;SIZE OF TABLE
003241  021524                          CALL    DSUB            ;GET BACK TO PREVIOUS COPY
003242  022714                          CALL    CVTSK           ;CONVERT AND SEEK
003243  104207  000721                  MOV     #WRBLK,R0       ;POINT TO COMMAND BLOCK
003245  104203  122400                  MOV     #WRCMD,R3       ;GET WRITE COMMAND
003247  104302  001112                  MOV     CURTRK,R2       ;GET CURRENT TRACK
003251  101023                          BIS     R2,R3           ;SET TRACK FOR WRITE
003252  100673  000004                  MOV     R3,RW.CMD(R0)   ;STORE IN COMMAND BLOCK
003254  104203  006621                  MOV     #RCTBUF,R3      ;POINT TO BUFFER
003256  100673  000001                  MOV     R3,RW.BUF(R0)   ;STICK IN COMMAND BLOCK
003260  104143                          MOV     (R4),R3         ;GET LOW ORDER HEADER
003261  100673  000002                  MOV     R3,RW.LOW(R0)   ;STORE IN WRITE BLOCK
003263  104643  000001                  MOV     1(R4),R3        ;GET HIGH ORDER
003265  105303  001335                  ADD     ST.LBN,R3       ;ADD STARTING LBN BITS
003267  100673  000003                  MOV     R3,RW.HI(R0)    ;STORE IN WRITE BLOCK
003271  104203  000726                  MOV     #HSLIM-1,R3     ;GET DUMMY SDI POINTER
003273  100673  000005                  MOV     R3,RW.DUM(R0)   ;POINT IN COMMAND BLOCK
003275  104303  001333          WRIT12: MOV     HPREA,R3        ;GET HEADER PREAMBLE
003277  104304  001334                  MOV     DPREA,R4        ;GET DATA PREAMBLE
003301  104302  000740                  MOV     UNIT,R2         ;SET UNIT
003303  104207  000721                  MOV     #WRBLK,R0       ;MAKE SURE POINTING AT BLOCK
003305  101207  100000                  BIS     #BIT15,R0       ;SET NO REVECTORING
003307  060012                          XFC     SIP             ;WAIT FOR SECTOR PULSE
003310  060003                          XFC     WRITE           ;WRITE SECTOR
003311  115001                          TST     R1              ;ANY ERROR ?
003312  013333                          BEQ     2$              ;NO - SKIP RETRY
003313  106300  001502  001504          CMP     RETRY,TMPTRY    ;MAX ?
003316  013322                          BEQ     1$              ;YES - TRY SOME RECOVERY
003317  115400  001504                  INC     TMPTRY          ;INC RETRY COUNT
003321  003275                          BR      WRIT12          ;DO RETRY
003322  104303  001505          1$:     MOV     RECTMP,R3       ;GET CURRENT ERROR RECOVERY LEVEL
003324  073333                          BMI     2$              ;IF NEGATIVE THEN FRIED
003325  022612                          CALL    ERRHND          ;TRY RECOVERY
003326  114000  001504                  CLR     TMPTRY          ;FOR INIT
003330  117400  001505                  DEC     RECTMP          ;DECREMENT IT
003332  003275                          BR      WRIT12          ;RETRY
```

```
;15 003333                          2$:
;16 003333  117405                          DEC     R5                      ;DEREMENT COUNTER
;17 003334  003230                          BR      OCDONE                  ;SEE IF ANY MORE TO DO
;18 003335                          RLDONE: POPA
;19 003343  000000                          RETURN                          ;ALL DONE
;20 003344  104012                  ORFTAL: MOV     R1,R2                   ;XFC ERROR CODE
;21 003345  104201  000016                  MOV     #14.,R1                 ;RCT READ ERROR
;22 003347  022552                          CALL    ERRMNT                  ;ERROR RETURN
```

```
                                .SBTTL  FCT->RCT CONVERSION OVERLAY (F5)
                        ;
                        ;       CONVERT FCT INTO RCT
                        ;
                                DMOVLY  F5,START
                        ;
                        ;
003047  003212                  JMP     START2          ;SKIP SUBROUTINES
```

```
                                        ;
                                        ;       COMPUTE RCT ADDRESS FOR GIVEN LBN
                                        ;
003050  115001                  HASH:   TST     R1                      ;NEED TO COMPUTE PRMARY RBN ?
003051  073101                          BMI     HKIP1                   ;NO - SKIP IT
003052  104207  001052                  MOV     #SCR,R0                 ;POINT TO CUBUNIT CHARACTERISTICS
003054  022757                          CALL    PRIMRB                  ;COMPUTE PRIMARY RBN
003055  104200  000200  000736  HKIP:   MOV     #128.,TEMP              ;DIVIDE BY 128 TO GET BLOCK NUMBER
003060  114000  000737                  CLR     TEMP+1                  ;FOR STORE
003062  104300  001151  000731          MOV     REVRBN,DDUMMY           ;GET PRIMARY RBN
003065  104300  001152  000732          MOV     REVRBN+1,DDUMMY+1       ;GET HIGH ORDER
003070  104204  000731                  MOV     #DDUMMY,R4              ;FOR DIVIDE
003072  104203  000736                  MOV     #TEMP,R3                ;DITTO
003074  021570                          CALL    DDIV                    ;DDUMMY=RCT BLOCK NUMBER
                                                                        ;TEMP=OFFSET
003075  104131                          MOV     (R3),R1                 ;GET OFFSET
003076  105011                          ADD     R1,R1                   ;MULTIPLY BY 2
003077  100131                          MOV     R1,(R3)                 ;STORE BACK
003100  000000                          RETURN
003101  104140  001151          HKIP1:  MOV     (R4),REVRBN             ;FORDIVIDE SETUP
003103  104640  000001  001152          MOV     1(R4),REVRBN+1          ;DITTO
003106  107300  001336  001152          SUB     ST.RBN,REVRBN+1         ;SUBTRACT STARTING RBN BITS
003111  003055                          BR      HKIP                    ;DO DIVIDE
```

```
                                        ;
                                        ;       SEARCH FOR OPEN ENTRY IN RCT
                                        ;
003112  114000  001222          PNGPNG: CLR     WRFLG                   ;CLEAR WRAP FLAG
003114  114002                  PNGBLK: CLR     R2                      ;FOR FLOP SET
003115  104303  000736                  MOV     OFFSET,R3               ;GET OFFSET
003117  104035                  PAGAIN: MOV     R3,R5                   ;MOV OFFSET INTO BUFF POINTER
003120  105025                          ADD     R2,R5                   ;ADD FLOP VALUE
003121  102205  000400                  BIT     #BIT8,R5                ;PAST ONE END (OR BOTH)
003123  053136                          BNE     FLIP                    ;YUP - FLIP OTHER DIRECTION
003124  104651  006622                  MOV     RCTBUF+1(R5),R1         ;GET HEADER CODE
003126  103201  007777                  BIC     #LO,R1                  ;CLEAR LOW ORDER STUFF
003130  106201  100000                  CMP     #RC.NUL,R1              ;END OF RCT ?
003132  013173                          BEQ     EORCT                   ;YUP - WRAP TO FIRST BLOCK
003133  106201  000000                  CMP     #RC.FRE,R1              ;FREE ?
003135  013205                          BEQ     PRET                    ;YUP - ALL DONE
003136  104025                  FLIP:   MOV     R2,R5                   ;GET FLIP VALUE
003137  114002                          CLR     R2
003140  107052                          SUB     R5,R2                   ;NEGATE IT
003141  073144                          BMI     NOINC                   ;IF NEGATIVE DON'T INC
003142  105202  000002                  ADD     #2,R2                   ;ADD TO NEXT VALUE
003144  106202  000400          NOINC:  CMP     #256.,R2                ;DONE EVERY SLOT IN BLOCK ?
003146  053117                          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
                                        ;
003147  115400  001477                  INC     RCTCNT                  ;INC TO NEXT ONE
003151  104303  001477          PNGRD:  MOV     RCTCNT,R3               ;FOR STORE
003153  104204  000731                  MOV     #DDUMMY,R4              ;FOR ADD
003155  100143                          MOV     R3,(R4)                 ;STORE BLOCK NUMBER
003156  114000  000732                  CLR     DDUMMY+1                ;FOR RESETS
003160  114000  000736                  CLR     OFFSET                  ;MAKE IT AT ZERO
003162  104203  001227                  MOV     #HOLD,R3                ;POINT TO FIRST RCT LBN
003164  021506                          CALL    DADD                    ;GET LBN OF THIS RCT BLOCK
003165  104040  001253                  MOV     R4,BUFPNT               ;STORE POINTER TO BLOCK NUMBER
003167  104201  000055                  MOV     #H1,R1                  ;RCT READ OVERLAY
003171  022444                          CALL    PAGE                    ;DO IT
003172  003114                          BR      PNGBLK                  ;SEARCH THIS BLOCK
003173  104303  001222          EORCT:  MOV     WRFLG,R3                ;GET WRAP FLAG
003175  053206                          BNE     PERR                    ;IF BEEN HERE ONCE THEN RCT FULL
003176  104200  000002  001477          MOV     #2,RCTCNT               ;FOR FIRST RCT BLOCK
003201  104200  000002  001222          MOV     #2,WRFLG                ;MAKE WRAP FLAG NON-ZERO
003204  003151                          BR      PNGRD                   ;READ IT AND CONTINUE
                                        ;
                                        ;
003205  000000                  PRET:   RETURN                          ;SUCCESSFUL RETURN
                                                                        ;R5=OFFSET
003206  104201  000020          PERR:   MOV     #16.,R1                 ;RCT FULL
003210  114002                          CLR     R2                      ;NO SUBCODE
003211  022552                          CALL    ERRMNT                  ;ERROR RETURN
```

```
                                        ;
                                        ;
                                        ;       THIS OVERLAY UPDATES THE RCT WITH THE SECTORS
                                        ;       IN THE ERRBUF BUFFER
                                        ;
                                        ;
003212                                  START2:
003212  104200  000014  001153  FCTRCT: MOV     #F5,CUROVL              ;GET OVERLAY NUMBER
003215  104303  001112                  MOV     CURTRK,R3               ;GET CURRENT TRACK
003217                                  PUSH    R3                      ;SAVE IT
003220  104303  001125                  MOV     CYLNUM,R3               ;GET LOW ORDER CYLINDR
003222                                  PUSH    R3                      ;SAVE FOR RESTORE
003223  104303  001126                  MOV     CYLNUM+1,R3             ;GET HIGH ORDER
003225                                  PUSH    R3                      ;SAVE FOR RESTORE
003226  104204  001253                  MOV     #FCTCNT,R4              ;FOR SUB
003230  104203  001256                  MOV     #ONE,R3                 ;DITTO
003232  021524                          CALL    DSUB                    ;SUB TO GET CURRENT FT BLOCK NUM
003233  104143                          MOV     (R4),R3                 ;GET IT
003234                                  PUSH    R3                      ;FOR LATER RESTORE
003235  114000  001256                  CLR     FCTCNT                  ;CLEAR FOR INIT
003237  104200  004535  001253          MOV     #RDBUF,BUFPNT           ;POINT TO BUFFER
003242  104201  000017                  MOV     #F6,R1                  ;READ A BLOCK OF THE FCT
003244  022444                          CALL    PAGE                    ;EXECUTE IT
003245  104204  004535                  MOV     #RDBUF,R4               ;MAKE SURE POINT TO IT
003247  104640  000016  001262          MOV     C512(R4),MNCNT          ;GET COUNT OF USED ENTRIES
003252  104200  000200  001454          MOV     #128.,SECCNT            ;ENTRIES IN A FCT SECTOR
003255  104300  001133  001227          MOV     LBNLBN,HOLD             ;GET LOW ORDER COUNT OF LBN'S
003260  104300  001134  001230          MOV     LBNLBN+1,HOLD+1         ;GET HIGH ORDER
003263  104203  001475                  MOV     #TOTRCT,R3              ;FOR SUBTRACT
003265  104204  001227                  MOV     #HOLD,R4                ;DITTO
003267  021524                          CALL    DSUB                    ;GET STARTING RCT LBN
003270  104201  000022                  MOV     #F7,R1                  ;RCT INIT OVERLAY
003272  022444                          CALL    PAGE                    ;INIT RCT
003273  104303  001262                  MOV     MNCNT,R3                ;GET COUNT
003275  013762                          BEQ     FCTSP                   ;QUIT IF NO ENTRIES
003276  104200  004535  001253  FBEGIN: MOV     #RDBUF,BUFPNT           ;POINT TO BUFFER
003301  104201  000017                  MOV     #F6,R1                  ;FCT READ OVERLAY
003303  022444                          CALL    PAGE                    ;DO IT
003304  104200  004535  001255          MOV     #RDBUF,FCTPTR           ;MAKE SURE POINT TO TI
003307  104304  001255                  MOV     FCTPTR,R4               ;FOR USE
003311  104140  001110          FBEG2:  MOV     (R4),CURPBN             ;GET LOW ORDER PBN
003313  104640  000001  001111          MOV     1(R4),CURPBN+1          ;GET HIGH ORDER
003316  103200  170000  001111          BIC     #HD.CLR,CURPBN+1        ;CLEAR THE FCT HEADER CODE
003321  104203  001110                  MOV     #CURPBN,R3              ;FOR COMAPRE
003323  104204  001064                  MOV     #SCR+LBNHOST,R4         ;DITTO
003325  021626                          CALL    DCMP                    ;IN RCT ??
003326  033720                          BPL     XYZ1                    ;YUP
003327  104201  000044                  MOV     #G5,R1                  ;OVERLAY TO CONVERT FORM PBN TO OTHER BN
003331  022444                          CALL    PAGE                    ;EXECUTE IT
003332  104303  001114                  MOV     CURBN+1,R3              ;GET HIGH ORDER CONVERTED BLOCK NUM
003334  103203  007777                  BIC     #LO,R3                  ;CLEAR ALL BUT HEADER
003336  106203  000000                  CMP     #HD.LBN,R3              ;IS IT AN LBN ?
003340  053510                          BNE     NOLBN                   ;NO - SKIP DOWN
003341  104204  001113                  MOV     #CURBN,R4               ;POINT TO BLOCK NUMBER
003343  023050                          CALL    HASH                    ;COMPUTE RCT ENTRY
003344  104143                          MOV     (R4),R3                 ;GET RCT BLOCK
003345  105203  000002                  ADD     #2,R3                   ;ADD TO GET BY FIRST 2 BLOCKS
```

```
 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,BUFPNT               ;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  000734          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.PRV,R3              ;MARK AS PRIMARY
 82 003421  100653  000001                  MOV     R3,1(R5)                ;STORE IT
 83 003423  103200  000400  001217          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  001115          MOV     DDUMMY,CURXBN           ;SAVE RCT BLOCK NUMBER
 92 003445  104300  000732  001116          MOV     DDUMMY+1,CURXBN+1       ;DITTO
 93 003450  024015                          CALL    FIXFCT                  ;YUP - GOT SOME GYRATIONS TO DO
 94 003451  104300  001115  000731          MOV     CURXBN,DDUMMY           ;RESTORE RCT BLOCK NUMBER
 95 003454  104300  001116  000732          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 CHARNGED 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,BUFPNT           ;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                    CMP     #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     0(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 UNUSUABLE
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                    FCTSLP: 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.PRV,R3      ;MARK AS PRIMARY
185 003701  100653  000001                  MOV     R3,1(R5)        ;STORE IN RCT
186 003703  102200  000400  001217          BIT     #DLL.FLAG       ;DID WE CREATE THE FCT ?
187 003706  013717                          BEQ     XYZ             ;NO - THEN DON'T CHANGE IT
188 003707  104303  001255                  MOV     FCTPTR,R3       ;GET POINTER TO FCT ENTRY
189 003711  104634  000001                  MOV     1(R3),R4        ;GET HIGH ORDER
190 003713  101204  100000                  BIS     #PRMY,R4        ;SET AS PRIMARY
191 003715  100634  000001                  MOV     R4,1(R3)        ;STORE BACK
192 003717  024056                  XYZ:    CALL    RCTWRT          ;WRITE OUT BUFFER
193 003720  105200  000002  001255  XYZ1:   ADD     #2,FCTPTR       ;POINT TO TO NEXT ENTRY
194 003723  117400  001262                  DEC     MNCNT           ;DEREMENT IT
195 003725  013750                          BEQ     FRDONE          ;IF ZERO THEN DONE
196 003726  117400  001454                  DEC     SECCNT          ;DECREMENT IT
197 003730  053744                          BNE     FRSKP           ;IF STILL IN BLOCK - CONTINUE
198 003731  102200  000400  001217          BIT     #DLL,FLAG       ;DID WE CREATE THE FCT ?
199 003734  013276                          BEQ     FBEGIN          ;NO - THEN DON'T CHANGE IT
200 003735  104200  004535  001253          MOV     #RDBUF,BUFPNT   ;POINT TO BUFFER
201 003740  104201  000030                  MOV     #F9,R1          ;FCT WRITE OVERLAY
202 003742  022444                          CALL    PAGE            ;EXECUTE IT
203 003743  003276                          BR      FBEGIN          ;AND GET A NEW ONE
204 003744  104304  001255          FRSKP:  MOV     FCTPTR,R4       ;ELSE GET CURRENT POINTER
205 003746  060022                          XFC     UPDATE          ;LET HOST KNOW STILL ALIVE
206 003747  003311                          BR      FBEG2           ;AND DO NEXT ENTRY
207 003750  102200  000400  001217  FRDONE: BIT     #DLL,FLAG       ;DID WE CREATE THE FCT ?
208 003753  013762                          BEQ     FCTSP           ;NO - THEN DON'T CHANGE IT
209 003754  104200  004535  001253          MOV     #RDBUF,BUFPNT   ;POINT TO BUFFER
210 003757  104201  000030                  MOV     #F9,R1          ;WRITE LAST FCT BLOCK
211 003761  022444                          CALL    PAGE            ;DO IT
212 003762                          FCTSP:  POP     R3              ;GET CURRENT FCT BLOCK NUM
213 003763  104030  001256                  MOV     R3,FCTCNT       ;RESTORE IT
214 003765  104200  005152  001253          MOV     #PBNBUF,BUFPNT  ;RE-READ IT IN CASE OF HEADER CHANGES
215 003770  104201  000017                  MOV     #F6,R1          ;FCT READ OVERLAY
216 003772  022444                          CALL    PAGE            ;DO IT
217 003773                                  POP     R3              ;GET HIGH ORDER CYLINDER
218 003774  104030  001126                  MOV     R3,CYLNUM+1     ;STORE IT
219 003776  104030  001077                  MOV     R3,ISEEK+2      ;STORE IN SEEK COMMAND
220 004000                                  POP     R3              ;GET LOW ORDER
221 004001  104030  001125                  MOV     R3,CYLNUM       ;RESTORE IT
222 004003  104030  001076                  MOV     R3,ISEEK+1      ;STORE IN SEEK COMMAND
223 004005                                  POP     R3              ;GET CURRENT TRACK
224 004006  104030  001112                  MOV     R3,CURTRK       ;RETORE IT
225 004010  104300  001463  001100          MOV     CURGRP,ISEEK+3  ;RESTORE GROUP
226 004013  022251                          CALL    SEEK            ;RESTORE TO PREVIOUS CYLINDER
227 004014  000000                          RETURN
```

```
                                                ;
                                                ;       THIS ROUTINE FIXES UP THE PREVIOUS BLOCK
                                                ;       OF THE FCT. IT CLEARS THE PRIMARY FLAG OF THE
                                                ;       LAST ENTRY.
                                                ;
004015  104204  001256          FIXFCT: MOV     #FCTCNT,R4              ;FOR SUBTRACT
004017  104203  001470                  MOV     #TWOC,R3                ;DOUBLE WORD OF 2
004021  021524                          CALL    DSUB                    ;SUBTRACT TO GET BACK TO RIGHT NUM
004022  104200  007275  001253          MOV     #IMAGE,BUFPNT           ;USE IMAGE BUFFER
004025  104201  000017                  MOV     #F6,R1                  ;FCT READ OVERLAY
004027  022444                          CALL    PAGE                    ;READ THE BLOCK
004030  104203  007275                  MOV     #IMAGE,R3               ;POINT TO BUFFER
004032  105203  000377                  ADD     #255.,R3                ;POINT TO LAST ENTRY
004034  104632  000001                  MOV     1(R3),R2                ;GET HIGH ORDER
004036  103202  100000                  BIC     #PRMY,R2                ;CLEAR FLAG
004040  100632  000001                  MOV     R2,1(R3)                ;STORE IT BACK
004042  104200  007275  001253          MOV     #IMAGE,BUFPNT           ;STORE BUFFER POINTER
004045  104201  000030                  MOV     #F9,R1                  ;FCT WRITE OVERLAY
004047  022444                          CALL    PAGE                    ;WRITE IT BACK OUT
004050                                  DJBINC  FCTCNT                  ;GET FCTCNT BACK TO NORMAL
004055  000000                          RETURN
```

```
                                        ;
                                        ;       WRITE AN RCT BLOCK
                                        ;       DDUMMY    BLOCK NUMBER TO BE WRITTEN
                                        ;
004056  114005                  RCTWRT: CLR     R5                      ;CLEAR ERROR COUNTER
004057  104050  001245                  MOV     R5,NEXT1                ;INIT NEXT COPY COUNTER
004061  104204  000731                  MOV     #DDUMMY,R4              ;POINT TO RCT LBN
004063  104203  001052          RCTRLP: MOV     #SCR,R3                 ;POINT TO CHARACTERISTICS
004065  104207  001430                  MOV     #CONBLK,R0              ;POINT TO CONVERT BLOCK
004067  104632  000011                  MOV     LBNTRK(R3),R2           ;GET LBN/TRACK
004071  103202  177400                  BIC     #HIBYTE,R2              ;CLEAR REST OF WORD
004073  100672  000004                  MOV     R2,V3(R0)               ;FOR CONVERT
004075  104632  000001                  MOV     STCYL(R3),R2            ;STARTING CLYLINDER
004077  103202  007777                  BIC     #LO,R2                  ;CLEAR REST OF WORD
004101  100672  000001                  MOV     R2,V1+1(R0)             ;STORE
004103  114002                          CLR     R2                      ;FOR STORE
004104  100672  000000                  MOV     R2,V1(R0)               ;LOW ORDER ALWAYS 0
004106  022714                          CALL    CVTSK                   ;CONVERT AND SEEK
004107  104207  000721                  MOV     #WRBLK,R0               ;POINT TO COMMAND BLOCK
004111  104203  122400                  MOV     #WRCMD,R3               ;GET WRITE COMMAND
004113  104302  001112                  MOV     CURTRK,R2               ;GET CURRENT TRACK
004115  101023                          BIS     R2,R3                   ;SET TRACK FOR WRITE
004116  100673  000004                  MOV     R3,RW.CMD(R0)           ;STORE IN COMMAND BLOCK
004120  104202  006621                  MOV     #RCTBUF,R2              ;POINT TO BUFFER
004122  022627                          CALL    CEDC                    ;COMPUTE EDC - RETURNED IN R3
004123  100623  000400                  MOV     R3,RW.EDC(R2)           ;STORE IT
004125  100672  000001                  MOV     R2,RW.BUF(R0)           ;STICK IN COMMAND BLOCK
004127  104143                          MOV     (R4),R3                 ;GET LOW ORDER HEADER
004130  100673  000002                  MOV     R3,RW.LOW(R0)           ;STORE IN WRITE BLOCK
004132  104643  000001                  MOV     1(R4),R3                ;GET HIGH ORDER
004134  105303  001335                  ADD     ST.LBN,R3               ;ADD STARTING LBN BITS
004136  101203  000000                  BIS     #HD.LBN,R3              ;SET HEADER
004140  100673  000003                  MOV     R3,RW.HI(R0)            ;STORE IN WRITE BLOCK
004142  104203  000726                  MOV     #HSLIM-1,R3             ;GET DUMMY SDI POINTER
004144  100673  000005                  MOV     R3,RW.DUM(R0)           ;POINT IN COMMAND BLOCK
004146  104303  001333          WRITE3: MOV     HPREA,R3                ;GET HEADER PREAMBLE
004150  104304  001334                  MOV     DPREA,R4                ;GET DATA PREAMBLE
004152  104302  000740                  MOV     UNIT,R2                 ;SET UNIT
004154  104207  000721                  MOV     #WRBLK,R0               ;MAKE SURE POINTING AT BLOCK
004156  101207  100000                  BIS     #BIT15,R0               ;SET NO REVECTORING
004160  060012                          XFC     SIP                     ;WAIT FOR SECTOR PULSE
004161  060003                          XFC     WRITE                   ;WRITE SECTOR
004162  115001                          TST     R1                      ;ANY ERROR ?
004163  014205                          BEQ     RWGOOD                  ;NOPE
004164  106300  001502  001504          CMP     RETRY,TMPTRY            ;MAX ?
004167  014173                          BEQ     1$                      ;YES - TRY SOME RECOVERY
004170  115400  001504                  INC     TMPTRY                  ;INC RETRY COUNT
004172  004146                          BR      WRITE3                  ;DO RETRY
004173  104303  001505          1$:     MOV     RECTMP,R3               ;GET CURRENT ERROR RECOVERY LEVEL
004175  074204                          BMI     2$                      ;IF NEGATIVE THEN FRIED
004176  022612                          CALL    ERRHND                  ;TRY RECOVERY
004177  114000  001504                  CLR     TMPTRY                  ;FOR INIT
004201  117400  001505                  DEC     RECTMP                  ;DECREMENT IT
004203  004146                          BR      WRITE3                  ;RETRY
004204                          2$:
004204  115405                          INC     R5                      ;YUP - INCREMENT COUNTER
004205  115400  001245          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 REAPEATS ?
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)
004250                                  DMOVLY  F7,START
                                ;
                                ;       INITIALIZE RCT TO ALL UNUSED
                                ;
003047  104200  000014  001153  RCTINI: MOV     #F5,CUROVL              ;FOR OVERLAY INIT
003052  104203  000400                  MOV     #256.,R3                ;SECTOR WORD COUNT
003054  114001                          CLR     R1                      ;FOR INIT OF RCT WORDS
003055  104204  004535                  MOV     #RDBUF,R4               ;BUFFER
003057  100241                  RCLP:   MOV     R1,(R4)+                ;STORE IN BUFFER
003060  117403                          DEC     R3                      ;DECREMENT COUNTER
003061  053057                          BNE     RCLP                    ;BRANCH BACK TILL DONE
003062  104201  001305                  MOV     #SERNUM,R1              ;POINT TO SERIAL NUMBER
003064  104204  004535                  MOV     #RDBUF,R4               ;POINT TO BUFFER
003066  105204  000000                  ADD     #RSER,R4                ;POINT TO SERIAL NUMBER
003070  104205  000004                  MOV     #4,R5                   ;COUNTER
003072  104212                  6$:     MOV     (R1)+,R2                ;GET WORD
003073  100242                          MOV     R2,(R4)+                ;STORE WORD
003074  117405                          DEC     R5                      ;DECREMENT COUNTER
003075  053072                          BNE     6$                      ;CONTINUE TILL DONE
003076  104303  001112                  MOV     CURTRK,R3               ;GET CURRENT TRACK
003100                                  PUSH    R3                      ;SAVE IT
003101  104303  001125                  MOV     CYLNUM,R3               ;GET LOW ORDER CYLINDR
003103                                  PUSH    R3                      ;SAVE FOR RESTORE
003104  104303  001126                  MOV     CYLNUM+1,R3             ;GET HIGH ORDER
003106                                  PUSH    R3                      ;SAVE FOR RESTORE
003107  104300  001135  000736          MOV     RBNLBN,TEMP             ;GET NUMBER OF RBN'S IN LBN AREA
003112  104300  001136  000737          MOV     RBNLBN+1,TEMP+1         ;HIGH ORDER
003115  104204  000736                  MOV     #TEMP,R4                ;FOR ADD
003117  104200  000400  000731          MOV     #256.,DDUMMY            ;2 BLOCKS(CONTROL) WORTH OF RBN'S
003122  114000  000732                  CLR     DDUMMY+1                ;CLEAR HIGH ORDER
003124  104203  000731                  MOV     #DDUMMY,R3              ;FOR ADD
003126  021506                          CALL    DADD                    ;ADD TO GET "REAL" NUMBER OF RBN'S
003127  104300  001133  001227          MOV     LBNLBN,HOLD             ;GET LOW ORDER COUNT OF LBN'S
003132  104300  001134  001230          MOV     LBNLBN+1,HOLD+1         ;GET HIGH ORDER
003135  104203  001475                  MOV     #TOTRCT,R3              ;FOR SUBTRACT
003137  104204  001227                  MOV     #HOLD,R4                ;DITTO
003141  021524                          CALL    DSUB                    ;GET STARTING RCT LBN
003142  104300  001227  001113          MOV     HOLD,CURBN              ;GET STARTING RCT BLOCK NUMBER
003145  104300  001227  001115          MOV     HOLD,CURLBN             ;ALSO SAVE
003150  104300  001230  001114          MOV     HOLD+1,CURBN+1          ;GET HIGH ORDER
003153  104300  001230  001116          MOV     HOLD+1,CURLBN+1         ;AND SAVE
003156  114000  001501                  CLR     COUNT                   ;CLEAR BLOCK COUNTER
003160  104203  001052                  MOV     #SCR,R3                 ;POINT TO CHARACTERISTICS
003162  104207  001430                  MOV     #CONBLK,R0              ;POINT TO CONVERT BLOCK
003164  104632  000011                  MOV     LBNTRK(R3),R2           ;GET LBN/TRACK
003166  103202  177400                  BIC     #HIBYTE,R2              ;CLEAR HIGH BYTE
003170  100672  000004                  MOV     R2,V3(R0)               ;FOR CONVERT
003172  104632  000001                  MOV     STCYL(R3),R2            ;STARTING CLYLINDER
003174  103202  007777                  BIC     #LO,R2                  ;CLEAR REST OF WORD
003176  100672  000001                  MOV     R2,V1+1(R0)             ;STORE
003200  114002                          CLR     R2                      ;FOR STORE
003201  100672  000000                  MOV     R2,V1(R0)               ;LOW ORDER ALWAYS 0
003203  114005                  RCINLP: CLR     R5                      ;CLEAR ERROR COUNTER
003204  104050  001245                  MOV     R5,NEXT1                ;INIT COPY COUNT
003206  104204  001113          RCLP2:  MOV     #CURBN,R4               ;FOR CONVERT
003210  022714                          CALL    CVTSK                   ;CONVERT AND SEEK
```

```
 58 003211  104207  000721                MOV     #WRBLK,R0       ;POINT TO COMMAND BLOCK
 59 003213  104205  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 003341  104200  000200  000731        MOV     #128.,DDUMMY    ;SUBTRACT ONE BLOCKS WORTH
109 003346  114000  000732                CLR     DDUMMY+1        ;FOR CLEAR
110 003352  104203  000731                MOV     #DDUMMY,R3      ;FOR SUBTRACT
111 003356  021524                        CALL    DSUB            ;SUBTRACT
112 003357  021626                        CALL    DCMP            ;IN LAST BLOCK ?
113 003360  073353                        BMI     4$              ;NOPE
114 003361  013353                        BEQ     4$              ;IF EQUAL - NO PARTIAL BLOCK
```

```
115 003352 023466                          CALL    FIXBLK          ;YES - CHANGE HEADERS TO NULL
116 003355 115000 001501           4$:     TST     COUNT           ;ON FIRST BLOCK ?
117 003355 053366                          BNE     7$              ;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                  RCLP4:  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     R3              ;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       ;RETORE 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
```

```
                                ;
                                ;
                                ;       CHANGE UNUSED ENTRIES TO NULL HEADERS
                                ;
                                ;
003466  114002                FIXBLK: CLR     R2                      ;FOR LOW ORDER
003467  114005                        CLR     R5                      ;FOR HIGH ORDER
003470  104303  000736                MOV     TEMP,R3                 ;GET REMAINDER FROM TEMP
003472  101205  100000                BIS     #RC.NUL,R5              ;SET IN NEW HEADER CODE
003474  104204  004535                MOV     #RDBUF,R4               ;POINT TO BUFFER
003476  104201  000200                MOV     #128.,R1                ;TOTAL COUNT
003500  107031                        SUB     R3,R1                   ;SUBTRACT USED ENTRIES
003501  105033                        ADD     R3,R3                   ;ADD TO GET OFFSET (MULT BY 2)
003502  105034                        ADD     R3,R4                   ;POINT TO FIRST UNUSED ENTRY
003503  100242                FIXLP:  MOV     R2,(R4)+                ;STORE LOW ORDER
003504  100245                        MOV     R5,(R4)+                ;STORE HIGH ORDER
003505  117401                        DEC     R1                      ;DECREMENT COUNT
003506  053503                        BNE     FIXLP                   ;LOOP TILL DONE
003507  101200  000040  001217       BIS     #RCINIT,FLAG            ;SET DONE IT
003512  000000                        RETURN
```

```
                                        .SBTTL  FCT READ OVERLAY (F6)
003513                                  DMOVLY  F6,START
                                ;
                                ;       READ A BLOCK OF THE FCT
                                ;       R5 -> BUFFER
                                ;
003047  104200  000017  001153          MOV     #F6,CUROVL              ;OVERLAY NUMBER
003052  104300  001256  000731          MOV     FCTCNT,DDUMMY           ;GET CURRENT COUNT
003055  114000  000732                  CLR     DDUMMY+1                ;FOR HIGH ORDER STORE
003057  114005                          CLR     R5                      ;CLEAR ERROR COUNTER
003060  104204  000731          FOLOOP: MOV     #DDUMMY,R4              ;FOR CONVERT
003062  104303  001143                  MOV     LBNCYL,R3               ;GET LBN CYLINDERS
003064  104207  001430                  MOV     #CONBLK,R0              ;POINT TO CONVERT BLOCK
003066  100673  000000                  MOV     R3,V1(R0)               ;STORE IT FOR CONVERT
003070  104303  001144                  MOV     LBNCYL+1,R3             ;HIGH ORDER
003072  100673  000001                  MOV     R3,V1+1(R0)             ;STORE IT
003074  104303  001127                  MOV     SECTRK,R3               ;GET SECTORS/TRACK
003076  100673  000004                  MOV     R3,V3(R0)               ;STORE FOR CONVERT
003100  022714                          CALL    CVTSK                   ;CONVERT FCT BLOCK NUMBER AND SEEK
003101  104207  000721                  MOV     #RDBLK,R0               ;PREPARE FOR READ SECTORS
003103  104203  000726                  MOV     #HSLIM-1,R3             ;POINTER TO DUMMY SDI BLOCK
003105  100673  000005                  MOV     R3,RW.DUM(R0)           ;STORE IN COMMAND BLOCK
003107  104303  000731                  MOV     DDUMMY,R3               ;LO ORDER BLOCK NUMBER
003111  100673  000002                  MOV     R3,RW.LOW(R0)           ;STORE IN READ COMD BLOCK
003113  104303  000732                  MOV     DDUMMY+1,R3             ;GET HIGH ORDER
003115  105303  001337                  ADD     ST.XBN,R3               ;ADD STARTING LBN BITS
003117  101203  120000                  BIS     #HD.XBN,R3              ;HEADER CODE
003121  100673  000003                  MOV     R3,RW.HI(R0)            ;STORE IN READ CMD BLOCK
003123  104303  001253                  MOV     BUFPNT,R3               ;GET BUFFER POINTER
003125  100673  000001                  MOV     R3,RW.BUF(R0)           ;STORE.BUFFER ADDRESS IN COMMAND BUFFER
003127  104203  013400                  MOV     #RWCMD,R3               ;LOAD SDI READ COMMAND
003131  104301  001112                  MOV     CURTRK,R1               ;GET CURRENT HEAD NUMBER IN R1
003133  101013                          BIS     R1,R3                   ;SET IT IN COMMAND
003134  100673  000004                  MOV     R3,RW.CMD(R0)           ;STORE BACK
003136  104207  000721          READ7:  MOV     #RDBLK,R0               ;MAKE SURE POINTING AT BLOCK
003140  104203  100000                  MOV     #RDCMD,R3               ;MARK AS ONLY REQUEST
003142  100173                          MOV     R3,(R0)                 ;STORE IN CMD BLOCK
003143  104302  000740                  MOV     UNIT,R2                 ;SDI INTERCONNECT
003145  101207  100000                  BIS     #BIT15,R0               ;SET NO REVECTORING
003147  060012                          XFC     SIP                     ;WAIT FOR PULSE
003150  060002                          XFC     READ                    ;READ 1 SECTOR
003151  115001                          TST     R1                      ;ANY ERRORS ?
003152  053170                          BNE     100$                    ;YES - TRY RECOVERY
003153  104173                          MOV     (R0),R3                 ;GET STATUS WORD
003154  102203  010000                  BIT     #ECCF,R3                ;ECC ERROR ?
003156  013162                          BEQ     101$                    ;NOPE - VERIFY EDC
003157  023027                          CALL    ECCCK                   ;CORRECT ECC
003160  115001                          TST     R1                      ;TEST FLAG
003161  053170                          BNE     100$                    ;UNCORRECTABLE
003162  104302  001253          101$:   MOV     BUFPNT,R2               ;POINT TO BUFFER
003164  022627                          CALL    CEDC                    ;COMPUTE EDC
003165  106623  000400                  CMP     RW.EDC(R2),R3           ;O.K. ?
003167  013227                          BEQ     102$                    ;YUP - CONSIDER GOOD
003170  106300  001502  001504  100$:   CMP     RETRY,TMPTRY            ;MAX ?
003173  013177                          BEQ     1$                      ;YES - TRY SOME RECOVERY
003174  115400  001504                  INC     TMPTRY                  ;INC RETRY COUNT
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      READ7           ;RETRY
 64 003210                          2$:
 65 003210  115405                          INC     R5              ;INCREMENT BAD COUNTER
 66 003211  106305  001244                  CMP     FCTCPY,R5       ;ALL BAD ?
 67 003213  013344                          BEQ     OFATAL          ;YUP - ALL OVER
 68 003214  104204  000731                  MOV     #DDUMMY,R4      ;POINT TO COUNT
 69 003216  104203  001240                  MOV     #FCTFMT,R3      ;SIZE OF TABLE
 70 003220  021506                          CALL    DADD            ;ADD TO POINT TO NEXT COPY
 71 003221  114000  001504                  CLR     TMPTRY          ;RESET RETRY LEVEL
 72 003223  104300  001503  001505          MOV     RECOV,RECTMP    ;DITTO RECOVERY LEVELS
 73 003226  003060                          BR      FOLOOP          ;BRANCH BACK
 74 003227                          102$:
 75 003227  114000  001504          FODONE: 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     BUFPNT,R3       ;GET BUFFER ADDDRESS
 89 003257  100673  000007                  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          WRITE8: 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 FRO 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                    OQUIT:  MOV     R1,R2           ;XFC ERROR CODE
130 003363  1042C1  000021                    MOV     #17.,R1         ;FCT READ ERROR
131 003365  022552                            CALL    ERRMNT          ;ERROR RETURN
```

```
                                        .SBTTL  GET FCT BLOCK FOR D/XBN FORMAT (G2) -
                                ;
                                ;
                                ;       GET'S ONE FCT BLOCK FROM HOST FOR D/BN FORMATTING
                                ;
                                ;
003366                          ;       DMOVLY  G2,START
                                ;
                                ;
003047  104200  000033  001453          MOV     #G2.CUROVL              ;SIGNAL OVERLAY 11
003052  104205  001263                  MOV     #DMBUF,R5               ;POINT TO MAINT BUFFER
003054  104303  001321                  MOV     FCMSG,R3                ;GET DUP CODE
003056  100153                          MOV     R3,(R5)                 ;STORE IT
003057  104303  001322                  MOV     FCMSG+1,R3              ;GET 'F ' IDENTIFIER
003061  100653  000001                  MOV     R3,1(R5)                ;STORE IT IN MESSAGE
003063  104303  001256                  MOV     FCTCNT,R3               ;GET BLOCK NUMBER DESIRED
003065  100653  000002                  MOV     R3,2(R5)                ;STORE IT
003067  022536                          CALL    SNDMNT                  ;SEND REQUEST
003070  022544                          CALL    RCVMNT                  ;RECEIVE ANSWER
003071  104653  000002                  MOV     2(R5),R3                ;GET BLOCK COUNT RECEIVED
003073  013113                          BEQ     DLERR                   ;ERROR IF ZERO
003074  104653  000003                  MOV     3(R5),R3                ;GET HOST ADDRESS
003076  104030  001425                  MOV     R3,OVLBLK+1             ;STORE IN OVERLAY BLOCK
003100  114000  001426                  CLR     OVLBLK+2                ;ZERO HIGH ORDER
003102  104200  000401  001424          MOV     #257.,OVLBLK            ;GET LENGTH
003105  104204  001424                  MOV     #OVLBLK,R4              ;FOR OVERLAY ROUTINE
003107  104203  005152                  MOV     #PBNBUF,R3              ;POINT TO BUFFER
003111  022526                          CALL    OVRLAY                  ;GET THE SECTOR
003112  000000                          RETURN
003113  104201  000023          DLERR:  MOV     #19.,R1                 ;SIGNAL DLL ERROR
003115  104302  001256                  MOV     FCTCNT,R2               ;BLOCK FAILED ON
003117  022552                          CALL    ERRMNT                  ;ERROR RETURN
```

```
                                        .SBTTL  GET FCT BLOCK FOR LBN FORMAT (G3)
                                ;
                                ;
                                ;       GET RIGHT FCT BLOCK FOR LBN FORMATTING
                                ;
                                ;
003120                          ;       DMOVLY  G3,START
                                ;
                                ;
                                ;
003047  104200  000036  001153          MOV     #G3,CUROVL              ;FOR CURRENT OVERLAY
003052  114000  001256                  CLR     FCTCNT                  ;FOR FIRST FCT BLOCK
003054  104200  005152  001253          MOV     #PBNBUF,BUFPNT          ;POINT TO BUFFER
003057  104201  000017                  MOV     #F6,R1                  ;FCT READ OVERLAY
003061  022444                          CALL    PAGE                    ;READ IT IN
003062  102200  000001  001217          BIT     #FCTAVL,FLAG            ;FCT STILL HERE ?
003065  013243                          BEQ     NGD                     ;NOPE - CAN IT
003066  104207  005152                  MOV     #PBNBUF,R0              ;POINT TO BUFFER
003070  104173                          MOV     (R0),R3                 ;GET FORMAT MEDIA WORD
003071  106203  126736                  CMP     #M512,R3                ;IS IT 512 ?
003073  013101                          BEQ     13$                     ;YUP - O.K.
003074  106203  074161                  CMP     #M576,R3                ;IS IT 576 ?
003076  013101                          BEQ     13$                     ;YUP - O.K.
003077  115003                          TST     R3                      ;IS IT FORMAT IN PROGRESS
003100  053243                          BNE     NGD                     ;NOPE - FCT NO GOOD
003101  104673  000025          13$:    MOV     FCTFLG(R0),R3           ;GET FLAG WORD
003103  102203  100000                  BIT     #NOFCT,R3               ;IS THERE REALLY AN FCT ??
003105  053243                          BNE     NGD                     ;NOPE - FIND OUT IF WE QUIT OR ROUGH IT
003106  104673  000016                  MOV     C512(R0),R3             ;GET COUNT OF USED ENTRIES
003110  104030  001237                  MOV     R3,FCNT                 ;STORE IT
003112  053116                          BNE     12$                     ;IF NOT ZERO THEN ENTRIES EXIST
003113  101200  000002  001217          BIS     #FCTEMT,FLAG            ;SET EMPTY FLAG
003116  114003                  12$:    CLR     R3                      ;FOR FCT INIT
003117  100173                          MOV     R3,(R0)                 ;SIGNAL FORMAT IN PROGRESS
003120  104203  005152                  MOV     #PBNBUF,R3              ;POINT TO BUFFER
003122  105203  000002                  ADD     #FSER,R3                ;POINT TO SERIAL NUMBER
003124  104204  001305                  MOV     #SERNUM,R4              ;SERIAL NUMBER BLOCK
003126  104205  000004                  MOV     #4,R5                   ;COUNTER
003130  104232                  8$:     MOV     (R3)+,R2                ;GET WORD
003131  100242                          MOV     R2,(R4)+                ;STORE WORD
003132  117405                          DEC     R5                      ;DECREMENT COUNTER
003133  053130                          BNE     8$                      ;CONT TILL DONE
003134  104200  005152  001253          MOV     #PBNBUF,BUFPNT          ;POINT TO BUFFER
003137  104201  000030                  MOV     #F9,R1                  ;FCT WRITE OVERLAY
003141  022444                          CALL    PAGE                    ;DO IT
003142  115000  001237                  TST     FCNT                    ;ANY ENTRIES ?
003144  013237                          BEQ     RDONE1                  ;NOPE - ALL DONE
003145  104200  000001  001256          MOV     #1,FCTCNT               ;FOR FCT COUNT INIT
003150  104200  005152  001253  RLOOP:  MOV     #PBNBUF,BUFPNT          ;POINT TO BUFFER
003153  104201  000017                  MOV     #F6,R1                  ;FCT READ OVERLAY
003155  022444                          CALL    PAGE                    ;DO THE READ
003156  102200  000001  001217          BIT     #FCTAVL,FLAG            ;STILL HAVE FCT ?
003161  013243                          BEQ     NGD                     ;NOPE - CAN IT
003162  104204  005152                  MOV     #PBNBUF,R4              ;POINT TO THE BUFFER
003164  105204  000376                  ADD     #254.,R4                ;POINT TO LAST ENTRY
003166  104203  001154                  MOV     #HGHPBN,R3              ;HIGHEST PBN IN LBN AREA
003170  021626                          CALL    DCMP                    ;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 00320'  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  00000'                  MOV     R0,1(R4)        ;STORE IT BACK
67 003212  104203  001154                  MOV     #HGHPBN,R3      ;POINT TO HIGHEST PBN
68 003214  021626                          CALL    DCMP            ;COMPARE
69 003215  033233                          BPL     RDONE           ;IF LESS THAN OR EQUAL THEN FOUND FIRST LBBN
70 003216  100641  00000'                  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  00'223                  MOV     R4,BADPBN       ;MAKE CURRENT BAD POINTER
79 003237  104300  001237  0013'1 RDONE': 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 00326'  022553                          CALL    ERRMNT          ;ERROR RETURN
```

```
                                                .SBTTL  RCT CLEANUP OVERLAY (G4)
                                        ;
                                        ;       RCT CLEANUP AND FINAL STATS
                                        ;
003262                                  ;       DMOVLY  G4,START
                                        ;
                                        ;
                                        ;
003047  104200  000041  001153          MOV     #G4,CUROVL              ;FOR OVERLAY IDENT
003052  104207  001263                  MOV     #DMBUF,R0               ;MESSAGE BUFFER
003054  104303  001472                  MOV     SNDCNT,R3               ;ANY SECONDARY REVECTORS ?
003056  100673  000003                  MOV     R3,SND1(R0)             ;STORE IT
003060  115003                          TST     R3                      ;ARE THERE ANY ?
003061  013201                          BEQ     CLSKP3                  ;NOPE - JUST EXIT
003062  104202  006204                  MOV     #REVBUF,R2              ;POINT TO REVECTOR BUFFER
003065  104200  000100  001501          MOV     #64.,COUNT              ;COUNT OF MAX TO REVECTOR AT ONCE
003067  114000  001106                  CLR     CURRBN                  ;CLEAR FOR INIT
003071  114000  001107                  CLR     CURRBN+1                ;HIGH ORDER TOO
003073  104200  000002  001477          MOV     #2,RCTCNT               ;INIT RCT BLOCK
003076  104200  000200  001454  CLELP2: MOV     #128.,SECCNT            ;GET COUNT OF RCT ENTRIES
003101  104304  001477                  MOV     RCTCNT,R4               ;GET BLOCK NUMBER TO READ
003103  023715                          CALL    RRC                     ;READ IT
003104  104205  006621                  MOV     #RCTBUF,R5              ;POINT TO BUFFER
003106  104653  000001          CLELP:  MOV     1(R5),R3                ;GET HEADER
003110  103203  007777                  BIC     #LO,R3                  ;CLEAR OUT LOW GARBAGE
003112  106203  030000                  CMP     #RC.SND,R3              ;IS IT A SECONDARY ?
003114  053144                          BNE     CLESKP                  ;NO - SKIP REVECTORING
003115  104153                          MOV     (R5),R3                 ;GET LOW ORDER
003116  100223                          MOV     R3,(R2)+                ;STORE IN REVECTOR BUFFER
003117  104653  000001                  MOV     1(R5),R3                ;GET HIGH ORDER
003121  103203  170000                  BIC     #HD.CLR,R3              ;CLEAR HEADER
003123  101203  030000                  BIS     #HD.REV,R3              ;SET AS AN LBN REVECTOR
003125  100223                          MOV     R3,(R2)+                ;STORE IT
003126  104303  001106                  MOV     CURRBN,R3               ;GET LOW ORDER RBN NUMBER
003130  100223                          MOV     R3,(R2)+                ;STORE IT
003131  104303  001107                  MOV     CURRBN+1,R3             ;GET HIGH ORDER
003133  100223                          MOV     R3,(R2)+                ;STORE IT
003134  117400  001501                  DEC     COUNT                   ;DEC NUM OF EMPTY REVECTOR SLOTS
003136  117400  001472                  DEC     SNDCNT                  ;DECREMENT IT
003140  013200                          BEQ     CLSKP4                  ;IF ZERO THEN DONE
003141  104303  001501                  MOV     COUNT,R3                ;FULL BLOCK ?
003143  013165                          BEQ     CLSKP2                  ;IF 0 - PROCESS BLOCK
003144                          CLESKP: DUBINC  CURRBN                  ;INCERMENT IT
003151  105205  000002                  ADD     #2,R5                   ;POINT TO NEXT RBN ENTR
003153  117400  001454                  DEC     SECCNT                  ;DECREMENT IT
003155  053106                          BNE     CLELP                   ;DO NEXT ENTRY IF NOT ZERO
003156  115400  001477                  INC     RCTCNT                  ;INCREMENT IT
003160  106300  001242  001477          CMP     RCTFMT,RCTCNT           ;DONE ?
003163  053076                          BNE     CLELP2                  ;NOPE - READ IN NEXT BLOCK
003164  003174                          BR      CLEDON                  ;ELSE DONE
003165  023270                  CLSKP2: CALL    CLEWRT                  ;PROCESS THE BLOCK
003166  104200  000100  001501          MOV     #64.,COUNT              ;FOR COUNTER INIT
003171  104202  006204                  MOV     #REVBUF,R2              ;RESET POINTER
003173  003144                          BR      CLESKP                  ;BRANCH BACK
003174  106200  000100  001501  CLEDON: CMP     #64.,COUNT              ;DOEN ANY ?
003177  013201                          BEQ     CLSKP3                  ;NO - DONE
003200  023270                  CLSKP4: CALL    CLEWRT                  ;WRITE OUT ANY LEFTOVERS
```

```
 58 003201  024222                  CLSKP3: CALL    FCTCK           ;VERIFY FCT
 59 003202  024510                          CALL    RCTCK           ;VERIFY RCT
 60 003203  104207  001263                  MOV     #DMBUF,R0       ;POINT TO DMBUF
 61 003205  104301  001327                  MOV     DONMSG,R1       ;GET DUP CODE
 62 003207  100171                          MOV     R1,(R0)         ;STORE IT
 63 003210  104301  001330                  MOV     DONMSG+1,R1     ;'D ' FLAG
 64 003212  100671  000001                  MOV     R1,1(R0)        ;STORE IT
 65 003214  114002                          CLR     R2              ;FOR CLEAR OF BUFFER
 66 003215  104303  001312                  MOV     LBNBAD,R3       ;GET COUNT OF REVECTORED LBN'S
 67 003217  100673  000002                  MOV     R3,LBD(R0)      ;STORE IN BLOCK
 68 003221  104303  001313                  MOV     RCTBAD,R3       ;RCT BLOCKS BAD
 69 003223  100673  000004                  MOV     R3,RBD(R0)      ;STORE IT
 70 003225  100672  000005                  MOV     R2,RBD+1(R0)    ;CLEAR OTHER ERROR
 71 003227  104303  001314                  MOV     DBBAD,R3        ;DBN BLOCKS BAD
 72 003231  100673  000006                  MOV     R3,DBD(R0)      ;STORE IT
 73 003233  100672  000007                  MOV     R2,DBD+1(R0)    ;CLEAR OTHER COUNT
 74 003235  104303  001315                  MOV     XBBAD,R3        ;XBN LOCKS BAD
 75 003237  100673  000010                  MOV     R3,XBD(R0)      ;STORE IT
 76 003241  100672  000011                  MOV     R2,XBD+1(R0)    ;CLEAR OTHER ERROR
 77 003243  104303  001473                  MOV     RTYCNT,R3       ;RETRY COUNT
 78 003245  100673  000012                  MOV     R3,RTCNT(R0)    ;STORE IT
 79 003247  102200  002000  001277          BIT     #BSTGS,FLAG     ;DID WE USE FCT ?
 80 003252  013260                          BEQ     GDFCT           ;YUP
 81 003253  104202  000002                  MOV     #2,R2           ;NOPE - FLAG AS 2
 82 003255  100672  000013                  MOV     R2,FCT(R0)      ;STORE IT
 83 003257  003264                          BR      ALDONE          ;SKIP GOOD FCT FLAGCIN
 84 003260  104202  000001          GDFCT:  MOV     #1,R2           ;GOOD IS A 1
 85 003262  100672  000013                  MOV     R2,FCT(R0)      ;STORE IT
 86 003264  022536                  ALDONE: CALL    SNDMNT          ;SEND IT
 87 003265  022324                          CALL    DISCON          ;DISCONNECT/SPINDOWN DRIVE
 88 003266  104007                          CLR     R0              ;MAKE SURE QUITS NICELY
 89 003267  060021                          XFC     DONE            ;DONE
```

```
                                                ;
                                                ;       PROCESS REVECTOR BLOCK
                                                ;
003270                                  CLEWRT: PUSHA
003276  104204  006204                        MOV     #REVBUF,R4      ;POINT TO BUFFER
003300  104300  001501  000736                MOV     COUNT,TEMP      ;GET COUNT
003303  104642  000002                  CLHERE: MOV     2(R4),R2        ;GET LOW ORDER RBN
003305  104643  000003                        MOV     3(R4),R3        ;GET HIGH ORDER RBN
003307  101203  060000                        BIS     #HD.RBN,R3      ;SET IN HDR CODE
003311  104205  005152                        MOV     #PBNBUF,R5      ;POINT TO BUFFER
003313  104201  000200                        MOV     #RBNRPT,R1      ;INIT COUNTER
003315  100252                        WLOOP:  MOV     R2,(R5)+        ;STORE LOW ORDER
003316  100253                                MOV     R3,(R5)+        ;STORE HIGH ORDER
003317  ?17401                                DEC     R1              ;DECREMENT COUNTER
003320  053315                                BNE     WLOOP           ;CONTINURE TILL DONE
003321  1042C3  001052                        MOV     #SCR,R3         ;POINT TO CHARACTERISTICS
003323  104632  000011                        MOV     LBNTRK(R3),R2   ;GET LBN/TRACK
003325  103202  177400                        BIC     #HIBYTE,R2      ;CLEAR HIGH BYTE
003327  104207  001430                        MOV     #CONBLK,R0      ;POINT TO CONVERT BLOCK
003331  100672  000004                        MOV     R2,V3(R0)       ;FOR CONVERT
003333  104632  000001                        MOV     STCYL(R3),R2    ;STARTING CLYLINDER
003335  103202  007777                        BIC     #LO,R2          ;CLEAR REST OF WORD
003337  100672  000000                        MOV     R2,V1+1(R0)     ;STORE
003341  114002                                CLR     R2              ;FOR STORE
003342  100672  000000                        MOV     R2,V1(R0)       ;LOW ALWAYS ZERO
003344  023455                                CALL    CS              ;CONVERT AND SEEK
003345  104207  000721                        MOV     #WRBLK,R0       ;POINT TO COMMAND BLOCK
003347  104203  122400                        MOV     #WRCMD,R3       ;GET WRITE COMMAND
003351  104302  001112                        MOV     CURTRK,R2       ;GET CURRENT TRACK
003353  101023                                BIS     R2,R3           ;SET TRACK FOR WRITE
003354  100673  000004                        MOV     R3,RW.CMD(R0)   ;STORE IN COMMAND BLOCK
003356  104202  005152                        MOV     #PBNBUF,R2      ;POINT TO BUFFER
003360  022627                                CALL    CEDC            ;COMPUTE EDC - RETURNED IN R3
003361  100623  000400                        MOV     R3,RW.EDC(R2)   ;STORE IT
003363  100672  000001                        MOV     R2,RW.BUF(R0)   ;STICK IN COMMAND BLOCK
003365  104143                                MOV     (R4),R3         ;GET LOW ORDER HEADER
003366  100673  000002                        MOV     R3,RW.LOW(R0)   ;STORE IN WRITE BLOCK
003370  104643  000001                        MOV     1(R4),R3        ;GET HIGH ORDER
003372  105303  001335                        ADD     ST.LBN,R3       ;ADD STARTING LBN BITS
003374  100673  000003                        MOV     R3,RW.HI(R0)    ;STORE IN WRITE BLOCK
003376  104203  000726                        MOV     #HSLIM-1,R3     ;GET DUMMY SDI POINTER
003400  100673  000005                        MOV     R3,RW.DUM(R0)   ;POINT IN COMMAND BLOCK
003402  104040  000731                        MOV     R4,DDUMMY       ;SAVE R4
003404  104303  001333                WRITE9: MOV     HPREA,R3        ;GET HEADER PREAMBLE
003406  104304  001334                        MOV     DPREA,R4        ;GET DATA PREAMBLE
003410  104302  000740                        MOV     UNIT,R2         ;SET UNIT
003412  104207  000721                        MOV     #WRBLK,R0       ;MAKE SURE POINTING AT BLOCK
003414  101207  100000                        BIS     #BIT15,R0       ;SET NO REVECTORING
003416  060012                                XFC     SIP             ;WAIT FOR SECTOR PULSE
003417  060003                                XFC     WRITE           ;WRITE SECTOR
003420  115001                                TST     R1              ;ANY ERROR ?
003421  013431                                BEQ     1$              ;NO - SKIP RETRY
003422  106200  000010  001504                CMP     #MAXTRY,TMPTRY  ;MAX ?
003425  013431                                BEQ     1$              ;YES - GIVE UP
003426  115400  001504                        INC     TMPTRY          ;INC RETRY COUNT
003430  003404                                BR      WRITE9          ;DO RETRY
003431                                  1$:                           .
```

```
 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     CLHERE                  ;NO - REPEAT
 64 003445  023522                       CALL    RBNWRT                  ;WRITE GOOD EDC'S TO RBN'S
 65 003446                                POPA
 66 003454  000000                       RETURN
 67                                ;
 68                                ;       CONVERT BLOCK NUMBER TO PHYSICAL ADDRESS AND SEEK
 69                                ;       R4 -> BLOCK NUMBER
 70                                ;
 71 003455  104207  001430        CS:     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 XYLINDER
 81 003501  104670  000007  001077        MOV     CYL+1(R0),ISEEK+2       ;HIGH ORDER CYLINDR
 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        CKR:    MOV     #10.,R1                 ;SEEK ERROR
 88 003515  104207  001430                MOV     #CONBLK,R0              ;CONVERT BLOCK
 89 003517  104672  000006                MOV     CYL(R0),R2              ;CYLINDER FAILD ON
 90 003521  022552                       CALL    ERRMNT                  ;ERROR RETURN
```

```
    1 003522  104204  006204          RBNWRT: MOV     #REVBUF,R4      ;POINT TO BUFFER
    2 003524  104203  005567                  MOV     #GDBLK,R3       ;POINT TO GOOD BLOCK
    3 003526  104302  001451                  MOV     EDC,R2          ;GET GOOD EDC
    4 003530  100632  000400                  MOV     R2,RW.EDC(R3)   ;STORE IT
    5 003532  104203  001052                  MOV     #SCR,R3         ;POINT TO CHARACTERISTICS
    6 003534  104632  000004                  MOV     RBNTRK(R3),R2   ;GET RBN/TRACK
    7 003536  103202  177600                  BIC     #HI1BYTE,R2     ;CLEAR HIGH GARBAGE
    8 003540  104207  001430                  MOV     #CONBLK,R0      ;POINT TO CONVERT BLOCK
    9 003542  100672  000004                  MOV     R2,V3(R0)       ;FOR CONVERT
   10 003544  104632  000011                  MOV     LBNTRK(R3),R2   ;GET LBN/TRACK
   11 003546  103202  177400                  BIC     #HIBYTE,R2      ;CLEAR HIGH BYTE
   12 003550  100672  000005                  MOV     R2,V4(R0)       ;SET UP FOR RBN'S
   13 003552  104632  000001                  MOV     STCYL(R3),R2    ;STARTING CLYLINDER
   14 003554  103202  007777                  BIC     #LO,R2          ;CLEAR REST OF WORD
   15 003556  100672  000001                  MOV     R2,V1+1(R0)     ;STORE
   16 003560  114002                          CLR     R2              ;FOR STORE
    7 003561  100672  000000                  MOV     R2,V1(R0)       ;LOW ALWAYS ZERO
   18 003563  104640  000002  000733  RNWHER: MOV     2(R4),TEMP2     ;GET LOW ORDER RBN
   19 003566  104640  000003  000734          MOV     3(R4),TEMP2+1   ;GET HIGH ORDER
   20 003571  104040  000731                  MOV     R4,DDJMMY       ;SAVE R4
   21 003573  104204  000733                  MOV     #TEMP2,R4       ;FOR CONVERT
   22 003575  022714                          CALL    CVTSK           ;CONVERT AND SEEK
   23 003576  104207  000721                  MOV     #WRBLK,R0       ;POINT TO COMMAND BLOCK
   24 003600  104203  122400                  MOV     #WRCMD,R3       ;GET WRITE COMMAND
   25 003602  104302  001112                  MOV     CURTRK,R2       ;GET CURRENT TRACK
   26 003604  101023                          BIS     R2,R3           ;SET TRACK FOR WRITE
   27 003605  100673  000004                  MOV     R3,RW.CMD(R0)   ;STORE IN COMMAND BLOCK
   28 003607  104202  005567                  MOV     #GDBLK,R2       ;POINT TO BLOCK
   29 003611  100672  000001                  MOV     R2,RW.BUF(R0)   ;STICK IN COMMAND BLOCK
   30 003613  104143                          MOV     (R4),R3         ;GET LOW ORDER HEADER
   31 003614  100673  000002                  MOV     R3,RW.LOW(R0)   ;STORE IN WRITE BLOCK
   32 003616  104643  000001                  MOV     1(R4),R3        ;GET HIGH ORDER
   33 003620  105303  001336                  ADD     ST.RBN,R3       ;ADD STARTING RBN BITS
   34 003622  101203  060000                  BIS     #HD.RBN,R3      ;GIVE RBN HEADER
   35 003624  100673  000003                  MOV     R3,RW.HI(R0)    ;STORE IN WRITE BLOCK
   36 003626  104203  000726                  MOV     #HSLIM-1,R3     ;GET DUMMY SDI POINTER
   37 003630  100673  000005                  MOV     R3,RW.DUM(R0)   ;POINT IN COMMAND BLOCK
   38 003632  104303  001333          WRIT13: MOV     HPREA,R3        ;GET HEADER PREAMBLE
   39 003634  104304  001334                  MOV     DPREA,R4        ;GET DATA PREAMBLE
   40 003636  104302  000740                  MOV     UNIT,R2         ;SET UNIT
   41 003640  104207  000721                  MOV     #WRBLK,R0       ;MAKE SURE POINTING AT BLOCK
   42 003642  101207  100000                  BIS     #BIT15,R0       ;SET NO REVECTORING
   43 003644  060012                          XFC     SIP             ;WAIT FOR SECTOR PULSE
   44 003645  060003                          XFC     WRITE           ;WRITE SECTOR
   45 003646  115001                          TST     R1              ;ANY ERROR ?
   46 003647  013670                          BEQ     2$              ;NO - SKIP RETRY
   47 003650  106300  001502  001504          CMP     RETRY,TMPTRY    ;MAX ?
   48 003653  013657                          BEQ     1$              ;YES - TRY SOME RECOVERY
   49 003654  115400  001504                  INC     TMPTRY          ;INC RETRY COUNT
   50 003656  003632                          BR      WRIT13          ;DO RETRY
   51 003657  104303  001505          1$:     MOV     RECTMP,R3       ;GET CURRENT ERROR RECOVERY LEVEL
   52 003661  073670                          BMI     2$              ;IF NEGATIVE THEN FRIED
   53 003662  022612                          CALL    ERRHND          ;TRY RECOVERY
   54 003663  114000  001504                  CLR     TMPTRY          ;FOR INIT
   55 003665  117400  001505                  DEC     RECTMP          ;DECREMENT IT
   56 003667  003632                          BR      WRIT13          ;RETRY
   57 003670                          2$:
```

```
 58 003670  114000  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     #REVLEN,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     RNWHER                  ;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
```

```
                                        ;
                                        ;       READ A BLOCK OF THE RCT
                                        ;       R4 ->BLOCK NUMBER OF RCT
                                        ;
003715                                  RRC:    PUSHA
003723  114005                                  CLR     R5                      ;CLEAR ERROR COUNTER
003724  104203  001052                          MOV     #SCR,R3                 ;POINT TO CHARACTERISTICS
003726  104632  000001                          MOV     STCYL(R3),R2            ;GET LOW ORDER STARTING CYLINDER
003730  103202  007777                          BIC     #LO,R2                  ;CLER REST OF WORD
003732  104207  001430                          MOV     #CONBLK,R0              ;POINT TO CONVERT BLOCK
003734  100672  000001                          MOV     R2,V1+1(R0)             ;STORE FOR CONVERT
003736  114002                                  CLR     R2                      ;FOR STORE
003737  100672  000000                          MOV     R2,V1(R0)               ;LOW ORDER ALWAYS ZERO
003741  104632  000011                          MOV     LBNTRK(R3),R2           ;GET LBN/TRK
003743  103202  177400                          BIC     #HIBYTE,R2              ;CLEAR HIGH BYTE
003745  100672  000004                          MOV     R2,V3(R0)               ;STORE IN CONVERT BLOCK
003747  104040  000731                          MOV     R4,DDUMMY               ;STORE BLOCK NUMBER
003751  114000  000732                          CLR     DDUMMY+1                ;FOR STORE
003753  104204  000731                          MOV     #DDUMMY,R4              ;POINT FOR ADD
003755  104203  001227                          MOV     #HOLD,R3                ;STARTING RCT LBN
003757  021506                                  CALL    DADD                    ;GET RCT LBN
003760  023455                                  CALL    CS                      ;CONVERT FCT BLOCK NUMBER AND SEEK
003761  104207  000721                          MOV     #RDBLK,R0               ;PREPARE FOR READ SECTORS
003763  104203  000726                          MOV     #HSLIM-1,R3             ;POINTER TO DUMMY SDI BLOCK
003765  100673  000005                          MOV     R3,RW.DUM(R0)           ;STORE IN COMMAND BLOCK
003767  104143                                  MOV     (R4),R3                 ;LO ORDER BLOCK NUMBER
003770  100673  000002                          MOV     R3,RW.LOW(R0)           ;STORE IN READ COMD BLOCK
003772  104643  000001                          MOV     1(R4),R3                ;HI ORDER BLOCK NUM AND CODE
003774  105303  001335                          ADD     ST.LBN,R3               ;ADD STARTING LBN BITS
003776  100673  000003                          MOV     R3,RW.HI(R0)            ;STORE IN READ CMD BLOCK
004000  104203  006621                          MOV     #RCTBUF,R3              ;LOAD ADDRESS OF DATA BUFFER
004002  100673  000001                          MOV     R3,RW.BUF(R0)           ;STORE IN COMMAND BUFFER
004004  104203  013400                          MOV     #RWCMD,R3               ;LOAD SDI READ COMMAND
004006  104301  001112                          MOV     CURTRK,R1               ;GET CURRENT HEAD NUMBER IN R1
004010  101013                                  BIS     R1,R3                   ;SET IT IN COMMAND
004011  100673  000004                          MOV     R3,RW.CMD(R0)           ;STORE BACK
004013  104207  000721                  READ8:  MOV     #RDBLK,R0               ;MAKE SURE POINTING AT BLOCK
004015  104203  100000                          MOV     #RDCMD,R3               ;MARK AS ONLY REQUEST
004017  100173                                  MOV     R3,(R0)                 ;STORE IN CMD BLOCK
004020  104302  000740                          MOV     UNIT,R2                 ;SDI INTERCONNECT
004022  101207  100000                          BIS     #BIT15,R0               ;SET NO REVECTORING
004024  060012                                  XFC     SIP                     ;WAIT FOR PULSE
004025  060002                                  XFC     READ                    ;READ 1 SECTOR
004026  115001                                  TST     R1                      ;ANY ERRORS ?
004027  054045                                  BNE     100$                    ;YES - TRY RECOVERY
004030  104173                                  MOV     (R0),R3                 ;GET STATUS WORD
004031  102203  010000                          BIT     #ECCF,R3                ;ECC ERROR ?
004033  014037                                  BEQ     101$                    ;NOPE - VERIFY EDC
004034  023027                                  CALL    ECCCK                   ;CORRECT ECC
004035  115001                                  TST     R1                      ;TEST FLAG
004036  054045                                  BNE     100$                    ;UNCORRECTABLE
004037  104202  006621                  101$:   MOV     #RCTBUF,R2              ;POINT TO BUFFER
004041  022627                                  CALL    CEDC                    ;COMPUTE EDC
004042  106623  000400                          CMP     RW.EDC(R2),R3           ;O.K. ?
004044  014102                                  BEQ     102$                    ;YUP - CONSIDER GOOD
004045  106300  001502  001504          100$:   CMP     RETRY,TMPTRY            ;MAX ?
004050  014054                                  BEQ     1$                      ;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 004065                            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            RCD:    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            WRIT10: 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                      MOV     R1,R2           ;XFC ERROR CODE
122 004217  104201  000016              MOV     #14.,R1         ;RCT READ ERROR
123 004221  022552                      CALL    ERRMNT          ;ERROR RETURN
```

```
 1                                          ;
 2                                          ;       VERIFY FCT
 3                                          ;
 4 004222  114000  001256          FCTCK:  CLR     FCTCNT          ;FOR INIT
 5 004224  115400  001260                  INC     FCTNPD          ;SO CHECK THE NULL BLOCK ALSO
 6 004226  104200  006204  001253          MOV     #PRMBUF,BUFPNT  ;POIN TO BUFFER
 7 004231  104201  000017                  MOV     #F6,R1          ;FCT READ OVERLAY
 8 004233  022444                          CALL    PAGE            ;READ FIRST BLOCK
 9 004234  104205  006204                  MOV     #PRMBUF,R5      ;POINT TO BUFFER
10 004236  104203  126736                  MOV     #M512,R3        ;GET MODE INDICATOR
11 004240  100153                          MOV     R3,(R5)         ;SIGNAL DONE FORMAT
12 004241  104050  001253                  MOV     R5,BUFPNT       ;STORE BUFFER POINTER
13 004243  104201  000030                  MOV     #F9,R1          ;FCT WRITE OVERLAY
14 004245  022444                          CALL    PAGE            ;WRITE IT OUT
15 004246  114000  001256                  CLR     FCTCNT          ;FOR FCTCNT INIT
16 004250  104300  001337  001116          MOV     ST.XBN,CURXBN+1 ;ALSO INITIALIZE XBN COUNTER
17 004253  104300  001337  001114          MOV     ST.XBN,CURBN+1  ;HIGH ORDER
18 004256  114000  001115                  CLR     CURXBN          ;OW ORDER ALWAYS 0
19 004260  114000  001113                  CLR     CURBN           ;AND BLOCK NUMBER
20 004262  104203  001052                  MOV     #SCR,R3         ;POINT TO CHARACTERISTICS
21 004264  104207  001430                  MOV     #CONBLK,R0      ;POINT TO CONVERT BLOCK
22 004266  104632  000100                  MOV     CYLBN(R3),R2    ;GET LOW ORDER CYLINDER
23 004270  100672  000000                  MOV     R2,V1(R0)       ;STORE IT
24 004272  104632  000001                  MOV     CYLBN+1(R3),R2  ;GET HIGH ORDER
25 004274  100672  000001                  MOV     R2,V1+1(R0)     ;STORE IT
26 004276  104303  001127                  MOV     SECTRK,R3       ;GET SECTORS/TRACK
27 004300  100673  000004                  MOV     R3,V3(R0)       ;STORE IT
28 004302  114005                  FCTCLP: CLR     R5              ;CLEAR WRITE ERROR COUNT
29 004303  104050  001245                  MOV     R5,NEXT1        ,CLEAR REPEAT COUNT
30 004305  104204  000736          FCTCL1: MOV     #TEMP,R4        ;POINT TO BLOCK
31 004307  104300  001113  000736          MOV     CURBN,TEMP      ;FOR CONVERSION
32 004312  104300  001114  000737          MOV     CURBN+1,TEMP+1  ;DITTO
33 004315  107300  001337  000737          SUB     ST.XBN,TEMP+1   ;SUBTRACT STARTING XBN BITS
34 004320  023455                          CALL    CS              ;CONVERT AND SEEK
35 004321  104207  000721                  MOV     #RDBLK,R0       ;POINT TO COMMAND BLOCK
36 004323  104203  013400                  MOV     #RWCMD,R3       ;GET READ COMMAND
37 004325  104302  001112                  MOV     CURTRK,R2       ;GET CURRENT TRACK
38 004327  101023                          BIS     R2,R3           ;SET TRACK FOR WRITE
39 004330  100673  000004                  MOV     R3,RW.CMD(R0)   ;STORE IN COMMAND BLOCK
40 004332  104203  006204                  MOV     #PRMBUF,R3      ;POINT TO BUFFER
41 004334  100673  000001                  MOV     R3,RW.BUF(R0)   ;STICK IN COMMAND BLOCK
42 004336  104303  001113                  MOV     CURBN,R3        ;GET LOW ORDER HEADER
43 004340  100673  000002                  MOV     R3,RW.LOW(R0)   ;STORE IN WRITE BLOCK
44 004342  104303  001114                  MOV     CURBN+1,R3      ;GET HIGH ORDER
45 004344  101203  120000                  BIS     #HD.XBN,R3      ;SET HEADER
46 004346  100673  000003                  MOV     R3,RW.HI(R0)    ;STORE IN WRITE BLOCK
47 004350  104203  000726                  MOV     #HSLIM-1,R3     ;GET DUMMY SDI POINTER
48 004352  100673  000005                  MOV     R3,RW.DUM(R0)   ;POINT IN COMMAND BLOCK
49 004354  104207  000721          READ9:  MOV     #RDBLK,R0       ;MAKE SURE POINTING AT BLOCK
50 004356  104203  100000                  MOV     #RDCMD,R3       ;GET READ COMMAND
51 004360  100673  000000                  MOV     R3,RW.STAT(R0)  ;STORE IT
52 004362  104302  000740                  MOV     UNIT,R2         ;SET UNIT
53 004364  101207  100000                  BIS     #BIT15,R0       ;SET NO REVECTORING
54 004366  060012                          XFC     SIP             ;WAIT FOR SECTOR PULSE
55 004367  060002                          XFC     READ            ;READ SECTOR
56 004370  115001                          TST     R1              ;ANY ERROR ?
57 004371  054407                          BNE     100$            ;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  004354                          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
```

```
                                                ;
                                                ;       VERIFY RCT
                                                ;
004510  115400  001261          RCTCK:  INC     RCTLBN          ;CHECK NON-PAD PLUS NULL BLOCK
004512  104300  001133  001227          MOV     LBNLBN,HOLD     ;GET LOW ORDER COUNT OF LBN'S
004515  104300  001134  001230          MOV     LBNLBN+1,HOLD+1 ;GET HIGH ORDER
004520  104203  001475                  MOV     #TOTRCT,R3      ;FOR SUBTRACT
004522  104204  001227                  MOV     #HOLD,R4        ;DITTO
004524  021524                          CALL    DSUB            ;GET STARTING RCT LBN
004525  104300  001227  001113          MOV     HOLD,CURBN      ;GET STARTING RCT BLOCK NUMBER
004530  104300  001227  001115          MOV     HOLD,CURLBN     ;ALSO SAVE
004533  104300  001230  001114          MOV     HOLD+1,CURBN+1  ;GET HIGH ORDER
004536  104300  001230  001116          MOV     HOLD+1,CURLBN+1 ;AND SAVE
004541  104200  000001  001501          MOV     #1,COUNT
004544  114005                  RCTCLP: CLR     R5              ;CLEAR ERROR COUNTER
004545  104050  001245                  MOV     R5,NEXT1        ;INIT COPY COUNT
004547  104203  001052          RCTCL1: MOV     #SCR,R3         ;POINT TO CHARACTERISTICS
004551  104632  000011                  MOV     LBNTRK(R3),R2   ;GET LBN/TRACK
004553  103202  177400                  BIC     #HIBYTE,R2      ;CLEAR HIGH BYTE
004555  104207  001430                  MOV     #CONBLK,R0      ;POINT TO CONVERT BLOCK
004557  100672  000004                  MOV     R2,V3(R0)       ;FOR CONVERT
004561  104632  000001                  MOV     STCYL(R3),R2    ;STARTING CLYLINDER
004563  103202  007777                  BIC     #LO,R2          ;CLEAR REST OF WORD
004565  100672  000001                  MOV     R2,V1+1(R0)     ;STORE
004567  114002                          CLR     R2              ;FOR STORE
004570  100672  000000                  MOV     R2,V1(R0)       ;LOW ORDER ALWAYS ZERO
004572  104204  001113                  MOV     #CURBN,R4       ;FOR CONVERT
004574  023455                          CALL    CS              ;CONVERT AND SEEK
004575  104207  000721                  MOV     #RDBLK,R0       ;POINT TO COMMAND BLOCK
004577  104203  013400                  MOV     #RWCMD,R3       ;GET READ COMMAND
004601  104302  001112                  MOV     CURTRK,R2       ;GET CURRENT TRACK
004603  101023                          BIS     R2,R3           ;SET TRACK FOR WRITE
004604  100673  000004                  MOV     R3,RW.CMD(R0)   ;STORE IN COMMAND BLOCK
004606  104203  006204                  MOV     #PRMBUF,R3      ;POINT TO BUFFER
004610  100673  000001                  MOV     R3,RW.BUF(R0)   ;STICK IN COMMAND BLOCK
004612  104303  001113                  MOV     CURBN,R3        ;GET LOW ORDER HEADER
004614  100673  000002                  MOV     R3,RW.LOW(R0)   ;STORE IN WRITE BLOCK
004616  104303  001114                  MOV     CURBN+1,R3      ;GET HIGH ORDER
004620  105303  001335                  ADD     ST.LBN,R3       ;ADD STARTING LBN BITS
004622  101203  000000                  BIS     #HD.LBN,R3      ;SET HEADER
004624  100673  000003                  MOV     R3,RW.HI(R0)    ;STORE IN WRITE BLOCK
004626  104203  000726                  MOV     #HSLIM-1,R3     ;GET DUMMY SDI POINTER
004630  100673  000005                  MOV     R3,RW.DIM(R0)   ;POINT IN COMMAND BLOCK
004632  104207  000721          READ10: MOV     #RDBLK,R0       ;MAKE SURE POINTING AT BLOCK
004634  104203  100000                  MOV     #RDCMD,R3       ;GET STATUS WORD
004636  100173                          MOV     R3,(R0)         ;STORE IT
004637  104302  000740                  MOV     UNIT,R2         ;SET UNIT
004641  101207  100000                  BIS     #BIT15,R0       ;SET NO REVECTORING
004643  060012                          XFC     SIP             ;WAIT FOR SECTOR PULSE
004644  060002                          XFC     READ            ;WRITE SECTOR
004645  115001                          TST     R1              ;ANY ERROR ?
004646  054664                          BNE     100$            ;YES - TRY RECOVERY
004647  104173                          MOV     (R0),R3         ;GET STATUS WORD
004650  102203  010000                  BIT     #ECCF,R3        ;ECC ERROR ?
004652  014656                          BEQ     101$            ;NOPE - VERIFY EDC
004653  023027                          CALL    ECCCK           ;CORRECT ECC
004654  115001                          TST     R1              ;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$:   CMP     RETRY,TMPTRY            ;MAX ?
 64 004667  014673                          BEQ     1$                      ;YES - TRY SOME RECOVERY
 65 004670  115400  001504                  INC     TMPTRY                  ;INC RETRY COUNT
 66 004672  004632                          BR      READ10                  ;DO RETRY
 67 004673  104303  001505          1$:     MOV     RECTMP,R3               ;GET CURRENT ERROR RECOVERY LEVEL
 68 004675  074705                          BMI     RCTNGD                  ;IF NEGATIVE SKIP GOOD INCREMENT
 69 004676  022612                          CALL    ERRHND                  ;TRY RECOVERY
 70 004677  114000  001504                  CLR     TMPTRY                  ;FOR INIT
 71 004701  117400  001505                  DEC     RECTMP                  ;DECREMENT IT
 72 004703  004632                          BR      READ10                  ;RETRY
 73 004704                          102$:
 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                   ;INCREMENT 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
```

```
                                        .SBTTL  FCT WRITE OVERLAY (F9)
004766                                  DMOVLY  F9,START
                                ;
                                ;       WRITE AN FCT BLOCK
                                ;
                                ;
003047  104200  000030  001153  FCTWRT: MOV     #F9.CUROVL              ;FOR INIT
003052  114005                          CLR     R5                      ;CLEAR ERROR COUNTER
003053  104050  001245                  MOV     R5,NEXT1                ;INIT NEXT COPY COUNTER
003055  104204  001256                  MOV     #FCTCNT,R4              ;POINT TO FCT BLOCK NUMBER
003057  104203  001466                  MOV     #ONE,R3                 ;FOR SUB
003061  021524                          CALL    DSUB                    ;SUB TO GET RIGHT ONE
003062  104305  001253                  MOV     BUFPNT,R5               ;GET BUFFER POINTER
003064  104303  001143          FCTRLP: MOV     LBNCYL,R3               ;GET LBN CYLINDERS
003066  104207  001430                  MOV     #CONBLK,R0              ;POINT TO CONVERT BLOCK
003070  100673  000000                  MOV     R3,V1(R0)               ;STORE FOR CONVERT
003072  104303  001144                  MOV     LBNCYL+1,R3             ;GET HIGH ORDER
003074  100673  00000                   MOV     R3,V1+1(R0)             ;STORE IT
003076  104303  001127                  MOV     SECTRK,R3               ;GET SECTORS/TRACK
003100  100673  000004                  MOV     R3,V3(R0)               ;STORE IT
003102  022714                          CALL    CVTSK                   ;CONVERT AND SEEK
003103  104207  000721                  MOV     #WRBLK,R0               ;POINT TO COMMAND BLOCK
003105  104203  122400                  MOV     #WRCMD,R3               ;GET WRITE COMMAND
003107  104302  001112                  MOV     CURTRK,R2               ;GET CURRENT TRACK
003111  101023                          BIS     R2,R3                   ;SET TRACK FOR WRITE
003112  100673  000004                  MOV     R3,RW.CMD(R0)           ;STORE IN COMMAND BLOCK
003114  104052                          MOV     R5,R2                   ;POINT TO BUFFER
003115  022627                          CALL    CEDC                    ;COMPUTE EDC - RETURNED IN R3
003116  100623  000400                  MOV     R3,RW.EDC(R2)           ;STORE IT
003120  100672  000001                  MOV     R2,RW.BUF(R0)           ;STICK IN COMMAND BLOCK
003122  104143                          MOV     (R4),R3                 ;GET LOW ORDER HEADER
003123  100673  000002                  MOV     R3,RW.LOW(R0)           ;STORE IN WRITE BLOCK
003125  104643  000001                  MOV     1(R4),R3                ;GET HIGH ORDER
003127  105303  001337                  ADD     ST.XBN,R3               ;ADD STARTING XBN BITS
003131  101203  120000                  BIS     #HD.XBN,R3              ;SET HEADER
003133  100673  000003                  MOV     R3,RW.HI(R0)            ;STORE IN WRITE BLOCK
003135  104203  000726                  MOV     #HSLIM-1,R3             ;GET DUMMY SDI POINTER
003137  100673  000005                  MOV     R3,RW.DUM(R0)           ;POINT IN COMMAND BLOCK
003141  104303  001333          WRITE5: MOV     HPREA,R3                ;GET HEADER PREAMBLE
003143  104304  001334                  MOV     DPREA,R4                ;GET DATA PREAMBLE
003145  104302  000740                  MOV     UNIT,R2                 ;SET UNIT
003147  104207  000721                  MOV     #WRBLK,R0               ;MAKE SURE POINTING AT BLOCK
003151  101207  100000                  BIS     #BIT15,R0               ;SET NO RVECTORING
003153  060012                          XFC     SIP                     ;WAIT FOR SECTOR PULSE
003154  060003                          XFC     WRITE                   ;WRITE SECTOR
003155  115001                          TST     R1                      ;ANY ERROR ?
003156  013200                          BEQ     FWGOOD                  ;NOPE
003157  106300  001502  001504          CMP     RETRY,TMPTRY            ;MAX ?
003162  013166                          BEQ     1$                      ;YES - TRY SOME RECOVERY
003163  115400  001504                  INC     TMPTRY                  ;INC RETRY COUNT
003165  003141                          BR      WRITE5                  ;DO RETRY
003166  104303  001505          1$:     MOV     RECTMP,R3               ;GET CURRENT ERROR RECOVERY LEVEL
003170  073177                          BMI     2$                      ;IF NEGATIVE THEN FRIED
003171  022612                          CALL    ERRHND                  ;TRY RECOVERY
003172  114000  001504                  CLR     TMPTRY                  ;FOR INIT
003174  117400  001505                  DEC     RECTMP                  ;DECREMENT IT
003176  003141                          BR      WRITE5                  ;RETRY
```

```
    58 003177                                          2$:
    59 003177  115405                                          INC     R5                      ;YUP - INCREMENT COUNTER
    60 003200  115400  001245                       FWGOOD:    INC     NEXT1                   ;INCREMENT IT
    61 003202  114000  001504                                  CLR     TMPTRY                  ;FOR RESET
    62 003204  104300  001503  001505                         MOV     RECOV,RECTMP            ;GET RECOVERY LEVELS
    63 003207  104204  001256                                  MOV     #FCTCNT,R4              ;FOR ADD
    64 003211  104203  001240                                  MOV     #FCTFMT,R3              ;FOR ADD
    65 003213  021506                                          CALL    DADD                    ;POINT TO NEXT COPY
    66 003214  106300  001244  001245                         CMP     FCTCPY,NEXT1            ;DONE THIS SECTOR ?
    67 003217  053064                                          BNE     FCTRLP                  ;NO - WRITE NEXT FCT COPY
    68 003220  106305  001244                                  CMP     FCTCPY,R5               ;ERROR ON EVERY WRITE ?
    69 003222  013244                                          BEQ     FCWERR                  ;YUP - BIG TROUBLE
    70 003223  104303  001245                       FCFXLP:    MOV     NEXT1,R3                ;ANY REAPEATS ?
    71 003225  013236                                          BEQ     FWTDON                  ;NO
    72 003226  104204  001256                                  MOV     #FCTCNT,R4              ;TO GET IT BACK
    73 003230  104203  001240                                  MOV     #FCTFMT,R3              ;DITTO
    74 003232  021524                                          CALL    DSUB
    75 003233  117400  001245                                  DEC     NEXT1                   ;SUB IT
    76 003235  003223                                          BR      FCFXLP                  ;REPEAT
    77 003236                                      FWTDON:     DUBINC  FCTCNT                  ;PUT BACK THE WAY IT WAS
    78 003243  000000                                          RETURN
    79 003244  104012                              FCWERR:     MOV     R1,R2                   ;XFC ERROR CODE
    80 003245  104201  000017                                  MOV     #15.,R1                 ;RCT WRITE ERROR
    81 003247  022552                                          CALL    ERRMNT                  ;ERROR RETURN
```

```
                                                .SBTTL  PBN->D.X.L.RBN CONVERSION OVERLAY (G5)
  003252                                        DMOVLY  G5,START
                                        ;
                                        ;       CONVERT FROM PBN TO D.X.L.RBN
                                        ;
  003047  104303  001111                PCON:   MOV     CURPBN+1,R3             ;GET PBN HEADER
  003051  103203  170000                        BIC     #HD.CLR,R3              ;CLEAR THE HEADER
  003053  104030  001111                        MOV     R3,CURPBN+1             ;STORE BACK
  003055  104300  001133  000731                MOV     LBNLBN,DDUMMY           ;GET TOTAL NUMBER OF LBN'S
  003060  104300  001134  000732                MOV     LBNLBN+1,DDUMMY+1       ;HIGH ORDER
  003063  104204  000731                        MOV     #DDUMMY,R4              ;FOR ADD
  003065  104203  001135                        MOV     #RBNLBN,R3              ;TOTAL NUMBER OF RBN'2
  003067  021506                                CALL    DADD                    ;ADD TO GET TOTAL BLOCKS IN LBN AREA
  003070  104203  001110                        MOV     #CURPBN,R3              ;CURRENT PBN
  003072  021626                                CALL    DCMP                    ;IS IT LBN OR RBN ?
  003073  073142                                BMI     LRBN                    ;YUP - GO COMPUTE IT
  003074  104204  001110                        MOV     #CURPBN,R4              ;ELSE X OR DBN - GET VALUE
  003076  104203  000731                        MOV     #DDUMMY,R3              ;TOTAL LBN+RBN BLOCKS
  003100  021524                                CALL    DSUB                    ;SUBTRACT
  003101  104203  001137                        MOV     #XBNSEC,R3              ;TOTAL XBN SECTORS
  003103  021626                                CALL    DCMP                    ;IS IT AN XBN ?
  003104  073125                                BMI     XBNFND                  ;YUP - GO FIXIT
  003105  104204  001110                        MOV     #CURPBN,R4              ;ELSE DBN - GET VALUE
  003107  104203  001137                        MOV     #XBNSEC,R3              ;TOTAL XBN SECTORS
  003111  021524                                CALL    DSUB                    ;SUBTRACT TO GET RELATIVE DBN
  003112  104641  000001                        MOV     1(R4),R1                ;GET HIGH ORDER
  003114  105301  001340                        ADD     ST.DBN,R1               ;ADD TO GET ABSOLUTE DBN
  003116  103201  170000                        BIC     #HD.CLR,R1              ;CLEAR THE HEADER
  003120  101201  140000                        BIS     #HD.DBN,R1              ;MARK AS DBN
  003122  104010  001111                        MOV     R1,CURPBN+1             ;STORE BACK
  003124  003276                                BR      PDONE                   ;CLEAN UP AND RETURN
  003125  104204  001110                XBNFND: MOV     #CURPBN,R4              ;POINT TO BLOCK NUMBER
  003127  104641  000001                        MOV     1(R4),R1                ;GET HIGH ORDER
  003131  105301  001337                        ADD     ST.XBN,R1               ;ADD TO GET ABSOLUTE XBN
  003133  103201  170000                        BIC     #HD.CLR,R1              ;CLEAR HEADER
  003135  101201  120000                        BIS     #HD.XBN,R1              ;MARK AS XBN
  003137  104010  001111                        MOV     R1,CURPBN+1             ;STORE BACK
  003141  003276                                BR      PDONE                   ;CLEAN UP AND RETURN
                                        ;
                                        ;
  003142  104300  001127  000731        LRBN:   MOV     SECTRK,DDUMMY           ;GET LOW ORDER SECOTRS/TRACK
  003145  104300  001130  000732                MOV     SECTRK+1,DDUMMY+1       ;HIGH ORDER
  003150  104203  000731                        MOV     #DDUMMY,R3              ;FOR DIVIDE
  003152  104204  001110                        MOV     #CURPBN,R4              ;DITTO
  003154  021570                                CALL    DDIV                    ;DIVIDE TO GET TRACK NUMBER AND REMAINDER
  003155  104205  001052                        MOV     #SCR,R5                 ;POINT TO CHARACTERISTICS
  003157  104653  000011                        MOV     LBNTRK(R5),R3           ;GET LBN/TRK
  003161  103203  177400                        BIC     #HIBYTE,R3              ;CLEAR HIGH BYTE
  003163  104201  000731                        MOV     #DDUMMY,R1              ;POINT TO REMAINDER
  003165  104114                                MOV     (R1),R4                 ;GET IT
  003166  106043                                CMP     R4,R3                   ;COMPARE
  003167  073242                                BMI     LBNFND                  ;IF MINUS THEN LBN
  003170  104653  000004                        MOV     RBNTRK(R5),R3           ;GET RBN/TRACK
  003172  103203  177600                        BIC     #HI1BYTE,R3             ;CLEAR OUT GARBAGE
  003174  104030  000736                        MOV     R3,TEMP                 ;STORE IT
  003176  114000  000737                        CLR     TEMP+1                  ;FOR STORE
  003200  104203  000736                        MOV     #TEMP,R3                ;FOR MULTIPLY
```

```
 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 SUBRACT
 61 003207  104653  000011                    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          LBNEND:   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
```

SYMBOL TABLE

```
ACC     001024
ACCESS  002235
AGAIN   003544
ALDONE  003264
ALLOVR  002603
ALLOV1  002611
AOUT    002362
ATTN  = 000002
ATTN1   002353
BADEDC  001452
BADFUN  001223
BADRBN  003645
BD    = 020000
BDHD  = 000040
BDIRCT  004136
BIT0  = 000001
BIT1  = 000002
BIT10 = 002000
BIT11 = 004000
BIT12 = 010000
BIT13 = 020000
BIT14 = 040000
BIT15 = 100000
BIT2  = 000004
BIT3  = 000010
BIT4  = 000020
BIT5  = 000040
BIT6  = 000100
BIT7  = 000200
BIT8  = 000400
BIT9  = 001000
BLKFND  003176
BMAX  = 007775
BOTTOM  003322
BRBN    004034
BREAK = 000000
BSTGS = 002000
BUFMSK= 007777
BUFPNT  001253
BUF1  = 004535
BUF2  = 005152
BUF3  = 005567
BUF4  = 006204
BUF5  = 006621
BUF6  = 007275
BUMER   003064
CBUF    001444
CDONE   004044
CEDC    002627
CHAR    001772
CHRDNE= 010000
CKR     003513
CLEAR   002243
CLEDON  003174
CLELP   003106
CLELP2  003076
CLESKP  003144
CLEWRT  003270
CLHERE  003303
CLSKP2  003165
CLSKP3  003201
CLSKP4  003200
CMDBUF= 006621
CMPDAT= 000006
CNT     001247
CNTCYL  001460
CONBLK  001430
CONINT  003067
COUNT   001501
CR      001037
CR.ACC  000765
CR.CLR  000771
CR.DIS  000755
CR.ERV  001005
CR.GCR  000745
CR.GSR  000751
CR.GST  000741
CR.ONL  001017
CR.RCL  001001
CR.RUN  000761
CR.SEK  000775
CS      003455
CSKIP   003522
CSKIP1  003534
CSKIP2  003533
CSKIP3  003773
CSKIP6  003743
CSKIP7  004000
CURBN   001113
CURGRP  001463
CURLBN  001115
CUROVL  001153
CURPBN  001110
CURRBN  001106
CURTRK  001112
CURXBN  001115
CUTOF   001235
CVT   - 000020
CVTERR  002750
CVTSK   002714
CYL   = 000006
CYLBN = 000000
CYLMSG  001316
CYLNUM  001125
C512  = 000016
DADD    001506
DADD1   001515
DAT   - 000011
DATA  = 000005
DATE    001301
DBBAD   001314
DBD   = 000006
DBN   = 000010
DBNCYL= 000022
DCLR    001026
DCMP    001626
DCMP1   001651
DCMP2   001644
DCMP3   001657
DCMP4   001637
DDIV    001570
DDUMMY  000731
DEAD  = 000020
DESC    003050
DFCT    004423
DINIT - 000011
DIS     001021
DISCON  002324
DLERR   003113
DLERT   003470
DLL     000400
DLLRET  003446
DLLRT1  003432
DMBUF   001263
DMBUFL= 000016
DMFLG   000004
DMOST = 000001
DMSG1   003054
DMSG1L= 000007
DMUL    001542
DMUNIT= 000003
DOLBN   003061
DONDLL  003451
DONE  = 000021
DONE1   003206
DONMSG  001327
DPBN    003307
DPREA   001334
DSUB    001524
DUPOVL= 007774
DWRD    001450
DXBN    003063
DXCH    003500
DXCHEC  003477
DXERR   003406
DXFCPG  003407
DXFCP1  003370
DXFORM  003114
DXTRK   003052
EAGAIN  002636
ECC   = 000015
ECCCK   003027
ECCF  = 010000
ECHO  - 000010
EDC     001451
EIMAGE  001231
EMAX    001225
EMSG    001324
ENTRY   000714
EORCT   003173
ERCV    000002
ERDN  - 000010
ERECOV  001102
ERFLAG  001221
ERLEN = 000002
ERPNT   001252
ERR     001226
ERRBUF  001224
ERRCNT  001453
ERRHND  002612
ERRLN = 000003
ERRLST  002624
ERRMNT  002552
ERROR   003465
ERRSYM- 000002
ERRT    002101
ER1     003753
FBDHD = 010000
FBEGIN  003276
FBEG2   003311
FCFXLP  003223
FCLR  = 177760
FCMSG   001321
FCNT    001237
FCPG    003657
FCT   - 000013
FCTAVL= 000001
FCTBAD= 000004
FCTCK   004222
FCTCKD  004454
FCTCKE  004503
FCTCLP  004302
FCTCL1  004305
FCTCNT  001256
FCTCPY  001244
FCTEMT= 000002
FCTFLG= 000025
FCTFMT  001240
FCTNPD  001260
FCTPTR  001255
FCTRCT  003212
FCTREV  001311
FCTRLP  003064
FCTSKP  003466
FCTSK1  003460
FCTSLP  003644
FCTSP   003762
FCTSZ = 000010
FCTWRT  003047
FCWERR  003244
FDAT  - 000012
FDLL    003272
FERR    003376
FILLIT  003666
FINI  - 040000
FIXBLK  003466
FIXFCT  004015
FIXIT   004046
FIXLP   003503
FKIP1   004121
FKIP10  004145
FKIP2   004102
FKIP9   004460
FKP1    004136
FLAG    001217
FLAG1   001220
FLGDON  004427
FLGSET  004373
FLGSKP  004405
FLIP    003136
FLKIP1  004434
FLKIP2  004415
FLKP1   004451
FODONE  003227
FOLOOP  003060
FORERR  003431
FORMAT= 000001
FRCPY = 000001
FRDONE  003750
FRSKP   003744
FSER  = 000002
FT.BUF= 000000
FT.HI = 000002
FT.LOW= 000001
FULL  = 040000
FWGOOD  003200
FWRD    001445
FWTDON  003236
F1    = 000000
F2    = 000003
F3    = 000006
F4    = 000011
F5    = 000014
F6    = 000017
F7    = 000022
F8    = 000025
F9    = 000030
GCR     001016
GDBLK = 005567
GDECC   003046
GDFCT   003260
GDONE   004337
GETUNT  004203
GOBAD = 000020
GOVER   004212
GRP   = 000010
GRPCNT  001464
GRPCYL= 000002
GSKIP   004315
GSKIP1  004232
GSKIP2  004250
GSKIP3  004267
GSR     001017
GST     001015
GSTATS  001662
```

SYMBOL TABLE

```
G1     = 000060
G2     = 000033
G3     = 000036
G4     = 000041
G5     = 000044
G7     = 000047
G8     = 000052
HASH   = 003050
HD.BAD= 110000
HD.CLR= 170000
HD.CUR  001462
HD.DBN= 140000
HD.LBN= 000000
HD.PRV= 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
INITS   002347
INSI   = 000001
IRECAL  001101
ISEEK   001075
ISKIP   003643
LAGAIN  003506
LAST   = 100000
LBD    = 000002
LBNBAD  001312
LBNCYL  001143
LBNFND  003242
LBNHOS= 000012
LBNLBN  001133
LBNPCY  001145
LBNTRK= 000011
LCHEC   003442
LDONE   004360
LEN    = 000000
LERR    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
LREDO   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
ORFTAL  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
OVLTBL  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
OVRLAY  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
```

SYMBOL TABLE

```
RCFIX   003442   READ9   004354   SECTRK  001127   STPNIC  002013   TRK    = 000011
RCFXLP  004230   RECAL   002154   SEEK    002251   STRBN = 000003   TRKCNT  001465
RCINDN= 000100   RECIR = 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   REVBUF= 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   REVLEN= 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.FO  = 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   RTYCNT  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.NUL= 100000   RWCMD  = 013400   STACK   001215   TILOP   004046   WRITE5  003141
RC.PRV= 020000   RWGD    003546   STARIT  001232   TILOP1  004053   WRITE8  003301
RC.SND= 030000   RWGOOD  004205   START   003047   TIMER   002656   WRITE9  003404
RC.UNU- 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   STDIAG  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
```

DATA FILE FOR PDP-11 UDA DISK D MACRO X04.00 23-JUL-81 15:08:54 PAGE 73-5 SEQ 0143

SYMBOL TABLE

XPERR	003711	XSKIP1	003061	XSKIP4	003235	XSLEEK	003115	XYZ1	003720
XPNGRD	003652	XSKIP2	003163	XSKIP5	003260	XSLEK2	003131	YES	004430
XPRE*	003707	XSKIP3	003160	XSKIP6	003255	XYZ	003717		

. ABS. 036432 000
000000 001
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 2971 WORDS (12 PAGES)
DYNAMIC MEMORY AVAILABLE FOR 71 PAGES
,B:ZUDFA0/C=[1,33]DMACRO,B:ZUDFA0.DOC,B:ZUDFA0

CROSS REFERENCE TABLE (CREF V01-05)

```
ACC     9-52    9-72#   32-74*
ACCESS  14-43   14-51   14-55   14-60   15-120#
AGAIN   35-70#  35-106
ALDONE  66-83   66-86#
ALLOV1  19-40   19-42#
ALLOVR  19-25   19-38#
AOUT    15-195  15-197  15-200#
ATTN    7-167#  15-144
ATTN1   15-194# 15-199
BADEDC  10-25#  42-231  44-130
BADPBN  9-143#  34-41*  35-8*   37-13   37-28   37-44*  39-8*   44-11   44-27   44-67*  44-80*  44-87*  44-155* 65-78*
BADRBN  59-73   59-168#
BD      7-324#  35-95   35-97   35-113  42-47   42-49   42-65
BDHD    7-244#  42-106  42-147  42-180  42-189  42-199  42-213
BDIRCT  42-110  42-182#  42-190
BIT0    7-103#  7-133   7-168   7-174   7-218   7-239   33-9
BIT1    7-104#  7-133   7-167   7-175   7-222   7-240   32-47   33-6
BIT10   7-113#  7-180   7-228
BIT11   7-114#  7-229
BIT12   7-115#  7-154   7-177   7-232   7-321
BIT13   7-116#  7-177   7-230   7-324   51-58
BIT14   7-117#  7-177   7-231   7-320   7-326
BIT15   7-118#  7-153   7-166   7-177   7-233   7-298   7-325   35-82   35-134  42-34   42-87   50-100  51-27   55-40
        55-100  60-40   61-78   63-39   63-102  67-48   68-42   69-41   69-101  70-53   71-48   72-43
BIT2    7-105#  7-133   7-134   7-181   7-225   7-241   32-50   33-11
BIT3    7-106#  7-133   7-135   7-182   7-219   7-242   32-53
BIT4    7-107#  7-178   7-183   7-223   7-243
BIT5    7-108#  7-176   7-224   7-244
BIT6    7-109#  7-220   7-245
BIT7    7-110#  7-136   7-226
BIT8    7-111#  7-137   7-227   54-10   58-9
BIT9    7-112#  7-179   7-221
BLKFND  65-58   65-61#
BMAX    7-367#  29-14
BOTTOM  51-59   51-71   51-8?   51-107#
BRBN    42-11?  42-149#
BREAK   7-51#
BSTGS   7-228#  33-13   34-36   37-26   40-74   42-8    44-101  44-136  44-169  50-21   50-127  63-126  65-83   66-79
        70-92
BUF1    7-344#  7-353
BUF2    7-345#  7-354
BUF3    7-346#  7-355
BUF4    7-347#  7-356   7-357
BUF5    7-348#  7-358   7-359   7-360
BUF6    7-349#  7-361
BUFMSK  7-132#  42-141  42-165  42-244
BUFPNT  9-169#  45-5*   51-39*  54-39*  55-8    58-37*  59-21*  59-36*  59-62*  59-127* 59-200* 59-209* 59-214* 59-237*
        59-245* 63-29   63-50   63-88   65-13*  65-43*  65-49*  70-6*   70-12*  72-13
BUMER   26-8    26-15#
C512    7-295#  34-42   34-43   59-25   65-29
CBUF    10-17#
CDONE   35-110  35-168  35-187#
CEDC    20-6#   35-148  42-145  44-69   50-73   53-24   55-52   60-25   61-64   63-51   67-33   69-53   70-65   71-60
        72-28
CHAR    14-59   14-62?
CHRDNE  7-232#  14-62   14-68
```

CROSS REFERENCE TABLE (CREF V01-05)

```
CKR       67-85    67-87#
CLEAR     14-22    15-128#  34-91    35-182   40-58    42-260
CLEDON    66-50    66-55#
CLELP     66-24#   66-46
CLELP2    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
CMPDAT     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
```

	50-84	51-12	51-118*	53-20	55-33	55-84	59-9	59-224*	60-21	61-21	61-60	61-142*	63-32	63-85
	67-29	67-79*	68-25	69-34	69-85	70-37	71-31	72-24						
CURXBN	9-99#	50-9*	50-11*	50-133*	50-133*	50-134	50-135	59-91*	59-92*	59-94	59-95	70-16*	70-18*	70-96*
	70-96*	70-97	70-98											
CUTOF	9-157#	27-15												
CVT	7-67#	21-11	67-78											
CVTERR	21-18	21-20#												
CVTSK	21-5#	50-81	53-17	55-21	55-81	60-18	61-57	63-19	63-82	68-22	72-21			
CYL	7-254#	21-13	21-14	21-22	67-80	67-81	67-89							
CYLBN	7-81#	26-69	26-70	26-89	26-90	26-131	26-132	26-133	50-15	50-17	70-22	70-24		
CYLMSG	9-185#	34-110	34-112	40-111	40-113									
CYLNUM	9-107#	26-155*	26-156*	34-64	34-65	34-114	34-120	34-123	34-132	40-14*	40-16*	40-19	40-31	40-32
	40-115	40-121	40-124	40-146	51-14	51-16	51-112*	51-115*	59-11	59-13	59-218*	59-221*	61-23	61-25
	61-136*	61-139*												
DADD	11-19#	12-48	25-12	26-140	26-147	26-166	26-187	26-227	28-24	38-12	38-14	38-16	38-35	38-37
	49-27	50-122	51-38	53-60	54-38	55-70	58-36	59-61	59-126	60-62	61-33	61-100	63-70	69-21
	69-71	70-86	71-83	72-65	73-13	73-69	73-86							
DADD1	11-24	11-26#												
DAT	7-278#	29-90												
DATA	7-96#	29-103												
DATE	9-178#	29-91	50-61											
DBBAD	9-183#	34-16*	66-71											
DBD	7-283#	66-72*	66-73*											
DBN	7-219#	34-11	34-19	34-30	37-17									
DBNCYL	7-90#	26-141	34-32	35-17										
DCLR	9-53	9-74#	14-21*											
DCMP	13-14#	59-46	61-112	65-57	65-68	73-15	73-21							
DCMP1	13-19	13-29#												
DCMP2	13-20	13-25#	13-36											
DCMP3	13-30	13-35#												
DCMP4	13-21#	13-37												
DDIV	12-53#	22-17	26-170	26-182	26-191	49-16	52-14	57-14	73-45					
DDUMMY	9-31#	22-14*	22-15*	22-16	22-20*	22-21*	22-22	25-6*	25-7*	25-8	26-28*	26-29*	26-34	26-78*
	26-79*	26-80	26-111*	26-112*	26-113	26-137*	26-138*	26-139	26-144*	26-145*	26-146	26-162*	26-163*	26-165
	26-167*	26-168*	26-169	26-178*	26-179*	26-181	26-183*	26-184*	26-186	26-188*	26-189*	26-190	26-192*	26-193*
	26-194	26-224*	26-225*	26-226	29-16*	29-17*	29-19	37-80*	37-81*	37-83	37-85*	37-87*	40-86*	40-87*
	40-88	40-95*	40-96*	40-97	42-119*	42-124*	42-127	44-201*	44-202*	44-204	44-206*	44-208*	47-16*	47-17*
	47-18	48-16*	48-17*	48-18	49-13*	49-14*	49-15	49-19*	49-20*	49-21	49-23*	49-24*	49-25*	49-26
	51-25*	51-26*	51-27*	51-30*	51-31	52-10*	52-11*	52-12	53-6	53-58	53-67	54-32	57-10*	57-11*
	57-12	58-31	58-33*	59-91	59-92	59-94*	59-95*	60-7	60-60	60-69	61-30*	61-31*	61-32	61-108*
	61-109*	61-110	63-8*	63-9*	63-11	63-23	63-25	63-68	63-79	63-90	63-92	67-43*	67-59	68-20*
	68-60	69-17*	69-18*	69-19	73-9*	73-10*	73-11	73-18	73-41*	73-42*	73-43	73-49	73-60	73-68
	73-85													
DEAD	7-243#	16-23	17-43	19-39										
DESC	24-5#													
DFCT	33-12	33-17#												
DINIT	7-60#	15-61	15-192	15-211										
DIS	9-50	9-65#												
DISCON	15-171#	19-37	66-87											
DLERR	64-22	64-31#												
DLERT	50-159#													
DLL	7-227#	33-17	35-56	37-15	37-52	59-83	59-160	59-186	59-198	59-207				
DLLRET	50-128	50-147#												
DLLRT1	50-130	50-141#												
DMBUF	9-177#	19-5	19-14	19-27	24-16	29-81	29-89	32-6	33-4	34-109	40-110	50-40	64-12	66-10
	66-60													
DMBUF1	7-267#	19-6	19-15	24-17										

CROSS REFERENCE TABLE (CREF V01-05)

DMFLG 7-266# 33-5
DMDST 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#
DONDLL 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 7-351# 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
EMAX 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#
ERCV 7-78# 29-28
ERDN 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#
ERRLST 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*

CROSS REFERENCE TABLE (CREF V01-05)

```
ERR     15-48#  15-57   15-59
F       7-196#  26-11   34-10
F2      7-197#  26-13   40-8    50-147
F3      7-198#  34-26   50-7
F4      7-199#  40-66   40-134  51-11
F5      7-200#  40-79   42-165* 59-8    61-6
F6      7-201#  45-6    59-22   59-37   59-215  59-238  63-7    65-14   65-50   70-7
F7      7-202#  40-76   59-32
F8      7-203#  40-43   44-6
F9      7-204#  59-201  59-210  59-246  65-44   70-13   72-7
FBDHD   7-154#  44-74
FBEG2   59-41#  59-206
FBEGIN  59-36#  59-199  59-203
FCFXLP  72-70#  72-76
FCLR    7-130#  28-7    29-48   29-63   50-145
FCMSG   9-187#  50-41   50-43   64-13   64-15
FCNT    9-159#  34-14   34-18   34-22   34-42*  37-37*  44-36*  44-100  44-148* 65-30*  65-46*  65-59*  65-72*  65-74*
        65-79
FCPG    44-106  44-181  45-4#
FCT     7-286#  66-82*  66-85*
FCTAVL  7-218#  26-9    33-15   40-9    44-23   50-141  63-127  65-16   65-52   65-84
FCTBAD  7-225#  63-126  65-83
FCTCK   66-58   70-4#
FCTCKD  70-83   70-92#
FCTCKE  70-91   70-102#
FCTCL1  70-30#  70-90
FCTCLP  70-28#  70-100
FCTCNT  9-172#  34-39*  34-45*  37-57*  59-15   59-20*  59-213* 59-234  59-248* 59-248* 63-8    63-120* 63-122* 64-17
        64-32   65-12*  65-48*  70-4*   70-15*  70-95*  70-99*  70-103  72-10   72-63   72-72   72-77*  72-77*
FCTCPY  9-162#  28-8*   50-123  50-125  50-138* 50-146* 53-61   53-65   55-67   60-63   60-65   61-101  61-103  63-66
        69-68   70-89   71-86   72-66   72-68
FCTEMT  7-222#  34-48   37-26   37-39   44-25   44-38   44-101  44-136  44-150  44-169  65-32
FCTFLG  7-297#  50-38*  65-26
FCTFMT  9-160#  28-10*  28-11*  50-121  50-139  55-79   63-69   63-80   69-80   70-85   72-64   72-73
FCTNPD  9-174#  26-196* 26-197* 50-136  70-5*   70-99
FCTPTR  9-171#  59-39*  59-40   59-85   59-162  59-188  59-193* 59-204
FCTRCT  59-8#
FCTREV  9-180#  34-13   34-18*  34-21   34-43*  65-79*
FCTRLP  72-14#  72-67
FCTSK1  59-90   59-97#
FCTSKT  59-84   59-96   59-101#
FCTSLP  59-161  59-167#
FCTSP   59-35   59-208  59-212#
FCTSZ   7-94#   28-10
FCTWRT  72-7#
FCWERR  72-69   72-79#
FDAT    7-296#  50-65
FDLL    50-5#
FERR    34-89   34-129# 35-180
FILLIT  59-71   59-179#
FINI    7-231#  40-133  51-119
FIXBLK  61-115  62-6#
FIXFCT  59-93   59-234#
FIXIT   35-65   36-6#
FIXLP   62-15#  62-18
FKIP1   36-21   36-32#
FKIP10  36-41   36-44#
```

FKIP2	36-15	36-17	36-22#											
FKIP9	43-41	43-44#												
FKF1	36-35	36-38	36-40#											
FLAG	9-139#	14-62*	14-68*	15-25*	15-42*	15-50*	15-52*	15-74*	26-9*	33-8*	33-13*	33-15*	33-17*	34-11*
	34-19*	34-30*	34-36*	34-48*	35-56*	37-15*	37-17*	37-26*	37-39*	37-52*	40-9*	40-71*	40-73*	40-74*
	40-133*	42-8*	42-107*	42-117*	42-120*	42-150*	42-181*	42-186*	44-18*	44-23*	44-25*	44-38*	44-40*	44-47*
	44-95*	44-99*	44-101*	44-136*	44-150*	44-156*	44-169*	50-21*	50-127*	50-129*	50-141*	50-155*	51-119*	59-83*
	59-160*	59-186*	59-198*	59-207*	61-105*	62-19*	63-124*	63-126*	63-127*	65-16*	65-32*	65-52*	65-81*	65-83*
	65-84*	66-79*	70-92*											
FLAG1	9-140#	14-40*	14-42*	16-23*	17-43*	19-24*	19-26*	19-39*	35-104*	42-56*	42-106*	42-121*	42-147*	42-179*
	42-180*	42-189*	42-199*	42-213*	42-233*	44-48*	44-162*	44-164*	61-124*	61-155*				
FLGDON	33-14	33-16	33-18	33-19#										
FLGSET	33-4#	33-27												
FLGSKP	33-7	33-9#												
FLIP	58-10	58-17#												
FLKIP1	43-21	43-32#												
FLKIP2	43-15	43-17	43-22#											
FLKP	43-25	43-38	43-40#											
FODO	63-75#	63-119												
FOLOOP	63-71#	63-73												
FORERR	40-56	40-143#	-258											
FORMAT	7-52#	34-84	35-175	40-51	42-253									
FRCPY	7-74#	28-5	50-143											
FRDONE	59-195	59-207#												
FRSKP	59-197	59-204#												
FSER	7-293#	50-28	65-36											
FT.BUF	7-16#	42-140	42-143*	42-164	42-167*	42-243	42-246*	44-20*	44-66*	44-133*	44-166*			
FT.HI	7-18#	35-115	35-153	35-156*	42-67	42-111	42-115	42-123*	42-153	42-159	42-162*	42-182	42-185*	42-192*
	42-198	42-237	44-52*	44-79*	44-86*	44-92*	44-154*	44-161*						
FT.LOW	7-17#	42-124	42-196	44-22*	44-135*									
FULL	7-320#													
FWGOOD	72-47	72-60#												
FWRD	10-18#	26-205	42-223	44-122										
FWTDON	72-71	72-77#												
G1	7-212#	24-20												
G2	7-205#	35-5	39-5	64-11										
G3	7-206#	40-11	65-11											
G4	7-207#	40-140	66-9											
G5	7-208#	59-48												
G7	7-209#	34-76	37-11											
G8	7-210#	40-23	46-9											
GCR	9-48	9-62#												
GDBLK	7-355#	26-201	37-22	42-163	42-236	44-15	44-19	44-165	68-2	68-28				
GDECC	23-14	23-16#												
GDFCT	66-80	66-84#												
GDONE	32-38	32-68#												
GETUNT	32-6#	33-26												
GOBAD	7-223#	33-8	63-124	65-81										
GOVER	32-9#	32-19	32-29	32-82										
GRP	7-255#	21-15	67-82											
GRPCNT	10-39#	34-60*	34-61*	34-62	34-105*	40-28*	40-104*							
GRPCYL	7-92#	26-30	34-60	40-26										
GSKIP	32-48	32-51	32-54	32-56#										
GSKIP1	32-15	32-20#												
GSKIP2	32-25	32-30#												
GSKIP3	32-36	32-39#												
GSR	9-49	9-63#	14-37	32-73*										

```
GET     9-47    9-61#
GETATS  14-12#  15-73   15-163  33-29
H1      7-211#  51-40   54-40   55-7    58-38   59-63   59-128
HASH    57-4#   59-55   59-120
HD.BAD  7-148#  35-155  37-42   42-184  44-85   44-153  51-48
HD..LR  7-150#  26-71   26-91   32-33   32-61   35-154  37-33   37-41   37-47   40-18   42-116  42-160  42-175  42-183
        42-201  44-32   44-50   44-77   44-84   44-90   44-152  44-159  47-7    48-7    51-30   51-52   51-68   51-82
        51-92   51-104  59-43   59-80   59-109  59-116  59-137  59-143  59-157  59-175  59-183  65-65   66-31   67-75
        73-7    73-28   73-35   73-72   73-90
HD.CUR  10-37#  34-49*  34-52*  35-117  37-48
HD.DBN  7-151#  34-49   36-16   73-29
HD.LBN  7-145#  42-69   43-14   44-91   53-31   59-52   60-32   61-71   71-40   73-91
HD.PRV  7-149#  42-122  42-155  42-172  44-51   51-72
HD.RBN  7-146#  42-71   42-113  42-131  42-239  43-16   44-58   44-160  59-113  67-9    68-34   73-73
HD.REV  7-147#  42-161  42-176  42-191  42-202  44-78   66-32
HD.XBN  7-152#  34-52   36-14   50-92   63-27   63-94   70-45   72-35   73-36
HEAD    7-97#   29-105
HERE    35-112# 35-161  35-166
HGHPBN  9-123#  28-29*  28-30*  65-56   65-67
HI1BYT  7-128#  22-19   26-55   26-102  40-94   44-116  48-15   49-18   68-7    73-54
HI2BYT  7-127#  29-37
HIBYTE  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
HKIP1   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*
HOLRBN  9-103#  40-92   40-99   40-100  46-19*  46-20*
HPREA   9-194#  29-108* 34-79   35-170  40-46   42-248  50-96   53-35   55-96   60-36   61-75   63-98   67-44   68-38
        69-97   72-39
HSLIM   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   69-24   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
INITS   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#
ISEEK   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*
ISKIP   27-19   27-22#
LAGAIN  42-22#  42-58
LAST    7-325#  37-98   44-219
```

BC	7-280#	66-67*												
.BNBAD	9-181#	40-136	40-139*	44-100*	66-66									
.BNCVL	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													
.BNFND	73-52	73-78#												
BN*OS	7-91#	26-78	26-79	59-45										
.BNLBN	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										
.BNPCV	9-118#	26-49*	26-50*	26-74	47-9									
.BNTRK	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									
.CHEC	40-70	42-6#	42-265											
DONE	42-62	42-218	42-266#											
.EN	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[illegible]#												
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												
MORE	43-12#	43-45												
LNOERR	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											

'ROSS REFERENCE TABLE (CREF V01-05)

```
LOOP1   15-15#   15-75
LOOP2   15-26    15-28#
LOOP3   50-152#  50-154
LOOPP   50-60    50-72#   50-124
LOOPP2  50-39    50-57#   50-156
LOVER   42-226#  42-230
LOVER1  42-255   42-264#
LOVER2  42-234   42-247#  42-263
LOVER3  42-240   42-243#
LOVER4  42-237#  42-242
LPBN    46-21    49-5#
LRBN    73-16    73-41#
LREDO   43-34    43-39#
LSKIP   42-9     42-14#
LSKIP1  42-18#   42-60
LSKIP2  42-13    42-17#
LSKIP3  42-188   42-208   42-211#
LSKIP4  40-53    40-61#
LSKIP5  40-124#
LSKIP6  42-96    42-100#
LSKIP7  42-66    42-72    42-213#
LSKIP8  42-70    42-73#
LSKIP9  42-200   42-204#
LSND    42-148   42-187   42-194#
LTO      9-166#  15-100   29-55*
LTRK    44-6#
M512     7-327#  65-20    70-10
M576     7-328#  65-22
MAINTR   7-65#   19-7     24-15
MAINTW   7-66#   19-16    24-18
MANU     7-226#  35-56    37-15    37-52
MARBAD  44-43    44-75    44-82#
MASK     7-337#  32-39
MAXTRY   7-330#  67-53
MNCNT    9-176#  59-25*   59-3,    59-194*
MORE    36-12#   36-45
MSG1    15-21    15-25#
MULPC    9-34#   12-43*   12-44*   12-45
N       10-32#   35-58*   35-62*   35-107*  42-10*   42-14*   42-59*
N1      10-33#   35-59*   35-60    35-63*   35-64    35-160*  35-162*  42-11*   42-12    42-15*   42-16    42-209*  42-211*
NDL      7-229#  50-129   50-155
NEXT    16-8#    24-21    26-14    34-27    40-141   50-148
NEXT1    9-163#  50-58*   50-117*  50-123*  53-5*    53-55*   53-61*   53-65    53-70*   60-6*    60-57*   60-63*   60-67    60-72*
        61-55*   61-95*   61-101*  70-29*   70-81*   70-89*   71-16*   71-76*   71-86*   72-9*    72-60*   72-66*   72-70    72-75*
NEXT5   16-11#   16-20
NGD     65-17    65-25    65-28    65-53    65-81#
NN1     10-34#   35-60*   35-64*   35-162   42-12*   42-16*   42-211
NO      50-104   50-117#
NOCRY   20-13    20-15#
NODLL   50-22    50-24#
NOERR   35-72    35-91    35-96    35-101#
NOFCT    7-298#  50-37    65-27
NOGOOD  61-82    61-95#
NOINC   58-20    58-22#
NOLBN   59-53    59-113#
NOSEK   51-120   51-123#
NOTHER  32-18    32-28    32-35    32-58    32-76#
```

CROSS REFERENCE TABLE (CREF V01-05)

```
NOTR      51-49   51-72#
NUM       10-16#  35-88   42-40   42-101
NUMDBN    34-50   35-17#
NUMLBN    46-11   47-5#
NUMRBN    46-16   48-5#
NUMXBN    34-54   35-37#
OCDONE    55-75#  55-117
OCLOOP    55-20#  55-73
OERR      16-13   16-17#
OERR2     16-18   16-21#
OFATAL    63-67   63-122#
OFFS       7-80#  37-70   44-191
OFFSET     9-36#  51-43   54-7    54-36*  58-6    58-34*  59-66   59-131
OK        35-151  35-160#
OLDONE    63-78   63-120#
ONE       10-41#  26-153  28-25   34-124  40-20   40-107  40-125  59-16   72-11
ONEFIL     2-4#    2-8     4-376    5-1
ONLIN     15-179#  33-28
ONLINE     9-57    9-86#
OOPS      33-31   33-34#
OQUIT     63-123  63-125  63-129#
ORFTAL    55-68   55-120#
OV...      9-3     9-3     9-3#    26-5    26-5    26-5    26-5#   34-8    34-8    34-8    34-8#   37-10   37-10   37-10
          37-10#  40-5    40-5    40-5    40-5#   44-2    44-2    44-2    44-2#   46-6    46-6    46-6    46-6#   50-5
          50-5    50-5    50-5#   51-2    51-2    51-2    51-2#   55-2    55-2    55-2    55-2#   56-7    56-7    56-7
          56-7#   61-2    61-2    61-2    61-2#   63-2    63-2    63-2    63-2#   64-7    64-7    64-7    64-7#   65-7
          65-7    65-7    65-7#   66-5    66-5    66-5    66-5#   72-2    72-2    72-2    72-2#   73-2    73-2    73-2
          73-2#   73-96   73-96#
OVCNT      7-191#  25-4
OVE.F1    34-8#   37-10
OVE.F2    40-5#   44-2
OVE.F3    50-5#   51-2
OVE.F4    51-2#   55-2
OVE.F5    56-7#   61-2
OVE.F6    63-2#   64-7
OVE.F7    61-2#   63-2
OVE.F8    44-2#   46-6
OVE.F9    72-2#   73-2
OVE.G1    26-5#   34-8
OVE.G2    64-7#   65-7
OVE.G3    65-7#   66-5
OVE.G4    66-5#   72-2
OVE.G5    73-2#   73-96
OVE.G7    37-10#  40-5
OVE.G8    46-6#   50-5
OVE.H1    55-2#   56-7
OVE.MN     9-3#   26-5
OVER      26-205#  26-212
OVER1     35-177  35-185#
OVER2     35-169#  35-184
OVL...     9-3     9-3#    26-5    26-5#   34-8    34-8#   37-10   37-10#  40-5    40-5#   44-2    44-2#   46-6    46-6#
          50-5    50-5#   51-2    51-2#   55-2    55-2#   56-7    56-7#   61-2    61-2#   63-2    63-2#   64-7    64-7#
          65-7    65-7#   66-5    66-5#   72-2    72-2#   73-2    73-2#   73-96   73-96#
OVL.F1     9-208   37-10   37-10   37-10#
OVL.F2     9-211   44-2    44-2    44-2#
OVL.F3     9-214   51-2    51-2    51-2#
OVL.F4     9-217   55-2    55-2    55-2#
```

OVL.F5	9-220	61-2	61-2	61-2#										
OVL.F6	9-223	64-7	64-7	64-7#										
OVL.F7	9-226	63-2	63-2	63-2#										
OVL.F8	9-229	46-6	46-6	46-6#										
OVL.F9	9-232	73-2	73-2	73-2#										
OVL.G1	9-256	34-8	34-8	34-8#										
OVL.G2	9-235	65-7	65-7	65-7#										
OVL.G3	9-238	66-5	66-5	66-5#										
OVL.G4	9-241	72-2	72-2	72-2#										
OVL.G5	9-244	73-96	73-96	73-96#										
OVL.G7	9-247	40-5	40-5	40-5#										
OVL.G8	9-250	50-5	50-5	50-5#										
OVL.H1	9-253	56-7	56-7	56-7#										
OVL.MN	9-3	9-3	26-5	26-5	26-5#									
OVLBLK	9-259#	50-51*	50-52*	50-53*	50-54	64-24*	64-25*	64-26*	64-27					
OVLEN	7-192#	25-13												
OVLTBL	9-208#	16-8	17-12	17-30	25-11									
OVRLAY	16-11	17-15	17-33	18-7#	50-56	64-29								
OVS.F1	9-209	34-8#												
OVS.F2	9-212	40-5#												
OVS.F3	9-215	50-5#												
OVS.F4	9-218	51-2#												
OVS.F5	9-221	56-7#												
OVS.F6	9-224	63-2#												
OVS.F7	9-227	61-2#												
OVS.F8	9-230	44-2#												
OVS.F9	9-233	72-2#												
OVS.G1	9-257	25-6	26-5#											
OVS.G2	9-236	64-7#												
OVS.G3	9-239	65-7#												
OVS.G4	9-242	66-5#												
OVS.G5	9-245	73-2#												
OVS.G7	9-248	37-10#												
OVS.G8	9-251	46-6#												
OVS.H1	9-254	55-2#												
OVS.MN	9-3#													
PAERR	17-18	17-36	17-39#											
PAERR1	17-40	17-45#												
PAGAIN	58-7#	58-23												
PAGE	17-10#	34-77	35-6	39-6	40-12	40-24	40-44	40-67	40-77	40-80	40-135	45-7	51-41	54-41
	58-39	59-23	59-33	59-38	59-49	59-64	59-129	59-202	59-211	59-216	59-239	59-247	65-15	65-45
	65-51	70-8	70-14											
PAGER	17-21	17-28#												
PALP1	17-15#	17-19												
PALP2	17-17	17-20#												
PALP3	17-35	17-38#												
PALP4	17-33#	17-37												
PARITI	7-137#	20-48												
PARITY	7-136#													
PBNBUF	7-354#	34-41	34-42	34-43	35-8	39-8	45-5	59-214	64-28	65-13	65-18	65-35	65-43	65-49
	65-54	65-62	67-10	67-32										
PCNT	10-51#	34-38*	44-35*	44-104	44-147*	44-179	45-8*	65-61*	65-71*					
PCON	73-6#													
PDONE	73-31	73-38	73-75	73-93#										
PERR	58-42	58-50#												
PNGBLK	58-5#	58-40												
PNGPG	51-61	51-97	54-4#											

```
PNGPNG  58-4#    59-102   59-150   59-168
PNGRD   58-30#   58-45
PRET    58-16    58-48#
PRIM     7-221#  42-117   42-120   42-150   42-181   44-18    44-47    44-156
PRIMRB  22-6#    42-128   44-55    52-7     57-7
PKMBUF   7-356#  42-132   42-139   42-144   44-59    44-65    44-68    70-6     70-9     70-40    70-64    71-34    71-59
PRMY     7-153#  44-45    59-87    59-99    59-165   59-190   59-243
PROD     7-11#    9-66
RB.BUF   7-43#   35-71    36-19*   42-23    43-19*
RB.CMD   7-46#   35-78    42-30
RB.HI    7-45#   35-76    42-28
RB.IM    7-47#   35-93    42-45
RB.LOW   7-44#   35-74    42-26
RBD      7-282#  66-69*   66-70*
RBNBUF   7-360#  42-219   44-118   44-132
RBNLBN   9-114#  26-96*   26-97*   26-160   26-161   26-176   26-177   28-23    38-13    38-34    61-27    61-28    73-12
RBNPLY   9-119#  26-61*   26-62*   26-94    48-9
RBNRPT   7-322#  42-133   44-60    67-11
RBNTRK   7-88#   22-18    26-54    26-101   40-93    44-115   48-14    49-17    68-6     73-53
RBNWRT  67-64    68-1#
RC.FRE   7-158#  51-76    54-16    58-15    59-70    59-135
RC.NUL   7-162#  54-14    58-13    61-150   62-9
RC.PRV   7-159#  51-83    51-93    59-81    59-184
RC.SND   7-160#  51-69    51-105   59-110   59-158   59-176   66-26
RC.UNU   7-161#  51-53    59-72    59-138   59-144
RCD     69-76#   69-118
RCFIX   61-130   61-146#
RCFXLP  60-67#   60-73
RCINDN   7-245#  61-124   61-155
RCINER  61-104   61-157#
RCINIT   7-224#  61-105   62-19
RCINLP  61-54#   61-134
RCL     69-22#   69-74
RCLP    61-10#   61-12
RCLP2   61-56#   61-102
RCLP3   61-151#  61-154
RCLP4   61-131#  61-156
RCLP6   61-125   61-135#
RCTBAD   9-182#  42-109*  44-42*   66-68
RCTBUF   7-359#  51-42    51-62    51-98    53-23    54-12    55-30    55-51    55-87    58-11    59-65    59-103   59-130   59-151
        59-169   60-24    66-23    69-31    69-52    69-88
RCTCK   66-59    71-4#
RCTCKD  71-80    71-89#
RCTCKE  71-88    71-97#
RCTCL1  71-17#   71-87
RCTCLP  71-15#   71-95
RCTCNT  10-50#   51-36*   54-30*   54-31    54-45*   58-29*   58-30    58-43*   59-59*   59-124*  66-19*   66-21    66-47*   66-48*
RCTERR  53-64    53-74#
RCTFMT   9-161#  28-12*   28-13*   53-59    53-68    55-69    60-61    60-70    61-99    61-133   66-48    69-70    71-82
RCTINI  61-6#
RCTLBN   9-175#  26-171*  26-172*  61-129   71-4*    71-94
RCTNGD  71-68    71-76#
RCTRLP  60-8#    60-64
RCTS2    7-95#   28-12
RCTTOT   9-156#  40-22*   44-97*
RCTUPD  51-11#
RCTWLP  53-7#    53-62
```

CROSS REFERENCE TABLE (CREF V01-05)

RCTWRT 59-101 59-149 59-192 60-5#
RCTWT 51-60 51-95 51-107 53-4#
RCV 7-56# 15-31 15-104 32-23
RCVMNT 19-14# 50-48 64-20
RCVRDY 7-168# 15-68 15-198 15-217
RCWERR 60-66 60-75#
RCXLP 53-65# 53-71
RDBLK 9-20# 23-9 35-70 35-82 35-121 35-134 42-22 42-34 42-75 42-87 55-22 55-36 63-20 63-35
69-23 69-37 70-35 70-49 71-29 71-44
RDBUF 7-353# 10-17 35-128 35-147 35-149 36-22 42-82 43-22 50-24 50-35 50-55 50-62 50-72 50-87
50-150 59-21 59-24 59-36 59-39 59-89 59-200 59-209 61-9 61-14 61-63 61-119 61-147 62-10
RDCMD 7-318# 35-83 35-135 42-35 42-88 55-37 63-36 69-38 70-50 71-45
RDLEN 7-313# 35-101 36-20 42-53 43-20
RDONE 65-69 65-75 65-77#
RDONE1 65-47 65-79#
READ 7-53# 35-85 35-138 42-37 42-91 55-42 63-41 69-43 70-55 71-50
READ1 35-82#
READ10 71-44# 71-66 71-72
READ11 55-36# 55-58 55-64
READ2 35-134#
READ3 42-34#
READ4 42-87#
READ7 63-35# 63-57 63-63
READ8 69-37# 69-59 69-65
READ9 70-49# 70-71 70-77
RECAL 15-79# 15-164 19-52 33-32
RECIR 7-326# 37-65 44-186
RECOV 10-54# 29-30* 35-103 42-55 50-119 53-57 55-72 55-76 60-59 61-97 63-72 63-76 68-59 69-73
69-77 70-88 71-78 71-85 72-62
RECTMP 10-56# 19-46* 19-48 29-31* 35-103* 42-55* 50-109 50-113* 50-119* 53-47 53-51* 53-57* 55-59 55-63*
55-72* 55-76* 55-109 55-113* 60-49 60-53* 60-59* 61-87 61-91* 61-97* 63-58 63-62* 63-72* 63-76*
63-111 63-115* 68-51 68-55* 68-59* 69-60 69-64* 69-73* 69-77* 69-110 69-114* 70-72 70-76* 70-88*
71-67 71-71* 71-78* 71-85* 72-52 72-56* 72-62*
REDO 36-34 36-39#
RETRY 10-53# 15-56 15-53 15-96 15-161 16-17 17-39 29-39* 32-17 32-27 34-88 35-179 40-55 42-257
50-105 53-43 55-55 55-105 60-45 61-83 63-54 63-107 68-47 69-56 69-106 70-68 71-63 72-48
REVBUF 7-357# 66-15 66-53 67-5 68-1
REVCNT 9-170# 40-63 40-68* 40-131 42-206* 51-109*
REVECT 7-220# 42-107 42-186 44-40 44-95 44-99
REVLEN 7-315# 67-60 68-61
REVRBN 9-120# 22-24* 22-25* 42-129 42-130 44-56 44-57 52-10 52-11 52-20* 52-21* 52-22* 57-10 57-11
57-20* 57-21* 57-22*
REVSEC 7-79# 27-7
RFCT 33-10 33-15#
RFTL 69-69 69-121#
RINC1 44-175 44-179#
RLD 69-79 69-119#
RLDONE 55-78 55-118#
RLOOP 65-49# 65-60
RLOOP1 65-63# 65-76
RNUMER 68-18# 68-64
ROVER 51-24# 51-110
ROVER1 51-28 51-30#
RPRIM 7-241# 42-121 42-179 42-233 44-48 44-162 44-164
RPT 44-61# 44-64
RPT1 42-134# 42-137
RQUIT 65-82 65-86#

CROSS REFERENCE TABLE (CREF V01-05)

RRC 66-22 69-5#
RRERR 35-87 35-92#
RRPL 59-136 59-141#
RSER 7-305# 61-15
RTCNT 7-285# 66-78*
RTDON 53-66 53-72#
RTRY 7-75# 29-32
RTY 7-233# 15-25 15-42 15-50 15-52 15-74
RTYCNT 10-46# 35-100* 42-51* 66-77
RTYDN 7-240# 35-104 42-56
RUN 9-51 9-71#
RW.BUF 7-29# 35-73* 35-129* 42-25* 42-83* 50-88* 53-26* 55-31* 55-88* 60-27* 61-66* 63-30* 63-89* 67-35*
68-29* 69-32* 69-89* 70-41* 71-35* 72-30*
RW.CMD 7-32# 35-79* 35-133* 42-31* 42-86* 50-86* 53-22* 55-35* 55-86* 60-23* 61-62* 63-34* 63-87* 67-31*
68-27* 69-36* 69-87* 70-39* 71-33* 72-26*
RW.DAT 7-33#
RW.DUM 7-28# 35-81* 35-123* 42-33* 42-77* 50-95* 53-34* 55-24* 55-95* 60-35* 61-74* 63-22* 63-97* 67-42*
68-37* 69-25* 69-96* 70-48* 71-43* 72-38*
RW.EDC 7-34# 35-150 42-146* 44-17* 44-70* 50-74* 53-25* 55-53 60-26* 61-65* 63-52 67-34* 68-4* 69-54
70-66 71-61 72-29*
RW.ER1 7-27#
RW.ER2 7-35#
RW.HI 7-31# 35-77* 35-127* 42-29* 42-81* 50-93* 53-32* 55-29* 55-93* 60-33* 61-72* 63-28* 63-95* 67-40*
68-35* 69-30* 69-94* 70-46* 71-41* 72-36*
RW.LOW 7-30# 35-75* 35-125* 42-27* 42-79* 50-90* 53-28* 55-26* 55-90* 60-29* 61-68* 63-24* 63-91* 67-37*
68-31* 69-27* 69-91* 70-43* 71-37* 72-32*
RW.STA 7-26# 35-84* 42-36* 70-51*
RWCMD 7-316# 35-130 36-28 42-84 43-28 55-32 63-31 69-33 70-36 71-30
RWGD 53-42 53-55#
RWGOOD 60-44 60-57#
RWRDY 7-166# 15-146
RWTDON 60-68 60-74#
SCR 9-30 9-49 9-78# 21-10 26-25 26-68 26-88 26-130 28-9 29-102 30-4 34-29 34-59 34-70
37-69 40-13 40-25 40-37 40-83 42-126 44-7 44-53 44-114 44-190 46-10 50-14 52-6 53-7
55-10 57-6 59-43 60-8 61-44 67-16 67-77 68-5 69-7 70-20 71-17 73-46 73-87
SECCNT 10-31# 35-68* 35-105* 36-6* 36-44* 37-12* 37-59* 42-20* 42-57* 43-6* 43-44* 44-10* 44-107* 44-117*
44-182* 59-26* 59-196* 66-20* 66-45*
SECNDY 51-73 51-97#
SECTCY 9-111# 26-116* 26-117* 35-23 35-40
SECTRK 9-109# 26-106* 26-107* 26-111 26-112 26-219 27-13 29-7 34-98 35-28 35-45 35-68 36-6 37-12
37-85 37-87 40-64 42-20 43-6 44-206 44-208 44-222 50-19 63-17 70-26 72-19 73-41 73-42
SEEK 15-136# 19-53 21-16 34-67 34-92 35-183 40-34 40-59 42-262 51-122 59-226 61-144 67-83
SEEK0 15-138# 15-165
SEEK1 15-142# 15-153
SEEK2 15-145 15-160#
SEEK3 15-149 15-162 15-166#
SEEK4 15-155# 15-167
SEEK5 15-143 15-159#
SEEK6 15-147 15-154#
SEEK7 15-151# 15-152
SEEKER 40-36 40-146#
SEND 7-55# 15-19 15-92 32-13
SER 7-277# 29-82
SERNUM 9-179# 29-83 50-27 61-13 65-37
SETOVL 24-19 25-4#
SFTRPT 32-65# 32-67
SHORTQ 7-73# 29-62

CROSS REFERENCE TABLE (CREF V01-05)

```
SIP      7-61#  35-67  35-120  42-19  42-74  50-101  53-39  55-41  55-101  60-41  61-79  63-40  63-103  67-49
        68-43   69-42  69-102  70-54  71-49  72-44
SIXTN    7-133#
SKERR   34-69   34-132#
SKIP1   34-51   34-55#
SKIP12  34-123#
SKIP19  34-44   34-48#
SKIP3   34-76#  34-103
SKIP4   34-37   34-47  54-49#
SKIP5   37-53   37-56  37-59#
SKIP6   34-86   34-94#
SKIP7   34-78#  34-93
SKPCNT   9-152# 27-18* 27-21*  36-9   43-9
SLEEK   34-59#  34-127
SLEEK2  34-64#  34-106
SLOOP   25-12#  25-15
SND     42-118  42-189#
SND1     7-281# 66-12*
SNDCNT  10-45#  42-178* 42-193* 42-203* 44-73* 66-1   66-39*
SNDMNT  19-5#   19-36   34-122  40-123  50-47  64-19  66 86
SRCK    14-38   14-45#
ST       9-47    9-50    9-51    9-52    9-53   9-54   9-55   9-56   9-57   9-76#  14-16  14-23  32-31
ST.DB    7-179# 14-53
ST.DBN   9-199# 30-36* 35-26   38-18   73-27
ST.DF    7-183# 14-18
ST.DR    7-176# 14-27
ST.ERB   7-185# 14-17
ST.ERR   7-184#
ST.FO    7-180# 14-49
ST.IN    7-181#
ST.LBN   9-196# 22-10   28-28  30-10*  47-12  49-9   49-25  51-81  51-91  51-103 53-30  55-28  55-92  59-79
        59-182  60-31   61-70  67-39   69-29  69-93  71-39  73-89
ST.PS    7-175# 14-33
ST.RBN   9-197# 30-16* 48-12   52-22   57-22  68-33  73-71
ST.RU    7-174# 14-30
ST.SR    7-178# 14-45
ST.WE    7-182# 14-58
ST.WP    7-177#
ST.XBN   9-198# 30-26* 35-43   38-39   50-9   50-10  50-79  63-26  63-93  70-16  70-17  70-33  72-34  73-34
STACK    9-132# 24-12
STARIT   9-151# 36-11* 36-40   36-43*  43-11* 43-40  43-43*
START    9-4    16-10  16-16   17-14   17-23  17-32  24-1#  26-5   26-5   34-8   34-8   37-10  37-10  40-5
        40-5    44-2   44-2    46-6    46-6   50-5   50-5   51-2   51-2   55-2   55-2   56-7   56-7   61-2
        61-2    63-2   63-2    64-7    64-7   65-7   65-7   66-5   66-5   72-2   72-2   73-2   73-2
START2  56-11   59-7#
START3  24-1    24-12#
STATFR  20-45   20-53#
STATRE  20-43#  20-47  20-49
STATRT  20-52#  20-55
STATST  14-14#  14-44  14-48  14-52  14-56  14-61
STATUS   7-58#  20-43
STATVL  15-62   15-66  15-142 15-194 15-213 20-41#
STCKSV   9-133# 15-53* 15-55
STCLR    7-131# 30-9   30-15  30-25  30-35
STCYL    7-86#  26-149 53-12  55-11  60-13  61-49  67-21  68-13  69-8   71-22
STDBN    7-85#  30-30
STDIAG  14-50   14-53#
```

CROSS REFERENCE TABLE (CREF V01-05)

STFORM 14-46 14-49#
STLBN 7-82# 30-8
STO 9-167# 15-141 29-74*
STPNIC 14-28 14-31 14-34 14-41 14-73#
STRBN 7-83# 30-14
STSC 7-257#
STSK1 14-20 14-23#
STSKP 14-63 14-69#
STWLK 14-54 14-57#
STXBN 7-84# 30-20
SWAP 51-77 51-86#
SWRD 10-19# 26-207 42-224 44-123
TALIP1 15-54 15-56#
TALK 14-15 14-65 14-67 15-13# 15-122 15-130 15-140 15-173 15-181 19-50
TALKDN 15-33 15-37#
TALKIP 15-51 15-55#
TALKP 15-43 15-45#
TALKRT 15-38 15-42#
TATTN1 15-69 15-73#
TBLK 9-153# 27-17* 27-20* 36-32 43-32
TCLEAR 15-24 15-36 15-41 15-50#
TEMP 9-37# 22-7* 22-9 26-32* 26-33* 26-35 26-44* 26-45* 26-46 26-56* 26-57* 26-58 26-69* 26-72*
26-73 26-89* 26-92* 26-93 26-121* 26-122* 26-123 26-132* 26-135* 26-136 26-160* 26-161* 26-164 26-176*
26-177* 26-180 26-185 26-220* 26-221* 26-222 27-10* 27-11* 27-12 27-15* 28-20* 28-21* 28-22 29-4*
29-5* 29-6 29-9 29-14* 29-15* 29-18 29-21 35-20* 35-21* 35-22 35-37* 35-38* 35-39 37-78*
37-79* 37-82 37-89* 37-90* 37-91 38-8* 38-9* 38-10 38-31* 38-32* 38-33 44-199* 44-200* 44-203
44-210* 44-211* 44-212 47-5* 47-6* 47-7* 47-8 48-5* 48-6* 48-7* 48-8 49-5* 49-6* 49-7
50-75* 50-76* 50-77 52-8* 52-9* 52-13 57-8* 57-9* 57-13 61-27* 61-28* 61-29 61-107 62-8
67-6* 68-62* 68-63* 70-30 70-31* 70-32* 70-33* 73-55* 73-56* 73-57 73-63* 73-64* 73-65 73-81*
73-82* 73-83
TEMP2 9-33# 51-50* 51-51* 51-58* 51-64 51-86* 51-87* 51-96 59-74* 59-75* 59-105 59-141* 59-142* 59-153
68-18* 68-19* 68-21
TERR 15-63 15-67 15-70#
THREB 7-312# 27-17
TILOP 29-67# 29-69
TILOP1 29-71# 29-73
TIMER 15-64 15-65 15-107 20-25#
TIMLOP 29-52# 29-54
TIMLP 20-2?# 20-32
TIMVAL 7-329# 20-26
TKIP1 37-18 37-21#
TKIP10 37-52#
TKIP11 37-68 37-75#
TKIP2 37-16 37-20 37-22# 37-61
TKIP3 37-27 37-31 37-35 37-46#
TKIP4 37-45 37-51#
TKIP5 37-74 37-78#
TKIP6 37-77 37-97#
TKIP7 37-38 37-40#
TKIP8 37-85# 37-88
TKIP9 37-86 37-89#
TMPTRY 10-55# 29-40* 35-102* 42-54* 50-105* 50-107* 50-112* 50-118* 53-43* 53-45* 53-50* 53-56* 55-55* 55-57*
55-62* 55-71* 55-75* 55-105* 55-107* 55-112* 60-45* 60-47* 60-52* 60-58* 61-83* 61-85* 61-90* 61-96*
63-54* 63-56* 63-61* 63-71* 63-75* 63-107* 63-109* 63-114* 67-53* 67-55* 67-58* 68-47* 68-49* 68-54*
68-58* 69-56* 69-58* 69-63* 69-72* 69-76* 69-106* 69-108* 69-113* 70-68* 70-70* 70-75* 70-87* 71-63*
71-65* 71-70* 71-77* 71-84* 72-48* 72-50* 72-55* 72-61*
TOTRC 10-48# 26-82* 26-83* 40-22 51-20 59-29 61-36 71-7

CROSS REFERENCE TABLE (CREF V01-05)

```
TRK       7-256#  21-12   67-79
TRKCNT   10-40#   34-73*  34-102* 40-40*  40-101*
TRKCYL    9-116#  26-37*  26-38*  26-47   26-59   26-114
TRKGRP    7-93#   26-26   34-71   40-38
TWO      27-16    27-20#
TWOB      7-311#  27-20
TWOC     10-43#   59-235
TWRD     10-20#   26-209  42-225  44-124
UHASH    51-32    52-4#
UHKIP    52-8#    52-23
UHKIP1   52-5     52-20#
UID       7-338#  32-32   32-60
UN.ERI    9-12#   15-86*  15-95*  15-96*  15-99*  32-16*  32-17*  32-20*  32-79*  34-87*  34-88*  34-94*  35-178* 35-179*
         35-185*  40-54*  40-55*  40-61*  42-256* 42-257* 42-264*
UN.ERR    9-10#   15-34*  15-39*  15-44*  15-58*  16-14*  16-17*  16-19*  17-20*  17-39*  17-41*
UN.ERT    9-11#   15-22*  15-27*  15-56*
UN.SEK    9-13#   15-155* 15-160* 15-161*
UNIT      9-43#   15-18   15-30   15-60   15-91   15-103  15-138  15-210  32-8*   32-12   32-76   32-78*  33-24   34-82
         35-66    35-119  35-137  35-173  40-49   42-18   42-73   42-90   42-251  50-98   53-37   55-39   55-98   60-38
         61-77    63-38   63-100  67-46   68-40   69-40   69-99   70-52   71-47   72-41
UNSEC     7-141#  15-37
UPDATE    7-69#   34-108  40-109  50-131  53-72   59-205  61-131  70-94   71-89
UPDPNT   10-47#   51-23*  51-24   51-45   51-108*
UREAD     7-62#   18-10
UWRITE    7-63#
V1        7-250#  50-16*  50-18*  53-14*  53-16*  55-14*  55-16*  60-15*  60-17*  61-51*  61-53*  63-14*  63-16*  67-23*
         67-25*   68-15*  68-17*  69-11*  69-13*  70-23*  70-25*  71-24*  71-26*  72-16*  72-18*
V2        7-251#  21-7*   21-9*   67-73*  67-76*
V3        7-252#  50-20*  53-11*  55-19*  60-12*  61-48*  63-18*  67-20*  68-9*   69-16*  70-27*  71-21*  72-20*
V4        7-253#  68-12*  68-67*
VLD       7-134#  20-46
VLD1      7-135#
WLOOP    67-12#   67-15
WP        7-239#  14-40   14-42
WRBLK     9-19#   50-82   50-99   53-18   53-38   55-82   55-99   60-19   60-39   61-58   61-78   63-83   63-101  67-27
         67-47    68-23   68-41   69-83   69-100  72-22   72-42
WRCMD     7-317#  50-83   53-19   55-83   60-20   61-59   63-84   67-28   68-24   69-84   72-23
WRFLG     9-142#  54-5*   54-43   54-46*  58-4*   58-41   58-44*
WRIT10   69-97#   69-109  69-115
WRIT12   55-96#   55-108  55-114
WRIT13   69-38#   68-50   68-56
WRITE     7-54#   50-102  53-40   55-102  60-42   61-80   63-104  67-50   68-44   69-103  72-45
WRITE1   50-96#   50-108  50-114
WRITE2   53-35#   53-46   53-52
WRITE3   60-36#   60-48   60-54
WRITE4   61-75#   61-86   61-92
WRITE5   72-39#   72-51   72-57
WRITE8   63-98#   63-110  63-116
WRITE9   67-44#   67-56
XAGAIN    54-8#   54-24
XBBAD     9-184#  34-24*  66-74
XBD       7-284#  66-75*  66-76*
XBNCYL    7-89#   26-121  26-137  34-53   35-37
XBNFND   73-22    73-32#
XBNIT    34-31    34-52#
XBNSEC    9-115#  26-125* 26-126* 38-8    38-9    73-20   73-24
XDONE    40-132   40-136#
```

R S REFERENCE TABLE (CREF V01-05)

XFCRET	54-15	54-43#					
XFLIP	54-11	54-18#					
XNGBLK	54-6#	54-4[illegible]					
XNGINC	54-21	54-23#					
XPBA	37-21	38-5*#					
XPERR	54-44	54-52#					
XPNGRD	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	40-8*#				
XSKIP6	40-75	40-79#					
XSLEEK	40-25#	40-128	40-130				
XSLEK2	40-31#	40-105					
XYZ	59-112	59-114	59-140	59-167	59-178	59-187	59-192#
XYZ1	59-47	59-193#					
YES	70-73	70-81#					

CROSS REFERENCE TABLE (CREF V01-05)

.MODE	1-3#	9-3												
.MEND	1-32#	73-96												
.MOV	8-65#													
.MOVLV	1-18#	9-3	26-5	34-8	37-10	40-5	44-2	46-6	50-5	51-2	55-2	56-7	61-2	63-2
	64-7	65-7	66-5	72-2	73-2									
DUBSNC	8-106#	37-51	37-54	44-94	44-103	44-167	50-133	59-248	61-126	66-43	70-96	71-91	72-77	
MSG	8-56#	9-47	9-48	9-49	9-50	9-51	9-52	9-53	9-54	9-55	9-56	9-57		
SVTERM	9-3	9-3#	9-3#	26-5	26-5#	34-8	34-8#	37-10	37-10#	40-5	40-5#	44-2	44-2#	46-6
	46-6#	50-5	50-5#	51-2	51-2#	55-2	55-2#	56-7	56-7#	61-2	61-2#	63-2	63-2#	64-7
	64-7#	65-7	65-7#	66-5	66-5#	72-2	72-2#	73-2	73-2#	73-96				
POP	8-32#	12-23	12-50	12-88	14-69	14-70	15-45	15-46	15-111	15-203	16-15	17-28	20-17	20-33
	20-50	20-53	22-26	35-9	35-187	39-9	45-9	51-111	51-114	51-117	54-50	59-212	59-217	59-220
	59-223	61-135	61-138	61-141										
POPA	8-85#	55-118	67-65	69-119										
PUSH	8-9#	12-14	12-37	12-69	14-12	14-13	15-13	15-14	15-88	15-188	17-11	20-6	20-25	20-41
	22-6	35-4	35-53	39-4	45-4	51-13	51-15	51-17	54-4	59-10	59-12	59-14	59-19	61-22
	61-24	61-26												
PUSHA	8-74#	55-6	67-4	69-5										
ROR	8-97#	20-10												