11/Z1+
RL01/02

RL01/2 PERF EXER

CNRLKA0

AH-T752A-MC

FICHE 1 OF 1

APR 1984

digital

Made In USA

```
                        .TITLE  CNRLKA0 RL01/02 PERF EXER
                        .NLIST TOC
                        .ENABLE AMA
000000                  .ENABLE ABS
                        .REM @
```

IDENTIFICATION

PRODUCT CODE: AC-T751A-MC

PRODUCT NAME: CNRLKA0 RL01/2 PERFORMANCE EXERCISER

PRODUCT DATE: DECEMBER 19, 1983

MAINTAINER: ISS DIAGNOSTIC SERVICES

AUTHOR: JAMES S. DOUCETTE

```
53
54
55
56
57                                          REVISION HISTORY
58                                          --------  ----
59
60          CHANGES MADE TO CZRLKB0 IN PRODUCING CNRLKA0 FOR THE SBC-11/21+ (FALCON-PLUS),
61          DEC. 19, 1983.  CHANGES ARE IDENTIFIED BY ";JSD REV A".
62
63                  1. CHANGED THE FORM OF THE ARGUMENT TO ALL "DELAY" AND "WAITUS" MACRO
64                        CALLS FROM #<VALUE> TO <VALUE>. THE FORMER GAVE ASSEMBLY ERRORS
65                        UNDER THE VAX/VMS DEVELOPMENT ENVIRONMENT (MCR MAC).
66
67                  2. CHANGED THE WAITMS MACRO DEFINITION SO THAT "ARG" IS USED AS A DELAY
68                        COUNTER, RATHER THAN "#ARG".
69
70                  3. CHANGED THE GENERAL OPERATING PRIORITY OF THE PROGRAM FROM LEVEL 7 TO
71                        LEVEL 6 TO ALLOW THE "BREAK" KEY TO INVOKE ODT. (THE TRAP
72                        HANDLER AND CLOCK INTERRUPT SERVICE ROUTINES STILL RUN BRIEFLY
73                        AT LEVEL 7).
74
75                  4. SET VECTOR 140 WITH THE ADDRESS OF ODT IN ROM (170000).
76
77                  5. FORCED THE PROGRAM TO ASSUME THAT NO CLOCK IS PRESENT.  AS A RESULT,
78                        (A) REMOVED THE ELAPSED TIME FROM ONLY THOSE FORMATTED MESSAGES
79                        WHICH COULD BE PRINTED  (I.E., SOME TIME MESSAGES WILL NEVER
80                        BE PRINTED), (B) REMOVED THE DROPPED / RUNNING MESSAGE PRINTED
81                        IN THE STATISTIC REPORT, AND (C) REMOVED THE SW QUESTION
82                        REGARDING THE TIME BETWEEN REPORTS.  REASON: UNDER FALCON-PLUS,
83                        CLOCK OPERATION IS NOT GUARANTEED.  CLOCK INTERRUPTS MAY OR MAY
84                        NOT BE HARD-ENABLED, AND EVEN IF THEY WERE, THE INTERRUPT RATE
85                        COULD BE 50, 60, OR 800 HERTZ.  FURTHERMORE, THE DRS CLOCK
86                        MACROS RETURN MISLEADING INFORMATION (UNDER FALCON-PLUS).
87
```

TABLE OF CONTENTS

1.0 GENERAL INFORMATION

1.1 PROGRAM ABSTRACT

1.1.1 STRUCTURE OF PROGRAM

THIS DIAGNOSTIC IS COMPATIBLE WITH BOTH CNDP+ AND ACT IT CAN BE RUN STANDALONE UNDER CNDP+, AND CAN BE CHAINED UNDER CNDP+, ACT AND APT IN ACT MODE (SEE 2.2 "CHAIN MODE OPERATION" FOR DETAILS OF CHAINING PROCEDURE). IT IS A SINGLE PROGRAM FROM THE STANDPOINT OF THE DIAGNOSTIC USER, WHICH AT RUN TIME IS APPENDED TO A COMMON FRONT-END PIECE OF SUPERVISOR SOFTWARE THROUGH WHICH THE DIAGNOSTIC PROGRAM INTERFACES TO THE ENVIRONMENT AS IT EXECUTES. (IN THIS DOCUMENT, "CNDP+" REFERS TO THE FALCON-SPECIFIC XXDP+ SYSTEM).

WHEN THIS DIAGNOSTIC IS STARTED, CONTROL GOES FIRST TO THE SUPERVISOR PORTION, WHICH WILL ASK CERTAIN "HARD CORE" QUESTIONS ABOUT THE ENVIRONMENT. THEN IT WILL ENTER COMMAND MODE, INDICATED BY A PROMPT CHARACTER (DR>). AT COMMAND MODE THE OPERATOR MAY ENTER ANY OF SEVERAL COMMANDS AS DESCRIBED IN 2.0 "OPERATING INSTRUCTIONS".

THE DIAGNOSTIC PROGRAM IS LOADED IN THE LOWER 8K OF MEMORY. THE DIAGNOSTIC SUPERVISOR CODING OCCUPIES 6.25K OF THE UPPER PART OF MEMORY JUST BELOW THE CNDP+ MONITOR WHICH RESIDES IN THE UPPERMOST 1.5K OF MEMORY SPACE.

1.1.2 DIAGNOSTIC INFORMATION

THE RL11/RLV11 RL01/02 EXERCISER IS A KXT-11 (SBC-11/21+) BASED PROGRAM. IT WILL RANDOMLY EXERCISE UP TO 2 CONTROLLERS AND 8 DRIVES. AFTER AN INITIAL WRITE OF EACH RL01/02, THE DRIVES ARE RANDOMLY PICKED AND GIVEN A RANDOM STRING FUNCTION OF:

1. SEEK, WRITE, WRITE-CHECK
2. SEEK, READ DATA, DATA COMPARE
3. SEEK, READ HEADERS, READ 1 SECTOR W/NO HEADER COMPARE, GET STATUS
4. SEEK, READ, READ

1.2 SYSTEM REQUIREMENTS

1.2.1 HARDWARE REQUIREMENTS

```
192
193                         • SBC 11/21• PROCESSOR, 28KW MEMORY, JUMPERED FOR MEMORY MAP 0
194
195                         • CONSOLE DEVICE (LA30,LA36,VT50,ETC.)
196
197                         • 1 OR 2 RL11/RLV11 CONTROLLER(S) WITH:
198
199                                 1 - 8 RL01 DRIVES WITH RL01K CARTRIDGES CONTAINING  A  'BAD
200                                     SECTOR FILE'
201                                 1   8 RL02 DRIVES WITH RL02K CARTRIDGES CONTAINING  A   BAD
202                                     SECTOR FILE'
203
204                         • CNDP• (XXDP•) LOAD DEVICE (RL02, RX02, ETC.)
205
206                         • LINE PRINTER (OPTIONAL)
207
208
209                 1.2.2       SOFTWARE REQUIREMENTS
210                             -------- ------------
211
212                         CNRLKAO RL11/RLV11 RL01/RL02 PERFORMANCE EXERCISER
213                                 (FORMERLY CZRLEB0)
214
215
216                 1.3         RELATED DOCUMENTS AND STANDARDS
217                             ------- ------- - --- -- ------
218
219                         RL01 DISK SUBSYSTEM USER'S GUIDE (EK-RL01-UG-002)
220                         XXDP•/SUPERVISOR USER'S MANUAL
221
222
223                 1.4         DIAGNOSTIC HIERARCHY PREREQUISITES
224                             -----  - - --------- - -----------
225
226                         1HE RL01/02 SUBSYSTEM SHOULD HAVE SUCCESSFULLY  RUN  THE  FOLLOWING
227                         PROGRAMS:
228
229                             CVRLABC                     RLV11 RL01 DISKLESS TEST (RLV11 ONLY)
230                             CNRLGAO                     RL11/RLV11 RL01/02 CONTROLLER TEST (PART 1)
231                             CNRLHAO                     RL11/RLV11 RL01/02 CONTROLLER TEST (PART 2)
232                             CNRLIAO                     RL01/02 DRIVE TEST (PART 1)
233                             CNRLJAO                     RL01/02 DRIVE TEST (PART 2)
234
235
236                 1.5         ASSUMPTIONS
237                             -----------
238
239                         THE HARDWARE OTHER THAN THE  RL01  SUBSYSTEM  IS  ASSUMED  TO  WORK
240                         PROPERLY.   FALSE ERRORS MAY BE REPORTED IF THE PROCESSOR, ETC., DO
241                         NOT FUNCTION PROPERLY.
242
243
244                 2.0         OPERATING INSTRUCTIONS
245                             --------- -- -- - ----
246
```

2.1 HOW TO RUN THIS DIAGNOSTIC

2.1.1 THE FIVE STEPS OF EXECUTION

THIS DIAGNOSTIC PROGRAM SHOULD BE LOADED AND STARTED USING NORMAL XXDP+ PROCEDURES. START THE EXECUTION OF THE CNDP+ MONITOR BY USING THE APPROPRIATE BOOTSTRAP PROGRAM. THE MONITOR WILL PRINT A MESSAGE IDENTIFYING ITSELF AND REQUESTING THAT THE CURRENT DATE BE ENTERED. AN EXAMPLE OF THIS MESSAGE IS GIVEN BELOW FOR THE CNDP+ MONITOR:

```
CNMDYA0 CNDP+ DY MONITOR
BOOTED VIA UNIT 0
ENTER DATE (DD-MMM-YY):
```

AFTER THE DATE HAS BEEN ACCEPTED BY THE MONITOR THE RESTART ADDRESS OF THE MONITOR IS PRINTED. THEN THE FOLLOWING TWO QUESTIONS ARE ASKED:

```
50 HZ ? N
LSI ?  N
```

THE DEFAULTS ARE BOTH "NO". TYPE "R" AND THE PROGRAM NAME TO RUN THE PROGRAM. DO NOT TYPE THE EXTENSION.

WHEN THIS DIAGNOSTIC IS STARTED THE FOLLOWING STEPS WILL OCCUR:

```
**********
* STEP 1 *
**********
```

THE DIAGNOSTIC WILL ISSUE THE PROMPT "DR>". FROM THIS POINT UNTIL THE TIME WHEN YOU RESTART CNDP+, YOU WILL BE TALKING TO THE DIAGNOSTIC, NOT CNDP+. WE WILL REFER TO THE PRESENCE OF THIS PROMPT AS BEING IN DIAGNOSTIC COMMAND MODE, AS OPPOSED TO CNDP+ COMMAND MODE.

AT THIS POINT YOU WILL ENTER A "START" COMMAND. THIS IS NOT THE SAME AS THE CNDP+ "START" COMMAND, WHICH YOU ALREADY ISSUED IN RESPONSE TO THE CNDP+ DOT PROMPT. THIS "START" COMMAND CAN TAKE A NUMBER OF SWITCHES AND FLAGS (ALL OPTIONAL) AND THE DETAILS OF THESE ARE SET FORTH IN 2.3 "DETAILS OF COMMANDS AND SYNTAX". HOWEVER, IN ORDER TO USE THE PROGRAM, ALL YOU NEED TO SAY IS SOMETHING LIKE THIS:

```
STA/PASS:1/FLAGS:HOE
```

THINGS TO NOTE HERE:

1. ONLY THE FIRST THREE CHARACTERS OF THIS OR ANY COMMAND AT THE "DR>" LEVEL NEED TO BE TYPED.

2. THE "PASS" SWITCH SPECIFIES HOW MANY PASSES YOU DESIRE. A PASS CONSISTS OF RUNNING THE FULL DIAGNOSTIC AGAINST ALL UNITS BEING TESTED (THIS WILL BE EXPLAINED SHORTLY). ONE PASS IS SPECIFIED IN THE ABOVE EXAMPLE.

3. THE "FLAGS" SWITCH MAY SPECIFY ANY OF A NUMBER OF FLAGS, BUT THE MAIN USEFUL ONES ARE:

PNT	PRINT NUMBER OF TEST BEING EXECUTED
LOE	LOOP ON ERROR
HOE	HALT ON ERROR
IER	INHIBIT ERROR PRINTOUT

THE HOE FLAG IS SPECIFIED IN THE ABOVE EXAMPLE (WE'LL SEE WHY SHORTLY).

****
* STEP 2 *
****

WHEN YOU HAVE TYPED IN A "START" COMMAND, THE DIAGNOSTIC WILL COME BACK WITH THE QUESTION "# UNITS?" TO WHICH YOU SHOULD RESPOND BY TYPING IN THE NUMBER OF DEVICES YOU WISH TO TEST.

A WORD OF WARNING HERE: THE NUMBER OF UNITS DEPENDS ON THE TARGET DEVICE OF THE DIAGNOSTIC. FOR EXAMPLE, IF THE DIAGNOSTIC IS DIRECTED AT A DISK DRIVE, THEN THE NUMBER OF UNITS WOULD BE THE NUMBER OF DRIVES TO BE TESTED. WHEREAS IF THE DIAGNOSTIC WAS DIRECTED AT THE DISK CONTROLLER, THEN THE NUMBER OF UNITS WOULD BE THE NUMBER OF CONTROLLERS. THE TARGET DEVICE OF A DIAGNOSTIC CAN ALWAYS BE DETERMINED BY INSPECTING THE "HEADER" STATEMENT NEAR THE BEGINNING OF THE SOURCE CODE. ONE OF THE OPERANDS OF THIS "HEADER" STATEMENT SHOULD BE THE DEVICE TYPE OF THE DIAGNOSTIC.

****
* STEP 3 *
****

WHEN YOU HAVE TYPED IN THE NUMBER OF UNITS TO BE TESTED, THE DIAGNOSTIC WILL ASK YOU THE "HARDWARE QUESTIONS". THE ANSWERS TO THESE QUESTIONS ARE USED TO BUILD TABLES IN CORE, CALLED "HARDWARE P-TABLES". ONE HARDWARE P-TABLE WILL BE BUILT FOR EACH UNIT TO BE TESTED.

THERE ARE SEVERAL HARDWARE QUESTIONS AND THE ENTIRE SERIES WILL BE POSED N TIMES, WHERE N IS THE NUMBER OF UNITS.

THIS REPRESENTS A NEW PHILOSOPHY IN DIAGNOSTIC ENGINEERING. DIAGNOSTICS IN THE FUTURE WILL NOT BE WRITTEN TO AUTOSIZE OR ASSUME STANDARD ADDRESSES: INSTEAD, THEY WILL ASK THE OPERATOR FOR ALL THE INFORMATION THEY NEED TO TEST THE DEVICE.

```
                                    **********
                                    * STEP 4 *
                                    **********

AFTER YOU HAVE ANSWERED ALL THE HARDWARE QUESTIONS (SEC 2.5) FOR
ALL THE UNITS, YOU WILL BE ASKED "CHANGE SW? IF YOU WANT TO BE
ASKED THE SOFTWARE QUESTIONS THAT DETERMINE THE BEHAVIOR OF THIS
PROGRAM, TYPE "Y". IF YOU WANT TO TAKE ALL THE DEFAULTS TO THESE
QUESTIONS, TYPE "N". IF YOU TYPE "Y' YOU WILL BE ASKED THE
SOFTWARE QUESTIONS (SEC 2.6), AND THE ANSWERS WILL BE PUT INTO THE
SOFTWARE P-TABLE IN THE PROGRAM. THE SERIES OF QUESTIONS WILL BE
ASKED JUST ONCE, REGARDLESS OF THE NUMBER OF UNITS TO BE TESTED.


                                    **********
                                    * STEP 5 *
                                    **********

AFTER YOU HAVE ANSWERED THE SOFTWARE QUESTIONS, THE DIAGNOSTIC WILL
BEGIN TO EXECUTE THE HARDWARE TEST CODE. THERE ARE SEVERAL THINGS
THAT CAN HAPPEN NEXT, DEPENDING ON WHETHER A HARDWARE ERROR IS
ENCOUNTERED AND ALSO ON WHAT SWITCH VALUES YOU SELECTED ON THE
START COMMAND. CONSIDER THE POSSIBILITIES:

    1.  IF NO ERROR IS ENCOUNTERED, THEN THE DIAGNOSTIC WILL SIMPLY
        EXECUTE THE DESIRED NUMBER OF PASSES AND RETURN TO COMMAND
        MODE (PROMPT DR>).

    2.  IF AN ERROR IS ENCOUNTERED, THEN ONE OF THREE THINGS
        HAPPENS, DEPENDING ON THE SETTINGS OF THE HOE AND LOE
        FLAGS.

        HOE SET:  THE ERROR WILL BE REPORTED ON THE CONSOLE AND THE
        DIAGNOSTIC WILL RETURN TO COMMAND MODE.

        LOE SET:  THE DIAGNOSTIC WILL LOOP ENDLESSLY ON THE BLOCK
        OF CODE THAT DETECTED THE ERROR.

        NEITHER HOE NOR LOE SET:  THE ERROR WILL BE REPORTED ON THE
        CONSOLE AND NORMAL EXECUTION WILL RESUME AS IF NO ERROR HAD
        OCCURRED.
```

2.1.2 SAMPLE RUN THROUGH

LET'S SEE HOW ALL THIS WORKS IN A REAL SITUATION. RECALL THAT WE ENTERED THE COMMAND "STA/PASS:1/FLAGS:HOE". THIS WOULD BE A VERY TYPICAL WAY TO RUN THE DIAGNOSTIC. IF NO ERRORS ARE ENCOUNTERED, THE SINGLE REQUESTED PASS WILL BE EXECUTED AND THE PROMPT WILL BE RE-ISSUED.

IF AN ERROR IS ENCOUNTERED, THE ERROR WILL BE REPORTED AND THE PROMPT WILL BE REISSUED (BECAUSE THE HOE FLAG IS SET). AT THIS POINT THERE ARE FOUR DIFFERENT WAYS YOU CAN GET THE PROGRAM GOING AGAIN:

1. ISSUE ANOTHER "START" COMMAND (THUS GOING THRU ALL OF STEPS 1, 2, 3, 4, AND 5 AGAIN)

2. ISSUE A "RESTART" COMMAND (SAME AS START COMMAND EXCEPT THAT THE HARDWARE QUESTIONS ARE NOT ASKED)

3. ISSUE A "CONTINUE" COMMAND (EXECUTION WILL RESUME AT THE BEGINNING OF THE PARTICULAR HARDWARE TEST (MOST DIAGNOSTICS CONSIST OF A NUMBER OF THESE) THAT IT WAS IN WHEN THE ERROR HALT OCCURRED. NO QUESTIONS ASKED.

4. ISSUE A "PROCEED" COMMAND: EXECUTION WILL RESUME AT THE INSTRUCTION FOLLOWING THE ERROR REPORT (THIS IS A SPECIAL COMMAND AND CAN BE ISSUED ONLY AT A HALT ON ERROR).

THE MOST TYPICAL THING TO DO HERE IS TO ISSUE THE PROCEED, BUT WITH DIFFERENT FLAG SETTINGS. PROBABLY YOU WOULD WANT TO SAY:

```
PRO/FLAGS:IER:LOE:HOE=0
```

THIS WILL DO THE FOLLOWING:

1. TURN ON THE IER (INHIBIT ERROR PRINTOUT) FLAG

2. TURN ON THE LOE FLAG

3. TURN OFF THE HOE FLAG

4. RESUME EXECUTION AT INSTRUCTION AFTER ERROR REPORT

THE DIAGNOSTIC WILL NOW LOOP ON THE BLOCK OF CODE THAT DETECTED AND REPORTED THE ERROR, BUT NO ERROR PRINTOUT WILL OCCUR. THUS YOU CAN STUDY THE ERROR OR SCOPE IT OR WHATEVER.

WHEN YOU VE SEEN ENOUGH, YOU MAY HIT CONTROL/C. THIS WILL TAKE YOU OUT OF THE LOOP AND PUT YOU BACK INTO COMMAND MODE. YOU NOW HAVE THREE CHOICES:

1. START

2. RESTART

3. CONTINUE

LET'S SAY YOU'VE REPAIRED THE DEFECT FOUND ABOVE AND WANT TO FINISH RUNNING THE DIAGNOSTIC. YOU WOULD TYPE

    CON/FLAGS:HOE:IER=0:LOE=0

THIS WILL RESTORE THE FLAGS TO THEIR ORIGINAL VALUES AND RESUME EXECUTION AT THE BEGINNING OF THE HARDWARE TEST YOU WERE IN. IF THE ERROR DOES NOT RECUR, THE EXECUTION WILL FLOW RIGHT ON THRU TO THE NEXT ERROR OR TO END OF PASS.

IF AT END OF PASS YOU WANT TO RUN THE DIAGNOSTIC AGAIN, YOU HAVE TWO CHOICES:

1. START

2. RESTART

YOU WOULD CHOOSE ONE, DEPENDING ON WHETHER YOU WANTED TO ANSWER THE HARDWARE QUESTIONS AGAIN.

```
492
493                                THE FULL PRINT-OUT FROM THE ABOVE DIALOGUE  MIGHT  LOOK  LIKE  THIS
494                                (O=OPERATOR, D=DIAGNOSTIC):
495
496
497                                                                                BY
498                                                                                WHOM
499                                                                                ENTERED:
500                                                                                --- -
501
502                                .R NRLKA0                                       O
503                                DRS LOADED                                      D
504                                DIAG. RUN-TIME SERVICES REV. D APR-79           D
505                                CNRLK-A-0                                       D
506                                CNRLK RANDOMLY PERFORMS DRIVE SEEK, READ,       D
507                                AND WRITE FUNCTIONS
508                                UNIT IS RL01, RL02                              D
509
510
511                                DR>STA/PASS:1/FLAGS:HOE                         D.O
512
513                                CHANGE HW (L) ? Y                               D.O
514
515                                # UNITS (D) ? 2                                D.O
516
517                                UNIT 0                                          D
518                                RL11 (L) Y ?                                    D.O
519                                BUS ADDRESS (O) 174400 ?                        D.O
520                                VECTOR (O) 160 ?                                D.O
521                                DRIVE (O) 0 ?                                   D.O
522                                DRIVE TYPE = RL01 (L) Y ?                       D.O
523                                BR LEVEL (O) 5 ?                                D.O
524
525                                UNIT 1                                          D
526                                RL11 (L) Y ?                                    D.O
527                                BUS ADDRESS (O) 174400 ?                        D.O
528                                VECTOR (O) 160 ?                                D.O
529                                DRIVE (O) 0 ? 1                                 D.O
530                                DRIVE TYPE = RL01 (L) ? N                       D.O (N=RL02)
531                                BR LEVEL (O) 5 ?                                D.O
532
533                                CHANGE SW (L) ? N                               D.O
534
535
536                                CNRLK HRD ERR 00004 TST 003 SUB 002 PC:004130
537                                ERR HLT
538
539                                DR>PRO/FLAGS:IER:LOE:HOE=0                      D.O
540
541
542                                ************************ ****************************
543                                AT THIS POINT THE DIAGNOSTIC IS LOOPING ON THE
544                                ERROR WITHOUT PRINTING ANYTHING.  YOU CAN SCOPE
545                                THE ERROR UNTIL YOU HAVE LOCATED IT, THEN ^C OUT.
546                                ************************************************
547
548                                ^C                                              O
```

```
DR>CON/FLAGS:HOE:IER:LOE=0                    D,0

CHANGE SW (L) ? N                             D,0

CNRLK EOP 1                                   D
^C

DR>RESTART/PASS:1                             D,0

CHANGE SW (L) ? N                             D,0
```

2.2 CHAIN MODE OPERATION

CHAIN MODE OPERATION CONSISTS OF THE SEQUFNTIAL EXECUTION OF PROGRAMS WITHOUT OPERATOR INTERVENTION. ONLY PROGRAMS THAT HAVE BEEN MODIFIED TO RUN IN CHAIN MODE CAN BE CHAINED. CHAINABLE PROGRAMS ARE IDENTIFIED IN THE DIRECTORY BY A BIC EXTENSION.

TO RUN CHAIN MODE, THE CNDP+ MONITOR USES AN ASCII FILE (KNOWN AS A CHAIN FILE) LISTING THE PROGRAMS TO BE RUN AND THE NUMBER OF PASSES EACH PROGRAM SHOULD RUN. THIS FILE MUST BE ON THE SYSTEM DEVICE.

A CHAIN FILE MAY BE GENERATED BY USE OF THE XTECO TEXT EDITOR. THIS FILE MUST HAVE A CCC EXTENSION. THE CHAIN FILE MAY CONTAIN ANY OF THE COMMANDS SUPPORTED BY THE CNDP+ MONITOR. THE COMMANDS IN THE ASCII FILE ARE EXECUTED IN THE ORDER IN WHICH THEY ARE ENCOUNTERED. COMMENTS MAY BE INCLUDED IN THE FILE.

TO EXECUTED A CHAIN FILE THE USER TYPES:

```
C FILNAM <CR> OR
C FILNAM/QV  <CR>
```

IN THE FIRST CASE THE PASS COUNT SPECIFIED IN THE CHAIN FILE IS USED BY THE CNDP+ MONITOR TO DETERMINE THE NUMBER OF PASSES TO EXECUTE EACH PROGRAM. IN THE SECOND CASE THE PROGRAM COUNT IS NOT USED AND EACH PROGRAM IS EXECUTED ONLY ONCE. THE /QV SWITCH PROVIDES A SINGLE EXECUTION MODE OF OPERATION OF QUICK VERIFY.

WHEN PROGRAMS ARE RUN IN CHAIN MODE, THE SOFTWARE SWITCH REGISTER SHOULD BE SET TO 000000. THE CNDP+ MONITOR PRINTS EACH COMMAND TAKEN FROM THF CHAIN FILE AND THEN EXECUTES THE COMMAND. WHEN THE LAST COMMAND OTHER THAN ANOTHER C COMMAND HAS BEEN EXECUTED THE CNDP+ MONITOR TERMINATES CHAIN MODE AND TYPES A PROMPT (.). IT IS READY TO ACCEPT ANOTHER COMMAND FROM THE CONSOLE. IF THE LAST COMMAND IS ANOTHER C COMMAND, THE CHAIN MODE WILL CONTINUE AND THE CHAIN FILE SPECIFIED BY THIS NEW C COMMAND WILL BE USED.

IF THE USER WISHES TO TERMINATE CHAIN MODE BEFORE ITS NORMAL TERMINATION HE MAY DO SO BY TYPING A CONTROL/C. HOWEVER, THE MONITOR WILL NOT ABORT THE CHAIN MODE UNTIL IT RECEIVES PROGRAM CONTROL FROM THE PROGRAM CURRENTLY RUNNING.

2.3 DETAILS OF COMMANDS AND SYNTAX

2.3.1 TABLE OF COMMAND VALIDITY

THERE ARE FOUR WAYS OF ENTERING DIAGNOSTIC COMMAND MODE, AND DIFFERENT SUBSETS OF THE DIAG COMMAND SET ARE AVAILABLE WITH EACH:

HOW ENTERED	LEGAL COMMANDS
1. OPERATOR ENTERED 'RUN DIAG'	START PRINT DISPLAY FLAGS ZFLAGS EXIT
2. DIAGNOSTIC HAS FINISHED ALL ITS REQUESTED PASSES	START RESTART PRINT DISPLAY FLAGS ZFLAGS EXIT
3. OPERATOR INTERRUPTED THE DIAGNOSTIC WITH CTRL/C	START RESTART CONTINUE PRINT DISPLAY FLAGS ZFLAGS EXIT
4. AN ERROR WAS ENCOUNTERED WITH THE HOE FLAG SET SET	START RESTART CONTINUE PROCEED PRINT DISPLAY FLAGS ZFLAGS EXIT

2.3.2 COMMAND SYNTAX

```
**********************************************************************
STA(RT)/TESTS:TEST-LIST/PASS:PASS-CNT/FLAGS:FLAG-LIST/EOP:EOP-INCR
**********************************************************************
```

THE DIAGNOSTIC IN CORE IS EXECUTED IN ACCORDANCE WITH THE SWITCHES SPECIFIED. THE MESSAGE "# UNITS?" IS PRINTED. THE START COMMAND MAY BE ISSUED WHEN DIAGNOSTIC COMMAND MODE HAS BEEN ENTERED VIA ONE OF THE FOLLOWING: A) OPERATOR TYPED "RUN DIAGNOSTIC" B) DIAGNOSTIC FINISHED EXECUTING C) ERROR WAS ENCOUNTERED WITH HOE FLAG SET D) OPERATOR ENTERED CONTROL/C. AFTER THE OPERATOR RESPONDS TO "# UNITS?", THE HARDWARE DIALOGUE IS INITIATED. WHEN IT IS COMPLETED, THE QUESTIONS "CHANGE SW?" IS ISSUED, AND THE ANSWERS, IF GIVEN, BECOME THE NEW DEFAULTS. THEREFORE IT IS NECESSARY TO RELOAD THE PROGRAM IN ORDER TO RETURN TO THE LOAD DEFAULTS.

THE SWITCH ARGUMENTS ARE AS FOLLOWS:

"TEST-LIST" IS A SEQUENCE OF DECIMAL NUMBERS (1:2 ETC.) OR RANGES OF DECIMAL NUMBERS (1-5:8-10 ETC.) THAT SPECIFY THE TESTS TO BE EXECUTED. THE NUMBERS ARE SEPARATED BY COLONS. THE NUMBERS RANGE FROM 1 TO THE LARGEST TEST NUMBER IN THE DIAGNOSTIC. THEY MAY BE SPECIFIED IN ANY ORDER. TESTS WILL BE EXECUTED IN NUMERICAL ORDER REGARDLESS OF THE ORDER OF SPECIFICATION. THE DEFAULT IS TO EXECUTE ALL TESTS.

"PASS-CNT" IS A DECIMAL NUMBER INDICATING THE DESIRED NUMBER OF PASSES. A PASS IS DEFINED AS THE EXECUTION OF THE FULL DIAGNOSTIC (ALL SELECTED TESTS) AGAINST ALL UNITS SUBMITTED. THE DEFAULT IS NON-ENDING TEST EXECUTION. "FLAG-LIST" IS A SEQUENCE OF ELEMENTS OF THE FORM <FLAG>, <FLAG=1>, OR <FLAG=0>, SEPARATED BY COLONS. WHERE <FLAG> HAS ONE OF THE FOLLOWING VALUES:

HOE HALT ON ERROR, CAUSING COMMAND MODE TO BE ENTERED WHEN AN ERROR IS ENCOUNTERED

LOE LOOP ON ERROR, CAUSING THE DIAGNOSTIC TO LOOP CONTINUOUSLY WITHIN THE SMALLEST DEFINED BLOCK OF CODING (SEGMENT, SUBTEST, OR TEST) CONTAINING THE ERROR

IER INHIBIT ERROR REPORTING

IBE INHIBIT BASIC ERROR REPORTS

IXE INHIBIT EXTENDED ERROR REPORTS

PRI DIRECT ALL MESSAGES TO A LINE PRINTER

PNT PRINT NUMBER OF TEST BEING EXECUTED

```
713
714                                 BOE BELL ON ERROR
715
716                                 UAM RUN IN UNATTENDED MODE, BYPASSING MANUAL INTERVENTION TESTS
717
718                                 ISR INHIBIT STATISTICAL REPORTS
719
720                                 IDU INHIBIT DROPPING OF UNITS BY DIAGNOSTIC
721
722                                 ADR EXECUTE AUTODROP CODE
723
724                                 LOT LOOP ON TEST
725
726                                 EVL EVALUATE
727
728                             THE FLAGS NAMED OR EQUATED TO 1 ARE SET, THOSE EQUATED TO 0 ARE
729                             CLEARED.  A FLAG NOT SPECIFIED IS CLEARED.  IF THE FLAGS SWITCH IS
730                             NOT GIVEN ALL FLAGS ARE CLEARED.
731
732                             "EOP-INCR" IS A DECIMAL NUMBER INDICATING HOW OFTEN (IN TERMS OF
733                             PASSES) IT IS DESIRED THAT THE END OF PASS MESSAGE BE PRINTED.  THE
734                             DEFAULT IS AT THE END OF EVERY PASS.
735
736                             ********************************************************************
737                             RES(TART)/TEST:TEST-LIST/PASS:PASS-CNT/FLAGS:FLAG-LIST/EOP:EOP-INCR/UNITS:UNIT-LIST
738                             ********************************************************************
739
740                             THE DIAGNOSTIC IN CORE IS EXECUTED IN ACCORDANCE WITH THE SWITCHES
741                             SPECIFIED.  HOWEVER, NEW "P-TABLES" ARE NOT BUILT.  INSTEAD, THE
742                             ONES IN CORE ARE USED.
743
744                             THE QUESTION "CHANGE SW?" IS ASKED AND THE ANSWERS GIVEN BECOME THE
745                             NEW DEFAULTS.  THE COMMAND MAY BE ISSUED WHEN COMAND MODE HAS BEEN
746                             ENTERED VIA A) DIAGNOSTIC IS FINISHED B) HALT ON ERROR C)
747                             CONTROL/C.
748
749                             THE SWITCH ARGUMENTS ARE AS IN THE START COMMAND EXCEPT:
750
751                                 1.  "UNIT-LIST" IS A SEQUENCE OF LOGICAL UNIT NUMBERS RANGING
752                                     FROM 1 THRU N (N = NUMBER OF UNITS BEING TESTED) SPECIFYING
753                                     WHICH UNITS ARE TO BE TESTED.  THE LOGICAL UNIT NUMBER
754                                     DESIGNATES THE POSITION OF THE P-TABLE IN CORE, ACCORDING
755                                     TO THE ORDER IN WHICH THEY WERE BUILT.  THE UNITS SPECIFIED
756                                     MUST NOT HAVE BEEN DROPPED BY THE OPERATOR DROP COMMAND.
757                                     THE UNIT-LIST DEFAULTS TO "ALL THAT HAVE NOT BEEN DROPPED
758                                     BY OPERATOR COMMAND".  THE EFFECT OF THE UNIT-LIST LASTS
759                                     UNTIL THE NEXT START (WHERE IT IS AUTOMATICALLY RESET TO
760                                     "ALL") OR THE NEXT RESTART.
761
762                                 2.  ALL UNSPECIFIED FLAG SETTINGS ARE UNCHANGED.
763
```

```
765
766                     ********************************************
767                     CON(TINUE)/PASS:<PASS-CNT/FLAGS:<FLAG LIST>
768                     ********************************************
769
770                     COMMAND MODE MUST HAVE BEEN ENTERED DUE TO A HALT ON ERROR OR A
771                     CONTROL/C.  THE EFFECT OF THE COMMAND IS TO GO TO THE BEGINNING OF
772                     THE TEST THAT WAS BEING EXECUTED WHEN THE HALT OR CONTROL/C TOOK
773                     PLACE.  SOFTWARE DIALOGUE MAY OPTIONALLY BE RE-EXECUTED.  HARDWARE
774                     PARAMETERS MAY NOT BE CHANGED.
775
776                     THE SWITCH ARGUMENTS ARE AS IN THE START COMMAND EXCEPT:
777
778                         1.  DEFAULT FOR PASS-CNT IS THE UNSATISFIED PASS-CNT FROM THE
779                             PREVIOUS START OR RESTART
780
781                         2.  UNSPECIFIED FLAG SETTINGS ARE UNCHANGED
782
783
784                     ****************************
785                     PRO(CEED)/FLAGS:<FLAG-LIST>
786                     ****************************
787
788                     COMMAND MODE MUST HAVE BEEN ENTERED VIA A HALT ON ERROR.  THE
789                     EFFECT OF THE COMMAND IS TO BEGIN EXECUTION AT THE LOCATION
790                     FOLLOWING THE ERROR CALL.  NEITHER HARDWARE NOR SOFTWARE PARAMETERS
791                     MAY BE ALTERED.
792
793                     THE SWITCH ARGUMENTS ARE THE SAME AS THE START COMMAND EXCEPT:
794
795                         1.  UNSPECIFIED FLAG SETTINGS ARE UNCHANGED
796
797
798                     ****
799                     EXIT
800                     ****
801
802                     RETURN TO CNDP> PROMPT MODE.
803
804
805                     ************************
806                     DRO(P)/UNITS:UNIT-LIST
807                     ************************
808
809                     THE UNITS SPECIFIED ARE DROPPED FROM TESTING UNTIL THEY ARE ADDED
810                     BACK OR UNTIL A START COMMAND IS GIVEN.  A DROP CANNOT BE FOLLOWED
811                     BY A PROCEED.
812
813                     THERE IS ALSO A "DROP" MACRO INTERNAL TO THE DIAGNOSTIC, WHICH
814                     GIVES THE FACILITY OF AUTO-DROPPING.  THE DURATION OF A PROGRAM
815                     DROP, HOWEVER, IS ONLY UNTIL THE NEXT START OR RESTART.
```

```
817
818                         **********
819                         ADD/UNITS:UNIT-LIST
820                         **********
821
822                         THE UNITS SPECIFIED ARE ADDED BACK (THEY MUST HAVE BEEN  PREVIOUSLY
823                         DROPPED  BY  THE DROP COMMAND) TO THE TEST SEQUENCE.  AN ADD CANNOT
824                         BE FOLLOWED BY A PROCEED.
825
826
827                         *******
828                         PRI(NT)
829                         *******
830
831                         ALL STATISTICS TABLES ACCUMULATED BY THE  DIAGNOSTIC  ARE  PRINTED.
832                         THE ISR (INHIBIT STATISTICAL REPORTING) FLAG IS CLEARED.
833
834                         ****************************
835                         DIS(PLAY)/UNITS:<UNIT-LIST>
836                         ****************************
837
838                         THE HARDWARE P-TABLES FOR ALL UNITS UNDER TEST ARE PRINTED  OUT  IN
839                         THE FORMAT IN WHICH THEY WERE ENTERED.  ANY UNITS THAT WERE DROPPED
840                         BY THE OPERATOR "DROP" COMMAND ARE SO DESIGNATED.
841
842
843                         *******
844                         FLA(GS)
845                         *******
846
847
848                         THE CURRENT SETTINGS OF ALL FLAGS ARE PRINTED.
849
850                         ********
851                         ZFL(AGS)
852                         ********
853
854                         ALL FLAGS ARE CLEARED.
855
856
857                2.4      EXTENDED P-TABLE DIALOGUE
858                         -------- ------- --------
859
860                         THE FULL CAPABILITY OF THE HARDWARE DIALOGUE  IS  REVEALED  BY  THE
861                         FOLLOWING DISCUSSION OF WHAT HAPPENS INTERNALLY.
862
863                         AS SOON AS THE QUESTION "# UNITS?" IS ANSWERED (WITH THE NUMBER N),
864                         SPACE  IN  CORE IS ALLOCATED FOR "N" P-TABLES.  ALL OF THE P-TABLES
865                         ARE OF THE SAME FORMAT, AND THERE IS  A  ONE-TO-ONE  CORRESPONDENCE
866                         BETWEEN  THE  HARDWARE  PARAMETER  QUESTIONS  AND  THE SLOTS IN THE
867                         P-TABLE FORMAT.
868
869                         IN GIVING A STRING OF VALUES, COMMAS WITHOUT INTERVENING VALUES MAY
870                         BE USED TO INDICATE A REPETITION OF THE LAST NAMED VALUE.
```

```
872
873                     A STRING OF VALUES MAY BE GIVEN AS A RANGE (6 10 FOR EXAMPLE).  IF
874                     THE  VALUES  REPRESENT  PURE  NUMERICAL  DATA,  THIS  SAMPLE  RANGE
875                     TRANSLATES TO THE STRING 6,7,8,9,10 (AN INCREMENT OF  1).  IF  THE
876                     VALUES  ARE  ADDRESSES,  THE  SAMPLE RANGE TRANSLATES TO THE STRING
877                     6,8,10 (AN INCREMENT OF 2).
878
879                     NOW LET US SEE HOW WE COULD USE THESE CAPABILITIES TO  CONSTRUCT  A
880                     SET  OF  P-TABLES.  ASSUME THAT WE HAVE 8 RL UNITS, AND THAT THERE
881                     ARE FIVE (5) HARDWARE PARAMETERS FOR EACH (5 SLOTS IN THE  P TABLE,
882                     5 HARDWARE QUESTIONS IN THE DIALOGUE).
883
884                     FOLLOWING  IS  THE DIALOGUE FOR  THIS  8  RLOX  DRIVE  SYSTEM.  THIS
885                     SYSTEM  HAS  TWO  (2)  RL11  TYPE  CONTROLLERS ALL TO BE SET AT "BR
886                     LEVEL" 5.  THE FIRST 4 DRIVES ARE RL01 S AND THE LAST 4 DRIVES  ARE
887                     RL02'S (ON THE SECOND CONTROLLER):
888
889                     # UNITS (D) ? 8
890
891                     UNIT 0
892                     RL11 (L) Y ?
893                     BUS ADDDRESS (O) 174400 ?
894                     VECTOR (O) 160 ?
895                     DRIVE (O) 0 ? 0-3
896                     DRIVE TYPE = RL01 (L) Y ?
897                     BR LEVEL (O) 5 ?
898
899                     UNIT 4
900                     RL11 (L) Y ?
901                     BUS ADDRESS (O) 174400 ? 175400
902                     VECTOR (O) 160 ? 164
903                     DRIVE (O) 0 ? 0-3
904                     DRIVE TYPE = RL01 (L) Y ? N
905                     BR LEVEL (O) 5 ?
906
907
908                     THE FIRST TIME THRU THE P-TABLE QUESTIONS THE  DEFAULT  VALUES  ARE
909                     USED  FOR  THE  CONTROLLER  TYPE  (QUESTION #1), CSR ADDRESS OF THE
910                     CONTROLLER  (QUESTION  #2),  THE  CONTROLLER  VECTOR  ASSIGNMENT
911                     (QUESTION  #3),  THE  DRIVE  TYPE (QUESTION #5), AND THE "BR LEVEL"
912                     (QUESTION #6).  THE ACTUAL UNIT NUMBERS OF THE RL01'S FOR  QUESTION
913                     #4 WAS ASSIGNED 0 THRU 3 FOR THE FIRST 4 P-TABLE SLOTS.
914
915                     THE SECOND TIME THRU THE P-TABLE QUESTIONS (FOR THE RL02 ASSIGNMENT
916                     ON  THE  SECOND CONTROLLER), THE FIRST QUESTION DEFAULTED TO "RL11"
917                     TYPE CONTROLLER.  THE SECOND QUESTION WAS ANSWERED TO  REFLECT  THE
918                     CHANGE IN CSR ADDRESS FOR THE RL02 CONTROLLER (175400).  THE SECOND
919                     CONTROLLER'S VECTOR WAS ALSO CHANGED TO 164 IN  QUESTION  #3.  THE
920                     RL02  TEST  UNIT NUMBERS WERE ASSIGNED VALUES 0 TO 3 IN QUESTION #4
921                     AND THE DRIVE TYPE WAS SET FOR RL02'S FOR THE REMAINING 4 UNITS  IN
922                     QUESTION  #5.  THE LAST QUESTION WAS DEFAULTED USING THE "BR LEVEL'
923                     FROM THE FIRST PASS.
```

2.5 HARDWARE PARAMETERS

THE FOLLOWING QUESTIONS WILL BE ASKED ON A START COMMAND. THE VALUE LOCATED TO THE LEFT OF THE QUESTION MARK IS THE DEFAULT VALUE THAT WILL BE TAKEN ON A CARRIAGE RETURN RESPONSE.

RL11 (L) Y?

ANSWER YES(Y) IF YOU HAVE AN RL11 CONTROLLER, NO(N) IF YOU HAVE AN RLV11 CONTROLLER.

BUS ADDRESS (O) 174400?

ANSWER WITH THE BUS ADDRESS OF THE CONTROLLER.

VECTOR (O) 160?

ANSWER WITH THE INTERRUPT VECTOR OF THE CONTROLLER.

DRIVE (O) 0?

ANSWER WITH THE DRIVE(S) CONNECTED TO THE CONTROLLER

DRIVE TYPE = RL01 (L) ?

ANSWER NO (N) IF DRIVE IS AN RL02

BR LEVEL (O) 5?

ANSWER WITH THE INTERRUPT PRIORITY OF THE CONTROLLER.

2.6 SOFTWARE PARAMETERS

THE FOLLOWING QUESTIONS ARE ASKED IF REQUESTED ON A START, RESTART, OR CONTINUE. THEY ALLOW FLEXIBILITY IN THE WAY THE PROGRAM BEHAVES. THE SOFTWARE PARAMETERS GIVE THE PROGRAM FLEXIBILITY IN THE WAY IT RUNS. THE PARAMETERS CAN BE MODIFIED ON A START, RESTART, OR CONTINUE BY ANSWERING (Y)ES TO THE FOLLOWING QUESTION:

"CHANGE S.W. ?"

A YES ANSWER WILL ASK THE FOLLOWING SOFTWARE PARAMETER QUESTIONS, WITH THE PRESENT DEFAULT VALUE PRINTED TO THE LEFT OF THE QUESTION MARK. (THE LAST ANSWER GIVEN IS THE DEFAULT) THE DEFAULT IS TAKEN ON A <CR>. CONTROL Z (↑Z) WILL DEFAULT ALL REMAINING QUESTIONS AND START THE TEST.

"RETRY LMT (D) 1 ?"

THIS IS THE NUMBER OF TIMES THE PROGRAM WILL ATTEMPT A COMMAND BEFORE IT QUITS AND REPORTS A HARD ERROR. IF THE RETRY IS SUCCESSFUL BEFORE THE RETRY LIMIT IS EXCEEDED IT WILL PRINT AND LOG A SOFT ERROR.

LIMITS 0 - 65,535

"SEEK RETRY LMT (D) 1 ?'

THIS IS THE NUMBER OF RETRYS THAT WILL BE ATTEMPTED TO SEEK TO A CYLINDER ON A MIS-SEEK. AFTER RETRY IS EXHAUSTED, WE WILL NOT TRY FOR THAT CYLINDER BUT CONTINUE WITH A NEW CYLINDER.

LIMITS 0 - 65,535

"DATA DMP ON DCK ERR (L) Y ?"

GIVES THE ABILITY TO SEE THE 1 SECTOR BUFFER THAT HAD A DATA CRC ERROR. THE RESULTS OF THE PRINTOUT ARE ONE OF TWO POSSIBILTIES.

1. ONLY THOSE WORDS OF THE SECTOR THAT WERE BAD ARE PRINTED WITH WHAT WAS EXPECTED.

2. IF ONE OF THE 1ST TWO WORDS IS BAD (USED TO KEY) THE ENTIRE BUFFER IS DUMPED.

LIMITS Y OR N

"# OF ERR DUMPED (D) 128 ?"

THIS IS THE NUMBER OF MISCOMPARES THAT WILL BE PRINTED.

LIMITS 0 - 128

****
*** THE FOLLOWING QUESTION, ORIGINALLY PART OF CZRLKB, HAS BEEN *
*** REMOVED FROM CNRLK. STATISTICAL REPORTS WILL ONLY BE PRINTED *
*** IF THE OPERATOR ISSUES THE DRS "PRINT" COMMAND. ;JSD REV A *
****

"TIME BETW REPORTS (MIN) (D) 240 ?"

THIS IS THE INTERVAL BETWEEN AUTOMATIC STATISTICAL REPORTS ON ALL DRIVES IF A CLOCK IS PRESENT AND WAS ANSWERED SO IN THE INITIAL DIALOGUE.

LIMITS 1 - 65,535

"DROP DR ON ERR LMTS REACHED (L) Y ?"

GIVES THE ABILITY TO AUTOMATICALLY STOP TESTING ON A DRIVE ONCE ONE OF THE ERROR LIMITS HAVE BEEN EXCEEDED (SEEK, DRIVE, HARD, SOFT). IF THE ANSWER IS YES THEN THE FOLLOWING FOUR QUESTIONS WILL BE ASKED, IF NO THEN THE NEXT QUESTION WILL BE 2.3.13.11.

LIMITS Y OR N

```
1039
1040                    'HRD ERR LMT (D) 3 ?"
1041
1042                    THIS IS THE LIMIT OF HARD ERRORS THAT A DRIVE WILL BE  DROPPED  ON.
1043                    A HARD ERROR IS ONE ON WHICH THE RETRY HAS BEEN EXHAUSTED.
1044
1045                                    LIMITS 1    65,535
1046
1047
1048                    "SFT ERR LMT (D) 10 ?"
1049
1050                    THIS IS THE LIMIT OF SOFT ERRORS THAT A DRIVE WILL BE  DROPPED  ON.
1051                    A SOFT ERROR IS AN ERROR ON AN OPERATION THAT WAS SUCCESSFUL WITHIN
1052                    THE RETRY LIMIT.
1053
1054                                    LIMITS 1 - 65,535
1055
1056
1057                    "DATA MISCOMPARE LIMIT (D) 10 ?"
1058
1059                    THIS IS THE LIMIT OF IN CORE MISCOMPARES THAT  THE  DRIVE  WILL  BE
1060                    DROPPED ON.
1061
1062                                    LIMITS 1 - 65,535
1063
1064
1065                    "SK ERR LMT (D) 3 ?"
1066
1067                    THIS IS THE LIMIT OF MIS-SEEK AND TRACKING ERRORS THAT A DRIVE WILL
1068                    BE DROPPED.
1069
1070                                    LIMITS 1 - 65,535
1071
1072
1073                    "DR ERR LMT (D) 3 ?"
1074
1075                    THIS IS THE LIMIT OF DRIVE ERRORS THAT A DRIVE WILL BE DROPPED ON.
1076
1077                                    LIMITS 1 - 65,535
1078
1079
1080                    "DROP DR ON OPER LMTS REACHED (L) N ?"
1081
1082                    GIVES THE ABILITY TO STOP TESTING ON  A  DRIVE  THAT  HAS  EXCEEDED
1083                    CERTAIN  OPERATION LIMITS (SEEK, BITS TRANSFERRED).  THE DRIVE WILL
1084                    BE DROPPED ONLY WHEN BOTH HAVE BEEN EXCEEDED.  IF THE ANSWER IS YES
1085                    THEN THE NEXT TWO QUESTIONS WILL BE ASKED.
1086
1087                                    LIMITS Y OR N
1088
1089
1090                    "DATA XFER LMT (*10(10)) (D) 25000 ?"
```

```
1092
1093                    THIS IS THE LIMIT OF COMBINED BITS READ/WRITTEN (*10(10)) ON  WHICH
1094                    THE DRIVE WILL BE DROPPED.
1095
1096                                    LIMITS 1    65,535
1097
1098
1099                    "SK LMT (*10(3)) (D) 10000 ?"
1100
1101                    THIS IS THE LIMIT OF SEEK OPERATIONS (*10(3)) ON  WHICH  THE  DRIVE
1102                    WILL BE DROPPED.
1103
1104                                    LIMITS 1 - 65,535     (*10(3))
1105
1106
1107                    "DO YOU WANT TO CHANGE SEEK, R/W PARAMETERS (L) N ?"
1108
1109                    THE NORMAL  OPERATION  IS  TO  SEEK  AND 'TRANSFER  ON  THE  ENTIRE
1110                    CARTRIDGE,  CYLINDERS 0 - 255.  (RL01) OR 511.  (RL02), SECTORS 0 -
1111                    39 AND BOTH SURFACES.  THE NORMAL TRANSFER IS RANDOM BETWEEN 3  AND
1112                    1280 WORDS.
1113
1114                    THE NEXT 8 PARAMETERS WILL ALLOW THE USER TO CONFINE THE TESTING TO
1115                    ANY CONTIGUOUS SECTION OF THE CARTRIDGE AND CONTROL THE SIZE OF THE
1116                    TRANSFERS.
1117
1118                    A YES ANSWER WILL ASK THE NEXT 13 QUESTIONS.
1119
1120
1121                    "STIPULATE R/W XFER SIZE (L) N ?"
1122
1123                    THE PROGRAM WILL NORMALLY MAXIMIZE THE TRANSFER SIZE BY  USING  ALL
1124                    OF  MEMORY  (<28K)  AVAILABLE.   THIS QUESTION IF ANSWERED YES WILL
1125                    RESTRICT THE BUFFER TO THOSE VALUES GIVEN IN  NEXT  TWO  QUESTIONS.
1126                    QUESTION IS 2.3.13.19.
1127
1128                                    LIMITS Y OR N
1129
1130
1131                    "MAX XFER (D) 2560 ?"
1132
1133                    REPRESENTS THE MAXIMUM AMOUNT OF WORDS TO READ OR WRITE
1134
1135                                    LIMITS 3 - 5120
1136
1137
1138                    "MIN XFER (D) 3 ?"
1139
1140                    REPRESENTS THE MINIMUM AMOUNT OF WORDS TO READ OR WRITE
1141
1142                                    LIMITS 3 - 5120
```

"RD ONLY (L) N ?"

GIVES THE ABILITY TO INHIBIT WRITING THE PACK WHILE TESTING. THE INITIAL WRITE OF THE PACK FROM THE START COMMAND WILL STILL OCCUR.

LIMITS Y OR N

"RAN PAT (L) Y ?"

NORMAL OPERATION SHOULD BE YES, BUT THIS PARAMETER WILL ALLOW THE WRITING OF ONLY ONE PATTERN OF EIGHT NORMAL PATTERNS. THE PATTERNS IN NEXT QUESTION.

LIMITS Y OR N

"WHICH ONE (O) 4 ?"

IT IS NOW POSSIBLE TO CONTAIN THE EXERCISER IN WRITING ONLY ONE OF THE FOLLOWING EIGHT PATTERNS:

0 - ALL 0'S

1 - 177777,177777,177777,52525,52525,52525
177777,177777,52525,52525,177777,52525
177252,177252,172765,172765

2 - 0,0,0,177777,177777,177777
0,0,177777,177777,0,177777,0,177777
0,177777

3 - 25252,52525,52525,125252,125252,125252
52525,52525,125252,125252,52525,125252
52525,125252,52525,125252

WORST CASE DATA

4 - 155555,133333,66666,155555,133333,66666
155555,133333,66666,155555,133333,66666
155555,133333,66666,155555

5 - 121105,150442,64221,132110,55044,26422
13211,105504,42642,21321,110550,44264
22132,11055,104426,42213

6 - ALL 1'S

7 - 45513,122645,151322,64551,132264,55132
26455,113226,45513,122645,151322,64551
132264,55132,26455,113226

LIMITS 0 - 7

"WORDS PER SECTOR COMPARED ON READ (D) 16 ?"

NORMAL TRANSFERS ARE RANDOM BETWEEN 3 AND 1280 WORDS. THIS PARAMETER WILL ALLOW YOU TO SPECIFY HOW MANY WORDS SHOULD BE COMPARED PER SECTOR IN CORE AFTER EACH READ. IF THE VALUE SPECIFIED IS GREATER THAN THAT READ IN ONLY THE NUMBER READ IN ARE COMPARED. THE FEWER WORDS COMPARED IN CORE ON EACH READ THE FASTER THROUGHPUT THE EXERCISER WILL HAVE.

LIMITS 0 - 128

"0 OF DATA ERR RPT'D PER BUF (D) 3 ?"

THIS PARAMETER WILL LIMIT THE NUMBER OF IN CORE MISCOMPARES PRINTED. THE PROGRAM WILL CONTINUE TO COMPARE AS MANY WORDS AS SPECIFIED BUT WILL INHIBIT THE PRINTOUT ONCE THIS LIMIT IS REACHED. AFTER ALL WORDS ARE CHECKED A SUMMARY WILL BE PRINTED:

X WORDS BAD OUT OF 128 WORDS READ

LIMITS 0 126

"MAX HD (D) 1 ?"

REPRESENTS MAXIMUM HEAD TO USE IN SEEK OPERATIONS.

LIMITS 0 - 1

"MIN HD (D) 0 ?"

REPRESENTS MINIMUM HEAD TO USE IN SEEK OPERATIONS

LIMITS 0 - 1

"CHANGE VALUES OF MXCYL & MINCYL (L) Y ?"

IF NO THEN THE NEXT TWO QUESTIONS WILL BE SKIPPED

"MAX CYL (D) 511 ?"

MAXIMUM INNER CYLINDER TO BE USED IN SEEK OPERATIONS.

LIMITS 0 - 255. (RL01) OR 511. (RL02)

"MIN CYL (D) 0 ?"

MINIMUM OUTER CYLINDER TO BE USED IN SEEK OPERATIONS.

LIMITS 0 - 255. (RL01) OR 511. (RL02)

"MAX SEC (D) 0 ?"

MAXIMUM SECTOR TO START TRANSFER ON

LIMITS 0 - 39

"MIN SEC (D) 0 ?"

MINIMUM SECTOR TO START TRANSFER ON

LIMITS 0 - 39

AFTER ANSWERING THE LAST SOFTWARE PARAMETER THE PROGRAM WILL START THE TESTING.

3.0 ERROR INFORMATION

ALL ERRORS ARE PRINTED VIA CONSOLE DEVICE. THE ERROR INCLUDES ERROR NUMBER, TYPE AND PROGRAM LOCATION. ERRORS INCLUDE REGISTERS BEFORE AND AT ERROR WITH RELEVENT DATA.

3.1 ERROR REPORTING

THE FOLLOWING ARE ERROR HEADINGS THAT MAY BE ENCOUNTERED WHILE RUNNING. A BRIEF DESCRIPTION IS GIVEN.

"SFT ERROR"

AN ERROR WAS DISCOVERED, BUT ON RETRY THE ERROR DID NOT PERSIST. INFO GIVEN IS ERROR, RLCS, RLBA, AND RLDA

"EXH'D RETRY ON SEEK"

THE NUMBER OF RETRIES GIVEN HAVE FAILED TO POSITION DRIVE TO THE GIVEN TRACK. INFO GIVEN IS RLCS,RLDA,RLBA, LAST POSITION,PRESENT POSITION, AND DRIVE STATUS

"VOL CHK WILL NOT RESET"

A DRIVE RESET WILL NOT RESET VOLUME CHECK BIT

"DID NOT REC'R FROM PWR UP'

DRIVE DID NOT COME BACK UP AFTER A POWER FAILURE

"DATA DMP - DATA CHECK/GARBBLED DATA"

THE PROGRAM ENCOUNTERED A DATA CHECK ERROR BUT WAS UNABLE TO MAKE SENSE OUT OF THE FIRST TWO WORDS, WHICH ARE USED TO KEY OFF OF. THEREFORE ALL WORDS OF SECTOR ARE DUMPED.(REFER TO SECTION 2.3.13.21)

"LIMITS EXCEEDED! HIGH - X LOW - Y"

ANSWER GIVEN IS NOT WITHIN LIMITS FOR QUESTION.

"NO DEFAULT PROVIDED!"

CANNOT <CR> TO THIS QUESTION

"ILLEGAL COMMAND"

START, RESTART, CONTINUE, PRINT TYPED IN WRONG FORM

"ILL ENTRY IN P-TABLE"

ANSWERS IN HARDWARE SECTION THAT ARE NOT LEGAL (I.E., MORE THAN TWO CONTROLLERS)

"CAN'T READ FACTORY BAD SECTOR FILE"

PROGRAM IS UNABLE TO READ ANY OF THE FACTORY FILES

"CAN'T READ FIELD BAD SECTOR FILE"

PROGRAM IS UNABLE TO READ ANY OF THE FIELD FILES

"MORE THAN 16 BAD SECTORS"

PROGRAM LIMITS EXERCISING CARTRIDGES TO THOSE WITH LESS THAN 16 BAD SECTORS.

```
1352
1353                    NO DRIVES ENTERED
1354
1355                    EITHER NO DRIVES WERE ENTERED OR ALL DRIVES THAT WERE ENTERED WERE
1356                    DROPPED FOR ONE REASON OR ANOTHER.  THE PROGRAM WILL LOOP AFTER
1357                    PRINTING THE ERROR, WAITING FOR ^C.  A START COMMAND IS NOW
1358                    NECESSARY.
1359
1360
1361                    'DRV NOT RDY W/O DRV ERR"
1362
1363                    ON COMPLETION OF A COMMAND, DRIVE READY IS CHECKED FOR A POSSIBLE
1364                    TRACKING DRIFT PROBLEM.  IF THERE IS NO DRIVE READY A GET STATUS IS
1365                    DONE TO VERIFY THAT THE DRIVE IS NOT IN PROCESS OF SEEKING.  IF IT
1366                    IS SEEKING THE CONDITION IS LEGAL.  THIS TYPEOUT IMPLIES THERE WERE
1367                    NO DRIVE ERRORS WHICH MAY HAVE CAUSED DRIVE READY TO GO AWAY.
1368
1369
1370                    'TRCK ERR"
1371
1372                    THIS ERROR MEANS THAT THE DRIVE IS NO LONGER ON THE TRACK SELECTED.
1373                    ANY SUBSEQUENT READ HEADER, READ OR WRITE COMMANDS WILL PRINT THIS
1374                    ERROR IF THE TRACK IS NOT CORRECT.  THIS ERROR WILL PRINT THE
1375                    POSITION BEFORE THE LAST SEEK, THE PRESENT POSITION AND THE
1376                    EXPECTED POSITION.
1377
1378
1379                    "MIS-SK ERR"
1380
1381                    AFTER A SEEK WAS DONE, READ HEADER IS DONE TO VERIFY THE SEEK.  THE
1382                    ERROR PRINTOUT WILL INCLUDE THE LAST POSTION BEFORE THE SEEK, THE
1383                    PRESENT POSITION AND THE EXPECTED POSTION.
1384
1385
1386                    "DRV STAT ERR"
1387
1388                    THE RESULT OF A GET STATUS OPERATION IS INCORRECT.  EITHER A ERROR
1389                    BIT IS SET OR THE STATE IS WRONG
1390
1391
1392                    "HRD ERR"
1393
1394                    THE NUMBER OF RETRIES WERE EXHAUSTED WITH OUT SUCCESS THE ERROR
1395                    PRINTOUT CONSISTS OF ALL REGISTERS BEFORE COMMAND AND AT TIME OF
1396                    ERROR.
1397
1398
1399                    "INIT WR OF SEC BAD"
1400
1401                    WHILE WRITING THE PACK INITIALLY, THE SECTOR INDICATED COULD NOT BE
1402                    WRITTEN AND VERIFIED.  THIS SECTOR WAS NOT IN THE BAD SECTOR FILE.
1403                    ONE OF THE FOLLOWING STEPS SHOULD BE ISSUED:  A) STOP THE EXERCISER
1404                    AND CHANGE CARTRIDGE.  B) STOP THE EXERCISER AND VERIFY THE
1405                    CARTRIDGE ON A PDP-11 (USE THE BAD SECTOR FILE TOOL - CZRLMA) OR
1406                    C) IGNORE ALL ERRORS FROM THAT SECTOR.
```

```
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
```

3.2 ERROR HALTS

ERROR HALTS ARE SUPPORTED PER DESCRIBED IN THE PREVIOUS SECTION WITH /FLAG:HOE. THERE ARE NO OTHER HALTS.

4.0 PERFORMANCE AND PROGRESS REPORTS

4.1 PERFORMANCE REPORTS

PERFORMANCE REPORTS ARE PRINTED BY OPERATOR REQUEST THROUGH THE DRS "PRINT" COMMAND. THE FORMAT IS:

```
        *** RL01 PERFORMANCE REPORT ***

        RLCS: XXXXXX  DRIVE: Y  DRIVE TYPE = RL0X
        PACK SERIAL #: DDDDDDDDDD
        TOTAL SEEKS:    IIIII
        WORDS READ:     JJJJJJJJ
        WORDS WRITTEN:  KKKKKKKKK

        ERRORS
        DRV-ER: N       SEEK:   N       TRACK:  N       DATA: N
        HARD:   N       SOFT:   N
        DCK:    N       HCRC:   N       NXM:    N       HNF: N
        DLT:    N       OPI:    N

        WHERE:

        XXXXXX  IS ADDRESS OF CONTROLLER
        Y       IS DRIVE NUMBER
        DDDDDDDDDD - IS 10 DIGIT OCTAL SERIAL NUMBER OF PACK
        IIII    IS TOTAL NUMBER OF SEEKS SINCE START
        JJJJ    IS TOTAL NUMBER OF WORDS READ SINCE START
        KKKK    IS TOTAL NUMBER OF WORDS WRITTEN SINCE START
        N       IS NUMBER OF THAT TYPE ERROR SINCE START
```

```
1450
1451                    4.2         PROGRESS REPORTS
1452                                ---------- -------
1453
1454                        THE ONLY PROGRESS REPORT IS THE PERFORMANCE REPORT.
1455
1456
1457                    5.0         DEVICE INFORMATION TABLES
1458                                ------ ----------- ------
1459
1460                        THE RL11/RLV11 CONTROLLER HAS THE FOLLOWING FOUR(4)  REGISTERS  FOR
1461                        CONTROL OF THE SUBSYSTEM.
1462
1463                                RLCS   CONTROL AND STATUS REGISTER (XXXXX0)
1464                                ---- - ------- --- ------ -               -
1465
1466                                BIT 15 - COMPOSITE ERROR
1467                                BIT 14   DRIVE ERROR
1468                                BIT 13 - NON EXISTANT MEMORY ERROR
1469                                BIT 12 - HEADER NOT FOUND (WITH BIT 10 SET)
1470                                       - DATA LATE (WITH BIT 10 CLEAR)
1471                                BIT 11 - HEADER CRC (WITH BIT 10 SET)
1472                                       - DATA CRC (WITH BIT 10 CLEAR)
1473                                BIT 10 - OPERATION INCOMPLETE
1474                                BIT  9/8 - DRIVE SELECT (0-3)
1475                                BIT  7 - CONTROLLER READY
1476                                BIT  6 - INTERRUPT ENABLE
1477                                BIT  5 - EXTENDED BUS ADDRESS (BIT 17)
1478                                BIT  4 - EXTENDED BUS ADDRESS (BIT 16)
1479                                BIT 3-1 - FUNCTION CODE
1480                                        0 - NOP (PDP-11) MAINT (LSI-11)
1481                                        1 - WRITE CHECK
1482                                        2 - GET DRIVE STATUS
1483                                        3 - SEEK
1484                                        4 - READ HEADER
1485                                        5 - WRITE DATA
1486                                        6 - READ DATA
1487                                        7 - READ WITHOUT HEADER COMPARE
1488
1489                                BIT  0 - DRIVE READY
1490
1491
1492                                RLBA - BUS ADDRESS REGISTER (XXXXX2)
1493                                ------------------------------ ------
1494
1495                                BITS 15-1 BUS ADDRESS OF DATA TRANSFER
1496                                BIT 0 SHOULD BE 0
1497
1498                                RLDA - DISK ADDRESS REGISTER (XXXXX4)
1499                                ------------------------------------
1500
1501                                FOR READ/WRITE FUNCTIONS
1502                                ------------------------
1503
1504                                BIT 15-7 - CYLINDER ADDRESS FOR TRANSFER
1505                                BIT 6 - SURFACE FOR TRANSFER
1506                                BIT 5-0 - SECTOR FOR TRANSFER (1-40.)
```

```
1508                    FOR SEEK FUNCTION
1509                     - -     - -
1510
1511                    BIT 15-7   DIFFERENCE TO NEW CY_INDER
1512                    BIT 6-5 - MUST BE ZERO (0)
1513                    BIT 4 - SURFACE (0=UPPER, 1=LOWER)
1514                    BIT 3 - MUST BE ZERO (0)
1515                    BIT 2 - SEEK DIRECTION( 1=IN / 0=OUT )
1516                    BIT 1 - MUST BE ZERO (0)
1517                    BIT 0 - MUST BE ONE (1)
1518
1519                    FOR GET STATUS FUNCTION
1520                    -----------------------
1521
1522                    BIT 15-4 - IGNORED SHOULD BE ZERO (0)
1523                    BIT 3 - DRIVE RESET
1524                    BIT 2   MUST BE ZERO (0)
1525                    BIT 1 - MUST BE ONE (1)
1526                    BIT 0 - MUST BE ONE (1)
1527
1528                    RLMP - MULTIPURPOSE REGISTER
1529                    ----------------------------
1530
1531                    FOR READ/WRITE FUNCTION
1532                    -----------------------
1533
1534                    BIT 15 - 0   WORD COUNT (TWO'S COMPLEMENT)
1535
1536                    FOR READ HEADER FUNCTION
1537                    ------ ---------- ----
1538
1539                    BIT 15-0 - DISK HEADER OF SECTOR (FIRST READ)
1540                             - ZERO WORD (SECOND READ)
1541                             - HEADER CRC (THIRD READ)
1542
1543                    FOR GET STATUS FUNCTION
1544                    ---------------------- -
1545
1546                    HAS DRIVE STATUS
1547
1548                    BIT 15 - WRITE DATA ERROR
1549                    BIT 14 - CURRENT HEAD ERROR (CHE)
1550                    BIT 13 - WRITE LOCK STATUS (WL)
1551                    BIT 12 - SEEK TIME OUT (SKTO)
1552                    BIT 11 - SPIN ERROR (SPE)
1553                    BIT 10 - WRITE GATE ERROR (WGE)
1554                    BIT  9 - VOLUME CHECK (VC)
1555                    BIT  8 - DRIVE SELECT ERROR (DSE)
1556                    BIT  7 - DRIVE TYPE IS RL02 IF SET
1557                    BIT  6 - SURFACE (0=UPPPER, 1=LOWER)
1558                    BIT  5 - COVER OPEN
1559                    BIT  4 - HEADS HOME
1560                    BIT  3 - BRUSHES HOME
1561                    PIT 2-0 -STATE BITS
1562                             0 - LOAD STATE
1563                             1 - SPIN UP
1564                             2 - BRUSH CYCLE
```

```
1566
1567                                                    3   LOAD HEADS
1568                                                    4 - SEEK   TRACK COUNTING
1569                                                    5   SEEK   LINEAR MODE
1570                                                    6   UNLOAD HEADS
1571                                                    7   SPIN DOWN
1572
1573
1574                    6.0 TEST SUMMARIES
1575                        -- - -----
1576
1577                        PROGRAM DESCRIPTION
1578
1579                        THE PROGRAM WILL TRY TO SIMULATE A  USER  ENVIRONMENT  WITH  RANDOM
1580                        SELECTION  OF  DRIVES  PERFORMING  RANDOM OPERATIONS OF GET STATUS,
1581                        SEEK, READ AND WRITE.
1582
1583                        INITIALLY THE BAD SECTOR FILE IS  RECOVERED  FROM  EACH  DRIVE  AND
1584                        STORED,  THEN  EACH  PACK  IS ENTIRELY WRITTEN RANDOMLY WITH ONE OF
1585                        EIGHT PREDETERMINED PATTERNS.
1586
1587                        THE MAIN LOOP IS A CONTINUOUS LOOP OF THE FOLLOWING STEPS
1588
1589                            1.  RANDOMLY SELECT A DRIVE
1590
1591                            2.  CHECK CONTROLLER OF SELECTED DRIVE IS NOT BUSY;
1592
1593                            3.  THEN STEP 3;  ELSE STEP 1
1594
1595                            4.  RANDOMLY SELECT FUNCTION FOR DRIVE
1596                                IF SEEK/WRITE/WRITE CHECK - THEN GO TO STEP 5
1597                                IF SEEK/READ - THEN GO TO STEP 11
1598                                IF SEEK/READ/READ - THEN GO TO STEP 15
1599                                IF SEEK/READ HDRS/READ W/NO HDR COMPARE/GET STATUS     THEN
1600                                GO TO STEP 21
1601
1602                            5.  GET A RANDOM CYLINDER ADDRESS (NOT THE BAD SECTOR FILE)
1603
1604                            6.  SEEK TO THE SELECTED CYLINDER AND WAIT TILL COMPLETED
1605
1606                            7.  GET A RANDOM WORD COUNT FOR THE WRITE FUNCTION - MAKE  SURE
1607                                THAT IT WON'T OVERFLOW THE TRACK
1608
1609                            8.  GET A RANDOM DATA PATTERN TO WRITE ON THE TRACK POINTED TO
1610
1611                            9.  ISSUE THE WRITE FUNCTION AND WAIT TILL COMPLETED
1612
1613                           10.  ISSUE A WRITE CHECK FUNCTION ON THE SAME  DISK  ADDRESS  TO
1614                                COMPARE THE DATA JUST WRITTEN BY THE WRITE FUNCTION THEN GO
1615                                TO STEP #1
1616
1617                           11.  GET A RANDOM CYLINDER # FOR THE SEEK
1618
1619                           12.  SEEK TO THE SELECTED CYLINDER AND WAIT TILL COMPLETED
1620
1621                           13.  GET A RANDOM WORD COUNT FOR THE READ FUNCTION -  MAKE  SURE
1622                                IT WILL NOT OVERFLOW THE SELECTED TRACK
```

14. ISSUE THE READ FUNCTION AND WAIT TILL COMPLETED ...THE INTERRUPT SERVICE WILL INITIATE A DATA COMPARE ON THE DATA READ (IF THE FUNCTION IS ENABLED) THEN GO TO STEP #1

15. GET A RANDOM CYLINDER FOR THE SEEK

16. SEEK AND WAIT TILL COMPLETED

17. GET A RANDOM WORD COUNT FOR THE READ COMMAND

18. ISSUE A READ COMMAND AND WAIT TILL COMPLETED

19. GET ANOTHER RANDOM WORD COUNT FOR SAME TRACK SELECTED

20. ISSUE A SECOND READ FUNCTION AND WAIT TILL COMPLETED THEN GOTO STEP #1

21. ISSUE A SEEK TO A RANDOM CYLINDER AND WAIT TILL COMPLETED

22. ISSUE A READ HEADER FUNCTION AND WAIT TILL COMPLETED

23. ISSUE A READ DATA WITH NO HEADER COMPARE (1 SECTOR TO BE READ) AND WAIT TILL COMPLETED

24. ISSUE A GET STATUS FUNCTION THEN GO TO STEP #1

THE PROGRAM WILL STAY WITHIN THAT MAIN LOOP UNTIL INTERRUPTED OUT BY A FUNCTION FINISHING AT WHICH TIME THE INTERRUPT SERVICE ROUTINE WILL START EXECUTION.

1. READ ALL REGISTERS OF CONTROLLER THAT INTERRUPTED AND SAVE IMAGES

2. IF NO ERROR SET; THEN STEP 3; ELSE STEP 14

5. CHECK FUNCTION WHICH CAUSED INTERRUPT
   IF WRITE CHECK; THEN STEP 3A
   IF GET STATUS; THEN STEP 5
   IF SEEK; THEN STEP 4A.
   IF READ HEADER; THEN STEP 7
   IF READ; THEN STEP 9
   IF WRITE; THEN STEP 3B

3A. CLEAR WRITE CHECK NEEDED FLAG, THEN STEP 4

3B. SET WRITE CHECK NEEDED FLAG IF REQUESTED THEN STEP 4

4. IF RETRY > 0 THEN REPORT SOFT ERROR, ELSE STEP 4A

4A. EXIT TO MAIN PROGRAM

5. CHECK STATUS FOR: NO ERRORS
                      COVER CLOSED
                      BRUSHES HOME
                      HEADS OUT

```
1682
1683                                        SEEK LINEAR/TRACKING
1684
1685            IF THEN STEP 4;  ELSE STEP 6
1686
1687        6.  REPORT STATUS ERROR;  GO TO STEP 4A
1688
1689        7.  SET VERIFICATION DONE FLAG COMPARE  PRESENT  POSITION  WITH
1690            HEADER WORD IF THEN STEP 4A;  ELSE STEP 8
1691
1692        8.  REPORT MIS-SEEK, SET NEW POSITION;  GO TO STEP 4
1693
1694        9.  IF DATA TO BE COMPARED;  THEN STEP 10;  ELSE STEP 4
1695
1696       10.  CHECK VALIDITY OF FIRST TWO WORDS;  IF THEN STEP 12;   ELSE
1697            STEP 11.
1698
1699       11.  REPORT GARBLED DATA;  GO TO STEP 4
1700
1701       12.  CHECK WORDS READ IN IF OKAY THEN STEP 4A ELSE STEP 13
1702
1703       13.  REPORT DATA ERROR, GO TO STEP 4
1704
1705       14.  IF DRIVE ERROR;  THEN STEP 33;  ELSE STEP 15
1706
1707       15.  IF NXM;  THEN STEP 18;  ELSE STEP 16
1708
1709       16.  IF OPI;  THEN STEP 18;  ELSE STEP 17
1710
1711       17.  IF DLT;  THEN STEP 18;  ELSE STEP 20
1712
1713       18.  IF RETRY < LIMIT THEN STEP 4A, ELSE STEP 19
1714
1715       19.  REPORT HARD ERROR;  CLEAR FLAGS;  GO TO STEP 4A
1716
1717       20.  IF HCRC;  THEN STEP 24;  ELSE STEP 21
1718
1719       21.  IF DCRC, THEN STEP 29;  ELSE STEP 22
1720
1721       22.  IF HNF, THEN STEP 30;  ELSE STEP 23
1722
1723       23.  YOU SHOULD NEVER GET HERE
1724
1725       24.  IF DOING READ/WRITE THEN STEP 25 IF DOING READ HEADER  THEN
1726            STEP 26
1727
1728       25.  CHECK IF DA IS BAD SECTOR THEN STEP 4A;  ELSE STEP 18.
1729
1730       26.  READ 40 HEADERS, IF ALL GOOD THEN STEP 27;  ELSE STEP 28
1731
1732       27.  REPORT SOFT HEADER CRC;  GO TO 4A
1733
1734       28.  FIGURE OUT BAD HEADER IF IN FILE THEN STEP 4A;   ELSE  STEP
1735            18
1736
1737       29.  CHECK IF DA-1 IS IN FILE IF THEN STEP 4A;  ELSE STEP 18
```

```
1739
1740                    30.  READ HEADER.  IF ON CORRECT TRACK THEN STEP 31;  ELSE  STEP
1741                         32
1742
1743                    31.  CHECK IF DA IS IN FILE IF THEN STEP 4A, ELSE STEP 18
1744
1745                    32.  REPORT TRACKING;  FIX POSITION, GO TO STEP 4
1746
1747                    33.  ACT UPON:     VC
1748                                       SKTO
1749                                       SPE
1750                                       WGE
1751                                       WDE
1752                                       CHE
1753
1754                    34.  GO TO STEP 4
1755
1756                @
```

HEADER

```
                                .SBTTL  HEADER

                        002000          .=2000

                                        .MCALL  SVC

002000                                  SVC
                        000000          SVCINS=0
                        000000          SVCTAG=0

002000                                  POINTER BGNRPT,BGNSW,BGNSFT,BGNAU,BGNDU

002000                          BGNMOD  MDHEDR
002000                                  HEADER  CNRLK,A,0,0,1,PRIO6             ;JSD REV A - ADDED PRIO6
002000  103                             .ASCII /C/
002001  116                             .ASCII /N/
002002  122                             .ASCII /R/
002003  114                             .ASCII /L/
002004  113                             .ASCII /K/
002005  000                             .BYTE   0
002006  000                             .BYTE   0
002007  000                             .BYTE   0
002010  101                             .ASCII  /A/
002011  060                             .ASCII  /0/
002012  000000                          .WORD   0
002014  000000                          .WORD   0
002016  031646                          .WORD   L$HARD
002020  032022                          .WORD   L$SOFT
002022  010560                          .WORD   L$HW
002024  010576                          .WORD   L$SW
002026  033406                          .WORD   L$LAST
002030  000000                          .WORD   0
002032  000000                          .WORD   0
002034  000001                          .WORD   1
002036  000000                          .WORD   0
002040  010676                          .WORD   L$DISPATCH
002042  000300                          .WORD   PRIO6
002044  000000                          .WORD   0
002046  000000                          .WORD   0
002050  003                             .BYTE   C$REVISION
002051  003                             .BYTE   C$EDIT
002052  000000                          .WORD   0
002054  000000                          .WORD   0
002056  000000                          .WORD   0
002060  002230                          .WORD   L$DVTYP
002062  010700                          .WORD   L$RPT
002064  000000                          .WORD   0
002066  000000                          .WORD   0
002070  013314                          .WORD   L$AU
002072  013400                          .WORD   L$DU
002074  000000                          .WORD   0
002076  002122                          .WORD   L$DESC
002100  104035                          EMT     E$LOAD
002102  000000                          .WORD   0
002104  010764                          .WORD   L$INIT
002106  013116                          .WORD   L$CLEAN
002110  012644                          .WORD   L$AUTO
```

```
HEADER

        002112  010756                          .WORD   L$PROT
        002114  000000                          .WORD   0
        002116  000000                          .WORD   0
        002120  000000                          .WORD   0
   1772
   1773 002122                                  ENDMOD
   1774
   1775 002122                          DESCRIPT <CNRLK PERFORMS RANDOM OPERATIONS OF GET STATUS, SEEK, READ, AND WRITE>
        002122      103     116     122         .ASCIZ  /CNRLK PERFORMS RANDOM OPERATIONS OF GET STATUS, SEEK, READ, AND WRITE/
        002125      114     113     040
        002130      120     105     122
        002133      106     117     122
        002136      115     123     040
        002141      122     101     116
        002144      104     117     115
        002147      040     117     120
        002152      105     122     101
        002155      124     111     117
        002160      116     123     040
        002163      117     106     040
        002166      107     105     124
        002171      040     123     124
        002174      101     124     125
        002177      123     054     040
        002202      123     105     105
        002205      113     054     040
        002210      122     105     101
        002213      104     054     040
        002216      101     116     104
        002221      040     127     122
        002224      111     124     105
        002227      000
                                            .EVEN
   1776
   1777 002230                              DEVTYP  <RL01,RL02>
        002230      122     114     060     .ASCIZ  #RL01,RL02#
        002233      061     054     122
        002236      114     060     062
        002241      000
                                            .EVEN
   1778
   1779                                 .SBTTL  BIT AND OFFSET DEFINITIONS
   1780
   1781                                 ;DEFINITIONS
   1782
   1783 002242                          BGNMOD  GLBEQAT
   1784
   1785 002242                          EQUALS
                                        ;
                                        ; BIT DIFINITIONS
                                        ;
                100000                  BIT15== 100000
                040000                  BIT14== 40000
                020000                  BIT13== 20000
                010000                  BIT12== 10000
                004000                  BIT11== 4000
                002000                  BIT10== 2000
```

BIT AND OFFSET DEFINITIONS

```
000400          BIT09== 1000
000400          BIT08== 400
000200          BIT07== 200
000100          BIT06== 100
000040          BIT05== 40
000020          BIT04== 20
000010          BIT03== 10
000004          BIT02== 4
000002          BIT01== 2
000001          BIT00== 1
                ;
001000          BIT9==  BIT09
000400          BIT8==  BIT08
000200          BIT7==  BIT07
000100          BIT6==  BIT06
000040          BIT5==  BIT05
000020          BIT4==  BIT04
000010          BIT3==  BIT03
000004          BIT2==  BIT02
000002          BIT1==  BIT01
000001          BIT0==  BIT00
                ;
                ; EVENT FLAG DEFINITIONS
                ;   EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION
                ;
                ;                                   ; BIT POSITION IN SECOND STATUS WORD
000040          EF.START==      32.                 ; (100000) START COMMAND WAS ISSUED
000037          EF.RESTART==    31.                 ; (040000) RESTART COMMAND WAS ISSUED
000036          EF.CONTINUE==   30.                 ; (020000) CONTINUE COMMAND WAS ISSUED
000035          EF.NEW==        29.                 ; (010000) A NEW PASS HAS BEEN STARTED
000034          EF.PWR==        28.                 ; (004000) A POWER-FAIL/POWER-UP OCCURRED
                ;
                ;
                ; PRIORITY LEVEL DEFINITIONS
                ;
000340          PRI07== 340
000300          PRI06== 300
000240          PRI05== 240
000200          PRI04== 200
000140          PRI03== 140
000100          PRI02== 100
000040          PRI01== 40
000000          PRI00== 0
                ;
                ;OPERATOR FLAG BITS
                ;
000004          EVL==     4
000010          LOT==    10
000020          ADR==    20
000040          IDU==    40
000100          ISR==   100
000200          UAM==   200
000400          BOE==   400
001000          PNT==  1000
002000          PRI==  2000
004000          IXE==  4000
010000          IBE== 10000
```

BIT AND OFFSET DEFINITIONS

```
                020000          IER=:    20000
                040000          LOE=:    40000
                100000          HOE=:   100000

1786
1787            000000          CS=0                            ;CONTROL AND STATUS OFFSET
1788            000002          BA=2                            ;BUSADDRESS OFFSET
1789            000004          DA=4                            ;DISK ADDRESS OFFSET
1790            000006          MP=6                            ;MULTI PURPOSE OFFSET
1791                            ;CONSTANT OFFSETS FOR INDIVIDUAL DRIVE BUFFERS
1792                            ;THE ONLY POSITION THAT IS CRITICAL IS THAT OF
1793                            ;""PRPOS""  IT M U S T (MUST) BE THE LAST ENTRY OF THE BUFFER
1794
1795            000000          SKCNT=0                 ;SEEK OPERATION COUNT
1796            000002          RXFR1=2                 ;READ OPERATION COUNT (BITS) LOW ORDER
1797            000004          RXFR2=4                 ;  "       "       "       "   HIGH ORDER
1798            000006          WXFR1=6                 ;WRITE OPERATION COUNT (BITS) LOW ORDER
1799            000010          WXFR2=10                ;  "       "       "       "   HIGH ORDER
1800            000012          ERRCNT=12               ;ERROR COUNT   HARD
1801            000014          SFTCNT=14               ;ERROR COUNT - SOFT
1802            000016          SKECNT=16               ;SEEK ERROR COUNT
1803            000020          DERCNT=20               ;DRIVE ERROR COUNT
1804            000022          DCRCER=22               ;DATA CRC ERROR COUNT
1805            000024          HCRCER=24               ;HEADER CRC ERROR COUNT
1806            000026          DLTCNT=26               ;DATA LATE ERROR COUNT
1807            000030          OPICNT=30               ;OPERATION INCOMPLETE ERROR COUNT
1808            000032          HNFERR=32               ;HEADER NOT FOUND ERROR COUNT
1809            000034          NXMCNT=34               ;NON EXISTENT MEMORY ERROR COUNT
1810            000036          RETRY=36                ;PRESENT RETRY NUMBER
1811            000040          BDA=40                  ;   "    DISK ADDRESS CONTENTS
1812            000042          BMP=42                  ;PRESENT MULTIPURPOSE CONTENTS
1813            000044          FUNC=44                 ;LAST FUNCTION LOADED
1814            000046          BCSADR=46               ;CSR IMAGE OF LAST COMMAND
1815            000050          LSTHDR=50               ;LAST POSITION ON DISK
1816            000052          RTYPE=52                ;ERROR ON WHICH RECOVERY IS BEING TRIED
1817            000054          SKCNT1=54               ;LOW SEEK COUNT
1818            000056          PRFLGS=56               ;INTERNAL FLAGS
1819            000060          RXFR3=60                ;THIRD ORDER READ COUNT
1820            000062          WXFR3=62                ;THIRD ORDER WRITE COUNT
1821            000064          LSTDA=64                ;DISK ADDRESS AT SOFT ERROR
1822            000066          DIFWD=66                ;LAST DIFFERENCE WORD OF SEEK
1823            000070          DPHOUR=70               ;HOUR OF DRIVE DROPPED
1824            000071          DPMIN=71                ;MINUTE OF DRIVE DROPPED
1825            000072          TRERR=72                ;TRACKING ERRORS COUNT
1826            000074          DATCER=74               ;DATA CMP ERRORS
1827            000076          DOWCK=76                ;PERFORM WRITE CHECK
1828            000100          SERNM1=100              ;SERIAL NUMBER OF CARTRIDGE
1829            000102          SERNM2=102              ;SERIAL NUMBER OF CARTRIDGE
1830            000104          DCS=104                 ;CSR ADDRESS
1831            000106          DRSEL=106               ;DRIVE SELECT BITS(8,9,10)
1832            000110          BBA=110                 ;PRESENT BUS ADDRESS CONTENTS
1833            000112          BSECPT=112              ;POINTER TO BAD SECTOR FILE
1834            000114          RSEEK=114               ;SEEK IN PROCESS OF RECOVERY
1835            000116          SOFTCS=116              ;CSR OF SOFT ERROR
1836            000120          TDR=120
1837            000122          WRIPG=122               ;WRITE IN PROGRESS FLAG
1838            000124          PRPOS=124               ;PRESENT POSITION ON DISK
1839
```

BIT AND OFFSET DEFINITIONS

```
1840            000001          SKDON=BIT0
1841            000001          DRDY=BIT0               ;DRIVE READY
1842            000100          INTEN=BIT6              ;INTERRUPT ENABLE
1843            100000          ERR=BIT15               ;COMPOSITE ERROR
1844            040000          DERR=BIT14              ;DRIVE ERROR
1845            100000          WDE=BIT15               ;WRITE DATA ERROR
1846            040000          HCE=BIT14               ;HEAD CURRENT ERROR
1847            020000          WL=BIT13                ;WRITE LOCK
1848            010000          SKTO=BIT12              ;SEEK TIMEOUT ERROR
1849            004000          SPE=BIT11               ;SPINDLE TIMEOUT/UNDER/OVER SPEED
1850            002000          WGE=BIT10               ;WRITE GATE ERROR
1851            001000          VC=BIT9                 ;VOLUME CHECK
1852            000400          DSE=BIT8                ;DRIVE SELECT ERROR
1853            020000          NXM=BIT13               ;NON-EXISTENT MEMORY ERROR
1854            010000          DLT=BIT12               ;DATA LATE
1855            004000          DCRC=BIT11              ;DATA CRC ERROR
1856            004000          HCRC=BIT11              ;HEADER CRC ERROR
1857            010000          HNF=BIT12               ;HEADER NOT FOUND ERROR
1858            002000          OPI=BIT10               ;OPERATION INCOMPLETE ERROR
1859            000200          CRDY=BIT7               ;CONTROLLER READY
1860            000040          BA17=BIT5               ;EXTENDED BUS ADDRESS BIT 17
1861            000020          BA16=BIT4               ;EXTENDED BUS ADDRESS BIT 16
1862            000002          WRCHK=BIT1              ;WRITE CHECK FUNCTION CODE
1863            000004          GSTAT=BIT2              ;GET DRIVE STATUS FUNCTION CODE
1864            000006          SEEK=BIT1!BIT2          ;SEEK FUNCTION CODE
1865            000010          RDHDR=BIT3              ;READ HEADER FUNCTION CODE
1866            000012          WRITE=BIT3!BIT1         ;WRITE FUNCTION CODE
1867            000014          READ=BIT3!BIT2          ;READ FUNCTION CODE
1868            000013          DRST=BIT3!BIT1!BIT0     ;DRIVE RESET COMMAND CODE FOR DRIVE COMMAND WORD
1869            000003          GSBIT=BIT1!BIT0         ;GET STATUS COMMAND CODE FOR DRIVE COMMAND WORD
1870            000001          MK=BIT0                 ;MARKER BIT FOR DRIVE COMMAND WORD(SEEK,GET STATUS)
1871            000004          SIGN=BIT2               ;DIRECTION FOR SEEK(0=AWAY FROM SPINDLE)
1872            000020          SKHS=BIT4               ;HEAD SELECT FOR SEEK
1873            000100          HEAD=BIT6               ;HEAD SELECT FOR READ,WRITE,GET STATUS
1874
1875                            ;OFFSET FOR HARDWARE P-TABLE
1876
1877            000000          CSR=0
1878            000002          VECT=2
1879            000004          PRIOR=4
1880            000006          TYPDR=6
1881            000010          DRBT=10
1882            000012          CNT=12
1883
1884                            ;OFFSET FOR SOFTWARE P-TABLE
1885
1886            000000          RLT=0
1887            000002          ELT=2
1888            000004          SET=4
1889            000006          DAT=6
1890            000010          SKT=10
1891            000012          TYT=12
1892            000014          RDT=14
1893            000016          DDT=16
1894            000020          CHFLG=20
1895            000022          MXB=22
1896            000024          MXH=24
```

BIT AND OFFSET DEFINITIONS

```
1897          000026                  MNH=26
1898          000030                  MXC=30
1899          000032                  MNC=32
1900          000034                  MXS=34
1901          000036                  MNS=36
1902          000040                  DCKFG=40
1903          000042                  DRFLG=42
1904          000044                  MNB=44
1905          000046                  SEL=46
1906          000050                  OPFLG=50
1907          000052                  DET=52
1908          000054                  ROF=54
1909          000056                  RAN=56
1910          000060                  PAT=60
1911          000062                  SRLT=62
1912          000064                  CLMT=64
1913          000066                  AUTO=66
1914          000070                  STIP=70
1915          000072                  WCK=72
1916          000074                  DCD=74
1917          000076                  ANS=76
1918
1919 002242                          ENDMOD
1920
1921                                 .SBTTL  MACRO DEFINITIONS
1922
1923                                 ;DELAY EXECUTION OF PROGRAM A SPECIFIED NUMBER OF 100-MILLISECOND TIME COUNTS
1924                                 .MACRO  WAITMS  ARG,?WAIT
1925                                 ;       MOV     @ARG,DLYCNT     ;INITIALIZE DELAY COUNTER                ;JSD REV A
1926                                         MOV     ARG,DLYCNT      ;INITIALIZE DELAY COUNTER                ;JSD REV A
1927                                         ASL     DLYCNT          ;MULTIPLY ARGUMENT BY 2
1928                                         ASL     DLYCNT          ;MULTIPLY ARGUMENT BY 2 AGAIN
1929                                 ;WAIT:  DELAY   @250.            ;IMPLEMENT 25-MS TIME DELAY              ;JSD REV A
1930                                 WAIT:   DELAY   250.             ;IMPLEMENT 25-MS TIME DELAY              ;JSD REV A
1931                                         DEC     DLYCNT          ;DECREMENT DELAY COUNT
1932                                         BNE     WAIT            ;BRANCH IF TIME DELAY NOT EXPIRED
1933                                 .ENDM
1934
1935                                 ;DELAY EXECUTION OF PROGRAM A SPECIFIED NUMBER OF 100-MICROSECOND TIME COUNTS
1936                                 .MACRO  WAITUS  ARG
1937                                 ;       DELAY   @ARG             ;IMPLEMENT 100-US TIME DELAY, ARGUMENT SPECIFIES ;JSD REV A
1938                                         DELAY   ARG              ;IMPLEMENT 100-US TIME DELAY, ARGUMENT SPECIFIES ;JSD REV A
1939                                                                  ;/THE NUMBER OF 100-US TIME COUNTS
1940                                 .ENDM
1941
1942                                 ;ACTIVATE THE CLOCK TO INITIATE THE GENERATION OF CLOCK INTERRUPTS
1943                                 .MACRO  CLKON                    ;ACTIVATE CLOCK WITH 1-SEC INCREMENTS
1944                                         JSR     PC,CLKINI       ;INITIALIZE CLOCK
1945                                         JSR     PC,CLKST        ;START CLOCK
1946                                 .ENDM
1947
1948                                 ;DEACTIVATE THE CLOCK TO HALT THE GENERATION  OF CLOCK INTERRUPTS
1949                                 .MACRO  CLKOFF
1950                                         CLR     CLKSON          ;INDICATE "CLOCK OFF"
1951                                         CMP     @1,CLKTYP       ;P-CLOCK?
1952                                         BNE     11$             ;BRANCH TO CHECK FOR L CLOCK
1953                                         CLR     @#172540        ;CLEAR P-CLOCK
```

MACRO DEFINITIONS

```
                                        11$:    CMP     #2,CLKTYP       ;L CLOCK?
                                                BNE     12$             ;BRANCH FOR NO CLOCK
                                                CLR     @#177546        ;CLEAR L-CLOCK
                                        12$:
                                        .ENDM

                                        ;REQUEST ELAPSED TIME IN SECONDS OCCURRING BETWEEN SUPERVISOR INITIATION
                                        ;AND THE GENERATION OF THE REQUEST
                                        .MACRO  REQTIM  ARG
                                                MOV     CLKACC,ARG
                                        .ENDM

                                        .SBTTL  GLOBAL DATA AND CONSTANTS

002242                                  BGNMOD  GLBDAT

002242  000000                          RECNT:  .WORD   0               ;READ ERROR COUNT
002244  000000                          RWCNT:  .WORD   0               ;R/W ERROR COUNT
002246  000000                          WHY:    .WORD   0               ;REASON FOR DROPPING DRIVE
002250  000000                          TSTDRV: .WORD   0               ;COPY OF SELECTED DRIVE FOR TESTING
002252  000                             DRUT:   .BYTE   0               ;DRIVES UNDER TEST
002253  000                             DRPRS:  .BYTE   0               ;DRIVES PRESENT
002254  000000                          T.DRIVE: .WORD  0               ;TYPE OF DRIVE FROM P-TABLE
002256  000000                          SYSMSK: .WORD   0               ;MASK FOR 0-7 DRIVES
002260  176543                          MINUM:  .WORD   176543          ;PRIME FOR RANDOM
002262  123456                          LONUM:  .WORD   123456          ;NUMBER GENERATOR
002264  100177                          CYLMSK: .WORD   100177          ;MASK FOR CYLINDER ONLY
002266  100077                          SECMSK: .WORD   100077          ;MASK OUT SECTOR BITS
002270  000177                          CMSK:   .WORD   000177
002272  000077                          SMSK:   .WORD   000077

002274  000000                          WRINIT: .WORD   0               ;WRITE INIT FLAG
002276  000000                          WRPOS:  .WORD   0               ;WRITE UNIT FLAG
002300  000000                          CYL:    .WORD   0               ;CYLINDER #
002302  000000                          SUR:    .WORD   0               ;SURFACE #
002304  000000                          SEC:    .WORD   0               ;SECTOR #
002306  000000                          REGEN:  .WORD   0               ;REGEN FLAG FOR BUFFERS
002310  000000                          KILLDC: .WORD   0               ;INHIBIT DATA COMP FLAG
002312  000000                          CLKFRQ: .WORD   0               ;CLOCK FREQUENCY FLAG, 1=60 HZ, 2=50 HZ
002314  000000                          CLKTYP: .WORD   0               ;CLOCK TYPE FLAG, 1=P-CLOCK, 2=L-CLOCK
002316  000000                          CLKADR: .WORD   0               ;POINTER TO ADDRESS OF SUPERVISOR CLOCK TABLE

                                        ;
                                        ;THE FOLLOWING LOCATIONS ARE CLEARED AS A GROUP (DOWN TO 'STFLG')
                                        ;THEREFORE DON'T INSERT ANY CONSTANTS
                                        ;
002320  174400                          CNTLR1: .WORD   174400          ;CSR OF CONTROLLER 1 (LUN 0-3)
002322  000000                          CNTLR2: .WORD   0               ;CSR OF CONTROLLER 2 (LUN 4-7)
002324  000000                          LSTDR1: .WORD   0               ;BUFFER POINTER OF DRIVE
002326  000000                          LSTDR2: .WORD   0               ;BUFFER POINTER OF DRIVE
002330  000000                          BCSR:   .WORD   0               ;CSR FROM P-TABLE
002332  000000                          BVEC:   .WORD   0               ;VECTOR "     "
002334  000000                          BPRIOR: .WORD   0               ;PRIORITY "     "
002336  000000                          BDRSEL: .WORD   0               ;DRIVE    "     "
002340  000000                          HDRFND: .WORD   0               ;FLAG TO INDICATE HDR IN BAD LIST
002342  000000                          CHKSEC: .WORD   0               ;SECTOR OF ERROR - USED BY BAD SECTOR LOCATION
002344  000000                          DECNT:  .WORD   0               ;DATA ERROR COUNT
002346  000000                          TEMPO:  .WORD   0               ;TEMP LOCATION
```

```
2011 002350  000000                TEMP1:   .WORD   0              ;TEMP LOCATION
2012 002352  000000                TEMP2:   .WORD   0              ;TEMP LOCATION
2013 002354  000000                TEMP3:   .WORD   0              ;  "      "
2014 002356  000000                TEMP4:   .WORD   0              ;  "      "
2015 002360  000000                TEMP5:   .WORD   0              ;  "      "
2016 002362  000000                TEMP6:   .WORD   0              ;  "      "
2017 002364  000000                TEMP7:   .WORD   0              ;  "      "
2018 002366  000000                TEMP8:   .WORD   0              ;  "      "
2019 002370  000000                TEMP9:   .WORD   0              ;  "      "
2020 002372  000160                VECT1:   .WORD   160            ;VECTOR OF FIRST CONTROLLER
2021 002374  000000                VECT2:   .WORD   0              ;VECTOR   "  2ND
2022 002376  000000                PRIOR1:  .WORD   0
2023 002400  000000                PRIOR2:  .WORD   0
2024 002402  000000                GDDAT:   .WORD   0              ;
2025 002404  000000                RNTEMP:  .WORD   0              ;    "    "
2026 002406  000000                INTERVAL: .WORD  0              ;KEEPS TRACK OF TIME BETWEEN STATISTICAL REPORTS
2027                                                              ;/(MINUTES RUNNING TIME)
2028 002410  000000                TICK:    .WORD   0              ;STORAGE FOR TICK COUNT
2029 002412  000000                SECOND:  .WORD   0              ;SECONDS OF SYSTEM CLOCK
2030 002414  000000                MINUTE:  .WORD   0              ;MINUTES OF SYSTEM CLOOK
2031 002416  000000                HOUR:    .WORD   0              ;HOURS OF SYSTEM CLOCK
2032 002420  000000                E.CS:    .WORD   0              ;IMAGES OF REGISTERS
2033 002422  000000                E.BA:    .WORD   0              ;ON INTERRUPT
2034 002424  000000                E.DA:    .WORD   0              ;
2035 002426  000000                E.MP:    .WORD   0              ;
2036 002430  000000                E.MP1:   .WORD   0              ;
2037 002432  000000                E.MP2:   .WORD   0              ;
2038 002434  000000                C.HDR:   .WORD   0              ;CURRENT HEADER - FOR ERROR REPORT
2039 002436  000000                BUF1:    .WORD   0              ;BUFFER FOR  FIRST CONTROLLER
2040 002440  000000                BUF2:    .WORD   0              ;BUFFER FOR SECOND CONTROLLER
2041 002442  000000                MAXWC:   .WORD   0              ;MAX WORD COUNT DETERMINED BY CORE
2042 002444  000000                UUT:     .WORD   0              ;NUMBER OF UNITS ON SYSTEM
2043 002446  000000                PWRFLG:  .WORD   0              ;POWER FAIL INDICATOR
2044 002450  000000                TRPFLG:  .WORD   0              ;INDICATES OCCURRENCE OF A TIME-OUT TRAP
2045 002452  000000                STFLG:   .WORD   0              ;START FLAG
2046                                ;
2047                                ;END OF MASS CLEAR
2048                                ;
2049 002454  000000                CNTFLG:  .WORD   0              ;CONTINUE FLAG
2050 002456  000000                FASCII:  .WORD   0              ;ASCII MESSAGE OF FUNCTION
2051 002460  000000                FASPNT:  .WORD   0              ;POINTER
2052 002462  000000                DWCNT:   .WORD   0              ;ERROR COUNT
2053 002464  000000                DWCNT1:  .WORD   0              ;ERROR COUNT
2054 002466  000004                ERRVEC:  .WORD   4              ;ERROR VECTOR
2055 002470  000034                ST1:     .WORD   34             ;STATES ALLOWED
2056 002472  000035                ST2:     .WORD   35             ;STATES ALLOWED
2057 002474  000000                OPCALL:  .WORD   0
2058 002476  000000                INCALL:  .WORD   0
2059 002500  000000                DLYCNT:  .WORD   0              ;DELAY COUNTER FOR WAITMS TIMING MACRO
2060 002502  000000                SYSCLK:  .WORD   0              ;FLAG INDICATING PRESENCE OF A SYSTEM CLOCK
2061 002504  000000                CLKSON:  .WORD   0              ;"CLOCK ON" INDICATOR
2062 002506  000000                CLKCNT:  .WORD   0              ;CLOCK COUNTER TO STORE TICK VALUE
2063 002510  000000                CLKBFR:  .WORD   0              ;CLOCK BUFFER TO STORE CLOCK TICK COUNT
2064 002512  000000                CLKACC:  .WORD   0              ;CLOCK ACCUMULATOR TO STORE ELAPSED TIME IN
2065                                                              ;/SECONDS OF SUPERVISOR TIME
2066 002514  000000                CLKFLD:  .WORD   0              ;CLOCK FIELD USED TO CHECK IF LSI-11 CLOCK
2067                                                              ;/IS "TICKING"
```

```
2068
2069 002516                            ENDMOD
2070
2071                                .SBTTL  GLOBAL MESSAGES
2072
2073 002516                         BGNMOD  GLBTXT
2074
2075                                ;GLOBAL TEXT
2076
2080
2081 002516  124  111  115  TIME:   .ASCIZ  "TIME: "
2082 002525  040  122  114  MRLCS:  .ASCIZ  " RLCS: "
2083 002535  040  050  122  CRLCS:  .ASCIZ  " (RLCS): "
2084 002547  076  076  040  MFUNC:  .ASCIZ  ">> FUNCTION: '
2085 002565  040  050  122  CRLBA:  .ASCIZ  ' 'RLBA): "
2086 002577  040  050  122  CRLDA:  .ASCIZ  " (RLDA): '
2087 002611  040  050  122  CRLMP:  .ASCIZ  " (RLMP): "
2088
2089 002623  104  111  106  DIFMSG: .ASCIZ  /DIF WD: /
2090 002634  120  101  103  CART:   .ASCIZ  /PACK SERIAL #: /
2091 002654  116  117  040  NOCRDY: .ASCIZ  /NO CRDY/
2092 002664  104  122  111  DNRDY:  .ASCIZ  /DRIVE NOT READY/
2093 002704  104  122  040  NORDY:  .ASCIZ  <DR NOT RDY W/O DR ERR>
2094 002732  102  125  107  PRGER:  .ASCIZ  /BUG/
2095 002736  111  116  111  NWRTS:  .ASCIZ  /INIT WR OF SEC BAD/
2096 002761  040  123  105  SMSG:   .ASCIZ  / SECTOR: /
2097 002773  116  117  040  EXHAUS: .ASCIZ  /NO GOOD HDR/
2098 003007  125  116  104  UDERR:  .ASCIZ  /UNDIAGNOSABLE ERR/
2099 003031  123  105  105  MSKER:  .ASCIZ  /SEEK ERR/
2100 003042  123  117  106  MSFER:  .ASCIZ  /SOFT ERR ENC'D/
2101 003061  104  122  040  DRVER:  .ASCIZ  /DR ERR/
2102 003070  104  122  040  MDERS:  .ASCIZ  /DR ERR WILL NOT RESET/
2103 003116  104  122  040  MDSER:  .ASCIZ  /DR STAT ERR/
2104 003132  126  117  114  MVCER:  .ASCIZ  /VOL CHK WILL NOT CLR/
2105 003157  127  122  040  WGEST:  .ASCIZ  /WR GATE ERR WILL NOT RESET/
2106 003212  104  122  040  MRDER:  .ASCIZ  /DR ERR - RECOVERED/
2107 003235  104  101  124  MDCER:  .ASCIZ  /DATA CMP ERR/
2108 003252  110  101  122  MHDER:  .ASCIZ  /HARD ERROR/
2109 003265  104  101  124  DMPDCK: .ASCIZ  /DATA DUMP - DCK/
2110 003305  124  122  101  TRACK:  .ASCIZ  /TRACKING ERR/
2111 003322  110  122  104  ERLMTM: .ASCIZ  /HRD ERR LMT EXC'D/
2112 003344  123  113  040  SERLMT: .ASCIZ  /SK ERR LMT EXC'D/
2113 003365  123  106  124  SFEMSG: .ASCIZ  /SFT ERR LMT EXC'D/
2114 003407  104  101  124  DCDMSG: .ASCIZ  /DATA ERR LMT EXC'D/
2115 003432  104  122  040  DERMSG: .ASCIZ  /DR ERR LMT EXC'D/
2116 003453  102  125  106  OVER:   .ASCIZ  /BUFFER CHOSEN TOO BIG   WAS /
2117 003510  122  105  121  REQ:    .ASCIZ  /REQ BY OPR/
2118 003523  105  130  110  SEXHAU: .ASCIZ  /EXH'D RETRY ON SEEK/
2119 003547  110  104  123  UNLOAD: .ASCIZ  /HDS NOT UNLD ON ERR/
2120 003573  104  122  040  NOLOAD: .ASCIZ  /DR WLD NOT LD/
2121 003611  117  120  105  SOPLMT: .ASCIZ  /OPER LMTS EXC'D/
2122 003631  107  101  122  NOREV:  .ASCIZ  /GARBLED DATA - CAN'T CHECK IT/
2123 003667  115  117  122  MBDMSC: .ASCIZ  /MORE THAN 16 BAD SECTORS/
2124 003720  116  117  040  NWSEC:  .ASCIZ  /NO FACTORY FILE/
2125 003740  116  117  040  SWSEC:  .ASCIZ  /NO FIELD FILE/
2126 003756  120  055  124  MPT:    .ASCIZ  /P-TABLE: /
2127 003770  111  114  114  ILLEG:  .ASCIZ  /ILL P-TABLE/
```

GLOBAL MESSAGES

```
2128 004004  040  126  105  MVEC:    .ASCIZ  / VECTOR: /
2129 004016  116  117  040  NODRIV:  .ASCIZ  /NO DRIVES/
2130 004030  040  104  122  DRNM:    .ASCIZ  / DRIVE: /
2131 004041  040  114  123  LPS:     .ASCIZ  / LST POS: /
2132 004054  105  130  120  EPS:     .ASCIZ  /EXP POS: /
2133 004066  040  122  105  RPS:     .ASCIZ  / REC POS: /
2134 004101  104  122  040  NOPWR:   .ASCIZ  /DR DID REC'R FROM PWR UP/
2135 004132  101  124  040  BUSAD:   .ASCIZ  /AT BUS ADDR: /
2136 004150  122  105  124  MRT:     .ASCIZ /RETRYS: /
2137 004161  040  105  122  ERT:     .ASCIZ  / ERROR TYPE: /
2138 004177  123  124  101  MST:     .ASCIZ  /STATUS WAS: /
2139 004214  040  123  110  MST1:    .ASCIZ  / SHOULD BE: /
2140 004231  040  122  105  RT1:     .ASCIZ  / RETRIES ATTEMPTED/
2141 004254  040  105  130  EXP:     .ASCIZ  / EXP'D: /
2142 004265  040  122  105  RCD:     .ASCIZ  / REC'D: /
2143 004276  104  122  111  DROP:    .ASCIZ  /DRIVE DROPPED/
2144 004314  040  110  116  MTHNF:   .ASCIZ  / HNF/
2145 004321  040  110  103  MTHCRC:  .ASCIZ  / HCRC/
2146 004327  040  104  103  MTDCRC:  .ASCIZ  / DCK/
2147 004334  040  104  114  MTDLT:   .ASCIZ  / DLT/
2148 004341  040  117  120  MTOPI:   .ASCIZ  / OPI/
2149 004346  040  116  130  MTNXM:   .ASCIZ  / NXM/
2150 004353  040  104  122  MTDRV:   .ASCIZ  / DRV/
2151 004360  124  105  123  MSTART:  .ASCIZ  /TESTING STARTED/
2152 004400  127  122  111  MSWRPK:  .ASCIZ  /WRITING PACK /
2153 004416  120  101  103  NORDDC:  .ASCIZ  /PACK NOT FULLY INIT'D...DATA COMPARE INHIBITTED/
2154 004476  103  125  122  ERRHDR:  .ASCIZ  /CURRENT POSITION (HDR) = /
2155 004530  123  131  123  NOCLK:   .ASCIZ  /SYSTEM CLOCK IS NOT AVAILABLE/
2156 004566  120  105  122  NOREPT:  .ASCIZ  /PERFORMANCE REPORTS WILL NOT BE PRINTED/
2157 004636  104  111  104  NOTRDY:  .ASCIZ  /DID NOT RESPOND WITH "READY"/
2158 004673  116  117  040  NOCTLR:  .ASCIZ  /NO CONTROLLER/
2159 004711  123  131  123  INSMEM:  .ASCIZ  /SYSTEM FATAL ERROR - INSUFFICIENT MEMORY BUFFER SPACE/
2160                          ;
2161                          ;THIS LIST OF ASCII TEXT IS USED AS A TABLE FOR PRINTING
2162                          ;FUNCTIONS IN ERROR MESSAGES   TABLE IS "MTCR - MTRD".
2163                          ;THE ORDER IS IMPORTANT AS WELL AS THE LENGTH OF EACH
2164                          ;ASCIZ STRING. EACH STRING IS SEVEN(10) BYTES PLUS ZERO
2165                          ;FILL BYTE (TOTAL 8(EIGHT) BYTES) LONG.  USED IN LINE1
2166                          ;SUBROUTINE.....................................
2167                          ;...............................................
2168
2169 004777  040  127  122  MTCR:    .ASCIZ  / WRCHK /
2170 005007  040  107  124  MTGS:    .ASCIZ  / GTSTAT/
2171 005017  040  123  105  MTSK:    .ASCIZ  / SEEK  /
2172 005027  040  122  104  MTRH:    .ASCIZ  / RDHDR /
2173 005037  040  127  122  MTWR:    .ASCIZ  / WRITE /
2174 005047  040  122  105  MTRD:    .ASCIZ  / READ  /
2175 005057  040  122  104  MTRNH:   .ASCIZ  / RD-NHD/
2176
2177                          ;...............................................
2178                          ;END OF LIST - YOU CAN PUT ANYTHING YOU WANT HERE
2179                          ;...............................................
2180
2181                          .NLIST  CND,MD,ME
2182
2183                          .EVEN
2184
```

GLOBAL MESSAGES

```
                                        ENDMOD
                                .SBTTL  ERROR MESSAGES
                                BGNMOD  GLBERR
                                                                        ;GENERAL ERROR REPORT
                                BGNMSG  ERR1
                                        JSR     PC,LINE3
                                        ENDMSG
                                L10000:
                                        TRAP    C$MSG
                                                                        ;MIS-SEEK ERROR REPORT
                                BGNMSG  ERR2
                                        JSR     PC,LINE3
                                        PRINTB  @FMT4,@DIFMSG,DIFWD(R4),@LPS,LSTHDR(R4),@EPS,PRPOS(R4),@RPS,R1
                                        MOV     R1,-(SP)
                                        MOV     @RPS,-(SP)
                                        MOV     PRPOS(R4),-(SP)
                                        MOV     @EPS,-(SP)
                                        MOV     LSTHDR(R4),-(SP)
                                        MOV     @LPS,-(SP)
                                        MOV     DIFWD(R4),-(SP)
                                        MOV     @DIFMSG,-(SP)
                                        MOV     @FMT4,-(SP)
                                        MOV     #11,-(SP)
                                        MOV     SP,R0
                                        TRAP    C$PNTB
                                        ADD     #24,SP
                                        ENDMSG
                                L10001:
                                        TRAP    C$MSG
                                                                        ;SOFT ERROR RECOVERABLE ERROR REPORT
                                BGNMSG  ERR3
                                        JSR     PC,LINE1
                                        PRINTB  @FMT2A,@CRLCS,SOFTCS(R4),@CRLBA,@BBA(R4),@CR_DA,LSTDA(R4)
                                        MOV     LSTDA(R4),-(SP)
                                        MOV     @CRLDA,-(SP)
                                        MOV     @BBA(R4),-(SP)
                                        MOV     @CRLBA,-(SP)
                                        MOV     SOFTCS(R4),-(SP)
                                        MOV     @CRLCS,-(SP)
                                        MOV     @FMT2A,-(SP)
                                        MOV     #7,-(SP)
                                        MOV     SP,R0
                                        TRAP    C$PNTB
                                        ADD     #20,SP
                                        MOV     LSTDA(R4),TEMP0 ;GET THE ADDRESS TO PRINT
                                        JSR     R5,TELCYL       ;CONVERT FOR PRINTING
                                        PRINTB  @FMT5,@MRT,RETRY(R4),@ERT,RTYPE(R4)
                                        MOV     RTYPE(R4),-(SP)
                                        MOV     @ERT,-(SP)
                                        MOV     RETRY(R4),-(SP)
```

ERROR MESSAGES

```
        005264  012746  004150                 MOV     #MRT, (SP)
        005270  012746  007077                 MOV     #FMT5,-(SP)
        005274  012746  000005                 MOV     #5, (SP)
        005300  010600                         MOV     SP,R0
        005302  104414                         TRAP    C$PNTB
        005304  062706  000014                 ADD     #14,SP
   2211 005310                                 ENDMSG
        005310                          L10002:
        005310  104423                         TRAP    C$MSG
   2212
   2213                                                         ;GET STATUS ERROR REPORT
   2214
   2215 005312                          BGNMSG ERR4
   2216 005312  004737  006270                 JSR     PC,LINE3
   2217 005316                                 PRINTB  #FMT6,#MST,E.MP,#MST1,ST1,ST2
        005316  013746  002472                 MOV     ST2,-(SP)
        005322  013746  002470                 MOV     ST1, (SP)
        005326  012746  004214                 MOV     #MST1,-(SP)
        005332  013746  002426                 MOV     E.MP,-(SP)
        005336  012746  004177                 MOV     #MST,-(SP)
        005342  012746  007113                 MOV     #FMT6,-(SP)
        005346  012746  000006                 MOV     #6,-(SP)
        005352  010600                         MOV     SP,R0
        005354  104414                         TRAP    C$PNTB
        005356  062706  000016                 ADD     #16,SP
   2218 005362                                 ENDMSG
        005362                          L10003:
        005362  104423                         TRAP    C$MSG
   2219
   2220                                                         ;DATA ERROR SUMMARY
   2221
   2222 005364                          BGNMSG ERR6
   2223 005364  004737  006200                 JSR     PC,LINE2
   2224 005370  016400  000042                 MOV     BMP(R4),R0
   2225 005374                                 PRINTB  #FMT9A,DECNT,R0
        005374  010046                         MOV     R0,-(SP)
        005376  013746  002344                 MOV     DECNT,-(SP)
        005402  012746  007223                 MOV     #FMT9A,-(SP)
        005406  012746  000003                 MOV     #3,-(SP)
        005412  010600                         MOV     SP,R0
        005414  104414                         TRAP    C$PNTB
        005416  062706  000010                 ADD     #10,SP
   2226 005422                                 ENDMSG
        005422                          L10004:
        005422  104423                         TRAP    C$MSG
   2227
   2228                                                         ;NON-RECOVERABLE ERROR REPORT
   2229
   2230 005424                          BGNMSG ERR7
   2231 005424                                 PRINTB  #FMT8,RETRY(R4),#RT1
        005424  012746  004231                 MOV     #RT1,-(SP)
        005430  016446  000036                 MOV     RETRY(R4),-(SP)
        005434  012746  007155                 MOV     #FMT8,-(SP)
        005440  012746  000003                 MOV     #3,-(SP)
        005444  010600                         MOV     SP,R0
        005446  104414                         TRAP    C$PNTB
        005450  062706  000010                 ADD     #10,SP
```

ERROR MESSAGES

```
2232 005454  004737  006270                    JSR     PC,LINE3
2233 005460                                    ENDMSG
     005460                              L10005:
     005460  104423                            TRAP    C$MSG
2234
2235                                                        ;BAD DATA COMPARE ERROR REPORT
2236
2237 005462                              BGNMSG ERR8
2238 005462  004737  006200                    JSR     PC,LINE2
2239 005466  016437  000040  002346            MOV     BDA(R4),TEMPO
2240 005474  004537  006450                    JSR     R5,TELCYL               ;REPORT THE CYL # & SECTOR/HEAD
2241 005500                                    PRINTB  #FMT10A,#CRLBA,@BBA(R4),#CRLDA,BDA(R4),#EXP,GDDAT,#RCD,(R2)
     005500  011246                            MOV     (R2),-(SP)
     005502  012746  004265                    MOV     #RCD,-(SP)
     005506  013746  002402                    MOV     GDDAT,-(SP)
     005512  012746  004254                    MOV     #EXP,-(SP)
     005516  016446  000040                    MOV     BDA(R4),-(SP)
     005522  012746  002577                    MOV     #CRLDA,-(SP)
     005526  017446  000110                    MOV     @BBA(R4),-(SP)
     005532  012746  002565                    MOV     #CRLBA,-(SP)
     005536  012746  007312                    MOV     #FMT10A,-(SP)
     005542  012746  000011                    MOV     #11,-(SP)
     005546  010600                            MOV     SP,R0
     005550  104414                            TRAP    C$PNTB
     005552  062706  000024                    ADD     #24,SP
2242 005556                                    PRINTB  #FMT10B,R2
     005556  010246                            MOV     R2,-(SP)
     005560  012746  007363                    MOV     #FMT10B,-(SP)
     005564  012746  000002                    MOV     #2,-(SP)
     005570  010600                            MOV     SP,R0
     005572  104414                            TRAP    C$PNTB
     005574  062706  000006                    ADD     #6,SP
2243 005600                                    ENDMSG
     005600                              L10006:
     005600  104423                            TRAP    C$MSG
2244                                                        ;DRIVE ERROR
2245
2246 005602                              BGNMSG ERR9
2247
2248 005602  004737  006270                    JSR     PC,LINE3
2249 005606                                    PRINTB  #FMT13,#MST,R1,#LPS,LSTHDR(R4)
     005606  016446  000050                    MOV     LSTHDR(R4),-(SP)
     005612  012746  004041                    MOV     #LPS,-(SP)
     005616  010146                            MOV     R1,-(SP)
     005620  012746  004177                    MOV     #MST,-(SP)
     005624  012746  007421                    MOV     #FMT13,-(SP)
     005630  012746  000005                    MOV     #5,-(SP)
     005634  010600                            MOV     SP,R0
     005636  104414                            TRAP    C$PNTB
     005640  062706  000014                    ADD     #14,SP
2250 005644                                    ENDMSG
     005644                              L10007:
     005644  104423                            TRAP    C$MSG
2251
2252                                                        ;INVALID ENTRY IN P-TABLE REPORT
2253
2254 005646                              BGNMSG ERR10
```

ERROR MESSAGES

```
 2255 005646                                    PRINTB  @FMT11,@MPT,R1,@MRLCS,BCSR,@MVEC,BVEC
      005646  013746  002332                    MOV     BVEC,-(SP)
      005652  012746  004004                    MOV     @MVEC,-(SP)
      005656  013746  002330                    MOV     BCSR,-(SP)
      005662  012746  002525                    MOV     @MRLCS, (SP)
      005666  010146                            MOV     R1,-(SP)
      005670  012746  003756                    MOV     @MPT,-(SP)
      005674  012746  007371                    MOV     @FMT11,-(SP)
      005700  012746  000007                    MOV     #7,-(SP)
      005704  010600                            MOV     SP,R0
      005706  104414                            TRAP    C$PNTB
      005710  062706  000020                    ADD     #20,SP
 2256 005714                                    ENDMSG
      005714                            L10010:
      005714  104423                            TRAP    C$MSG
 2257
 2258 005716                            BGNMSG  ERR1<
 2259
 2260 005716  004737  006270                    JSR     PC,LINE3
 2261
 2262 005722                                    ENDMSG
      005722                            L10011:
      005722  104423                            TRAP    C$MSG
 2263
 2264 005724                            BGNMSG  ERR13
 2265 005724  004737  006270                    JSR     PC,LINE3
 2266 005730  016403  000104                    MOV     DCS(R4),R3
 2267 005734  016337  000006  002426            MOV     MP(R3),E.MP             ;GET HEADER
 2268 005742                                    PRINTB  @FMT14C                 ;CRLF
      005742  012746  007501                    MOV     @FMT14C,-(SP)
      005746  012746  000001                    MOV     #1,-(SP)
      005752  010600                            MOV     SP,R0
      005754  104414                            TRAP    C$PNTB
      005756  062706  000004                    ADD     #4,SP
 2269 005762                                    PRINTB  @FMT12,@ERRHDR,C.HDR ;PRINT THE HEADER MESSAGE
      005762  013746  002434                    MOV     C.HDR,-(SP)
      005766  012746  004476                    MOV     @ERRHDR,-(SP)
      005772  012746  007411                    MOV     @FMT12,-(SP)
      005776  012746  000003                    MOV     #3,-(SP)
      006002  010600                            MOV     SP,R0
      006004  104414                            TRAP    C$PNTB
      006006  062706  000010                    ADD     #10,SP
 2270 006012                                    ENDMSG
      006012                            L10012:
      006012  104423                            TRAP    C$MSG
 2271
 2272 006014  016437  000044  002460  LINE1:  MOV     FUNC(R4),FASPNT         ;GET FUNCTION
 2273 006022  012737  004777  002456            MOV     @MTCR,FASCII            ;FIRST FUNCTION ASCIZ
 2274 006030  042737  000100  002460            BIC     @INTEN,FASPNT           ;CLEAR INTERRUPT ENABLE
 2275 006036  006237  002460                    ASR     FASPNT                  ;ALIGN - NOW = 1 TO 7
 2276 006042  005337  002460            1$:     DEC     FASPNT                  ;DOWN COUNT FUNCTION
 2277 006046  001404                            BEQ     2$                      ;FOUND?
 2278 006050  062737  000010  002456            ADD     #8.,FASCII              ;NO NEXT ONE
 2279 006056  000771                            BR      1$                      ;LOOP
 2280
 2281                                   ;2$:     PRINTB  @FMT16,@TIME,HOUR,MINUTE,SECOND,@MRLCS,DCS(R4),@DRNM,<8,DRSEL+1(R4)>     ;JSD
REV A
 2282 006060                            2$:     PRINTB  @FMT10,@MRLCS,DCS(R4),@DRNM,<8,DRSEL+1(R4)>                              ;JSD
REV A
```

ERROR MESSAGES

```
        006060  005046                          CLR     (SP)
        006062  156416  000107                  BISB    DRSEL+1(R4),(SP)
        006066  012746  004030                  MOV     #DRNM,-(SP)
        006072  016446  000104                  MOV     DCS(R4),-(SP)
        006076  012746  002525                  MOV     #MRLCS,-(SP)
        006102  012746  007277                  MOV     #FMT10,-(SP)
        006106  012746  000005                  MOV     #5,-(SP)
        006112  010600                          MOV     SP,R0
        006114  104414                          TRAP    C$PNTB
        006116  062706  000014                  ADD     #14,SP
   2283 006122                                  PRINTB  #FMTDT,TDR(R4)
        006122  016446  000120                  MOV     TDR(R4),-(SP)
        006126  012746  010103                  MOV     #FMTDT,-(SP)
        006132  012746  000002                  MOV     #2,-(SP)
        006136  010600                          MOV     SP,R0
        006140  104414                          TRAP    C$PNTB
        006142  062706  000006                  ADD     #6,SP
   2284 006146                                  PRINTB  #FMT1A,#MFUNC,FASCII
        006146  013746  002456                  MOV     FASCII,-(SP)
        006152  012746  002547                  MOV     #MFUNC,-(SP)
        006156  012746  006645                  MOV     #FMT1A,-(SP)
        006162  012746  000003                  MOV     #3,-(SP)
        006166  010600                          MOV     SP,R0
        006170  104414                          TRAP    C$PNTB
        006172  062706  000010                  ADD     #10,SP
   2285 006176  000207                          RTS     PC
   2286
   2287                             ;LINE2:    PRINTB  #FMT10,#TIME,HOUR,MINUTE,SECOND,#MRLCS,DCS(R4),#DRNM,<B,DRSEL+1(R4)>     ;JSD
REV A
   2288 006200                             LINE2:  PRINTB  #FMT10,#MRLCS,DCS(R4),#DRNM,<B,DRSEL+1(R4)>                          ;JSD
REV A
        006200  005046                          CLR     -(SP)
        006202  156416  000107                  BISB    DRSEL+1(R4),(SP)
        006206  012746  004030                  MOV     #DRNM,-(SP)
        006212  016446  000104                  MOV     DCS(R4),-(SP)
        006216  012746  002525                  MOV     #MRLCS,-(SP)
        006222  012746  007277                  MOV     #FMT10,-(SP)
        006226  012746  000005                  MOV     #5,-(SP)
        006232  010600                          MOV     SP,R0
        006234  104414                          TRAP    C$PNTB
        006236  062706  000014                  ADD     #14,SP
2289    006242                                  PRINTB  #FMTDT,TDR(R4)
        006242  016446  000120                  MOV     TDR(R4),-(SP)
        006246  012746  010103                  MOV     #FMTDT,-(SP)
        006252  012746  000002                  MOV     #2,-(SP)
        006256  010600                          MOV     SP,R0
        006260  104414                          TRAP    C$PNTB
        006262  062706  000006                  ADD     #6,SP
2290    006266  000207                          RTS     PC
2291
2292    006270  004737  006014          LINE3:  JSR     PC,LINE1
2293    006274                                  PRINTB  #FMT2,#CRLCS,BCSADR(R4),#CRLBA,@BBA(R4),#CRLDA,BDA(R4),#CRLMP,BMP(R4)
        006274  016446  000042                  MOV     BMP(R4),-(SP)
        006300  012746  002611                  MOV     #CRLMP,-(SP)
        006304  016446  000040                  MOV     BDA(R4),-(SP)
        006310  012746  002577                  MOV     #CRLDA,-(SP)
        006314  017446  000110                  MOV     @BBA(R4),-(SP)
        006320  012746  002565                  MOV     #CRLBA,-(SP)
        006324  016446  000046                  MOV     BCSADR(R4),-(SP)
```

ERROR MESSAGES

```
                006330  012746  002535                  MOV     #CRLCS,(SP)
                006334  012746  006654                  MOV     #FMT2,-(SP)
                006340  012746  000011                  MOV     #11,-(SP)
                006344  010600                          MOV     SP,R0
                006346  104414                          TRAP    C$PNTB
                006350  062706  000024                  ADD     #24,SP
          2294  006354                                  PRINTB  #FMT3,#CRLCS,E.CS,#CRLBA,E.BA,#CRLDA,E.DA,#CRLMP,E.MP
                006354  013746  002426                  MOV     E.MP,-(SP)
                006360  012746  002611                  MOV     #CRLMP,-(SP)
                006364  013746  002424                  MOV     E.DA,-(SP)
                006370  012746  002577                  MOV     #CRLDA,-(SP)
                006374  013746  002422                  MOV     E.BA,-(SP)
                006400  012746  002565                  MOV     #CRLBA,-(SP)
                006404  013746  002420                  MOV     E.CS,-(SP)
                006410  012746  002535                  MOV     #CRLCS,-(SP)
                006414  012746  006717                  MOV     #FMT3,(SP)
                006420  012746  000011                  MOV     #11,-(SP)
                006424  010600                          MOV     SP,R0
                006426  104414                          TRAP    C$PNTB
                006430  062706  000024                  ADD     #24,SP
          2295  006434  013737  002424  002346          MOV     E.DA,TEMP0      ;GET ADDRESS TO PRINT
          2296  006442  004537  006450                  JSR     R5,TELCYL       ;PRINT IT
          2297  006446  000207                          RTS     PC              ;EXIT
          2298
          2299  006450  013737  002346  002300  TELCYL: MOV     TEMP0,CYL       ;GET THE ADDRESS
          2300  006456  042737  000177  002300          BIC     #177,CYL        ;SAVE ONLY CYLINDER BITS
          2301  006464  000337  002300                  SWAB    CYL
          2302  006470  000241                          CLC
          2303  006472  006137  002300                  ROL     CYL
          2304  006476  103002                          BCC     1$
          2305  006500  005237  002300                  INC     CYL
          2306  006504  013737  002346  002304  1$:     MOV     TEMP0,SEC       ;GET SECTOR #
          2307  006512  042737  177700  002304          BIC     #177700,SEC     ;SAVE ONLY THE SECTOR BITS
          2308  006520  005037  002302                  CLR     SUR             ;INIT TO HEAD 0
          2309  006524  032737  000100  002424          BIT     #100,E.DA       ;HEAD 1?
          2310  006532  001405                          BEQ     2$              ;NO
          2311  006534  005237  002302                  INC     SUR             ;YUP
          2312  006540  042737  177776  002302          BIC     #177776,SUR
          2313  006546                          2$:     PRINTB  #FMT3A,#DRVER,CYL,SUR,SEC
                006546  013746  002304                  MOV     SEC,-(SP)
                006552  013746  002302                  MOV     SUR,-(SP)
                006556  013746  002300                  MOV     CYL,-(SP)
                006562  012746  003061                  MOV     #DRVER,-(SP)
                006566  012746  006760                  MOV     #FMT3A,-(SP)
                006572  012746  000005                  MOV     #5,-(SP)
                006576  010600                          MOV     SP,R0
                006600  104414                          TRAP    C$PNTB
                006602  062706  000014                  ADD     #14,SP
          2314  006606  000205                          RTS     R5
          2315
          2316                          ;FORMAT STATEMENTS
          2317
          2321
          2322  006610      045     124     045  FMT1:   .ASCIZ  /%T%Z2%A:%Z2%A:%Z2/
          2323  006632      045     124     045  FMT17:  .ASCIZ  /%T%06%T%01/
          2324  006645      045     124     045  FMT1A:  .ASCIZ  /%T%T%N/
          2325  006654      045     101     102  FMT2:   .ASCII  /%ABEFORE ERR%T%06/
```

ERROR MESSAGES

```
2326 006675  045  124  045  FMT2A:  .ASCIZ  /%T%06%T%06%T%06%T%06%N/
2327 006717  045  101  101  FMT3:   .ASCIZ  /%AAT ERR    %T%06%T%06%T%06%T%06%T%06/
2328 006760  045  116  045  FMT3A:  .ASCIZ  /%N%T%A ADDR = CYL: %Z3%A. SUR: %01%A SECT: %Z2%A.%N/
2329 007044  045  116  045  FMT4:   .ASCIZ  /%N%T%06%T%06%N%T%06%T%06%N/
2330 007077  045  124  045  FMT5:   .ASCIZ  /%T%06%T%T%N/
2331 007113  045  124  045  FMT6:   .ASCIZ  /%T%06%T%06%A OR %06%N/
2332 007141  045  124  045  FMT7:   .ASCIZ  /%T%A - %T%N/
2333 007155  045  104  066  FMT8:   .ASCIZ  /%D6%T%N/
2334 007165  045  124  045  FMT9:   .ASCIZ  /%T%Z2%A:%Z2%A:%Z2%T%06%T%01%N/
2335 007223  045  104  066  FMT9A:  .ASCIZ  /%D6%A. WORDS BAD OUT OF %D6%A. WORDS READ%N/
2336                       ;FMT10: .ASCIZ  /%T%Z2%A:%Z2%A:%Z2%T%06%T%01/                    ;JSD REV A
2337 007277  045  124  045  FMT10:  .ASCIZ  /%T%06%T%01/                                    ;JSD REV A
2338 007312  045  124  045  FMT10A: .ASCIZ  /%T%06%T%06%N%T%06%T%06%A AT BUS ADDRESS /
2339 007363  045  117  066  FMT10B: .ASCIZ  /%06%N/
2340 007371  045  124  045  FMT11:  .ASCIZ  /%T%02%T%06%T%03/
2341 007411  045  124  045  FMT12:  .ASCIZ  /%T%06%N/
2342 007421  045  124  045  FMT13:  .ASCIZ  /%T%06%T%06%N/
2343 007436  045  124  045  FMT13D: .ASCIZ  /%T%Z4%A NOW IS %Z4%N/
2344 007463  045  116  045  FMT14:  .ASCIZ  /%N%T%N/
2345 007472  045  117  066  FMT14A: .ASCIZ  /%06%A /
2346 007501  045  116  000  FMT14C: .ASCIZ  /%N/
2347 007504  045  101  127  FMT14B: .ASCIZ  ?%AWORD %D3%A. S/B %06%A WAS %06%N?
2348 007546  045  101  105  FMT15:  .ASCIZ  /%AERROR(S) SET:%T%N%ARECOVERY BEING ATTEMPTED/
2349 007624  045  116  045  FRMT16: .ASCIZ  /%N%ANOT TESTING CS=%06%A DR=%01%N/
2350 007666  045  116  045  FMT18:  .ASCIZ  /%N%T/
2351 007673  045  116  045  FMTXS:  .ASCIZ  /%N%AXFER SIZE = %Z6%A. WORDS%N/
2352 007732  045  116  045  FMTS1:  .ASCIZ  /%N%N%S10%A*** RL01-RL02 PERFORMANCE REPORT ***%N%N/
2353 010015  045  101  052  FMTS1A: .ASCIZ  /%A*** RUNNING%N/
2354 010035  045  101  052  FMTS1B: .ASCIZ  /%A*** DROPPED  %Z2%A:%Z2%N/
2355 010070  045  124  045  FMTS2:  .ASCIZ  /%T%05%05%N/
2356 010103  045  101  040  FMTDT:  .ASCIZ  /%A DRIVE TYPE = RL0%01%N/
2357 010134  045  101  124  FMTS2A: .ASCIZ  /%ATOTAL SEEKS:    %D6%Z3%N%AWORDS READ:    %D6%Z4%Z4%N/
2358 010223  045  101  127  FMTS2B: .ASCIZ  /%AWORDS WRITTEN: %D6%Z4%Z4%N/
2359 010260  045  116  045  FMTS3:  .ASCIZ  /%N%AERRORS%N%ADRV-ER:%D6%A   SEEK: %D6%A   TRACK: %D6%A   DATA:  %D6%N/
2360 010367  045  101  110  FMTS3A: .ASCIZ  /%AHARD:  %D6%A   SOFT: %D6%N/
2361 010424  045  101  104  FMTS4:  .ASCIZ  /%ADCK:   %D6%A   HCRC: %D6%A   NXM:   %D6%A   HNF:   %D6%N/
2362 010517  045  101  104  FMTS5:  .ASCIZ  /%ADLT:   %D6%A   OPI:  %D6%N%N/
2363
2367
2368                       .EVEN
2369
2370 010556                       ENDMOD
2371
2372                       .SBTTL  DEFAULT HARDWARE P-TABLE PARAMETERS
2373
2374 010556                BGNMOD  HPTCODE
2375
2376 010556                BGN%HW
     010556  000006               .WORD   L10013-L%HW/2
2377
2378 010560  174400               .WORD   174400          ;DRIVE CSR
2379 010562  000160               .WORD   160             ;DRIVE VECTOR
2380 010564  000240               .WORD   240             ;DRIVE PRIORITY
2381 010566  000001               .WORD   1               ;DRIVE TYPE
2382 010570  000000               .WORD   0               ;DRIVE NUMBER
2383 010572  000001               .WORD   1               ;CONTROLLER TYPE
2384
```

DEFAULT HARDWARE P TABLE PARAMETERS

```
2385 010574                        ENDHW
     010574                        L10013:
2386
2387 010574                                ENDMOD
2388
2389                               .SBTTL  DEFAULT SOFTWARE P TABLE PARAMETERS
2390
2391 010574                        BGNMOD  SPTCODE
2392
2393 010574                        BGNSW
     010574  000037                        .WORD   L10014-L$SW/2
2394
2395 010576  000001                LIMIT:  .WORD   1                ;RETRY LIMIT
2396 010600  000003                ERLMT:  .WORD   3                ;ERROR LIMIT
2397 010602  000003                SELMT:  .WORD   3                ;SEEK ERROR LIMIT
2398 010604  060650                DALMT:  .WORD   25000.           ;DATA XFER LIMIT (*(10*3)) (BITS)
2399 010606  023420                SKLMT:  .WORD   10000.           ;SEEK LIMIT
2400 010610  000360                TYINT:  .WORD   240.             ;TIME INTERVAL IN MINS. BETWEEN STATISTICAL
2401                                                                ;/REPORTS (4 HRS. TOTAL)
2402 010612  000020                CMRD:   .WORD   16.              ;WORDS TO COMPARE ON READ
2403 010614  000003                DELMT:  .WORD   3                ;ERRORS TO REPORT ON DATA COMPARE
2404 010616  000000                XCHFLG: .WORD   0                ;CHANGE OTHER PARAMETERS
2405 010620  002400                T.MXB:  .WORD   1280.            ;MAXIMUM R/W TRANSFER BUFFER
2406 010622  000100                T.MXH:  .WORD   100              ;MAXIMUM HEAD SELECT
2407 010624  000000                T.MNH:  .WORD   0                ;MINIMUM HEAD SELECT
2408 010626  177600                T.MXC:  .WORD   177600           ;MAXIMUM CYLINDER
2409 010630  000000                T.MNC:  .WORD   0                ;MINIMUM CYLINDER
2410 010632  000000                T.MXS:  .WORD   0                ;MAXIMUM START SECTOR
2411 010634  000000                T.MNS:  .WORD   0                ;MINIMUM START SECTOR
2412 010636  000001                T.DCK:  .WORD   1                ;DATA DUMP ON DATA CHECK ERROR
2413 010640  000001                T.DRP:  .WORD   1                ;DROP ON LIMIT REACHED
2414 010642  000003                T.MNB:  .WORD   3                ;MINIMUM BUFFER TRANSFER SIZE
2415 010644  000012                SFLMT:  .WORD   10.              ;SOFT ERROR LIMIT
2416 010646  000000                T.STA:  .WORD   0                ;DROP DRIVE ON PERFORMANCE REACHED
2417 010650  000003                DRLMT:  .WORD   3                ;DRIVE ERROR LIMIT
2418 010652  000000                T.ROF:  .WORD   0                ;READ ONLY FLAG
2419 010654  000001                T.RAN:  .WORD   1                ;RANDOM SELECT OF PATTERNS
2420 010656  000004                T.PAT:  .WORD   4                ;ONLY ONE PATTERN  4 = WORST CASE
2421 010660  000001                T.SLT:  .WORD   1                ;SEEK RETRY LIMIT
2422 010662  000200                T.CLT:  .WORD   128.             ;NUMBER OF ERRORS ON DCK DUMP
2423 010664  000000                T.STIP: .WORD   0                ;RESTRICT BUFFER SIZE
2424 010666  000001                T.WCK:  .WORD   1                ;DO WRITE CHECK
2425 010670  000012                T.DCD:  .WORD   10.
2426 010672  000001                T.ANS:  .WORD   1
2427
2428 010674                        ENDSW
     010674                        L10014:
2429
2430 010674                                ENDMOD
2431
2432 010674                        BGNMOD  DSPCODE
2433
2434 010674                        DISPATCH        1
     010674  000001                        .WORD   1
     010676  014464                        .WORD   T1
2435
2436 010700                                ENDMOD
```

DEFAULT SOFTWARE P-TABLE PARAMETERS

```
2437
2438                                    .SBTTL  STATISTICAL CODE
2439
2440 010700                             BGNMOD  RPTCODE
2441
2442 010700                             BGNRPT
2443 010700                                     PRINTS  @FMTS1                  ;PRINT STATISTICAL HEADER
     010700  012746  007732                     MOV     @FMTS1,(SP)
     010704  012746  000001                     MOV     #1,(SP)
     010710  010600                             MOV     SP,R0
     010712  104416                             TRAP    C$PNTS
     010714  062706  000004                     ADD     #4,SP
2444                                                                            ;'RL01 RL02 PERFORMANCE REPORT"
2445
2446 010720  010446                             MOV     R4,-(SP)               ;SAVE PRESENT VALUE OF R4
2447 010722  012704  030362                     MOV     #DRBUF,R4              ;START OF DRIVE BUFFER
2448 010726  005764  000104             1$:     TST     DCS(R4)                ;IS THERE A DRIVE?
2449 010732  001402                             BEQ     2$                     ;NO, GET NEXT ONE
2450 010734  004737  013772                     JSR     PC,REPORT              ;TYPE OUT SUMMARY
2451 010740  062704  000126             2$:     ADD     #PRPOS*2,R4            ;NEXT DRIVE
2452 010744  020427  031642                     CMP     R4,#ENDBUF             ;AT THE END?
2453 010750  001366                             BNE     1$                     ;NO, TRY NEXT
2454 010752  012604                             MOV     (SP)+,R4               ;RESTORE R4
2455 010754                             ENDRPT
     010754                             L10015:
     010754  104425                             TRAP    C$RPT
2456
2457 010756                                     ENDMOD
2458
2459                                    .SBTTL  LOAD PROTECTION TABLE
2460 010756                             BGNPROT
2461 010756  000000                     .WORD    0                             ;P-TABLE OFFSET OF CSR
2462 010760  177777                     .WORD   -1                             ;NOT A MASS-BUS DRIVE
2463 010762  000010                     .WORD   10                             ;P-TABLE OFFSET OF DRIVE
2464 010764                             ENDPROT
2465
2466                                    .SBTTL  INITIALIZATION CODE
2467
2468 010764                             BGNMOD  INITCODE                       ;START OF INITIALIZE CODE
2469
2470 010764                                     BGNINIT
2471
2472 010764                                     SETVEC  #140,#170000,#340      ;ODT STARTING ADDR              ;JSD REV A
     010764  012746  000340                     MOV     #340,-(SP)
     010770  012746  170000                     MOV     #170000,-(SP)
     010774  012746  000140                     MOV     #140,-(SP)
     011000  012746  000003                     MOV     #3,-(SP)
     011004  104437                             TRAP    C$SVEC
     011006  062706  000010                     ADD     #10,SP
2473                                    ;        SETPRI  #340                   ;PRI TO 7 TO INHIBIT INT'S      ;JSD REV A
2474 011012                                     SETPRI  #300                   ;PRI TO 6 TO INHIBIT INT'S      ;JSD REV A
     011012  012700  000300                     MOV     #300,R0
     011016  104441                             TRAP    C$SPRI
2475
2476 011020                                     BRESET                         ;FOR LSI-11 CPU'S
     011020  104433                             TRAP    C$RESET
2477                                    ;CLEAR OPERATION FLAGS
```

INITIALIZATION CODE

```
2478 011022  005037  000050                    CLR     OPFLG
2479 011026  005037  002476                    CLR     INCALL
2480 011032  005037  002452                    CLR     STFLG
2481 011036  005037  002454                    CLR     CNTFLG                  ;CLEAR CONT
2482                                    ;CHECK FOR PRESENCE OF A SYSTEM CLOCK
2483 011042  005037  002502                    CLR     SYSCLK                  ;CLEAR SYSTEM CLOCK FLAG
2484 011046  000536                            BR      PWRCH                   ;FORCE THE PROGRAM TO ASSUME NO CLOCK    ;JSD REV A
2485 011050                                    CLOCK   P,CLKADR                ;P-CLOCK?
     011050  012700  000120                    MOV     #'P,R0
     011054  104462                            TRAP    C$CLCK
     011056  010037  002316                    MOV     R0,CLKADR
2486 011062                                    BNCOMPLETE      LCLKCH  ;BRANCH IF NO P-CLOCK
     011062  103006                            BCC     LCLKCH
2487 011064  012737  000001  002314            MOV     #1,CLKTYP               ;IDENTIFY P-CLOCK TYPE
2488 011072  005237  002502                    INC     SYSCLK                  ;INDICATE PRESENCE OF A SYSTEM CLOCK
2489 011076  000522                            BR      PWRCH                   ;BRANCH TO CHECK POWER
2490 011100                            LCLKCH: CLOCK   L,CLKADR                ;L-CLOCK?
     011100  012700  000114                    MOV     #'L,R0
     011104  104462                            TRAP    C$CLCK
     011106  010037  002316                    MOV     R0,CLKADR
2491 011112                                    BCOMPLETE       1$      ;BRANCH IF L-CLOCK
     011112  103401                            BCS     1$
2492 011114  000467                            BR      NILCLK                  ;ELSE, INDICATE CLOCK IS NOT PRESENT
2493 011116                            1$:     READBUS                         ;CHECK TYPE OF BUS
     011116  104407                            TRAP    C$RDBU
2494 011120                                    BNCOMPLETE      2$      ;BRANCH IF NOT Q-BUS
     011120  103057                            BCC     2$
2495 011122  005037  002514                    CLR     CLKFLD                  ;CLEAR CLOCK FIELD FOR STORING "TICKS"
2496 011126                                    SETVEC  #100,#CLKTIK,#340       ;SET UP L-CLOCK INTERRUPT VECTOR TO CHECK
     011126  012746  000340                    MOV     #340,-(SP)
     011132  012746  017040                    MOV     #CLKTIK,-(SP)
     011136  012746  000100                    MOV     #100,-(SP)
     011142  012746  000003                    MOV     #3,-(SP)
     011146  104437                            TRAP    C$SVEC
     011150  062706  000010                    ADD     #10,SP
2497                                                            ;/IF CLOCK IS "TICKING"
2498 011154                                    SETPRI  #240            ;SET PRIORITY TO 5 TO ALLOW CLOCK INTERRUPTS
     011154  012700  000240                    MOV     #240,R0
     011160  104441                            TRAP    C$SPRI
2499 011162                                    WAITMS  #5              ;PAUSE TO ALLOW CLOCK INTERRUPTS
     011200  012727  000372                    MOV     #250.,(PC)+
     011204  000000                            .WORD   0
     011206  013727  002116                    MOV     L$DLY,(PC)+
     011212  000000                            .WORD   0
     011214  005367  177772                    DEC     -6(PC)
     011220  001375                            BNE     .-4
     011222  005367  177756                    DEC     -22(PC)
     011226  001367                            BNE     .-20
2500                                    ;      SETPRI  #340            ;RESTORE PRI TO 7 TO INHIBIT INT'S       ;JSD REV A
2501 011236                                    SETPRI  #300            ;RESTORE PRI TO 6 TO INHIBIT INT'S       ;JSD REV A
     011236  012700  000300                    MOV     #300,R0
     011242  104441                            TRAP    C$SPRI
2502 011244                                    CLRVEC  #100            ;CLEAR L-CLOCK INTERRUPT VECTOR
     011244  012700  000100                    MOV     #100,R0
     011250  104436                            TRAP    C$CVEC
2503 011252  005737  00251A                    TST     CLKFLD          ;L-CLOCK "TICKS"?
2504 011256  001406                            BEQ     NILCLK          ;BRANCH IF NO "TICKS'
```

INITIALIZATION CODE

```
2505 011260  012737  000002  002314  2$:     MOV     #2,CLKTYP       ;IDENTIFY L-CLOCK TYPE
2506 011266  005237  002502                  INC     SYSCLK          ;INDICATE PRESENCE OF A SYSTEM CLOCK
2507 011272  000424                          BR      PWRCH           ;BRANCH TO CHECK POWER
2508 011274                          NILCLK: PRINTF  #FMT14,#NOCLK   ;REPORT "SYSTEM CLOCK IS NOT AVAILABLE"
     011274  012746  004530                  MOV     #NOCLK,-(SP)
     011300  012746  007463                  MOV     #FMT14,-(SP)
     011304  012746  000002                  MOV     #2,-(SP)
     011310  010600                          MOV     SP,R0
     011312  104417                          TRAP    C$PNTF
     011314  062706  000006                  ADD     #6,SP
2509 011320                                  PRINTF  #FMT14,#NOREPT  ;PRINT 'PERFORMANCE REPORTS WILL NOT BE PRINTED"
     011320  012746  004566                  MOV     #NOREPT,-(SP)
     011324  012746  007463                  MOV     #FMT14,-(SP)
     011330  012746  000002                  MOV     #2,-(SP)
     011334  010600                          MOV     SP,R0
     011336  104417                          TRAP    C$PNTF
     011340  062706  000006                  ADD     #6,SP
2510                                 ;POWER FAIL SEQUENCE
2511 011344                          PWRCH:  READEF  #EF.PWR         ;POWER FAILURE?
     011344  012700  000034                  MOV     #EF.PWR,R0
     011350  104447                          TRAP    C$REFG
2512 011352                                  BNCOMPLETE      3$      ;BRANCH IF NO POWER FAILURE
     011352  103121                          BCC     3$
2513 011354  005237  002446                  INC     PWRFLG          ;INDICATE POWER FAIL
2514 011360  012704  030362                  MOV     #DRBUF,R4       ;INITIALIZE POINTER TO DRIVE PARAMETER BUFFERS
2515 011364  012702  000001                  MOV     #1,R2
2516 011370  130237  002252          11$:    BITB    R2,DRUT
2517 011374  001471                          BEQ     13$
2518 011376  016400  000106                  MOV     DRSEL(R4),R0
2519 011402  052700  000200                  BIS     #200,R0
2520 011406  010074  000104                  MOV     R0,@DCS(R4)
2521 011412  012701  000170                  MOV     #120.,R1        ;INITIALIZE WAIT COUNT
2522 011416  032774  000001  000104  12$:    BIT     #1,@DCS(R4)
2523 011424  001037                          BNE     15$
2524 011426                                  WAITMS  #10.            ;IMPLEMENT 1 SECOND TIME DELAY
     011444  012727  000372                  MOV     #250.,(PC)+
     011450  000000                          .WORD   0
     011452  013727  002116                  MOV     L$DLY,(PC)+
     011456  000000                          .WORD   0
     011460  005367  177772                  DEC     -6(PC)
     011464  001375                          BNE     .-4
     011466  005367  177756                  DEC     -22(PC)
     011472  001367                          BNE     .-20
2525 011502  005301                          DEC     R1
2526 011504  001344                          BNE     12$
2527 011506  012737  004101  002246          MOV     #NOPWR,WMY      ;MSG. "DR DID REC'R FROM PWR UP"
2528 011514  004537  023450                  JSR     R5,DRDRV
2529 011520  000137  011560                  JMP     13$
2530
2531 011524  004537  024376          15$:    JSR     R5,ISDRST
2532 011530  004537  025620                  JSR     R5,HDHOME
2533 011534  005064  000056                  CLR     PRFLGS(R4)
2534 011540  005064  000036                  CLR     RETRY(R4)
2535 011544  005064  000076                  CLR     DOWCK(R4)
2536 011550  005064  000052                  CLR     RTYPE(R4)
2537 011554  005064  000114                  CLR     RSEEK(R4)
2538 011560  062704  000126          13$:    ADD     #PRPOS+2,R4
```

INITIALIZATION CODE

```
2539 011564  106302                    ASLB    R2
2540 011566  103300                    BCC     11$
2541 011570  005737  002502            TST     SYSCLK          ;SYSTEM CLOCK AVAILABLE?
2542 011574  001406                    BEQ     4$
2543 011576                            CLKON                   ;ACTIVATE CLOCK WITH 1-SECOND INCREMENTS
2544 011606                            REQTIM  R0              ;REQUEST ELAPSED SUPERVISOR TIME
2545 011612  000137  012642      4$:   JMP     INIEND
2546                                   ;"CONTINUE" COMMAND SEQUENCE
2547 011616                      3$:   READEF  @#EF.CONTINUE           ;CONTINUE FROM CONSOLE?
     011616  012700  000036            MOV     @#EF.CONTINUE,R0
     011622  104447                    TRAP    C$REFG
2548 011624                            BNCOMPLETE      1$              ;NO, CONTINUE W/ INIT CODE
     011624  103004                    BCC     1$
2549
2550 011626  005237  002454            INC     CNTFLG                  ;YES SET CONT FLAG, GO TO END OF INIT
2551 011632  000137  012170            JMP     END
2552
2553 011636  004537  027124      1$:   JSR     R5,CLEAR                ;CLEAR ALL DRIVE BUFFERS
2554 011642  012737  176543  002260    MOV     #176543,HINUM           ;PRIME RANDOM GENERATOR
2555 011650  012737  123456  002262    MOV     #123456,LONUM           ;   "       "       "
2556 011656  012700  002320      2$:   MOV     #CNTLR1,R0              ;INITIALIZE POINTER TO GLOBAL DATA AREA
2557 011662  005020              CLRDAT: CLR   (R0)+                   ;MASS CLEAR OF GLOBAL DATA AREA
2558 011664  020027  002454            CMP     R0,#STFLG+2             ;AT END OF GLOBAL DATA AREA?
2559 011670  001374                    BNE     CLRDAT                  ;
2560
2561 011672  012704  030362            MOV     #DRBUF,R4               ;SET UP DRIVE INFORMATION BUFFER POINTER
2562 011676  012702  027314            MOV     #BSECO,R2               ;SET UP BAD SECTOR POINTER
2563 011702  013703  002012            MOV     L$UNIT,R3               ;GET NUMBER OF UNITS
2564 011706  010337  002444            MOV     R3,UUT                  ;SAVE L$UNIT
2565 011712  005001                    CLR     R1                      ;INITIALIZE P-TABLE FOR LOGICAL UNIT
2566 011714  005703              1$:   TST     R3                      ;ANY    P-TABLES LEFT?
2567 011716  001524                    BEQ     END                     ;NO,GO TO END
2568 011720                            GPHARD  R1,R0                   ;REQUEST A P-TABLE FOR DRIVE
     011720  010100                    MOV     R1,R0
     011722  104442                    TRAP    C$GPHRD
2569 011724                            BNCOMPLETE      12$
     011724  103112                    BCC     12$
2570                                   ;MOVE P-TABLE CONTENTS TO LOCAL STORAGE
2571 011726  012037  002330            MOV     (R0)+,BCSR              ;GET CSR
2572 011732  012037  002332            MOV     (R0)+,BVEC              ;GET VECTOR
2573 011736  012037  002334            MOV     (R0)+,BPRIOR            ;GET PRIORITY
2574 011742  012037  002254            MOV     (R0)+,T.DRIVE           ;GET DRIVE TYPE
2575 011746  011037  002336            MOV     (R0),BDRSEL             ;GET DRIVE NUMBER
2576 011752  005737  002320            TST     CNTLR1                  ;DO WE HAVE CSR 1 YET?
2577 011756  001011                    BNE     2$                      ;YES,THEN SEE IF IT THIS DRIVE IS
2578                                                                   ;/ASSOCIATED WITH CNTLR1
2579 011760  013737  002334  002376    MOV     BPRIOR,PRIOR1
2580 011766  013737  002330  002320    MOV     BCSR,CNTLR1             ;NO,MAKE THIS ONE CSR 1
2581 011774  013737  002332  002372    MOV     BVEC,VECT1              ;MAKE THIS VECTOR VECT1
2582 012002  023737  002330  002320  2$:   CMP     BCSR,CNTLR1         ;IS THIS CSR CNTLR1?
2583 012010  001012                    BNE     5$                      ;NO,GO CHECK AGAINST #2
2584 012012  023737  002332  002372    CMP     BVEC,VECT1              ;IS VECTOR PROPER?
2585 012020  001050                    BNE     10$                     ;NO, REPORT ERROR
2586 012022  012737  002436  002350    MOV     #BUF1,TEMP1             ;FIRST CONTROLLER/FIRST BUFFER
2587 012030  004537  013442            JSR     R5,FILINF               ;FILL BUFFER
2588 012034  000450                    BR      11$                     ;GO GET NEXT P-TABLE
2589 012036  005737  002322      5$:   TST     CNTLR2                  ;HAVE WE GOT CSR #2 YET?
```

INITIALIZATION CODE

```
2590 012042  001015                          BNE     6$                      ;YES, CHECK THIS ONE AGAINST IT
2591 012044  023737  002372  002330          CMP     VECT1,BCSR              ;IS THIS VECTOR SAME AS CNTLR1
2592 012052  001433                          BEQ     10$                     ;IF SO, DON'T ALLOW IT
2593 012054  013737  002330  002322          MOV     BCSR,CNTLR2             ;MAKE THIS ONE CSR 2
2594 012062  013737  002332  002374          MOV     BVEC,VECT2              ;SETUP SECOND VECTOR
2595 012070  013737  002334  002400          MOV     BPRIOR,PRIOR2
2596 012076  023737  002330  002322  6$:     CMP     BCSR,CNTLR2             ;IS THIS CSR # 2?
2597 012104  001016                          BNE     10$                     ;NO, WELL WE DON'T ALLOW 3
2598 012106  023737  002332  002374          CMP     BVEC,VECT2              ;DOES IT HAVE PROPER VECTOR
2599 012114  001012                          BNE     10$                     ;NO, GO REPORT ERROR
2600 012116  023737  002374  002372          CMP     VECT2,VECT1             ;IS VECTOR OF FIRST EQUAL TO
2601 012124  001406                          BEQ     10$                     ;VECTOR OF SECOND, YES REPORT ERROR
2602 012126  012737  002440  002350          MOV     #BUF2,TEMP1             ;OTHER CNTLR/OTHER BUFFER
2603 012134  004537  013442                  JSR     R5,FILINF               ;LOAD BUFFER
2604 012140  000406                          BR      11$                     ;NEXT
2605 012142                          10$:     ERRDF   160.,ILLEG,ERR10        ;BAD P-TABLE
     012142  104455                          TRAP    C$ERDF
     012144  000240                          .WORD   160
     012146  003770                          .WORD   ILLEG
     012150  005646                          .WORD   ERR10
2606 012152  005064  000104          12$:     CLR     DCS(R4)
2607 012156  005201                  11$:     INC     R1                      ;POINT TO NEXT
2608 012160  005303                          DEC     R3                      ;DOWN COUNT
2609 012162  062702  000042                  ADD     #34.,R2                 ;NEXT BAD SECTOR FILE
2610 012166  000652                          BR      1$                      ;DO WHILE
2611
2612 012170                          END:
2613
2614 012170  012737  177770  002256          MOV     #177770,SYSMSK          ;SETUP FOR EIGHT DRIVES
2615 012176  023727  002444  000004          CMP     UUT,#4                  ;MORE THAN FOUR
2616 012204  003012                          BGT     2$                      ;YES, THEN MASK IS OKAY
2617 012206  052737  000004  002256          BIS     #4,SYSMSK               ;SETUP FOR FOUR DRIVES
2618 012214  023727  002444  000002          CMP     UUT,#2                  ;MORE THAN TWO
2619 012222  003003                          BGT     2$                      ;YES, IT'S OKAY
2620 012224  052737  000002  002256          BIS     #2,SYSMSK               ;SET FOR ONE OR TWO
2621
2622                                 ;"START" COMMAND SEQUENCE
2623 012232                          2$:     READEF  #EF.START               ;START COMMAND
     012232  012700  000040                  MOV     #EF.START,R0
     012236  104447                          TRAP    C$REFG
2624 012240                                  BNCOMPLETE      RESTART         ;NO, CHK RESTART
     012240  103006                          BCC     RESTART
2625 012242  005237  002452                  INC     STFLG                   ;SET START INDICATOR
2626 012246  005037  002274                  CLR     WRINIT                  ;CLEAR THE WRITE INIT FLAG ON START
2627 012252  005037  002310                  CLR     KILLDC                  ;CLEAR DATA COMP FLAG ON START ONLY
2628
2629 012256                          RESTART:
2630 012256  005737  002454                  TST     CNTFLG                  ;CONTINUING
2631 012262  001047                          BNE     3$                      ;YES GO TO 3$
2632 012264  005737  002274                  TST     WRINIT                  ;IN PROCESS OF INITTING THE PACK?
2633 012270  001420                          BEQ     11$                     ;NO
2634 012272  005037  002274                  CLR     WRINIT                  ;YES - CLEAR THE FLAG
2635 012276  005237  002310                  INC     KILLDC                  ;INHIBIT DATA COMPARES!
2636 012302  005037  010612                  CLR     CMRD                    ;AND SET DAT COMPARE TO 0 WORDS
2637 012306                                  PRINTF  #FMT18,#NORDDC          ;TELL OPR PACK NOT INITTED YET
     012306  012746  004416                  MOV     #NORDDC,-(SP)
     012312  012746  007666                  MOV     #FMT18,-(SP)
```

INITIALIZATION CODE

```
                012316  012746  000002                  MOV     #2,-(SP)
                012322  010600                          MOV     SP,R0
                012324  104417                          TRAP    C$PNTF
                012326  062706  000006                  ADD     #6,SP
2638
2639                                            ;LET'S CREATE INTERNAL BITMAP
2640
2641 012332     012701  000001          11$:    MOV     #1,R1                   ;BIT MASK
2642 012336     105037  002253                  CLRB    DRPRS                   ;CLEAR OUT DRIVES PRESENT
2643 012342     012704  030362                  MOV     #DRBUF,R4               ;START OF DRIVE BUFFERS
2644 012346     005764  000104          1$:     TST     DCS(R4)                 ;ANY CSR?
2645 012352     001402                          BEQ     2$                      ;NO, NO DRIVE THEN
2646 012354     150137  002253                  BISB    R1,DRPRS                ;INDICATE DRIVE IN BITMAP
2647 012360     006301                  2$:     ASL     R1                      ;NEXT POSITION
2648 012362     062704  000126                  ADD     #PRPOS*2,R4             ;NEXT DRIVE BUFFER
2649 012366     022704  031642                  CMP     #ENDBUF,R4              ;DONE
2650 012372     001365                          BNE     1$                      ;NO
2651
2652 012374     113737  002253  002252          MOVB    DRPRS,DRUT              ;SET UP DRIVES UNDER TEST
2653
2654 012402                             3$:
2655
2656 012402                                     SETVEC  VECT1,#INTR1,PRIOR1     ;SET CONTROLLER 1'S VECTOR
                012402  013746  002376          MOV     PRIOR1,-(SP)
                012406  012746  017046          MOV     #INTR1,-(SP)
                012412  013746  002372          MOV     VECT1,-(SP)
                012416  012746  000003          MOV     #3,-(SP)
                012422  104437                  TRAP    C$SVEC
                012424  062706  000010          ADD     #10,SP
2657
2658 012430     005737  002322                  TST     CNTLR2                  ;RUNNING TWO CONTROLLERS?
2659 012434     001413                          BEQ     4$                      ;NO
2660
2661 012436                                     SETVEC  VECT2,#INTR2,PRIOR2     ;YES SET CONTROLLER 2'S VECTOR
                012436  013746  002400          MOV     PRIOR2,-(SP)
                012442  012746  017056          MOV     #INTR2,-(SP)
                012446  013746  002374          MOV     VECT2,-(SP)
                012452  012746  000003          MOV     #3,-(SP)
                012456  104437                  TRAP    C$SVEC
                012460  062706  000010          ADD     #10,SP
2662
2663 012464     005737  002454          4$:     TST     CNTFLG                  ;CONTINUE?
2664 012470     001412                          BEQ     FINDBF                  ;NO, GO PAST RESTART OF CLOCK
2665
2666 012472     005737  002502                  TST     SYSCLK                  ;DO WE HAVE SYSTEM CLOCK?
2667 012476     001461                          BEQ     INIEND                  ;NO
2668
2669 012500                                     CLKON                           ;ACTIVATE SYSTEM CLOCK
2670 012510                                     REQTIM  R0                      ;REQUEST ELAPSED SUPERVISOR TIME
2671 012514     000452                          BR      INIEND                  ;GO TO END
2672
2673                                    ;REQUEST MEMORY BUFFER SPACE TO PERFORM READ/WRITE OPERATIONS
2674 012516                             FINDBF: MEMORY  R2                      ;REQUEST MEMORY BUFFER SPACE
                012516  104431                  TRAP    C$MEM
                012520  010002                  MOV     R0,R2
2675 012522     022712  002400                  CMP     #1280.,(R2)             ;DO WE HAVE A MINIMUM OF 1280 WORDS?
2676 012526     003413                          BLE     1$                      ;YES   BRANCH
```

INITIALIZATION CODE

```
2677 012530                                    PRINTF  #FMT14,#INSMEM         ;NO   PRINT MSG. "SYSTEM FATAL ERROR
     012530  012746  004711                    MOV     #INSMEM,-(SP)
     012534  012746  007463                    MOV     #FMT14, (SP)
     012540  012746  000002                    MOV     #2,-(SP)
     012544  010600                            MOV     SP,R0
     012546  104417                            TRAP    C$PNTF
     012550  062706  000006                    ADD     #6,SP
2678                                                                        ;/INSUFFICIENT MEMORY BUFFER SPACE"
2679 012554  000000                            HALT
2680 012556  010237  002436            1$:     MOV     R2,BUF1             ;GET ADDRESS OF FREE MEMORY
2681 012562  005737  002322                    TST     CNTLR2              ;TWO CONTROLLERS?
2682 012566  001410                            BEQ     2$                  ;NO - ASSIGN ALL BUFFER TO SINGLE CONTROLLER
2683 012570  042712  000001                    BIC     #1,(R2)             ;MAKE LENGTH OF FREE MEMORY EVEN
2684 012574  013737  002436  002440            MOV     BUF1,BUF2           ;SET UP FOR BUFFER 2
2685 012602  061237  002440                    ADD     (R2),BUF2           ;ADD HALF OF BUFFER
2686 012606  006212                            ASR     (R2)                ;DIVIDE BUFFER SPACE BY 2
2687 012610  011237  002442            2$:     MOV     (R2),MAXWC          ;INITIALIZE MAXIMUM WORD COUNT
2688 012614  023727  002442  012000            CMP     MAXWC,#5120.        ;IS WORD COUNT LESS THAN OR EQUAL TO 5120?
2689 012622  003403                            BLE     3$                  ;BRANCH IF TRUE
2690 012624  012737  012000  002442            MOV     #5120.,MAXWC        ;NO - INITIALIZE VALUE TO 5120 WORDS
2691
2692 012632                            3$:     CLKON                       ;ACTIVATE SYSTEM CLOCK TO INITIATE GENERATION
2693                                                                        ;/OF TIMING INTERVALS
2694 012642                            INIEND:
2695 012642                                    ENDINIT
     012642                            L10017:
     012642  104411                            TRAP    C$INIT
2696 012644                            ENDMOD
2697
2698                                   .SBTTL  AUTO DROP SECTION
2699
2700                                   ;THE AUTO DROP SECTION IS CONDITIONALLY EXECUTED AFTER THE INITIALIZATION CODE
2701                                   ;WHEN THE OPERATOR "ADR" FLAG IS SET. EACH DRIVE IS CHECKED TO DETERMINE IF IT
2702                                   ;IS READY TO TRANSFER DATA. IF THE DRIVE DOES NOT RESPOND WITH "READY" IT IS
2703                                   ;DROPPED FROM THE TEST CYCLE. THE HARDWARE TESTS ARE PERFORMED IMMEDIATELY
2704                                   ;AFTER THE READY STATUS OF ALL DRIVES HAVE BEEN CHECKED.
2705
2706 012644                            BGNAUTO
2707 012644  010346                            MOV     R3,-(SP)            ;SAVE REGISTERS
2708 012646  010446                            MOV     R4,-(SP)
2709 012650  013703  002012                    MOV     L$UNIT,R3           ;INITIALIZE NUMBER OF DRIVES UNDER TEST
2710 012654  012704  030362                    MOV     #ORBUF,R4           ;INITIALIZE START OF DRIVE BUFFERS
2711 012660  005037  002450            1$:     CLR     TRPFLG              ;CLEAR TRAP FLAG
2712 012664                                    SETVEC  ERRVEC,#TRPHAN,#340     ;SET UP TIME-OUT VECTOR TO DETECT
     012664  012746  000340                    MOV     #340,-(SP)
     012670  012746  013764                    MOV     #TRPHAN,-(SP)
     012674  013746  002466                    MOV     ERRVEC,-(SP)
     012700  012746  000003                    MOV     #3,-(SP)
     012704  104437                            TRAP    C$SVEC
     012706  062706  000010                    ADD     #10,SP
2713                                                                        ;/NON-EXISTENT CONTROLLER
2714 012712  005774  000104                    TST     @DCS(R4)            ;ACCESS CONTROLLER
2715 012716  005737  002450                    TST     TRPFLG              ;DID TRAP OCCUR?
2716 012722  001425                            BEQ     2$                  ;BRANCH TO CHECK DRIVE IF TRAP DID NOT OCCUR
2717 012724                                    PRINTF  #FRMT16,DCS(R4),<B,DRSEL+1(R4)> ;PRINT CONTROL STATUS AND DRIVE
     012724  005046                            CLR     -(SP)
     012726  156416  000107                    BISB    DRSEL+1(R4),(SP)
```

AUTO DROP SECTION

```
        012732  016446  000104                  MOV     DCS(R4),-(SP)
        012736  012746  007624                  MOV     #FRMT16, (SP)
        012742  012746  000003                  MOV     #3, (SP)
        012746  010600                          MOV     SP,R0
        012750  104417                          TRAP    C$PNTF
        012752  062706  000010                  ADD     #10,SP
2718                                                                    ;/NUMBER INFORMATION
2719 012756  012737  004673  002246             MOV     #NOCTLR,WHY     ;PROVIDE REASON FOR DROPPING DRIVE
2720                                                                    ;/"NO CONTROLLER"
2721 012764  004537  023450                     JSR     R5,DRDRV        ;DO DROP UNIT ON DRIVE FROM TEST CYCLE
2722 012770  005064  000104                     CLR     DCS(R4)         ;TAKE DRIVE OUT OF BUFFER
2723 012774  000436                             BR      3$              ;BRANCH TO GET NEXT DRIVE
2724 012776  056474  000106  000104  2$:       BIS     DRSEL(R4),@DCS(R4)      ;GET SELECTED DRIVE NUMBER
2725 013004  052774  000200  000104            BIS     #200,@DCS(R4)   ;SET CONTROLLER READY
2726 013012  032774  000001  000104            BIT     #1,@DCS(R4)     ;IS DRIVE READY?
2727 013020  001024                             BNE     3$              ;BRANCH TO CHECK NEXT DRIVE IF READY
2728 013022                                     PRINTF  #FRMT16,DCS(R4),<B,DRSEL+1(R4)> ;PRINT CONTROL STATUS AND DRIVE
        013022  005046                          CLR     -(SP)
        013024  156416  000107                  BISB    DRSEL+1(R4),(SP)
        013030  016446  000104                  MOV     DCS(R4),-(SP)
        013034  012746  007624                  MOV     #FRMT16,-(SP)
        013040  012746  000003                  MOV     #3,-(SP)
        013044  010600                          MOV     SP,R0
        013046  104417                          TRAP    C$PNTF
        013050  062706  000010                  ADD     #10,SP
2729                                                                    ;/NUMBER INFORMATION
2730 013054  012737  004636  002246             MOV     #NOTRDY,WHY     ;PROVIDE REASON FOR DROPPING DRIVE
2731                                                                    ;/"DID NOT RESPOND WITH "READY"
2732 013062  004537  023450                     JSR     R5,DRDRV        ;DO DROP UNIT ON DRIVE FROM TEST CYCLE
2733 013066  005064  000104                     CLR     DCS(R4)         ;TAKE DRIVE OUT OF BUFFER
2734 013072                          3$:        CLRVEC  ERRVEC          ;RELEASE THE ERROR VECTOR
        013072  013700  002466                  MOV     ERRVEC,R0
        013076  104436                          TRAP    C$CVEC
2735 013100  062704  000126                     ADD     #PRPOS+2,R4     ;UPDATE POINTER TO ACCESS DRIVE BUFFER
2736                                                                    ;/FOR NEXT DRIVE
2737 013104  005303                             DEC     R3              ;DECREMENT DRIVE COUNT
2738 013106  001264                             BNE     1$              ;BRANCH TO GET NEXT DRIVE IF MORE
2739 013110  012604                             MOV     (SP)+,R4        ;RESTORE REGISTERS
2740 013112  012603                             MOV     (SP)+,R3
2741 013114                          ENDAUTO
        013114                       L10020:
        013114  104461                          TRAP    C$AUTO
2742
2743 013116                          BGNMOD     CLNCODE
2744
2745 013116                                     BGNCLN
2746
2747 013116                                     SETVEC  ERRVEC,#TRPHAN,#340
        013116  012746  000340                  MOV     #340,-(SP)
        013122  012746  013764                  MOV     #TRPHAN,-(SP)
        013126  013746  002466                  MOV     ERRVEC,-(SP)
        013132  012746  000003                  MOV     #3,-(SP)
        013136  104437                          TRAP    C$SVEC
        013140  062706  000010                  ADD     #10,SP
2748 013144                                     SETPRI  #PRI00                  ;PRIORITY TO ZERO
        013144  012700  000000                  MOV     #PRI00,R0
        013150  104441                          TRAP    C$SPRI
```

AUTO DROP SECTION

```
2749
2750 013152  032777  000200  167140  1$:     BIT     #CRDY,@CNTLR1           ;WAIT FOR CONTROLLER TO FINISH
2751 013160  001774                          BEQ     1$                      ;
2752 013162  042777  000100  167130          BIC     #INTEN,@CNTLR1          ;CLEAR INTERRUPT IF PENDING
2753 013170                                  CLRVEC  VECT1                   ;RELEASE VECTOR OF FIRST CONTROLLER
     013170  013700  002372                  MOV     VECT1,R0
     013174  104436                          TRAP    C$CVEC
2754
2755 013176  005737  002322                  TST     CNTLR2                  ;TWO CONTROLLERS
2756 013202  001412                          BEQ     3$                      ;NO
2757
2758 013204  032777  000200  167110  2$:     BIT     #CRDY,@CNTLR2           ;WAIT FOR OTHER CONTROLLER TO FINISH
2759 013212  001774                          BEQ     2$                      ;
2760 013214  042777  000100  167100          BIC     #INTEN,@CNTLR2          ;CLEAR OUT INTERRUPT ENABLE
2761 013222                                  CLRVEC  VECT2                   ;YES, WELL RELEASE ITS VECTOR
     013222  013700  002374                  MOV     VECT2,R0
     013226  104436                          TRAP    C$CVEC
2762
2763 013230  005037  002476          3$:     CLR     INCALL
2764 013234  005037  002474                  CLR     OPCALL
2765 013240                                  CLRVEC  ERRVEC
     013240  013700  002466                  MOV     ERRVEC,R0
     013244  104436                          TRAP    C$CVEC
2766 013246  005737  002502                  TST     SYSCLK
2767 013252  001416                          BEQ     4$
2768 013254                                  CLKOFF                          ;DEACTIVATE SYSTEM CLOCK
2769 013310                          4$:     BRESET                          ;TAKE CARE OF LSI 11
     013310  104433                          TRAP    C$RESET
2770 013312                                  ENDCLN
     013312                          L10021:
     013312  104412                          TRAP    C$CLEAN
2771
2772 013314                                  ENDMOD
2773
2774 013314                          BGNMOD  ADDCODE
2775
2776 013314                          BGNAU
2777
2778 013314  012704  030362                  MOV     #DRBUF,R4               ;START OF DRIVE BUFFERS
2779 013320  012701  000001                  MOV     #1,R1                   ;MASK TO FIND DRIVE
2780 013324  010002                          MOV     R0,R2                   ;SAVE WHICH TO FIND
2781 013326  005700                  1$:     TST     R0                      ;THIS ONE
2782 013330  001405                          BEQ     2$                      ;YES
2783 013332  062704  000126                  ADD     #PRPOS+2,R4             ;NEXT
2784 013336  006301                          ASL     R1                      ;NEXT MASK
2785 013340  005300                          DEC     R0
2786 013342  000771                          BR      1$
2787 013344  150137  002252          2$:     BISB    R1,DRUT                 ;INSERT IN DRIVE UNDER TEST
2788 013350                                  GPHARD  R2,R1
     013350  010200                          MOV     R2,R0
     013352  104442                          TRAP    C$GPHRD
     013354  010001                          MOV     R0,R1
2789 013356  011164  000104                  MOV     (R1),DCS(R4)
2790 013362  012700  000100                  MOV     #SERNM1,R0              ;SETUP TO CLEAR STATUS
2791 013366  006200                          ASR     R0
2792 013370  005024                  4$:     CLR     (R4)+
2793 013372  005300                          DEC     R0
```

AUTO DROP SECTION

```
2794 013374  001375                          BNE     4$
2795 013376                          5$:
2796
2797 013376                          ENDAU
     013376                          L10022:
     013376  104452                          TRAP    C$AU
2798
2799 013400                          ENDMOD
2800
2801 013400                          BGNMOD  DROPCODE
2802
2803 013400                          BGNDU
2804
2805 013400  005737  002476                  TST     INCALL
2806 013404  001015                          BNE     3$
2807 013406  012704  030362                  MOV     #DRBUF,R4
2808 013412  005700                  2$:     TST     R0
2809 013414  001404                          BEQ     1$
2810 013416  005300                          DEC     R0
2811 013420  062704  000126                  ADD     #PRPOS+2,R4
2812 013424  000772                          BR      2$
2813
2814 013426  012737  003510  002246  1$:     MOV     #REQ,WHY
2815 013434  004537  023444                  JSR     R5,ODRDRV
2816 013440                          3$:
2817
2818 013440                          ENDDU
     013440                          L10023:
     013440  104453                          TRAP    C$DU
2819
2820 013442                          ENDMOD
2821
2822                                 .SBTTL  GLOBAL SUBROUTINES
2823
2824 013442                          BGNMOD  GLBSUB
2825                                 ;
2826                                 ;ROUTINE TO FILL DRIVE PARAMETER BUFFERS WITH INFORMATION
2827
2828 013442  013764  002336  000106  FILINF: MOV     BDRSEL,DRSEL(R4)        ;SET DRIVE SELECT BITS
2829 013450  022737  000001  002254          CMP     #1,T.DRIVE              ;DRIVE = RL01?
2830 013456  001403                          BEQ     FILTD                   ;YES
2831 013460  012737  000002  002254          MOV     #2,T.DRIVE              ;DRIVE IS AN RL02
2832 013466  013764  002254  000120  FILTD:  MOV     T.DRIVE,TDR(R4)
2833 013474  013764  002330  000104          MOV     BCSR,DCS(R4)            ;SET CSR
2834 013502  013764  002350  000110          MOV     TEMP1,BBA(R4)           ;SET R/W BUFFER
2835 013510  010264  000112                  MOV     R2,BSECPT(R4)           ;SETUP BAD SECTOR POINTER
2836 013514  062704  000126                  ADD     #PRPOS+2,R4             ;UPDATE POINTER
2837 013520  000205                          RTS     R5
2838
2839                                 ;SETS UP CLOCK INTERRUPT VECTOR, CLOCK COUNT, AND IDENTIFIES CLOCK FREQUENCY
2840
2841 013522  010346                  CLKINI: MOV     R3,-(SP)                ;SAVE R3
2842 013524  022737  000001  002314          CMP     #1,CLKTYP               ;P-CLOCK?
2843 013532  001014                          BNE     LCLK                    ;BRANCH IF NOT P-CLOCK
2844 013534                          SETVEC  #104,#UPDATE,#340               ;SET P-CLOCK INTERRUPT VECTOR
     013534  012746  000340                  MOV     #340,-(SP)
     013540  012746  016636                  MOV     #UPDATE,-(SP)
```

GLOBAL SUBROUTINES

```
                013544  012746  000104                  MOV     #104,-(SP)
                013550  012746  000003                  MOV     #3,-(SP)
                013554  104437                          TRAP    C$SVEC
                013556  062706  000010                  ADD     #10,SP
   2845         013562  000417                          BR      FRQCHK                  ;BRANCH FOR SYSTEM FREQUENCY CHECK
   2846         013564  022737  000002  002314  LCLK:   CMP     #2,CLKTYP               ;L-CLOCK?
   2847         013572  001036                          BNE     ENDINI                  ;BRANCH IF NO CLOCCK
   2848         013574                                  SETVEC  #100,#UPDATE,#340       ;SET L-CLOCK INTERRUPT VECTOR
                013574  012746  000340                  MOV     #340,-(SP)
                013600  012746  016636                  MOV     #UPDATE,-(SP)
                013604  012746  000100                  MOV     #100,-(SP)
                013610  012746  000003                  MOV     #3,-(SP)
                013614  104437                          TRAP    C$SVEC
                013616  062706  000010                  ADD     #10,SP
   2849         013622  013703  002316          FRQCHK: MOV     CLKADR,R3               ;GET BASE ADDRESS OF THE SUPERVISOR CLOCK TABLE
   2850         013626  022763  000074  000006          CMP     #60.,6(R3)              ;60 HZ?
   2851         013634  001007                          BNE     FRQ50                   ;BRANCH FOR 50 HZ
   2852         013636  012737  000074  002506          MOV     #60.,CLKCNT             ;INITIALIZE CLOCK COUNT FOR 60 TICKS
   2853                                                                                  ;/PER SECOND
   2854         013644  012737  000001  002312          MOV     #1,CLKFRQ               ;IDENTIFY CLOCK FREQUENCY IS 60 HZ
   2855         013652  000406                          BR      ENDINI                  ;RETURN
   2856         013654  012737  000062  002506  FRQ50:  MOV     #50.,CLKCNT             ;INITIALIZE CLOCK COUNT FOR 50 TICKS
   2857                                                                                  ;/PER SECOND
   2858         013662  012737  000002  002312          MOV     #2,CLKFRQ               ;IDENTIFY CLOCK FREQUENCY IS 50 HZ
   2859         013670  012603                  ENDINI: MOV     (SP)+,R3                ;RESTORE R3
   2860         013672  000207                          RTS     PC
   2861
   2862                                          ;DETERMINES CLOCK TYPE AND INITIALIZES THE CLOCK FOR OPERATION IN REPEAT
   2863                                          ;INTERRUPT MODE AT LINE FREQUENCY
   2864
   2865         013674  005037  002512          CLKST:  CLR     CLKACC                  ;CLEAR CLOCK ELAPSED TIME INDICATOR
   2866         013700  022737  000002  002314          CMP     #2,CLKTYP               ;L-CLOCK?
   2867         013706  001006                          BNE     1$                      ;BRANCH FOR P-CLOCK
   2868         013710  012737  000100  177546          MOV     #100,@#177546           ;SET INTERRUPT ENABLE BIT TO 1
   2869         013716  005237  002504                  INC     CLKSON                  ;INDICATE "CLOCK ON"
   2870         013722  000414                          BR      2$                      ;BRANCH TO SET UP TIME INCREMENTS
   2871         013724  022737  000001  002314  1$:     CMP     #1,CLKTYP               ;P-CLOCK?
   2872         013732  001013                          BNE     3$                      ;BRANCH IF NO CLOCK
   2873         013734  012737  000001  172542          MOV     #1,@#172542             ;SET UP P-CLOCK FOR 1 INTERRUPT PER TICK
   2874         013742  012737  000115  172540          MOV     #115,@#172540           ;SET INTERRUPT ENABLE,REPEAT INTERRUPT MODE,
   2875                                                                                  ;/LINE FREQUENCY RATE,START CLOCK
   2876         013750  005237  002504                  INC     CLKSON                  ;INDICATE "CLOCK ON"
   2877         013754  013737  002506  002510  2$:     MOV     CLKCNT,CLKBFR           ;SET UP TIME INCREMENTS
   2878         013762  000207                  3$:     RTS     PC                      ;RETURN
   2879
   2880         013764  005237  002450          TRPHAN: INC     TRPFLG
   2881         013770  000002                          RTI
   2882
   2883                                          .SBTTL  REPORT ROUTINE
   2884                                          ;ROUTINE TO PRINT STATISTICAL REPORT OF DRIVE(S)
   2885
   2886                                          ;REPORT:PRINTS  #FMT10,#TIME,HOUR,MINUTE,SECOND,#MRLCS,DCS(R4),#DRNM,<B,DRSEL+1(R4)>     ;JSD
REV A
   2887         013772                           REPORT: PRINTS  #FMT10,#MRLCS,DCS(R4),#DRNM,<B,DRSEL+1(R4)>                             ;JSD
REV A
                013772  005046                          CLR     -(SP)
                013774  156416  000107                  BISB    DRSEL+1(R4),(SP)
                014000  012746  004030                  MOV     #DRNM,-(SP)
                014004  016446  000104                  MOV     DCS(R4),-(SP)
```

REPORT ROUTINE

```
                014010  012746  002525                  MOV     #MRLCS, (SP)
                014014  012746  007277                  MOV     #FMT10,-(SP)
                014020  012746  000005                  MOV     #5, (SP)
                014024  010600                          MOV     SP,R0
                014026  104416                          TRAP    C$PNTS
                014030  062706  000014                  ADD     #14,SP
           2888 014034                                  PRINTS  #FMTDT,TDR(R4)
                014034  016446  000120                  MOV     TDR(R4),-(SP)
                014040  012746  010103                  MOV     #FMTDT,-(SP)
                014044  012746  000002                  MOV     #2,-(SP)
                014050  010600                          MOV     SP,R0
                014052  104416                          TRAP    C$PNTS
                014054  062706  000006                  ADD     #6,SP
           2889 014060  000432                          BR      2$                  ;SKIP THIS BECAUSE DPHOUR(R4) WILL ALWAYS       ;JSD REV A
           2890                                                                     ;...BE ZERO, EVEN FOR DROPPED UNIT              ;JSD REV A
           2891 014062  005764  000070                  TST     DPHOUR(R4)              ;DO WE HAVE ANY DROPPED TIME
           2892 014066  001417                          BEQ     1$                      ;NO, THEN PRINT "RUNNING'
           2893
           2894                                 ;PRINT THE TIME THE DRIVE WAS DROPPED FROM TESTING
           2895
           2896 014070                                  PRINTS  #FMTS1B,<B,DPHOUR(R4)>,<B,DPMIN(R4)>
                014070  005046                          CLR     -(SP)
                014072  156416  000071                  BISB    DPMIN(R4),(SP)
                014076  005046                          CLR     -(SP)
                014100  156416  000070                  BISB    DPHOUR(R4),(SP)
                014104  012746  010035                  MOV     #FMTS1B,-(SP)
                014110  012746  000003                  MOV     #3,-(SP)
                014114  010600                          MOV     SP,R0
                014116  104416                          TRAP    C$PNTS
                014120  062706  000010                  ADD     #10,SP
           2897 014124  000410                          BR      2$
           2898
           2899 014126                          1$:     PRINTS  #FMTS1A             ;PRINT '*** RUNNING'
                014126  012746  010015                  MOV     #FMTS1A,-(SP)
                014132  012746  000001                  MOV     #1,-(SP)
                014136  010600                          MOV     SP,R0
                014140  104416                          TRAP    C$PNTS
                014142  062706  000004                  ADD     #4,SP
           2900
           2901 014146                          2$:     PRINTS  #FMTS2,#CART,SERNM2(R4),SERNM1(R4)
                014146  016446  000100                  MOV     SERNM1(R4),-(SP)
                014152  016446  000102                  MOV     SERNM2(R4),-(SP)
                014156  012746  002634                  MOV     #CART,-(SP)
                014162  012746  010070                  MOV     #FMTS2,-(SP)
                014166  012746  000004                  MOV     #4,-(SP)
                014172  010600                          MOV     SP,R0
                014174  104416                          TRAP    C$PNTS
                014176  062706  000012                  ADD     #12,SP
           2902 014202                                  PRINTS  #FMTS2A,SKCNT(R4),SKCNT1(R4),RXFR3(R4),RXFR2(R4),RXFR1(R4)
                014202  016446  000002                  MOV     RXFR1(R4),-(SP)
                014206  016446  000004                  MOV     RXFR2(R4),-(SP)
                014212  016446  000060                  MOV     RXFR3(R4),-(SP)
                014216  016446  000054                  MOV     SKCNT1(R4),-(SP)
                014222  016446  000000                  MOV     SKCNT(R4),-(SP)
                014226  012746  010134                  MOV     #FMTS2A,-(SP)
                014232  012746  000006                  MOV     #6,-(SP)
                014236  010600                          MOV     SP,R0
```

REPORT ROUTINE

```
        014240  104416                  TRAP    C$PNTS
        014242  062706  000016          ADD     #16,SP
   2903 014246                          PRINTS  #FMTS2B,WXFR3(R4),WXFR2(R4),WXFR1(R4)
        014246  016446  000006          MOV     WXFR1(R4),-(SP)
        014252  016446  000010          MOV     WXFR2(R4),-(SP)
        014256  016446  000062          MOV     WXFR3(R4),-(SP)
        014262  012746  010223          MOV     #FMTS2B,-(SP)
        014266  012746  000004          MOV     #4,-(SP)
        014272  010600                  MOV     SP,R0
        014274  104416                  TRAP    C$PNTS
        014276  062706  000012          ADD     #12,SP
   2904 014302                          PRINTS  #FMTS3,DERCNT(R4),SKECNT(R4),TRERR(R4),DATCER(R4)
        014302  016446  000074          MOV     DATCER(R4),-(SP)
        014306  016446  000072          MOV     TRERR(R4),-(SP)
        014312  016446  000016          MOV     SKECNT(R4),-(SP)
        014316  016446  000020          MOV     DERCNT(R4),-(SP)
        014322  012746  010260          MOV     #FMTS3,-(SP)
        014326  012746  000005          MOV     #5,-(SP)
        014332  010600                  MOV     SP,R0
        014334  104416                  TRAP    C$PNTS
        014336  062706  000014          ADD     #14,SP
   2905 014342                          PRINTS  #FMTS3A,ERRCNT(R4),SFTCNT(R4)
        014342  016446  000014          MOV     SFTCNT(R4),-(SP)
        014346  016446  000012          MOV     ERRCNT(R4),-(SP)
        014352  012746  010367          MOV     #FMTS3A,-(SP)
        014356  012746  000003          MOV     #3,-(SP)
        014362  010600                  MOV     SP,R0
        014364  104416                  TRAP    C$PNTS
        014366  062706  000010          ADD     #10,SP
   2906 014372                          PRINTS  #FMTS4,DCRCER(R4),HCRCER(R4),NXMCNT(R4),HNFERR(R4)
        014372  016446  000032          MOV     HNFERR(R4),-(SP)
        014376  016446  000034          MOV     NXMCNT(R4),-(SP)
        014402  016446  000024          MOV     HCRCER(R4),-(SP)
        014406  016446  000022          MOV     DCRCER(R4),-(SP)
        014412  012746  010424          MOV     #FMTS4,-(SP)
        014416  012746  000005          MOV     #5,-(SP)
        014422  010600                  MOV     SP,R0
        014424  104416                  TRAP    C$PNTS
        014426  062706  000014          ADD     #14,SP
   2907 014432                          PRINTS  #FMTS5,DLTCNT(R4),OPICNT(R4)
        014432  016446  000030          MOV     OPICNT(R4),-(SP)
        014436  016446  000026          MOV     DLTCNT(R4),-(SP)
        014442  012746  010517          MOV     #FMTS5,-(SP)
        014446  012746  000003          MOV     #3,-(SP)
        014452  010600                  MOV     SP,R0
        014454  104416                  TRAP    C$PNTS
        014456  062706  000010          ADD     #10,SP
   2908 014462  000207                  RTS     PC
   2909
   2910 014464                          ENDMOD
   2911
   2912                         .SBTTL  PROGRAM MAIN LOOP
   2913 014464                  BGNTST
   2914 014464                  STARS
                                ;;**********************************************************************
   2915                         ;PROGRAM WILL RANDOMLY PICK ONE OF THE DRIVES TO
   2916                         ;PERFORM AN OPERATION.  WE WILL ALWAYS PICK ONE OF FOUR
```

PROGRAM MAIN LOOP

```
2917                                    ;OR EIGHT DRIVES (ONE OR TWO CONTROLLERS) "DRUT" WILL BE
2918                                    ;CHECKED TO SEE IF DRIVE IS ON SYSTEM.  ONCE DRIVE IS PICKED
2919                                    ;THEN A FUNCTION WILL BE SELECTED RANDOMLY FOR THAT
2920                                    ;DRIVE.  FUNCTIONS OF CONTROLLER RESET, GET STATUS, SEEK, READ, WRITE
2921                                    ;WILL BE SELECTED, EACH FUNCTION WILL HAVE ITS OWN ROUTINE
2922                                    ;TO GET PARAMETERS FOR THE DRIVE.
2923 014464                            STARS
                                        ;;*****************************************************************
2924
2925 014464                            MTEST:  SETPRI  #240                    ;PRIORITY TO 5 TO ALLOW CLOCK INTERRUPTS
     014464  012700  000240                     MOV     #240,R0
     014470  104441                             TRAP    C$SPRI
2926                                                                            ;/AND TO INHIBIT DRIVE INTERRUPTS
2927
2928 014472  005737  002274                     TST     WRINIT                 ;HERE AFTER PWR FAIL DURING WRITE
2929 014476  001407                             BEQ     161$                   ;NO
2930 014500  013704  002274                     MOV     WRINIT,R4              ;YES - RESET R4
2931 014504  013701  002276                     MOV     WRPOS,R1               ;AND R1 POINTERS
2932 014510  005237  002452                     INC     STFLG                  ;FAKE OUT THE START FLAG
2933 014514  000410                             BR      16$                    ;AND CONTINUE WRITE INIT CODE
2934 014516  012704  030362            161$:   MOV     #DRBUF,R4              ;GET DRIVE BUFFERS
2935 014522  012701  000001                     MOV     #1,R1                  ;MASK
2936 014526  010437  002274                     MOV     R4,WRINIT              ;COPY THE R4 AND
2937 014532  010137  002276                     MOV     R1,WRPOS               ;POINTERS
2938
2939 014536  130137  002252            16$:    BITB    R1,DRUT                ;DRIVE UNDER TEST
2940 014542  001442                             BEQ     15$                    ;NO
2941
2942 014544  012774  000200  000104            MOV     #200,@DCS(R4)          ;CHECK IF DRIVE THERE
2943 014552  056474  000106  000104            BIS     DRSEL(R4),@DCS(R4)
2944 014560  012700  000000                     MOV     #0.,R0                 ;STALL
2945 014564  005300                    13$:    DEC     R0
2946 014566  001376                             BNE     13$
2947 014570  032774  000001  000104            BIT     #DRDY,@DCS(R4)         ;WAIT FOR DRIVE TO BECOME 'READY
2948 014576  001006                             BNE     14$                    ;AFTER THE HEADS HOME COMMAND
2949
2950 014600  012737  002664  002246            MOV     #DNRDY,WHY             ;MSG. "DRIVE NOT READY"
2951 014606  004537  023450                     JSR     R5,DRDRV
2952 014612  000416                             BR      15$
2953
2954 014614  004537  022604            14$:    JSR     R5,RDBDSC              ;GO GET BAD SECTORS
2955 014620  005064  000056                     CLR     PRFLGS(R4)
2956 014624  005064  000114                     CLR     RSEEK(R4)
2957 014630  005764  000122                     TST     WRIPG(R4)              ;SEE IF WRITE IN PROGRESS FLAG SET
2958 014634  001003                             BNE     99$                    ;JUMP IF SET
2959 014636  005737  002452                     TST     STFLG
2960 014642  001402                             BEQ     15$
2961
2962 014644  004537  024552            99$:    JSR     R5,WRPACK
2963
2964 014650  062704  000126            15$:    ADD     #PRPOS+2,R4            ;NEXT DRIVE
2965 014654  010437  002274                     MOV     R4,WRINIT              ;SAVE CURRENT R4 POINTER
2966 014660  006337  002276                     ASL     WRPOS                  ;AND SHIFT COPY OF R1 POINTER
2967 014664  106301                             ASLB    R1                     ;DONE?
2968 014666  103323                             BCC     16$                    ;NO GO FOR NEXT ONE
2969
2970                                    ;HERE WHEN ALL FINISHED WITH THE WRITE INIT CODE
```

PROGRAM MAIN LOOP

```
2971
2972 014670  005037  002274            12$:    CLR     WRINIT                  ;CLEAR THE WRITE INIT FLAG
2973 014674                                     PRINTF  @FMT14,@MSTART          ;MSG. "TESTING STARTED'
     014674  012746  004360                     MOV     @MSTART,-(SP)
     014700  012746  007463                     MOV     @FMT14,(SP)
     014704  012746  000002                     MOV     #2,(SP)
     014710  010600                             MOV     SP,R0
     014712  104417                             TRAP    C$PNTF
     014714  062706  000006                     ADD     #6,SP
2974 014720                                     SETPRI  #0                      ;PRIORITY TO 0 TO ALLOW BOTH
     014720  012700  000000                     MOV     #0,R0
     014724  104441                             TRAP    C$SPRI
2975                                                                            ;/CLOCK AND DRIVE INTERRUPTS
2976
2977 014726  004537  024454            MAIN:   JSR     R5,RAND                 ;GET A DRIVE?(LUN)
2978 014732  013702  002262                     MOV     LONUM,R2                ;GET THE SELECTED DRIVE (LUN)
2979 014736  043702  002256            PEROTH: BIC     SYSMSK,R2               ;MASK TO DRIVES ON SYSTEM
2980 014742  012701  000001                     MOV     #1,R1                   ;LET'S SEE IF DRIVE IS THERE
2981 014746  005702                    1$:     TST     R2                      ;HAVE WE GOT PROPER MASK YET
2982 014750  001403                             BEQ     2$                      ;YES, GO TO 2$
2983 014752  006301                             ASL     R1                      ;NO, SHIFT FOR NEXT DRIVE
2984 014754  005302                             DEC     R2                      ;DECREMENT DRIVE NUMBER
2985 014756  000773                             BR      1$                      ;GO CHECK NEW DRIVE NUMBER
2986 014760  105737  002252            2$:     TSTB    DRUT                    ;ANY DRIVES ON LINE
2987 014764  001006                             BNE     5$                      ;YES, CHECK
2988
2989 014766                                     ERRSF   170.,NODRIV             ;NO DRIVES
     014766  104454                             TRAP    C$ERSF
     014770  000252                             .WORD   170
     014772  004016                             .WORD   NODRIV
     014774  000000                             .WORD   0
2990
2991 014776  000137  030354                     JMP     ENDOFPROGRAM
2992
2993 015002  130137  002252            5$:     BITB    R1,DRUT                 ;IS THIS DRIVE PRESENT?
2994 015006  001747                             BEQ     MAIN                    ;NO, GO BACK TRY AGAIN
2995 015010  010137  002250                     MOV     R1,TSTDRV               ;COPY UNIT UNDER TEST FOR LATER CHECK
2996
2997                                    ;WE NOW HAVE A DRIVE, CHECK TO SEE IF ITS CONTROLLER
2998                                    ;IS FREE BEFORE WE GO ANY FURTHER
2999
3000 015014  023737  002406  010610            CMP     INTERVAL,TYINT          ;TIME FOR STATISTICAL REPORT?
3001 015022  002403                             BLT     6$                      ;NO, PERFORM FUNCTION
3002 015024  005037  002406                     CLR     INTERVAL                ;CLEAR INTERVAL TO INITIALIZE TIME INTERVAL
3003                                                                            ;/BEFORE THE NEXT STATISTICAL REPORT
3004
3005 015030                                     DORPT                           ;PRINT STATISTICAL REPORT
     015030  104424                             TRAP    C$DRPT
3006
3007 015032  012704  030362            6$:     MOV     #DRBUF,R4               ;GET START OF DRIVE BUFFERS
3008 015036  013702  002262                     MOV     LONUM,R2                ;GET RANDOM DRIVE BACK (LUN)
3009 015042  043702  002256                     BIC     SYSMSK,R2               ;MASK TO SYSTEM SYS
3010 015046  005702                    3$:     TST     R2                      ;DO WE HAVE BUFFER FOR THAT DRIVE
3011 015050  001404                             BEQ     4$                      ;YES, GO CHECK ITS CONTROLLER
3012 015052  062704  000126                     ADD     #PRPOS+2,R4             ;NO, UPDATE FOR NEXT BUFFER
3013 015056  005302                             DEC     R2                      ;DOWN COUNT DRIVE NUMBER (LUN)
3014 015060  000772                             BR      3$                      ;GO BACK AND CHECK FOR FOUND
```

PROGRAM MAIN LOOP

```
3015 015062  032774  000200  000104  4$:     BIT     @BIT7,@DCS(R4)  ;CONTROLLER ASSOCIATED WITH DRIVE
3016 015070  001716                          BEQ     MAIN            ;BUSY
3017 015072  032774  000100  000104          BIT     @BIT6,@DCS(R4)  ;INTERRUPT BEEN SERVICED?
3018 015100  001312                          BNE     MAIN            ;NO   WAIT FOR THE INTERRUPT
3019
3020                                 ;WE CAN NOW PROCEED IN GETTING A FUNCTION AND RELATED DATA
3021                                 ;FOR THE DRIVE RANDOMLY.  R4 HAS DRIVE BUFFER POINTER
3022
3023 015102                          TAGX:
3024 015102  005737  010640                  TST     T.DRP                   ;DROP ON ERROR LIMITS REACHED?
3025 015106  001456                          BEQ     GETFNC                  ;NO
3026 015110  026437  000012  010600          CMP     ERRCNT(R4),ERLMT        ;HARD REACHED?
3027 015116  103404                          BLO     9$
3028 015120  012737  003322  002246          MOV     @ERLMTM,WHY
3029 015126  000442                          BR      11$
3030 015130  026437  000014  010644  9$:     CMP     SFTCNT(R4),SFLMT        ;SOFT REACHED?
3031 015136  103404                          BLO     10$
3032 015140  012737  003365  002246          MOV     @SFEMSG,WHY
3033 015146  000432                          BR      11$
3034 015150  026437  000074  010670  10$:    CMP     DATCER(R4),T.DCD
3035 015156  103404                          BLO     110$
3036 015160  012737  003407  002246          MOV     @DCDMSG,WHY
3037 015166  000422                          BR      11$
3038 015170  016401  000016          110$:   MOV     SKECNT(R4),R1
3039 015174  066401  000072                  ADD     TRERR(R4),R1
3040 015200  020137  010602                  CMP     R1,SELMT
3041 015204  103404                          BLO     7$
3042 015206  012737  003344  002246          MOV     @SERLMT,WHY
3043 015214  000407                          BR      11$
3044 015216  026437  000020  010650  7$:     CMP     DERCNT(R4),DRLMT        ;DRIVE ERROR REACHED?
3045 015224  103407                          BLO     GETFNC                  ;NO - TIME TO DO SOMETHING
3046 015226  012737  003432  002246          MOV     @DERMSG,WHY
3047
3048 015234  004537  023450          11$:    JSR     R5,DRDRV                ;DROP THIS DRIVE!!!
3049 015240  000137  014726                  JMP     MAIN                    ;GO GET ANOTHER
3050
3051                                 ;HERE TO GET A 'STRING' FUNCTION - LIST OF COMMANDS TO ISSUE
3052
3053 015244                          GETFNC:
3054 015244  005737  010646          6$:     TST     T.STA           ;DO WE WISH TO DROP ON OPR LIMITS
3055 015250  001422                          BEQ     98$             ;NO
3056
3057 015252  026437  000000  010606          CMP     SKCNT(R4),SKLMT ;PAST THE SEEK LIMIT??
3058 015260  103416                          BLO     98$             ;NO, THEN GO TEST
3059 015262  016400  000060                  MOV     RXFR3(R4),R0    ;GET READ COUNT
3060 015266  066400  000062                  ADD     WXFR3(R4),R0    ;ADD IN WRITE COUNT
3061 015272  020037  010604                  CMP     R0,DALMT        ;LIMIT  REACHED??
3062 015276  103407                          BLO     98$             ;NO, THEN GO TEST
3063 015300  012737  003611  002246          MOV     @SOPLMT,WHY
3064 015306  004537  023450                  JSR     R5,DRDRV        ;DROP THE DRIVE
3065 015312  000137  014726                  JMP     MAIN            ;GO FOR ANOTHER DRIVE
3066
3067 015316  004537  024454          98$:    JSR     R5,RAND         ;GET A RANDOM FUNCTION INDEX NUMBER
3068                                                                 ;0 & 7 ARE NOT LEGIT
3069 015322  013702  002262                  MOV     LONUM,R2        ;GET IT
3070 015326  042702  177770                  BIC     #177770,R2      ;MASK TO 0-7
3071 015332  001001                          BNE     6$              ;IF 0, MAKE 1
```

```
3072 015334  005202                        INC     R2
3073 015336  022702  000007          6$:   CMP     #7,R2           ;IS IT 7?
3074 015342  001001                        BNE     5$              ;IF 7, MAKE 6
3075 015344  005302                        DEC     R2
3076 015346  006302                  5$:   ASL     R2              ;SHIFT LEFT (X2)
3077 015350  000172  022566                JMP     @LIST(R2)       ;GO TO FUNCTION ROUTINE
3078
3079 015354                        STARS
                                   ;;*****************************************************************
3080                               ;SKWRT -- ISSUE:
3081                               ;       SEEK TO A CYLINDER
3082                               ;       WRITE DATA
3083                               ;       WRITE CHECK
3084 015354                        STARS
                                   ;;*****************************************************************
3085
3086 015354  004537  015716        SKWRT: JSR     R5,SKFNC        ;RANDOM SEEK LOAD
3087 015360  004537  015470               JSR     R5,OPROK        ;WAIT TILL DONE
3088 015364  004537  016360               JSR     R5,WRTFNC       ;WRITE DATA LOAD
3089 015370  004537  015470               JSR     R5,OPROK
3090 015374  004537  015654               JSR     R5,WRTCKF       ;WRITE CHECK LOAD
3091 015400  004537  015470               JSR     R5,OPROK
3092 015404  000137  014726               JMP     MAIN            ;GET NEXT COMMAND
3093
3094 015410                        STARS
                                   ;;*****************************************************************
3095                               ;SKRD -- ISSUE:
3096                               ;       RANDOM SEEK TO A CYLINDER
3097                               ;       READ DATA
3098 015410                        STARS
                                   ;;*****************************************************************
3099
3100 015410  004537  015716        SKRD:  JSR     R5,SKFNC        ;LOAD SEEK
3101 015414  004537  015470               JSR     R5,OPROK
3102 015420  004537  016430               JSR     R5,RDDFNC       ;LOAD READ DATA CMD
3103 015424  004537  015470               JSR     R5,OPROK
3104 015430  000137  014726               JMP     MAIN            ;GET THE NEXT COMMAND
3105
3106 015434                        STARS
                                   ;;*****************************************************************
3107                               ;SKRDRD -- ROUTINE TO DO:
3108                               ;       SEEK TO A CYLINDER
3109                               ;       READ (AND COMPARE DATA)
3110                               ;       READ (AGAIN)
3111 015434                        STARS
                                   ;;*****************************************************************
3112
3113 015434  004537  015716        SKRDRD: JSR    R5,SKFNC        ;LOAD SEEK
3114 015440  004537  015470               JSR     R5,OPROK
3115 015444  004537  016430               JSR     R5,RDDFNC       ;LOAD READ
3116 015450  004537  015470               JSR     R5,OPROK
3117 015454  004537  016430               JSR     R5,RDDFNC       ;LOAD READ
3118 015460  004537  015470               JSR     R5,OPROK
3119 015464  000137  014726               JMP     MAIN            ;EXIT
3120
3121 015470                        STARS
                                   ;;*****************************************************************
```

PROGRAM MAIN LOOP

```
3122                                    ;OPROK    ROUTINE TO ISSUE THE FUNCTION AND WAIT FOR 'READY'...IF AN
3123                                    ;       ERROR RETRY IS NEEDED   THEN ISSUE THE FUNCTION AGAIN.
3124 015470                             STARS
                                        ;;*****************************************************************
3125
3126 015470  004537  016524  000000    OPROK:  JSR     R5,LDFUNC       ;ISSUE THE FUNCTION
3127 015474  004537  024270  000000            JSR     R5,WTRDY        ;WAIT TILL READY
3128 015500  133737  002250  002252            BITB    TSTDRV,DRUT     ;DRIVE STILL AVAILABLE?
3129 015506  001003                            BNE     1$              ;YUP - CONTINUE
3130 015510  005726                            TST     (SP)+           ;NO - FIX THE STACK
3131 015512  000137  014726  000000            JMP     MAIN            ;BACK TO THE MAIN LOOP   FORCED EXIT FROM
3132                                                                    ;THE STRING FUNCTION
3133 015516  005764  000036  000000    1$:     TST     RETRY(R4)       ;NEED TO RETRY FUNCTION?
3134 015522  001403                            BEQ     3$              ;NO
3135 015524  004537  016472  000000    2$:     JSR     R5,ISSUE        ;YES - ISSUE THE FUNCTION AGAIN
3136 015530  000757                            BR      OPROK           ;AND DO IT
3137 015532  005764  000114  000000    3$:     TST     RSEEK(R4)       ;SEEK RETRY?
3138 015536  001403                            BEQ     4$              ;NO - EXIT NOW
3139 015540  004537  015716  000000            JSR     R5,SKFNC        ;DO A SEEK AGAIN
3140 015544  000751                            BR      OPROK           ;ISSUE & EXECUTE THE SEEK
3141 015546  000205                    4$:     RTS     R5              ;EXIT
3142
3143 015550                             STARS
                                        ;;*****************************************************************
3144                                    ;SKRM -- ISSUE:
3145                                    ;       RANDOM SEEK
3146                                    ;       READ HEADERS
3147                                    ;       READ DATA W/NO HDR CMP
3148                                    ;       GET STATUS
3149 015550                             STARS
                                        ;;*****************************************************************
3150
3151 015550  004537  015716  000000    SKRM:   JSR     R5,SKFNC        ;LOAD SEEK
3152 015554  004537  016524  000000            JSR     R5,LDFUNC       ;ISSUE
3153 015560  004537  024270  000000            JSR     R5,WTRDY
3154 015564  004537  016346  000000            JSR     R5,RDHFNC       ;LOAD READ HDRS
3155 015570  004537  016524  000000            JSR     R5,LDFUNC       ;ISSUE
3156 015574  004537  024270  000000            JSR     R5,WTRDY
3157 015600  004537  015634  000000            JSR     R5,RDNHC        ;LOAD READ W/NO HDRS
3158 015604  004537  016524  000000            JSR     R5,LDFUNC       ;ISSUE
3159 015610  004537  024270  000000            JSR     R5,WTRDY
3160 015614  004537  015676  000000            JSR     R5,GSTFNC       ;LOAD GET STATUS
3161 015620  004537  016524  000000            JSR     R5,LDFUNC       ;ISSUE
3162 015624  004537  024270  000000            JSR     R5,WTRDY
3163 015630  000137  014726  000000            JMP     MAIN            ;GET THE NEXT COMMAND
3164
3165 015634                             STARS
                                        ;;*****************************************************************
3166                                    ;READ DATA W/NO HDR COMPARE
3167 015634                             STARS
                                        ;;*****************************************************************
3168
3169 015634  012764  177600  000042    RDNHC:  MOV     #-128.,BMP(R4)  ;SET FOR A 1 SECTOR READ
3170 015642  012764  000016  000044            MOV     #16,FUNC(R4)    ;LOAD THE COMMAND
3171 015650  000137  016472  000000            JMP     ISSUE           ;PROCESS IT
3172
3173 015654                             STARS
```

PROGRAM MAIN LOOP

```
                                        ;;*****************************************************************
                                        ;WRTCKF - WRITE CHECK FUNCTION
                                        STARS
                                        ;;*****************************************************************

WRTCKF: TST     T.ROF           ;READ ONLY SET?
        BEQ     1$              ;NO - DO THE WRITE-CHECK FUNCTION
        RTS     R5              ;YES - EXIT NOW

1$:     MOV     #WRCHK,FUNC(R4) ;SAVE CMD
        JMP     ISSUE           ;PROCESS IT

                                .SBTTL  ROUTINES TO SETUP AND ISSUE GET STATUS & SEEK
                                STARS
                                ;;*****************************************************************
                                ;GET STATUS FUNCTION
                                STARS
                                ;;*****************************************************************

GSTFNC: MOV     #GSTAT,FUNC(R4) ;LOAD GET STATUS
        MOV     #GSBIT,BDA(R4)  ;SET GSBIT IN COMMAND WORD
        JMP     ISSUE           ;GO ISSUE FUNCTION

                                STARS
                                ;;*****************************************************************
                                ;SEEK FUNCTION
                                STARS
                                ;;*****************************************************************

                                ;WE WILL CALL "RAND" FOR A NEW DISK ADDRESS TO SEEK
                                ;TO.  ANY TRACK BUT LAST IS LEGAL.  WE WILL ALSO INCREMENT
                                ;ITS SEEK COUNT

SKFNC:  TST     RSEEK(R4)       ;TRYING TO RECOVER
        BNE     10$             ;YES - DO IT
        TST     RETRY(R4)       ;RECOVERY FROM A 'DRIVE' ERROR?
        BEQ     98$             ;NO - NORMAL SEEK REQUIRED
10$:    MOV     LSTHDR(R4),R1   ;YES SET UP FOR RESEEK
        MOV     PRPOS(R4),R2    ;TO CYLINDER
        BIC     #100,R1         ;HEAD SET IN LATER
        BIC     #100,R2         ;
        BR      4$              ;SKIP RANDOM PART
98$:    JSR     R5,RAND         ;GET A RANDOM NUMBER
        MOV     LONUM,R2        ;GET THE RANDOM NUMBER
        BIC     SMSK,R2         ;LEAVE CYL AND HEAD
        CMP     R2,PRPOS(R4)    ;ON THAT TRACK ALREADY
        BEQ     98$             ;YES - RESELECT

980$:   CMP     #1,TDR(R4)      ;THIS DRIVE AN RL01?
        BNE     981$            ;NO - MUST BE AN RL02
        BIC     #BIT15,R2       ;KILL UPPER BIT OF CYL ADDRESS
        CMP     #077700,R2      ;POINTING TO THE BAD SEC FILE?
        BNE     96$             ;NO - PROCEED
        BR      982$            ;YUP - CORRECT THE POSITION
981$:   CMP     #177700,R2      ;RL02 BAD SECTOR FILE?
        BNE     96$             ;NO - PROCEED
982$:   NOP                     ;TRAP
```

ROUTINES TO SETUP AND ISSUE GET STATUS & SEEK

```
3225 016032  042702  000100                BIC     #HEAD,R2        ;POINT TO HEAD 0 LAST TRACK
3226
3227 016036  010237  002342          96$:  MOV     R2,CHKSEC       ;SAVE THE ADDRESS FOR THE BAD SEC FILE CHECK
3228 016042  004537  027224                JSR     R5,CKBDTK       ;SEE IF THIS ADDR IN BAD SECTOR FILE
3229 016046  005737  002340                TST     HDRFND          ;WAS IT?
3230 016052  001340                       BNE     98$             ;YES - RESELECT THE ADDRESS
3231 016054  005003                 90$:  CLR     R3
3232 016056  010200                       MOV     R2,R0           ;COPY ADDRESS - NO SECTOR YET
3233 016060  042700  177677                BIC     #1776/7,R0      ;LEAVE ONLY HEAD
3234 016064  023737  010626  010630        CMP     T.MXC,T.MNC     ;MIN AND MAX CYLINDERS THE SAME
3235 016072  001011                       BNE     95$             ;NO, BRANCH AND STAY IN LIMITS
3236 016074  013702  010626                MOV     T.MXC,R2        ;MAKE CYLINDER MAX/MIN
3237 016100  022764  000001  000120        CMP     #1,TDR(R4)      ;DRIVE = RL01?
3238 016106  001031                       BNE     92$             ;NO
3239 016110  042702  100000                BIC     #BIT15,R2       ;FORCE CYL TO PROPER LIMIT
3240 016114  000426                       BR      92$             ;GO CALCULATE DIFF AND SEEK
3241 016116  042702  000100          95$:  BIC     #HEAD,R2        ;STRIP OUT H.S. BIT
3242 016122  023702  010626          94$:  CMP     T.MXC,R2        ;IS ADDRESS LESS/EQUAL THAN MAX
3243 016126  103010                       BHIS    93$             ;YES, CHECK LOW END
3244 016130  005203                       INC     R3              ;BUMP A TALLY COUNTER
3245 016132  020327  000012                CMP     R3,#10.         ;IF CAN'T FIND ADDRESS IN 10 TIMES THEN RESELECT
3246 016136  001706                       BEQ     98$             ;RESELECT
3247 016140  006202                       ASR     R2              ;HALF IT AND CHECK AGAIN
3248 016142  062702  000200          91$:  ADD     #BIT7,R2        ;JUST TO MAKE NON ZERO
3249 016146  000763                       BR      95$             ;GO BACK AND CHECK AGAIN
3250 016150  023702  010630          93$:  CMP     T.MNC,R2        ;IS MIN GREATER/EQUAL THAN ADDRESS
3251 016154  101406                       BLOS    92$             ;YES, CALCULATE DIFF AND SEEK
3252 016156  005203                       INC     R3
3253 016160  020327  000012                CMP     R3,#10.         ;TIME TO RESELECT?
3254 016164  001673                       BEQ     98$             ;YUP - DO IT NOW
3255 016166  006302                       ASL     R2              ;NO, DOUBLE IT
3256 016170  000764                       BR      91$             ;GO CHECK MAX/MIN AGAIN
3257 016172  016401  000124          92$:  MOV     PRPOS(R4),R1    ;GET PRESENT DISK POSITION
3258 016176  042701  000177                BIC     #177,R1
3259 016202  022764  000001  000120        CMP     #1,TDR(R4)      ;RL01=1
3260 016210  001002                       BNE     25$             ;BRANCH...MUST BE RL02
3261 016212  042702  100000                BIC     #BIT15,R2       ;CLEAR THE HIGH BIT FOR RL02 CYL #
3262 016216  016464  000124  000050  25$:  MOV     PRPOS(R4),LSTHDR(R4)
3263 016224  010264  000124                MOV     R2,PRPOS(R4)    ;NEW HEADER AFTER SEEK
3264 016230  050064  000124                BIS     R0,PRPOS(R4)    ;SET IN RANDOM HEAD GOTTEN
3265 016234  023737  010622  010624        CMP     T.MXH,T.MNH     ;MIN AND MAX HEAD SELECT THE SAME
3266 016242  001012                       BNE     4$              ;NO, THEN WE CAN USE BOTH SURFACES
3267 016244  005737  010622                TST     T.MXH           ;WHICH IS OUR SURFACE FOR USE
3268 016250  001004                       BNE     97$             ;TOP SURFACE BRANCH
3269 016252  042764  000100  000124        BIC     #HEAD,PRPOS(R4) ;LOWER SURFACE ONLY
3270 016260  000403                       BR      4$
3271 016262  052764  000100  000124  97$:  BIS     #HEAD,PRPOS(R4) ;TOP SURFACE ONLY
3272
3273 016270                             STARS
                                        ;;*****************************************************************
3274                                    ;CALCULATE THE DIFFERENCE WORD AND STORE IT IN BDA
3275 016270                             STARS
                                        ;;*****************************************************************
3276
3277 016270  160102                4$:  SUB     R1,R2           ;SUBTRACT PRESENT FROM NEXT
3278 016272  103002                       BCC     1$              ;IF POSITIVE RESULT GO TO 1$
3279 016274  005402                       NEG     R2              ;NEG RESULT, NEGATE IT
```

ROUTINES TO SETUP AND ISSUE GET STATUS & SEEK

```
3280 016276  000402                          BR      2$              ;GO SET DIRECTION OUT
3281 016300  052702  000004          1$:     BIS     #SIGN,R2        ;DIRECTION OUT, MARKER
3282 016304  052702  000001          2$:     BIS     #MK,R2          ;MARKER BIT
3283 016310  032764  000100  000124          BIT     #HEAD,PRPOS(R4) ;WHICH SURFACE SELECTED?
3284 016316  001402                          BEQ     3$              ;TOP, THEN 3$
3285 016320  052702  000020                  BIS     #SKMS,R2        ;BOTTOM SET HEAD BIT
3286 016324  010264  000040          3$:     MOV     R2,BDA(R4)      ;MOVE DIFFERENCE WORD TO DA
3287 016330  010264  000066                  MOV     R2,DIFWD(R4)    ;LOAD DIFFERENCE WORD
3288 016334  012764  000006  000044          MOV     #SEEK,FUNC(R4)  ;LOAD SEEK
3289 016342  000137  016472                  JMP     ISSUE
3290
3291                                 .SBTTL  ROUTINE TO LOAD READ HEADER AND ISSUE IT
3292
3293 016346  012764  000010  000044  RDHFNC: MOV     #RDHDR,FUNC(R4) ;LOAD READ HEADER
3294 016354  000137  016472                  JMP     ISSUE
3295
3296                                 .SBTTL  ROUTINE TO LOAD WRITE DATA COMMAND
3297
3298 016360  005737  010652          WRTFNC: TST     T.ROF           ;READ ONLY
3299 016364  001021                          BNE     RDDFNC          ;YES
3300 016366  004537  025714                  JSR     R5,GWCDA        ;GET WORD COUNT,DA
3301 016372  005737  010612                  TST     CMRD            ;COMPARE DATA ON A READ?
3302 016376  001404                          BEQ     1$              ;NO - SO DON'T GEN A WRITE BUFFER
3303 016400  005237  002306                  INC     REGEN           ;YES - SET THE GENERATE DATA FLAG
3304                                 ;
3305                                 ;WE NOW HAVE SECTOR AND WORD COUNT, LET'S WRITE BUFFER IN MEMORY
3306                                 ;TO WRITE OUT TO DISK
3307                                 ;FORMAT:        WORD 1 - # OF WORDS IN SECTOR
3308                                 ;       :       WORD 2 - ADDRESS OF PATTERN WRITTEN ON SECTOR
3309                                 ;       :       WORD 3 - 127  DATA PATTERN
3310                                 ;
3311
3312 016404  004537  022320                  JSR     R5,WRBUF        ;WRITE BUFFER INTO MEMORY
3313 016410  012764  000012  000044  1$:     MOV     #WRITE,FUNC(R4) ;LOAD WRITE
3314 016416  012764  000001  000122          MOV     #1,WRIPG(R4)    ;SET THE WRITE IN PROGRESS FLAG
3315 016424  000137  016472                  JMP     ISSUE           ;GO ISSUE FUNCTION
3316
3317                                 .SBTTL  ROUTINE TO LOAD READ DATA COMMAND
3318
3319                                 ;THIS ROUTINE WILL FIRST CLEAR OUT THE BUFFER AREA,
3320                                 ;SELECT A RANDOM NUMBER OF WORDS TO READ AND A
3321                                 ;RANDOM SECTOR ON THE PRESENT CYLINDER TO READ FROM
3322
3323 016430  004537  025714          RDDFNC: JSR     R5,GWCDA        ;GET WORD COUNT, DA
3324 016434  005737  010612                  TST     CMRD            ;GOING TO COMPARE DATA AFTER READING?
3325 016440  001407                          BEQ     2$              ;NO - SO SKIP THE CLEAR BUFFER CODE
3326 016442  016402  000042                  MOV     BMP(R4),R2      ;CLEAR OUT BUFFER AREA
3327 016446  017401  000110                  MOV     @BBA(R4),R1     ;SO WE KNOW READ
3328 016452  005021                  1$:     CLR     (R1)+           ;WORKED!!
3329 016454  005202                          INC     R2
3330 016456  001375                          BNE     1$
3331 016460  012764  000014  000044  2$:     MOV     #READ,FUNC(R4)  ;LOAD READ
3332 016466  000137  016472                  JMP     ISSUE
3333
3334                                 .SBTTL  SETUP CONTROLLER AND DRIVE INFO FOR INTERRUPT PROCESSING
3335
3336                                 ;WE COME HERE BEFORE ISSUING ANY FUNCTION SO THAT ON INTERRUPT
```

SETUP CONTROLLER AND DRIVE INFO FOR INTERRUPT PROCESSING

```
3337                                    ;WE CAN PROPERLY PROCESS THE INTERRUPT.  WE WILL CHECK WHICH
3338                                    ;CONTROLLER WE ARE WORKING WITH AND STORE OFF THE DRIVE BUFFER
3339                                    ;POINTER IN ITS "LSTDR"
3340                                    ;
3341
3342 016472  026437  000104  002320  ISSUE:  CMP     DCS(R4),CNTLR1  ;DRIVE ON CONTROLLER 1?
3343 016500  001003                           BNE     1$              ;NO, ASSUME ON CONTROLLER 2
3344 016502  010437  002324                   MOV     R4,LSTDR1       ;PUT BUFFER POINTER IN 1
3345 016506  000402                           BR      2$              ;SKIP OVER NEXT INSTRUCTION
3346 016510  010437  002326           1$:     MOV     R4,LSTDR2       ;PUT BUFFER POINTER IN 2
3347 016514  052764  000100  000044  2$:     BIS     #INTEN,FUNC(R4) ;ALLOW INTERRUPTS
3348 016522  000205                           RTS     R5              ;EXIT
3349
3350                                    .SBTTL  ROUTINE TO LOAD FUNCTION
3351 016524                             STARS
                                        ;;*****************************************************************
3352                                    ;CALL   JSR     R5,LDFUNC
3353                                    ;ALL INFORMATION MUST BE SET UP IN DRIVE BUFFER
3354                                    ;R4 HAS POINTER TO BUFFER
3355 016524                             STARS
                                        ;;*****************************************************************
3356
3357 016524  016403  000104             LDFUNC: MOV     DCS(R4),R3      ;GET CSR FOR DRIVE
3358 016530  032713  000200                     BIT     #BIT7,(R3)      ;CAN WE ISSUE COMMAND?
3359 016534  001004                             BNE     1$              ;YES, GO ISSUE COMMAND
3360
3361 016536                                     ERRSF   200.,PRGER      ;THIS ERROR SHOULD NEVER PRINT
     016536  104454                             TRAP    C$ERSF
     016540  000310                             .WORD   200
     016542  002732                             .WORD   PRGER
     016544  000000                             .WORD   0
3362
3363 016546  017463  000110  000002  1$:     MOV     @BBA(R4),BA(R3) ;LOAD BUS ADDRESS REGISTER
3364 016554  016463  000040  000004          MOV     BDA(R4),DA(R3)  ;LOAD DISK ADDRESS REGISTER
3365 016562  016463  000042  000006          MOV     BMP(R4),MP(R3)  ;LOAD MULTI-PURPOSE REGISTER
3366 016570  016464  000044  000046          MOV     FUNC(R4),BCSADR(R4)     ;GET FUNCTION
3367 016576  056464  000106  000046          BIS     DRSEL(R4),BCSADR(R4)    ;SET DRIVE SELECT BITS
3368 016604  052764  000201  000046          BIS     #CRDY!DRDY,BCSADR(R4)   ;SET CRDY:DRDY IN IMAGE
3369 016612  042764  002000  000046          BIC     #OPI,BCSADR(R4)         ;WE'RE CLEAR BIT 10 FOR DRIVE 7-4 (OKAY?)
3370 016620  016463  000046  000000          MOV     BCSADR(R4),CS(R3)       ;LOAD CSR
3371 016626  042763  000200  000000          BIC     #CRDY,CS(R3)            ;ISSUE FUNCTION
3372 016634  000205                          RTS     R5                      ;EXIT
3373
3374                                    .SBTTL  INTERRUPT SERVICE ROUTINES
3375
3376                                    ;CLOCK INTERRUPT HANDLER
3377                                    ;UPDATES TIME EVERY 1/60 SECOND (60 HZ) OR EVERY 1/50 SECOND (50 HZ)
3378 016636                             BGNSRV  UPDATE
3379 016636  010446                             MOV     R4,-(SP)        ;SAVE R4
3380                                    ;CLEAR CLOCK INTERRUPT ENABLE TO INHIBIT CLOCK INTERRUPTS DURING UPDATING
3381                                    ;OF TIME FIELDS
3382 016640  022737  000001  002314             CMP     #1,CLKTYP       ;P-CLOCK?
3383 016646  001004                             BNE     1$              ;BRANCH IF NOT P-CLOCK
3384 016650  042737  000100  172540             BIC     #100,@#172540   ;DISABLE P-CLOCK INTERRUPT FACILITY
3385                                    ;UPDATE TIME FIELDS
3386 016656  000403                             BR      2$
3387 016660  042737  000100  177546  1$:     BIC     #100,@#177546   ;DISABLE L-CLOCK INTERRUPT FACILITY
```

```
INTERRUPT SERVICE ROUTINES

3388 016666  012704  002410          2$:     MOV     #TICK,R4        ;INITIALIZE TICK ADDRESS
3389 016672  005214                          INC     (R4)            ;INCREMENT TICK TIME FIELD
3390 016674  023727  002312  000002          CMP     CLKFRQ,#2       ;50 HZ CLOCK?
3391 016702  001005                          BNE     3$              ;NO--BRANCH FOR SERVICING 60 HZ CLOCK
3392 016704  021427  000062                  CMP     (R4),#50        ;((R4))=50?
3393 016710  001026                          BNE     EXIT2           ;IF NOT,UPDATING IS COMPLETE
3394 016712  005014                          CLR     (R4)            ;ELSE,((R4))=0 (RESET COUNT)
3395 016714  000404                          BR      4$              ;BRANCH TO UPDATE "SECOND" TIME FIELD
3396 016716  021427  000074          3$:     CMP     (R4),#60.       ;((R4))=60?
3397 016722  001021                          BNE     EXIT2           ;IF NOT,UPDATING IS COMPLETE
3398 016724  005014                          CLR     (R4)            ;ELSE,((R4))=0 (RESET COUNT)
3399 016726  005724                  4$:     TST     (R4)+           ;(R4)=(R4)+2 (GO TO NEXT TIME FIELD)
3400 016730  005214                          INC     (R4)            ;INCREMENT "SECOND" TIME FIELD
3401 016732  021427  000074                  CMP     (R4),#60.       ;((R4))=60?
3402 016736  001013                          BNE     EXIT2           ;IF NOT,UPDATING IS COMPLETE
3403 016740  005237  002406                  INC     INTERVAL        ;INCREMENT INTERVAL TIME FIELD (STORES
3404                                                                 ;/RUNNING TIME BETWEEN STATISTICAL REPORTS)
3405 016744  005014                          CLR     (R4)            ;ELSE,((R4))=0 (RESET COUNT)
3406 016746  005724                          TST     (R4)+           ;ACCESS "MINUTE" TIME FIELD
3407 016750  005214                          INC     (R4)            ;INCREMENT "MINUTE" TIME FIELD
3408 016752  021427  000074                  CMP     (R4),#60.       ;((R4))=60?
3409 016756  001003                          BNE     EXIT2           ;IF NOT,UPDATING IS COMPLETE
3410 016760  005014                          CLR     (R4)            ;ELSE,((R4))=0 (RESET COUNT)
3411 016762  005724                          TST     (R4)+           ;ACCESS "HOUR" TIME FIELD
3412 016764  005214                          INC     (R4)            ;INCREMENT "HOUR" TIME FIELD
3413 016766  005337  002510          EXIT2:  DEC     CLKBFR          ;COUNT CLOCK TICKS
3414 016772  003005                          BGT     5$              ;TIME NOT EXPIRED
3415 016774  005237  002512                  INC     CLKACC          ;BUMP ELAPSED TIME
3416 017000  013737  002506  002510          MOV     CLKCNT,CLKBFR   ;RE-INITIALIZE TIME INCREMENT
3417                         ;RE-ENABLE CLOCK INTERRUPT FACILITY
3418 017006  022737  000001  002314  5$:     CMP     #1,CLKTYP       ;P-CLOCK?
3419 017014  001004                          BNE     6$              ;BRANCH IF NOT P-CLOCK
3420 017016  052737  000100  172540          BIS     #100,@#172540   ;SET P-CLOCK INTERRUPT ENABLE BIT
3421 017024  000403                          BR      7$              ;EXIT
3422 017026  052737  000100  177546  6$:     BIS     #100,@#177546   ;SET L-CLOCK INTERRUPT ENABLE BIT
3423 017034  012604                  7$:     MOV     (SP)+,R4        ;RESTORE R4
3424 017036                          ENDSRV
     017036                          L10025:
     017036  000002                          RTI
3425
3426                                 ;L-CLOCK "TICK" CHECK ROUTINE FOR LSI-11
3427 017040                          BGNSRV  CLKTIK
3428
3429 017040  005237  002514                  INC     CLKFLD          ;INCREMENT CLOCK FIELD TO INDICATE THAT
3430                                                                 ;/CLOCK IS "TICKING"
3431
3432 017044                          ENDSRV
     017044                          L10026:
     017044  000002                          RTI
3433
3434 017046                          BGNSRV  INTR1
3435
3436                                 ;ON INTERRUPT WE CHECK FOR ERRORS FIRST. IF NO ERRORS WE
3437                                 ;CHECK FUNCTION PERFORMED. WE ACT ACCORDING IF FUNCTION IS:
3438                                 ; 1-    WRITE CHECK - NOTHING IF NO ERROR
3439                                 ; 2-    GET STATUS - READ AND CHECK DRIVE STATUS
3440                                 ; 3-    SEEK - NOTHING RTI;  SET RD HDR AS NEXT COMMAND
```

INTERRUPT SERVICE ROUTINES

```
3441                                    ; 4     RDHDR   COMPARE HEADER TO PRESENT POSITION
3442                                    ; 5-    WRITE   UPDATE XFER COUNT, EXIT
3443                                    ; 6     READ   COMPARE DATA IF REQUESTED, UPDATE XFER COUNT, EXIT
3444                                    ; 7-    READ W/NO HDR COMPARE - UPDATE XFER COUNT, EXIT
3445                                    ;
3446                                    ;ALL SUCCESSFUL EXITS FROM INTERRUPT ROUTINE TEST RETRY
3447                                    ;LIMIT IF RETRY IS LESS THEN LIMIT THEN LOG SOFT ERROR, CLEAR RETRY
3448                                    ;IF RETRY = 0, THEN NOTHING
3449                                    ;
3450                                    ;ON ERRORS - IF DRIVE ERROR - UNDER NON-INTERRUPT
3451                                    ;       DO:     GET STATUS - INVESTIGATE ERROR TYPE
3452                                    ;
3453                                    ;       DO:     DRIVE RESET - IF ERROR OCCURS AGAIN   FATAL ERROR
3454                                    ;                               IF NO ERROR, EXIT
3455                                    ;       DRIVE ERROR IS LOGGED UNDER ALL CIRCUMSTANCES
3456                                    ;
3457                                    ;
3458                                    ;       IF DCRC, HCRC, HNF CHECK BAD SECTOR LIST, IF IN LIST
3459                                    ;       IGNORE ERROR EXIT AS NORMAL, IF NOT IN LIST
3460                                    ;       INCREMENT RETRY; IF RETRY LIMIT EXCEEDED
3461                                    ;       LOG HARD ERROR, ELSE RETRY FUNCTION
3462                                    ;
3463                                    ;       IF OPI,NXM INCREMENT RETRY CHECK RETRY LIMIT
3464                                    ;       IF RETRY EXCEEDED LOG HARD ERROR EXIT
3465                                    ;       IF RETRY NOT EXCEEDED RETRY FUNCTION
3466                                    ;
3467                                    ;
3468
3469 017046  010446                    INTR1:  MOV     R4, (SP)        ;SAVE PRESENT R4 VALUE
3470 017050  013704  002324                    MOV     LSTDR1,R4       ;GET THE DRIVE BUFFER OF INTERRUPTING DRIVE
3471 017054  000403                            BR      SAVE            ;GO SAVE R0-R3
3472 017056  010446                    INTR2:  MOV     R4,-(SP)        ;SAVE PRESENT R4 VALUE
3473 017060  013704  002326                    MOV     LSTDR2,R4       ;GET THE DRIVE BUFFER OF INTERRUPTING DRIVE
3474 017064  013746  002420            SAVE:   MOV     E.CS,-(SP)
3475 017070  013746  002422                    MOV     E.BA,-(SP)
3476 017074  013746  002424                    MOV     E.DA,-(SP)
3477 017100  013746  002426                    MOV     E.MP,-(SP)
3478 017104  013746  002430                    MOV     E.MP1,-(SP)
3479 017110  013746  002432                    MOV     E.MP2,-(SP)
3480 017114  013746  002342                    MOV     CHKSEC,-(SP)
3481 017120  013746  002340                    MOV     HDRFND,-(SP)
3482 017124  013746  002350                    MOV     TEMP1,-(SP)
3483 017130  013746  002246                    MOV     WHY,-(SP)
3484 017134  013746  002474                    MOV     OPCALL,-(SP)
3485 017140  013746  002476                    MOV     INCALL,-(SP)
3486 017144  010346                          MOV     R3,-(SP)        ;SAVE R3
3487 017146  010246                          MOV     R2,-(SP)        ;R2
3488 017150  010146                          MOV     R1,-(SP)        ;R1
3489 017152  010046                          MOV     R0,-(SP)        ;R0
3490 017154  005064  000122                  CLR     WRIPG(R4)       ;CLEAR THE WRITE IN PROGRESS FLAG
3491 017160  016403  000104                  MOV     DCS(R4),R3      ;GET CSR FOR INTERRUPT
3492 017164  016337  000000  002420          MOV     CS(R3),E.CS     ;SAVE ALL REGISTERS NOW!!
3493 017172  016337  000002  002422          MOV     BA(R3),E.BA
3494 017200  016337  000004  002424          MOV     DA(R3),E.DA
3495 017206  016337  000006  002426          MOV     MP(R3),E.MP
3496 017214  016337  000006  002430          MOV     MP(R3),E.MP1
3497 017222  016337  000006  002432          MOV     MP(R3),E.MP2
```

```
INTERRUPT SERVICE ROUTINES

 3498 017230  005737  002420                    TST     E.CS            ;ANY ERRORS?
 3499 017234  100402                            BMI     1$              ;YES, GO SOLVE ERROR MYSTERY
 3500 017236  000137  020362                    JMP     CHKFNC          ;NO, GO SEE IF WE HAVE TO DO ANYTHING
 3501
 3502                                 .SBTTL  CONTROLLER ERROR CHECK ROUTINE
 3503
 3504                                 ;WE HAVE SOME SORT OF ERROR LET'S FIND OUT WHICH ONE
 3505                                 ;IT IS.
 3506
 3507 017242  013764  002424  000064  1$:     MOV     E.DA,LSTDA(R4)  ;SAVE DA FOR SOFT ERROR PRINT
 3508 017250  032737  040000  002420          BIT     #DERR,E.CS      ;DRIVE ERROR?
 3509 017256  001402                          BEQ     2$              ;NO, CONTINUE
 3510 017260  000137  021350                  JMP     CKDERR          ;YES, GO CHECK DRIVE ERROR
 3511 017264  032737  000001  002420  2$:     BIT     #DRDY,E.CS      ;DRIVE READY THERE
 3512 017272  001017                          BNE     23$             ;YES, CONTINUE CHECKING
 3513 017274  004537  024362                  JSR     R5,GETDST       ;NO,GET DRIVE STATUS
 3514 017300  042701  000100                  BIC     #100,R1         ;GET RID OF HEAD
 3515 017304  020127  000034                  CMP     R1,#34          ;ALLOW ONLY SEEK TRACKING STATE
 3516 017310  001410                          BEQ     23$             ;WAS 34 SKIP ERROR
 3517
 3518 017312  005264  000012                  INC     ERRCNT(R4)      ;INDICATE HARD ERROR
 3519 017316                                  ERRDF   1000.,NORDY,ERR9
      017316  104455                          TRAP    C$ERDF
      017320  001750                          .WORD   1000
      017322  002704                          .WORD   NORDY
      017324  005602                          .WORD   ERR9
 3520
 3521 017326  000137  021204                  JMP     EXIT1
 3522
 3523 017332  032737  020000  002420  23$:    BIT     #NXM,E.CS       ;NON-EXISTENT MEMORY?
 3524 017340  001407                          BEQ     3$              ;NO, KEEP CHECKING
 3525 017342  012764  004346  000052          MOV     #MTNXM,RTYPE(R4)  ;ERROR MESSAGE
 3526 017350  005264  000034                  INC     NXMCNT(R4)      ;LOG ERROR
 3527 017354  000137  017766                  JMP     111$            ;CHECK RETRY, EXIT BACK
 3528
 3529 017360  032737  014000  002420  3$:     BIT     #BIT12!BIT11,E.CS  ;QUALIFING BITS SET?
 3530 017366  001020                          BNE     5$              ;YES, CAN'T BE OPI ALONE
 3531
 3532 017370  032737  002000  002420          BIT     #OPI,E.CS       ;OPI SET?
 3533 017376  001006                          BNE     4$              ;YES, CONTINUE
 3534
 3535 017400                                  ERRSF   10.,UDERR,ERR1  ;WE HAVE AN UNDIAGNOSABLE CONDITION, ONLY COMPOSITE SET
      017400  104454                          TRAP    C$ERSF
      017402  000012                          .WORD   10
      017404  003007                          .WORD   UDERR
      017406  005070                          .WORD   ERR1
 3536 017410                          33$:     BREAK
      017410  104422                          TRAP    C$BRK
 3537 017412  000776                          BR      33$
 3538
 3539 017414  012764  004341  000052  4$:     MOV     #MTOPI,RTYPE(R4);SET UP FOR "OPI" PRINT
 3540 017422  005264  000030                  INC     OPICNT(R4)      ;LOG ERROR
 3541 017426  000557                          BR      111$            ;CHECK RETRY EXIT BACK
 3542
 3543                                 ;WE KNOW IT'S NOW EITHER DLT, DCRC,HNF, OR HCRC
 3544                                 ;CHECK FOR EACH
 3545
```

CONTROLLER ERROR CHECK ROUTINE

```
3546 017430  032737  002000  002420  5$:     BIT     #OPI,E.CS       ;OPI QUALIFIER SET?
3547 017436  001060                          BNE     7$              ;YES, THEN IT S HCRC OR HNF
3548
3549                                 ;IT'S NOW DOWN TO DLT OR DCRC
3550
3551 017440  032737  010000  002420          BIT     #DLT,E.CS       ;DATA LATE?
3552 017446  001406                          BEQ     6$              ;NO, MUST BE DATA CRC
3553 017450  012764  004334  000052          MOV     #MTDLT,RTYPE(R4);SET UP FOR "DLT"PRINT
3554 017456  005264  000026                  INC     DLTCNT(R4)      ;LOG ERROR
3555 017462  000541                          BR      111$            ;CHECK RETRY, EXIT
3556
3557 017464  013737  002424  002342  6$:     MOV     E.DA,CHKSEC     ;SET UP SECTOR TO LOOK FOR
3558 017472  005364  000064                  DEC     LSTDA(R4)       ;DOWN COUNT FOR PRINT OUT
3559 017476  005337  002342                  DEC     CHKSEC          ;DOWN COUNT FOR LOOP UP
3560 017502  004537  027146                  JSR     R5,CKBDSC       ;CHECK BAD SECTOR LIST
3561 017506  005737  002340                  TST     HDRFND          ;WAS HEADER THERE?
3562 017512  001117                          BNE     110$            ;IGNORE ERROR, RETURN
3563 017514  005264  000022          117$:   INC     DCRCER(R4)      ;ACCOUNT FOR ERROR
3564 017520  012764  004327  000052          MOV     #MTDCRC,RTYPE(R4);SET UP FOR "DCRC" PRINT
3565 017526  022764  000102  000044          CMP     #INTEN!WRCHK,FUNC(R4)
3566 017534  001001                          BNE     118$
3567 017536  000513                          BR      111$
3568
3569 017540  005737  010636          118$:   TST     T.DCK           ;DUMP BUFFER?
3570 017544  001510                          BEQ     111$            ;NO, EXIT
3571 017546                                  PRINTF  #FMT14,#DMPDCK
     017546  012746  003265                  MOV     #DMPDCK,-(SP)
     017552  012746  007463                  MOV     #FMT14,-(SP)
     017556  012746  000002                  MOV     #2,-(SP)
     017562  010600                          MOV     SP,R0
     017564  104417                          TRAP    C$PNTF
     017566  062706  000006                  ADD     #6,SP
3572 017572  004537  026222                  JSR     R5,DMPBUF       ;DUMP BUFFER
3573
3574 017576  000473                          BR      111$            ;EXIT
3575
3576                                 ;IT'S NOW EITHER HNF OR HCRC.
3577                                 ;IF HCRC AND RDHDR, DETERMINE IF BAD SECTOR BY DOING 40 RDHDRS
3578                                 ;IF HCRC AND R/W, CHECK IF DA IS IN BAD SECTOR FILE
3579                                 ;IF HNF READ HEADER TO VERIFY IF ON CORRECT CYLINDER
3580                                 ;THEN IF ON CORRECT CYLINDER SEE IF DA IS A BAD SECTOR
3581                                 ;IF NOT ON CORRECT CYLINDER REPORT MISSEEK, LOG MISEEK
3582                                 ;AND PRESENT POSITION UPDATE.
3583
3584 017600  032737  010000  002420  7$:     BIT     #HNF,E.CS       ;HEADER NOT FOUND SET?
3585 017606  001470                          BEQ     112$            ;NO IT MUST BE HCRC
3586 017610  012701  000051                  MOV     #41.,R1         ;ALLOW FORTY READ HEADERS TO
3587 017614  004537  024376          8$:     JSR     R5,ISDRST
3588 017620  016402  000106                  MOV     DRSEL(R4),R2    ;FIND CYLINDER
3589 017624  052702  000010                  BIS     #RDHDR,R2       ;READ HEADER
3590 017630  016403  000104                  MOV     DCS(R4),R3
3591 017634  010263  000000                  MOV     R2,CS(R3)       ;ISSUE READ HEADER
3592 017640  004537  024270                  JSR     R5,WTRDY        ;WAIT
3593 017644  005301                          DEC     R1              ;DONE 40 OF THESE?
3594 017646  001424                          BEQ     9$              ;YES, GIVE UP WE DON'T HAVE ALL DAY!
3595 017650  005763  000000                  TST     CS(R3)          ;IS ERROR SET?
3596 017654  100757                          BMI     8$              ;YES, GO DO IT AGAIN
```

CONTROLLER ERROR CHECK ROUTINE

```
3597
3598 017656  016301  000006                  MOV     MP(R3),R1       ;GET HEADER
3599 017662  010137  002434                  MOV     R1,C.HDR        ;SAVE FOR ERROR REPORTING
3600 017666  043701  002272                  BIC     SMSK,R1         ;MASK OUT SECTOR BITS
3601 017672  020164  000124                  CMP     R1,PRPOS(R4)    ;IS CYLINDER HEAD CORRECT?
3602 017676  001415                         BEQ     10$             ;YES, GO CHECK BAD SECTOR LIST
3603
3604 017700  005264  000072                  INC     TRERR(R4)
3605 017704                                  ERRHRD  20.,TRACK,ERR2  ;TRACKING DRIFT ERROR
     017704  104456                         TRAP    C$ERHRD
     017706  000024                         .WORD   20
     017710  003305                         .WORD   TRACK
     017712  005076                         .WORD   ERR2
3606
3607 017714  000137  020700                  JMP     SKRETRY         ;FIX TRACKING ERROR
3608
3609 017720                          9$:     ERRHRD  30.,EXHAUS,ERR1 ;WE CAN'T FIND GOOD HEADER ON THIS TRACK
     017720  104456                         TRAP    C$ERHRD
     017722  000036                         .WORD   30
     017724  002773                         .WORD   EXHAUS
     017726  005070                         .WORD   ERR1
3610
3611 017730  000410                         BR      110$
3612
3613 017732  013737  002424  002342  10$:    MOV     E.DA,CHKSEC
3614 017740  004537  027224                  JSR     R5,CKBDTK       ;GO CHECK BAD SECTOR FILE
3615 017744  005737  002340                  TST     HDRFND          ;WAS IT THERE
3616 017750  001401                         BEQ     11$             ;NO, LOG IT EXIT
3617 017752  000577                  110$:   BR      GOERRX          ;YES IGNORE ERROR
3618
3619 017754  005264  000032          11$:    INC     HNFERR(R4)      ;LOG IT
3620 017760  012764  004314  000052          MOV     #MTHNF,RTYPE(R4);SET UP FOR "HNF" PRINT
3621 017766  000573                  111$:   BR      GOFIN           ;EXIT
3622
3623                                         ;
3624                                         ;IT WAS A HEADER CRC ERROR, FIGURE OUT IF IT WAS
3625                                         ;ON A READ HEADER OR READ/WRITE
3626                                         ;
3627
3628 017770  022764  000110  000044  112$:   CMP     #INTEN!RDHDR,FUNC(R4)   ;READ HEADER?
3629 017776  001417                         BEQ     13$             ;YES, GO FIND OUT MORE ABOUT IT
3630                                                                 ;NO, IT MUST BE R/W
3631 020000  013737  002424  002342          MOV     E.DA,CHKSEC
3632 020006  004537  027146                  JSR     R5,CKBDSC       ;BAD SECTOR SEARCH
3633 020012  005737  002340                  TST     HDRFND          ;WAS OUR DA THERE?
3634 020016  001401                         BEQ     12$             ;NO, MUST BE LEGIT ERROR
3635 020020  000554                         BR      GOERRX          ;YES, IGNORE ERROR
3636
3637 020022  005264  000024          12$:    INC     HCRCER(R4)      ;LOG ERROR
3638 020026  012764  004321  000052          MOV     #MTHCRC,RTYPE(R4)
3639 020034  000550                         BR      GOFIN
3640
3641 020036  017401  000110          13$:    MOV     @BBA(R4),R1     ;USE IT'S BUFFER TO STORE HDRS
3642 020042  012737  000050  002350          MOV     #40.,TEMP1      ;40 CONSECUTIVE HEADERS
3643 020050  012702  000010          14$:    MOV     #RDHDR,R2       ;READ HEADER
3644 020054  056402  000106                  BIS     DRSEL(R4),R2    ;
3645 020060  016403  000104                  MOV     DCS(R4),R3      ;
```

CONTROLLER ERROR CHECK ROUTINE

```
3646 020064  010263  000000                MOV     R2,CS(R3)          ;
3647 020070  004537  024270                JSR     R5,WTRDY           ;WAIT FOR READY
3648 020074  016321  000000                MOV     CS(R3),(R1)+       ;READ ALL REGISTERS
3649 020100  016321  000006                MOV     MP(R3),(R1)+       ;
3650 020104  016321  000006                MOV     MP(R3),(R1)+       ;
3651 020110  016321  000006                MOV     MP(R3),(R1)+       ;
3652 020114  005337  002350                DEC     TEMP1              ;DONE 40 YET?
3653 020120  001353                        BNE     14$                ;NO, GO BACK
3654
3655                                ;WE HAVE 40 HEADERS NOW LETS SEE IF WE CAN VERIFY WHETHER
3656                                ;OR NOT A BAD SECTOR CAUSED THE ERROR.  CHECK FIRST TO SEE
3657                                ;IF WE HAVE ANY BAD SECTORS ON THIS TRACK.
3658
3659 020122  017402  000110         99$:   MOV     @BBA(R4),R2        ;GET BUFFER START
3660 020126  012701  000050                MOV     #40.,R1            ;FORTY HEADERS
3661 020132  032712  002000         15$:   BIT     #OPI,(R2)          ;IS OPI SET IN CS
3662 020136  001403                        BEQ     16$                ;NO, WELL CAN'T BE HCRC
3663 020140  032712  004000                BIT     #HCRC,(R2)         ;INSURE HCRC W/OPI
3664 020144  001005                        BNE     17$                ;FOUND GO SEE IF IT COMPARES
3665 020146  062702  000010         16$:   ADD     #10,R2             ;NEXT CS IMAGE
3666 020152  005301                        DEC     R1                 ;DONE 40
3667 020154  001366                        BNE     15$
3668 020156  000721                        BR      12$
3669
3670 020160  020274  000110         17$:   CMP     R2,@BBA(R4)        ;IS HEADER FIRST ONE?
3671 020164  001046                        BNE     21$                ;NO, READ PREVIOUS HEADER
3672                                                                  ;YES, WE'LL HAVE TO GO THRU
3673                                                                  ;AND CHECK OTHERS BEFORE WE
3674                                                                  ;CAN SAFELY CALCULATE
3675                                                                  ;"SUPPOSED" BAD SECTOR
3676 020166  017401  000110                MOV     @BBA(R4),R1
3677 020172  012703  000001                MOV     #1,R3
3678 020176  062701  000010         18$:   ADD     #10,R1
3679 020202  032711  002000                BIT     #OPI,(R1)
3680 020206  001416                        BEQ     19$
3681 020210  032711  004000                BIT     #HCRC,(R1)
3682 020214  001413                        BEQ     19$
3683 020216  005203                        INC     R3
3684 020220  022703  000017                CMP     #15.,R3
3685 020224  001364                        BNE     18$
3686
3687 020226  012737  003667  002246        MOV     #MBOMSC,WHY        ;DROP DRIVE DUE TO
3688 020234  004537  023450                JSR     R5,DRDRV           ;MORE THAN 16 BAD SECTORS
3689 020240  000137  021204                JMP     EXIT1
3690
3691 020244  005012                 19$:   CLR     (R2)               ;CLEAR THIS CS
3692 020246  062701  000002                ADD     #2,R1              ;GET IT'S HEADER ADDRESS
3693 020252  011102                        MOV     (R1),R2            ;GET HEADER
3694 020254  010201                        MOV     R2,R1              ;SAVE HEADER
3695 020256  042702  177700                BIC     #177700,R2         ;MASK   ONLY SECTOR
3696 020262  160301                        SUB     R3,R1              ;BACK UP TO SECTOR WHICH IS BAD
3697 020264  100402                        BMI     20$                ;IF MINUS DO MAGIC
3698 020266  160302                        SUB     R3,R2              ;NO THEN SUBTRACT IS LEGAL
3699 020270  000421                        BR      22$                ;BRANCH TO CHECK FILE
3700 020272  160302                 20$:   SUB     R3,R2              ;THIS SUB PRODUCES WRONG ANSWER
3701 020274  062702  000050                ADD     #50,R2             ;FIX IT UP
3702 020300  000415                        BR      22$                ;GO CHECK FILE
```

CONTROLLER ERROR CHECK ROUTINE

```
3703
3704 020302  005012                  21$:   CLR     (R2)              ;CLEAR THIS CS OUT
3705 020304  162702  000006                SUB     #6,R2             ;GET PREVIOUS HEADER
3706 020310  011201                        MOV     (R2),   R1
3707 020312  005201                        INC     R1
3708 020314  010102                        MOV     R1,R2
3709 020316  042701  177700                BIC     #177700,R1
3710 020322  022701  000050                CMP     #40.,R1
3711 020326  002402                        BLT     22$
3712 020330  162702  000050                SUB     #40.,R2
3713 020334  010237  002342          22$:  MOV     R2,CHKSEC
3714 020340  004537  027146                JSR     R5,CKBUSC
3715 020344  005737  002340                TST     HDRFND
3716 020350  001664                        BEQ     99$
3717 020352  000137  021210          GOERRX: JMP     ERREX
3718
3719 020356  000137  021312          GOFIN:  JMP     FINERR
3720
3721                                  .SBTTL  COMMAND SERVICE ROUTINES
3722
3723                                  ;THERE WAS NO ERROR SO.....
3724                                  ;NOW WE WILL FIND OUT WHICH FUNCTION WE DID TO CAUSE
3725                                  ;INTERRUPT AND ACT ACCORDINGLY.
3726                                  ;
3727
3728 020362  016401  000044          CHKFNC: MOV     FUNC(R4),R1       ;GET FUNCTION OF DRIVE
3729 020366  006201                        ASR     R1                ;ALIGN THE FUNCTION CODE
3730 020370  042701  000040                BIC     #40,R1            ;WIPE OUT INT. ENAB (SHIFTED)
3731 020374  005301                        DEC     R1                ;WRITE CHECK??
3732 020376  001002                        BNE     2$                ;NO. BRANCH
3733 020400  000137  020540                JMP     AFWRCK            ;FUNCTION #1
3734
3735 020404  005301                  2$.   DEC     R1                ;GET STATUS?
3736 020406  001565                        BEQ     AGSTAT            ;BRANCH IF SO...FUNCTION #2
3737 020410  005301                        DEC     R1                ;SEEK?
3738 020412  001421                        BEQ     ASEEK             ;BRANCH IF SO...FUNCTION #3
3739 020414  005301                        DEC     R1                ;RDHDR?
3740 020416  001500                        BEQ     ARDHDR            ;BRANCH IF SO...FUNCTION #4
3741 020420  005301                        DEC     R1                ;WRITE?
3742 020422  001002                        BNE     1$                ;NO. BRANCH
3743 020424  000137  021066                JMP     AWRITE            ;FUNCTION #5
3744 020430  005301                  1$:   DEC     R1                ;READ?
3745 020432  001432                        BEQ     AFREAD            ;BRANCH IF SO...FUNCTION #6
3746 020434  005301                        DEC     R1                ;READ W/NO HDR COMPARE?
3747 020436  001440                        BEQ     AFWRCK            ;YES   TREAT AS IF WRITE CHECK
3748
3749 020440                               ERRSF   210.,PRGER        ;SHOULD NEVER GET HERE!!!
     020440  104454                       TRAP    C$ERSF
     020442  000322                       .WORD   210
     020444  002732                       .WORD   PRGER
     020446  000000                       .WORD   0
3750 020450  000000                       HALT
3751 020452  000137  021152          XEXIT:  JMP     EXIT
3752
3753                                  .SBTTL  SEEK INTERRUPT SERVICE
3754
3755 020456  052764  000001  000056  ASEEK:  BIS     #SKDON,PRFLGS(R4)  ;SET SEEK VERIFY NEEDED
```

SEEK INTERRUPT SERVICE

```
3756 020464  005064  000114                    CLR     RSEEK(R4)          ;CLEAR THE RETRY FLAG
3757 020470  005264  000054                    INC     SKCNT1(R4)         ;INCREMENT COUNT
3758 020474  026427  000054  001750            CMP     SKCNT1(R4),#1000.  ;10(3) REACHED
3759 020502  002404                            BLT     99$                ;NO, EXIT
3760 020504  005264  000000                    INC     SKCNT(R4)          ;YES, BUMP THOUSANDS
3761 020510  005064  000054                    CLR     SKCNT1(R4)
3762 020514  000137  021152          99$:      JMP     EXIT
3763
3764                                .SBTTL  READ INTERRUPT SERVICE
3765
3766 020520  042764  000001  000056  AFREAD: BIC     #SKDON,PRFLGS(R4)       ;CLEAR THE SEEK VERIFY FLAG
3767                                ;       SETPRI  #340                                        ;JSD REV A
3768 020526                                 SETPRI  #300                                        ;JSD REV A
     020526  012700  000300                MOV     #300,R0
     020532  104441                        TRAP    C$SPRI
3769 020534  004537  023672                JSR     R5,CKDATA          ;CHECK DATA
3770
3771 020540  016401  000042          AFWRCK: MOV     BMP(R4),R1         ;BUMP UP XFER COUNT
3772 020544  005401                        NEG     R1                 ;MAKE POSITIVE
3773 020546  060164  000002                ADD     R1,RXFR1(R4)       ;ADD THE BITS
3774 020552  022764  023420  000002        CMP     #10000.,RXFR1(R4)  ;10(8) REACHED YET
3775 020560  101016                        BHI     2$                 ;NO, EXIT
3776 020562  005264  000004                INC     RXFR2(R4)          ;BUMP 10(10)
3777 020566  162764  023420  000002        SUB     #10000.,RXFR1(R4)  ;START 10(8) AT 0
3778 020574  022764  023420  000004        CMP     #10000.,RXFR2(R4)  ;10(10) REACHED YET
3779 020602  101005                        BHI     2$                 ;NO, EXIT
3780 020604  005264  000060                INC     RXFR3(R4)          ;YES BUMP 65K 10(10)
3781 020610  162764  023420  000004        SUB     #10000.,RXFR2(R4)  ;MAKE 10(10) 0
3782 020616  000555                2$:     BR      EXIT               ;EXIT
3783
3784                                .SBTTL  READ HEADER INTERRUPT SERVICE
3785
3786 020620  013701  002426          ARDHDR: MOV     E.MP,R1            ;GET HEADER
3787 020624  043701  002272                  BIC     SMSK,R1            ;MASK OUT SECTOR BITS
3788 020630  026401  000124                  CMP     PRPOS(R4),R1       ;IS HEADER CORRECT?
3789 020634  001442                        BEQ     1$                 ;YES, CONTINUE
3790
3791 020636  032764  000001  000056        BIT     #SKDON,PRFLGS(R4)  ;IS THIS MIS-SEEK OR TRACKING ERROR
3792 020644  001407                        BEQ     2$                 ;BRANCH IF TRACKING
3793
3794 020646  005264  000016                INC     SKECNT(R4)         ;ACCOUNT FOR SEEK ERROR
3795 020652                                 ERRHRD  50.,MSKER,ERR2
     020652  104456                        TRAP    C$ERHRD
     020654  000062                        .WORD   50
     020656  003031                        .WORD   MSKER
     020660  005076                        .WORD   ERR2
3796 020662  000406                        BR      3$                 ;BRANCH AROUND TRACKING ERROR REPORT
3797
3798 020664  005264  000072          2$:     INC     TRERR(R4)          ;ACCOUNT FOR TRACKING ERROR
3799 020670                                 ERRHRD  55.,TRACK,ERR2     ;TRACKING ERROR
     020670  104456                        TRAP    C$ERHRD
     020672  000067                        .WORD   55
     020674  003305                        .WORD   TRACK
     020676  005076                        .WORD   ERR2
3800
3801          020700                SKRETRY=.
3802
```

READ HEADER INTERRUPT SERVICE

```
3803 020700  005264  000114          3$:     INC     RSEEK(R4)  ;SET RETRY IN PROGRESS
3804 020704  026437  000114  010660          CMP     RSEEK(R4),T.SLT ;RETRY EXHAUSTED?????
3805 020712  101405                          BLOS    4$              ;NO, THEN RETRY
3806
3807 020714                                  ERRHRD  333.,SEXHAU,ERR2
     020714  104456                          TRAP    C$ERHRD
     020716  000515                          .WORD   333
     020720  003523                          .WORD   SEXHAU
     020722  005076                          .WORD   ERR2
3808 020724  000406                          BR      1$
3809
3810 020726  010164  000050          4$:     MOV     R1,LSTHDR(R4)   ;SET UP RETRY
3811 020732  042764  000001  000056          BIC     #SKDON,PRFLGS(R4)  ;ALLOW SEEK
3812 020740  000504                          BR      EXIT            ;EXIT
3813 020742  042764  000001  000056  1$:     BIC     #SKDON,PRFLGS(R4)  ;SET VERIFICATION DONE
3814 020750  005064  000114                  CLR     RSEEK(R4)
3815 020754  010164  000124                  MOV     R1,PRPOS(R4)    ;MAKE THIS HEADER PRESENT POSITION
3816 020760  000474                          BR      EXIT            ;EXIT
3817
3818                                 .SBTTL  GET STATUS INTERRUPT SERVICE
3819
3820 020762  013701  002426          AGSTAT: MOV     E.MP,R1         ;GET STATUS
3821 020766  042701  000100                  BIC     #100,R1         ;CLEAR OUT HEAD SELECT
3822 020772  005737  010652                  TST     T.ROF           ;READ ONLY
3823 020776  001402                          BEQ     2$
3824 021000  042701  020000                  BIC     #WL,R1
3825 021004  032701  177400          2$:     BIT     #177400,R1      ;ANY BITS WRONG
3826 021010  001406                          BEQ     1$              ;NO, CONTINUE
3827
3828 021012  005264  000012                  INC     ERRCNT(R4)      ;STATUS BITS WRONG
3829 021016                                  ERRHRD  60.,MDSER,ERR4
     021016  104456                          TRAP    C$ERHRD
     021020  000074                          .WORD   60
     021022  003116                          .WORD   MDSER
     021024  005312                          .WORD   ERR4
3830
3831 021026  010102                  1$:     MOV     R1,R2           ;COPY STATUS WORD
3832 021030  042702  177700                  BIC     #177700,R2      ;GET STATE BITS
3833 021034  022702  000034                  CMP     #34,R2          ;COVER CLSD, HEADS OUT, BRUSHES HOME, SEEK TRACK COUNTING?
3834 021040  001444                          BEQ     EXIT            ;YES, EXIT
3835 021042  022702  000035                  CMP     #35,R2          ;COVER CLSD, HEADS OUT, BRUSHES HOME, SEEK LINEAR MODE
3836 021046  001441                          BEQ     EXIT            ;YES, EXIT
3837
3838 021050  005264  000012                  INC     ERRCNT(R4)
3839 021054                                  ERRHRD  70.,MDSER,ERR4
     021054  104456                          TRAP    C$ERHRD
     021056  000106                          .WORD   70
     021060  003116                          .WORD   MDSER
     021062  005312                          .WORD   ERR4
3840
3841 021064  000432                          BR      EXIT
3842
3843                                 .SBTTL  WRITE INTERRUPT SERVICE
3844
3845 021066  042764  000001  000056  AWRITE: BIC     #SKDON,PRFLGS(R4) ;CLEAR SEEK VERIFY FLAG
3846 021074  016401  000042                  MOV     BMP(R4),R1      ;GET WORD COUNT
3847 021100  005401                          NEG     R1              ;MAKE POSITIVE
```

WRITE INTERRUPT SERVICE

```
3848 021102  060164  000006                ADD     R1,WXFR1(R4)      ;ADD THE BITS
3849 021106  022764  023420  000006        CMP     #10000.,WXFR1(R4) ;10(5) YET?
3850 021114  101016                        BHI     EXIT              ;NO - EXIT
3851 021116  005264  000010                INC     WXFR2(R4)         ;YES BUMP 10(10)
3852 021122  162764  023420  000006        SUB     #10000.,WXFR1(R4) ;10(5) GOES TO ZERO
3853 021130  022764  023420  000010        CMP     #10000.,WXFR2(R4) ;10(10) YET?
3854 021136  101005                        BHI     EXIT              ;NO - EXIT
3855 021140  005264  000062                INC     WXFR3(R4)         ;INC 65K (10)(10)
3856 021144  162764  023420  000010        SUB     #10000.,WXFR2(R4) ;MAKE 10(10)
3857
3858 021152  005764  000036        EXIT:   TST     RETRY(R4)         ;IN PROCESS OF RETRYING?
3859 021156  001414                        BEQ     ERREX             ;NO
3860 021160  026427  000052  004353        CMP     RTYPE(R4),#MTDRV
3861 021166  001406                        BEQ     EXIT1
3862 021170  005264  000014                INC     SFTCNT(R4)        ;YES, LOG SOFT ERROR
3863
3864 021174                                ERRSOFT 80.,MSFER,ERR3    ;REPORT SOFT ERROR
     021174  104457                        TRAP    C$ERSOFT
     021176  000120                        .WORD   80
     021200  003042                        .WORD   MSFER
     021202  005162                        .WORD   ERR3
3865
3866 021204  005064  000036        EXIT1:  CLR     RETRY(R4)         ;CLEAR RETRY
3867
3868                               .SBTTL  EXIT FOR INTERRUPT SERVICE
3869
3870 021210  042774  000100  000104  ERREX:  BIC     #INTEN,@DCS(R4)
3871 021216  012600                        MOV     (SP)+,R0
3872 021220  012601                        MOV     (SP)+,R1
3873 021222  012602                        MOV     (SP)+,R2
3874 021224  012603                        MOV     (SP)+,R3
3875 021226  012637  002476                MOV     (SP)+,INCALL
3876 021232  012637  002474                MOV     (SP)+,OPCALL
3877 021236  012637  002246                MOV     (SP)+,WHY
3878 021242  012637  002350                MOV     (SP)+,TEMP1
3879 021246  012637  002340                MOV     (SP)+,HDRFND
3880 021252  012637  002342                MOV     (SP)+,CHKSEC
3881 021256  012637  002432                MOV     (SP)+,E.MP2
3882 021262  012637  002430                MOV     (SP)+,E.MP1
3883 021266  012637  002426                MOV     (SP)+,E.MP
3884 021272  012637  002424                MOV     (SP)+,E.DA
3885 021276  012637  002422                MOV     (SP)+,E.BA
3886 021302  012637  002420                MOV     (SP)+,E.CS
3887 021306  012604                        MOV     (SP)+,R4
3888 021310                                ENDSRV
     021310                        L10027:
     021310  000002                        RTI
3889
3890 021312  004537  022546        FINERR: JSR     R5,RCNT           ;CHECK TO SEE IF WE HAVE EXCEEDED
3891 021316  000405                        BR      1$                ;RETRY LIMIT, IF SO 1$ AND REPORT HARD
3892 021320  013764  002420  000116        MOV     E.CS,SOFTCS(R4)
3893 021326  000137  021210                JMP     ERREX             ;NOT EXCEEDED EXIT
3894 021332  005264  000012        1$:     INC     ERRCNT(R4)        ;INDICATE ERROR
3895
3896 021336                                ERRHRD  90.,MHDER,ERR13   ;NON-RECOVERABLE ERROR
     021336  104456                        TRAP    C$ERHRD
     021340  000132                        .WORD   90
```

```
EXIT FOR INTERRUPT SERVICE

        021342  003252                          .WORD   MHDER
        021344  005724                          .WORD   ERR13
   3897 021346  000716                          BR      EXIT1
   3898
   3899                                 .SBTTL  DRIVE ERROR INTERRUPT SERVICE
   3900
   3901                                 ;WE HAVE A DRIVE ERROR, LET'S GET THE STATUS
   3902
   3903 021350  005264  000020          CKDERR: INC     DERCNT(R4)      ;ACCOUNT FOR ERROR
   3904 021354  004537  024362                  JSR     R5,GETDST       ;GET DRIVE STATUS
   3905                                 ;REPORT DRIVE ERROR
   3906 021360                                  ERRHRD  224.,DRVER,ERR9 ;DRIVE ERROR
        021360  104456                          TRAP    C$ERHRD
        021362  000340                          .WORD   224
        021364  003061                          .WORD   DRVER
        021366  005602                          .WORD   ERR9
   3907
   3908                                 ;ACT ACCORDINGLY TO DRIVE ERROR
   3909
   3910 021370  032701  001000                  BIT     #VC,R1          ;VOLUME CHECK?
   3911 021374  001027                          BNE     9$              ;YES, GO ISSUE RESET
   3912 021376  032701  010000                  BIT     #SKTO,R1        ;SEEK TIME OUT?
   3913 021402  001070                          BNE     12$             ;YES, ISSUE RESET
   3914 021404  032701  144000                  BIT     #WDE!HCE!SPE,R1 ;WRITE DATA, CURRENT HEAD, SPINDLE?
   3915 021410  001153                          BNE     15$             ;GO WAIT FOR HEADS TO UNLOAD
   3916 021412  032701  002000                  BIT     #WGE,R1         ;WRITE GATE ERROR
   3917 021416  001003                          BNE     20$             ;YES, ISSUE  RESET
   3918 021420  004537  024376                  JSR     R5,ISDRST       ;ISSUE RESET
   3919 021424  000431                          BR      10$             ;GO CHECK DRIVE READY
   3920 021426  004537  024376          20$:    JSR     R5,ISDRST       ;ISSUE RESET
   3921 021432  004537  024362                  JSR     R5,GETDST       ;RESET WORK?
   3922 021436  032701  002000                  BIT     #WGE,R1         ;WGE CLEAR
   3923 021442  001422                          BEQ     10$             ;YES GO CHECK DRIVE READY
   3924 021444  012737  003157  002246          MOV     #WGEST,WHY      ;REPORT WGE DIDN'T CLR
   3925 021452  000412                          BR      91$             ;DROP DRIVE
   3926
   3927 021454  004537  024376          9$:     JSR     R5,ISDRST       ;ISSUE RESET
   3928 021460  004537  024362                  JSR     R5,GETDST       ;RESET WORK
   3929 021464  032701  001000                  BIT     #^C,R1          ;VOL CHK CLEAR
   3930 021470  001407                          BEQ     10$             ;YES, CHECK DRIVE READY
   3931 021472  012737  003132  002246          MOV     #MVCER,WHY      ;DROP THE DRIVE
   3932
   3933 021500  004537  023450          91$:    JSR     R5,DRDRV
   3934 021504  000137  021204                  JMP     EXIT1
   3935 021510  032763  000001  000000  10$:    BIT     #DRDY,CS(R3)    ;DRIVE READY POSTED?
   3936 021516  001004                          BNE     101$            ;YES, PRINT RECOVERED
   3937
   3938 021520  012737  002664  002246          MOV     #DNRDY,WHY
   3939 021526  000764                          BR      91$             ;NO, DROP DRIVE
   3940
   3941 021530                          101$:   PRINTB  #FMT14,#MRDER   ;PRINT DRIVE RECOVERED
        021530  012746  003212                  MOV     #MRDER,-(SP)
        021534  012746  007463                  MOV     #FMT14,-(SP)
        021540  012746  000002                  MOV     #2,-(SP)
        021544  010600                          MOV     SP,R0
        021546  104414                          TRAP    C$PNTB
        021550  062706  000006                  ADD     #6,SP
```

DRIVE ERROR INTERRUPT SERVICE

```
3942 021554  004537  022246            JSR     R5,GHDR         ;GET THE CURRENT DISK POSITION - HEADER
3943 021560  000137  021312            JMP     FINERR
3944 021564  012702  000004      12$:  MOV     #4,R2           ;SEEK TIME OUT
3945 021570  004537  024376      13$:  JSR     R5,ISDRST       ;ISSUE DRIVE RESET
3946                                                           ;FOUR TIMES BEFORE
3947 021574                            WAITMS  #15.            ;DROPPING DRIVE
     021612  012727  000372            MOV     #250.,(PC)+
     021616  000000                    .WORD   0
     021620  013727  002116            MOV     L$DLY,(PC)+
     021624  000000                    .WORD   0
     021626  005367  177772            DEC     -6(PC)
     021632  001375                    BNE     .-4
     021634  005367  177756            DEC     -22(PC)
     021640  001367                    BNE     .-20
3948
3949 021650  032763  000001  000000    BIT     #DRDY,CS(R3)    ;DRIVE READY YET?
3950 021656  001006                    BNE     14$             ;YES, CHECK IF ERROR CLEARED
3951 021660  005302                    DEC     R2              ;NO, HAVE WE DONE IT FOUR TIMES
3952 021662  001342                    BNE     13$             ;YET
3953
3954 021664  012737  003070  002246  141$:  MOV     #MDERS,WHY      ;YES, DROP DRIVE
3955 021672  000702                    BR      91$
3956
3957 021674  032763  040000  000000  14$:  BIT     #DERR,CS(R3)    ;DRIVE ERROR SET STILL
3958 021702  001370                    BNE     141$            ;YES, DROP DRIVE
3959 021704                            PRINTB  #FMT14,#MRDER
     021704  012746  003212            MOV     #MRDER,-(SP)
     021710  012746  007463            MOV     #FMT14,-(SP)
     021714  012746  000002            MOV     #2,-(SP)
     021720  010600                    MOV     SP,R0
     021722  104414                    TRAP    C$PNTB
     021724  062706  000006            ADD     #6,SP
3960 021730  004537  022246            JSR     R5,GHDR
3961 021734  000137  021152            JMP     EXIT
3962
3963 021740  012702  000004      15$:  MOV     #4,R2           ;WAIT FOR HEADS TO UNLOAD
3964 021744  004537  024362      16$:  JSR     R5,GETDST       ;GET STATUS
3965 021750  032701  000020            BIT     #BIT4,R1        ;UNLOAD STATE
3966 021754  001434                    BEQ     17$             ;YES, CONTINUE W/ RECOVERY
3967 021756                            WAITMS  #15.            ;WAIT A WHILE
     021774  012727  000372            MOV     #250.,(PC)+
     022000  000000                    .WORD   0
     022002  013727  002116            MOV     L$DLY,(PC)+
     022006  000000                    .WORD   0
     022010  005367  177772            DEC     -6(PC)
     022014  001375                    BNE     .-4
     022016  005367  177756            DEC     -22(PC)
     022022  001367                    BNE     .-20
3968 022032  005302                    DEC     R2              ;WAIT LONG ENOUGH
3969 022034  001343                    BNE     16$             ;NO, GO BACK
3970 022036  012737  003547  002246    MOV     #UNLOAD,WHY     ;DROP DRIVE
3971 022044  000615                    BR      91$
3972
3973 022046  004537  024376      17$:  JSR     R5,ISDRST       ;ISSUE RESET
3974 022052                            WAITMS  #1.             ;
     022070  012727  000372            MOV     #250.,(PC)+
     022074  000000                    .WORD   0
```

DRIVE ERROR INTERRUPT SERVICE

```
        022076  013727  002116                  MOV     L$DLY,(PC)+
        022102  000000                          .WORD   0
        022104  005367  177772                  DEC     -6(PC)
        022110  001375                          BNE     .-4
        022112  005367  177756                  DEC     -22(PC)
        022116  001367                          BNE     .-20
  3975  022126  032763  040000  000000          BIT     #DERR,CS(R3)    ;DRIVE ERROR CLEAR?
  3976  022134  001253                          BNE     141$            ;NO, DROP DRIVE
  3977  022136  012702  000075                  MOV     #61.,R2         ;YES, WAIT 60 SECONDS
  3978  022142                          18$:    WAITMS  #10.            ;FOR DRIVE READY TO
        022160  012727  000372                  MOV     #250.,(PC)+
        022164  000000                          .WORD   0
        022166  013727  002116                  MOV     L$DLY,(PC)+
        022172  000000                          .WORD   0
        022174  005367  177772                  DEC     -6(PC)
        022200  001375                          BNE     .-4
        022202  005367  177756                  DEC     -22(PC)
        022206  001367                          BNE     .-20
  3979  022216  032763  000001  000000          BIT     #DRDY,CS(R3)    ;COME BACK
  3980  022224  001223                          BNE     14$             ;
  3981  022226  005302                          DEC     R2
  3982  022230                                  BREAK                   ;INITIATE PROGRAM CALL TO SUPERVISOR
        022230  104422                          TRAP    C$BRK
  3983  022232  001343                          BNE     18$
  3984  022234  012737  003573  002246          MOV     #NOLOAD,WHY     ;NO READY DROP DRIVE
  3985  022242  000137  021500                  JMP     91$
  3986
  3987  022246  012763  000210  000000  GHDR:   MOV     #CRDY!RDHDR,CS(R3)
  3988  022254  056463  000106  000000          BIS     DRSEL(R4),CS(R3)
  3989  022262  042763  000200  000000          BIC     #200,CS(R3)
  3990  022270  004537  024270                  JSR     R5,WTRDY
  3991  022274  016301  000006                  MOV     MP(R3),R1
  3992  022300  043701  002272                  BIC     SMSK,R1
  3993  022304  010164  000124                  MOV     R1,PRPOS(R4)
  3994  022310  012764  004353  000052          MOV     #MTDRV,RTYPE(R4)  ;SETUP DRIVE ERROR
  3995  022316  000205                          RTS     R5
  3996
  3997                                  .SBTTL  BUFFER GENERATION ROUTINE FOR THE 'WRITE' FUNCTION
  3998  022320                          STARS
                                        ;;*****************************************************************
  3999                                  ;WRBUF -- ROUTINE TO WRITE A BUFFER INTO MEMORY.  USES WORD COUNT AND BUS
  4000                                  ;       ADDRESS FROM DRIVE BUFFER (R4).  WILL WRITE RANDOM FROM ONE OF
  4001                                  ;       8 PATTERNS.  USED BY WRITE FUNCTION AND WRPACK ROUTINE.
  4002  022320                          STARS
                                        ;;*****************************************************************
  4003
  4004  022320  005737  002306  WRBUF:  TST     REGEN           ;REBUILD THE DATA BUFFER?
  4005  022324  001507                  BEQ     9$              ;NO --EXIT
  4006  022326  010346                  MOV     R3,-(SP)        ;SAVE REGISTERS
  4007  022330  010246                  MOV     R2,-(SP)
  4008  022332  010146                  MOV     R1,-(SP)
  4009  022334  010046                  MOV     R0,-(SP)
  4010  022336  016402  000042          MOV     BMP(R4),R2      ;R2 HAS TOTAL WORDS TO SET UP FOR
  4011  022342  005402                  NEG     R2              ;POSITIVE NUMBER
  4012  022344  017401  000110          MOV     @RBA(R4),R1     ;WHERE BUFFER IS
  4013  022350  020227  000200  2$:     CMP     R2,#128.        ;MORE THAN 128 WORDS
  4014  022354  002015                  BGE     4$              ;YES. BRANCH
```

BUFFER GENERATION ROUTINE FOR THE 'WRITE' FUNCTION

```
4015 022356  020227  000003                    CMP     R2,#3           ;GREATER THAN THREE WORDS
4016 022362  002005                            BGE     3$              ;YES, BRANCH
4017 022364  062702  000003                    ADD     #3,R2           ;ADD 3
4018 022370  162764  000003  000042            SUB     #3,BMP(R4)      ;WC UP BY 3
4019 022376  010221                    3$:     MOV     R2,(R1)+        ;STORE WC
4020 022400  005302                            DEC     R2              ;ACCOUNT FOR WC
4021 022402  010237  002362                    MOV     R2,TEMP6        ;LOAD DOWN COUNTER
4022 022406  000405                            BR      5$
4023 022410  012737  000177  002362    4$:     MOV     #127.,TEMP6     ;LOAD DOWN COUNTER
4024 022416  012721  000200                    MOV     #128.,(R1)+
4025 022422  005737  010654            5$:     TST     T.RAN           ;RANDOM SELECT OF PATTERNS
4026 022426  001003                            BNE     55$             ;YEA
4027 022430  013703  010656                    MOV     T.PAT,R3        ;NO GET PATTERN OPERATOR
4028 022434  000406                            BR      56$             ;WANTS TO USE
4029 022436  004537  024454            55$:    JSR     R5,RAND         ;GET RANDOM # FOR PATTERN
4030 022442  013703  002262                    MOV     LONUM,R3        ;GET RANDOM PATTERN
4031 022446  042703  177770                    BIC     #177770,R3      ;0,7
4032 022452  006303                    56$:    ASL     R3              ;WORD OFFSET
4033 022454  062703  027734                    ADD     #PATLST,R3      ;GET PATTERN LIST
4034 022460  011303                            MOV     (R3),R3         ;GET LIST ADDRESS
4035 022462  010337  002364                    MOV     R3,TEMP7        ;STOR FOR RECALL
4036 022466  010321                            MOV     R3,(R1)+        ;LOAD IT
4037 022470  005337  002362                    DEC     TEMP6           ;ACCOUNT FOR IT
4038 022474  013703  002364            6$:     MOV     TEMP7,R3        ;PATTERN START
4039 022500  012737  000020  002366            MOV     #16.,TEMP8      ;16 ENTRIES
4040 022506  012321                    7$:     MOV     (R3)+,(R1)+     ;STORE PATTERN
4041 022510  005337  002362                    DEC     TEMP6           ;DOWN COUNT
4042 022514  001404                            BEQ     8$              ;DONE?
4043 022516  005337  002366                    DEC     TEMP8           ;DONE WITH PATTERN
4044 022522  001371                            BNE     7$              ;NO, GO BACK
4045 022524  000763                            BR      6$              ;RESTART PATTERN
4046 022526  162702  000200            8$:     SUB     #128.,R2        ;ANOTHER SECTOR TO USE
4047 022532  003306                            BGT     2$              ;YES GO BACK
4048 022534  012600                            MOV     (SP)+,R0        ;RESTORE REGISTERS
4049 022536  012601                            MOV     (SP)+,R1
4050 022540  012602                            MOV     (SP)+,R2
4051 022542  012603                            MOV     (SP)+,R3
4052 022544  000205                    9$:     RTS     R5
4053
4054                                    .SBTTL  RETRY LIMIT ROUTINE
4055
4056                                    ;RETRY BUMP, TWO RETURNS - CALL +2 - RETRY EXCEEDED
4057                                    ;                          CALL +4 - CONTINUE RETRY
4058
4059 022546  026437  000036  010576    RCNT:   CMP     RETRY(R4),LIMIT ;LIMIT REACHED?
4060 022554  001403                            BEQ     1$              ;YES TAKE FIRST RETURN
4061 022556  005264  000036                    INC     RETRY(R4)       ;ACCOUNT FOR RETRY
4062 022562  005725                            TST     (R5)+           ;NEXT RETURN
4063 022564  000205                    1$:     RTS     R5              ;RETURN
4064
4065                                    .SBTTL  LIST OF FUNCTION ROUTINES
4066
4067                                    ;WE GO THRU THIS LIST WHEN CALLED IN "GETFNC"
4068                                    ;LIST IS IN NUMERICAL ORDER 1-6
4069
4070 022566  000000                    LIST:   .WORD   0
4071 022570  015354                            SKWRT           ;SEEK - WRITE DATA - WRITE CHECK
```

LIST OF FUNCTION ROUTINES

```
4072 022572  015410                    SKRD            ;SEEK - READ DATA
4073 022574  015550                    SKRH            ;SEEK   READ HDR   READ W/NO HDR CMP   GET STATUS
4074 022576  015354                    SKWRT           ;SEEK - WRITE DATA   WRITE CHECK
4075 022600  015410                    SKRD            ;SEEK - READ DATA
4076 022602  015434                    SKRDRD          ;SEEK - READ DATA   READ DATA
4077
4078                                .SBTTL  BAD SECTOR FILE ROUTINE
4079 022604                         STARS
                                    ;;*****************************************************************
4080                                ;RDBDSC -- ROUTINE TO RECOVER BAD SECTOR FILE AND SAVE IT FOR
4081                                ;COMPARISON UPON ERROR ON READS/WRITES & FOR THE SEEK FUNCTION. WE
4082                                ;WILL ONLY RESERVE SPACE FOR 16 BAD SECTORS PER DRIVE AND 1 ENTRY FOR
4083                                ;THE BAD SECTOR FILE AREA POINTER - LAST TRACK ON THE CARTRIDGE.
4084                                ;WE WILL ISSUE A DRIVE RESET FIRST, READ HEADER, POSITION TO THE LAST
4085                                ;TRACK (CYLINDER 255. OR 511., SURFACE 1) AND READ IN THE FIRST SECTOR
4086                                ;FOR FACTORY BAD, AND THE 20TH FOR FIELD BAD SECTORS.  R4 WILL CONTAIN
4087                                ;THE BUFFER POINTER TO THE DRIVE WE WANT TO READ.
4088                                ;
4089                                ;CALL   JSR     R5,RDBDSC               ;GET THE BAD SECT FILE ENTRYS
4090                                ;
4091                                ;THE BAD SECTOR FILE (BOTH FACTORY AND FIELD) LOOKS LIKE THIS:
4092                                ;
4093                                ;       SERIAL NUMBER LOW 5 DIGITS (OCTAL SERIAL NUMBER)
4094                                ;       SERIAL NUMBER HIGH 5 DIGITS
4095                                ;
4096                                ;       0'S
4097                                ;       0'S
4098                                ;
4099                                ;       ENTRY - CYLINDER # FROM 0 TO 1777 MAX (RL02) OR 777 (RL01)
4100                                ;       ENTRY - HEAD & SECTOR NUMBER
4101                                ;
4102                                ;       ENTRY - CYL
4103                                ;       ENTRY - HEAD & SECTOR
4104                                ;
4105                                ;       -1 ...END OF ENTRYS
4106                                ;       -1 ...TO WORD 256. (END OF SECOND SECTOR IN PAIR)
4107 022604                         STARS
                                    ;;*****************************************************************
4108
4109 022604  010046                RDBDSC: MOV     R0,-(SP)        ;SAVE REGISTERS
4110 022606  010146                        MOV     R1,-(SP)        ;
4111 022610  010246                        MOV     R2,-(SP)
4112 022612  010346                        MOV     R3,-(SP)
4113 022614  004537  024376        21$:    JSR     R5,ISDRST       ;ISSUE A DRIVE RESET
4114 022620  012764  000010  000044        MOV     #RDHDR,FUNC(R4) ;READ HEADER TO FIND POSITION
4115 022626  004537  016524                JSR     R5,LDFUNC       ;ON  DISK
4116 022632  004537  024270                JSR     R5,WTRDY
4117
4118 022636  016300  000006                MOV     MP(R3),R0       ;GET HEADER AND CALCULATE
4119 022642  022764  000001  000120        CMP     #1,TDR(R4)      ;RL02 TYPE DRIVE?
4120 022650  001005                        BNE     23$             ;JUMP IF RL02
4121 022652  043700  002264                BIC     CYLMSK,R0       ;HERE FOR RL01
4122 022656  012701  077600                MOV     #77600,R1
4123 022662  000404                        BR      25$
4124 022664  043700  002270        23$:    BIC     CMSK,R0         ;HERE FOR RL02
4125 022670  012701  177600                MOV     #177600,R1
4126 022674  160001                25$:    SUB     R0,R1
```

BAD SECTOR FILE ROUTINE

```
4127 022676  010164  000040                MOV     R1,BDA(R4)
4128 022702  052764  000025  000040        BIS     #SKMS!SIGN!MK,BDA(R4)
4129 022710  012764  000006  000044        MOV     #SEEK,FUNC(R4)
4130 022716  004537  016524                JSR     R5,LDFUNC       ;SEEK TO THE BAD SECTOR FILE AREA
4131 022722  004537  024270                JSR     R5,WTRDY        ;WAIT FOR DRIVE READY
4132 022726  012764  000010  000044        MOV     #RDHDR,FUNC(R4)
4133 022734  004537  016524                JSR     R5,LDFUNC       ;READ A HEADER ON THE BSF
4134 022740  004537  024270                JSR     R5,WTRDY        ;WAIT FOR DRIVE READY
4135 022744  016300  000006                MOV     MP(R3),R0       ;GET THE HEADER WORD READ
4136 022750  042700  000077                BIC     #77,R0          ;CLEAR SECTOR NUMBER READ
4137 022754  022764  000001  000120        CMP     #1,TDR(R4)      ;DRIVE = RL01?
4138 022762  001007                        BNE     300$            ;NO - MUST BE AN RL02
4139 022764  022700  077700                CMP     #77700,R0       ;YES - ON BSF AREA?
4140 022770  001311                        BNE     21$             ;NO - SEEK AGAIN
4141 022772  012764  077700  000040        MOV     #77700,BDA(R4)  ;SAVE THIS HEADER FOR READ COMMAND
4142 023000  000406                        BR      555$
4143 023002  022700  177700        300$:   CMP     #177700,R0      ;RL02 BSF AREA?
4144 023006  001302                        BNE     21$             ;NO - SEEK AGAIN
4145 023010  012764  177700  000040        MOV     #177700,BDA(R4) ;YES - SAVE FOR THE READ COMMAND
4146 023016  012764  177400  000042  555$: MOV     #-256.,BMP(R4)  ;SETUP FOR A 2 SECTOR READ IN BSF
4147 023024  012764  000014  000044        MOV     #READ,FUNC(R4)  ;GET THE READ FUNCTION #
4148
4149 023032  005037  002354                CLR     TEMP3           ;MANUFACTURING/FIELD FILE SWITCH
4150 023036  012737  003720  002246        MOV     #HWSEC,WHY      ;START WITH MANUFACTURING BAD
4151 023044  016402  000112                MOV     BSECPT(R4),R2   ;INITIALIZE LIST TO ALL 1'S
4152 023050  012700  000021                MOV     #17.,R0         ;SIXTEEN ENTRIES + 1 FOR BSF POINTER
4153 023054  012722  177777        11$:    MOV     #-1,(R2)+       ;INIT STORAGE TO -1'S
4154 023060  005300                        DEC     R0              ;DONE?
4155 023062  001374                        BNE     11$             ;NO - DO THE NEXT ONE
4156
4157 023064  016402  000112                MOV     BSECPT(R4),R2   ;GET POINTER TO LIST TO STORE BSF ENTRYS
4158 023070  016422  000040                MOV     BDA(R4),(R2)+   ;SAVE 1ST ENTRY AS BSF POINTER
4159 023074  012700  000020                MOV     #16.,R0         ;SIXTEEN ENTRIES
4160 023100  004537  016524          4$:   JSR     R5,LDFUNC       ;READ THE BSF SECTOR PAIR
4161 023104  004537  024270                JSR     R5,WTRDY        ;WAIT FOR DRIVE READY
4162
4163 023110  005774  000104                TST     @DCS(R4)        ;WAS THE READ GOOD?
4164 023114  100042                        BPL     3$              ;YES
4165
4166 023116  004537  024376                JSR     R5,ISDRST       ;NO - ISSUE A DRIVE RESET
4167 023122  062764  000004  000040        ADD     #4,BDA(R4)      ;POINT TO NEXT SECTOR
4168 023130  005737  002354                TST     TEMP3           ;MANUFACTURING OR FIELD BAD
4169 023134  001414                        BEQ     5$              ;MANUFACTURING = 0
4170 023136  012737  003740  002246        MOV     #SWSEC,WHY      ;FIELD BAD
4171 023144  022764  000001  000120        CMP     #1,TDR(R4)      ;DRIVE = RL01?
4172 023152  001011                        BNE     400$            ;NO - MUST BE RL02
4173 023154  022764  077750  000040        CMP     #77750,BDA(R4)  ;YES - AT END OF FIELD FILE?
4174 023162  001346                        BNE     4$              ;NO - CONTINUE
4175 023164  000516                        BR      6$              ;DROP DRIVE AND EXIT
4176
4177 023166  026427  000040  077724  5$:   CMP     BDA(R4),#77724  ;AT END OF MANUFACTURING BAD
4178 023174  000410                        BR      55$             ;SEE IF DONE
4179 023176  022764  177750  000040  400$: CMP     #177750,BDA(R4) ;AT END OF FIELD BAD FOR RL02
4180 023204  001335                        BNE     4$              ;NO GO BACK FOR NEXT
4181 023206  000505                        BR      6$              ;DROP THE DRIVE AND EXIT
4182 023210  026427  000040  177724        CMP     BDA(R4),#177724 ;AT END OF MANUFACTURING BAD?
4183 023216  001330                55$:    BNE     4$              ;BR IF NOT DONE
```

BAD SECTOR FILE ROUTINE

```
4184 023220  000500                          BR      6$              ;YES - REPORT ERROR AND EXIT
4185
4186 023222  017401  000110          3$:     MOV     @BBA(R4),R1     ;START OF BSF ENTRY LIST
4187 023226  012164  000100                  MOV     (R1)+,SERNM1(R4)  ;GET LOW PART OF SERIAL #
4188 023232  012164  000102                  MOV     (R1)+,SERNM2(R4)  ;GET HIGH PART OF SERIAL #
4189 023236  022121                          CMP     (R1)+,(R1)+     ;SKIP PAST JUNK
4190 023240  012137  002350          1$:     MOV     (R1)+,TEMP1     ;GET CYLINDER
4191 023244  100444                          BMI     2$              ;END OF THE ENTRYS?
4192 023246  012137  002352                  MOV     (R1)+,TEMP2     ;NO - GET HEAD (0 OR 1) & SECTOR NUMBER
4193 023252  000337  002350                  SWAB    TEMP1           ;PUT CYLINDER IN HIGH BYTE
4194 023256  000241                          CLC
4195 023260  006037  002350                  ROR     TEMP1
4196 023264  103003                          BCC     111$
4197 023266  052737  100000  002350          BIS     #BIT15,TEMP1
4198 023274  013712  002350          111$:   MOV     TEMP1,(R2)      ;STORE THE CYLINDER PART
4199 023300  013737  002352  002350          MOV     TEMP2,TEMP1     ;GET SECTOR
4200 023306  042737  177700  002350          BIC     #177700,TEMP1   ;LEAVE ONLY SECTOR
4201 023314  053712  002350                  BIS     TEMP1,(R2)      ;SET IN SECTOR BITS
4202 023320  006237  002352                  ASR     TEMP2
4203 023324  006237  002352                  ASR     TEMP2           ;POSITION THE HEAD SELECT BIT
4204 023330  042737  177677  002352          BIC     #177677,TEMP2   ;CLEAR ALL OTHER BITS
4205 023336  053722  002352                  BIS     TEMP2,(R2)+     ;SET IN HEAD
4206 023342  005300                          DEC     R0              ;COUNT THIS ENTRY FROM BSF
4207 023344  001335                          BNE     1$              ;ALLOW MORE ENTRYS?
4208 023346  012737  003667  002246          MOV     #MBDMSC,WHY     ;MORE THAN 16 BAD SECTORS
4209 023354  000422                          BR      6$              ;DROP THE DRIVE & ERROR EXIT
4210
4211 023356  005737  002354          2$:     TST     TEMP3           ;SWITCH TO FIELD BAD OR QUIT
4212 023362  001021                          BNE     7$              ;QUIT, 7$
4213 023364  022764  000001  000120          CMP     #1,TDR(R4)      ;DRIVE = RL01?
4214 023372  001004                          BNE     350$            ;NO - MUST BE AN RL02
4215 023374  012764  077724  000040          MOV     #77724,BDA(R4)  ;YES   POINT TO FIELD BSF 1ST SECTOR
4216 023402  000403                          BR      36$
4217 023404  012764  177724  000040  350$:   MOV     #177724,BDA(R4) ;POINT TO 1ST SECT IN FIELD FILE FOR RL02
4218 023412  012737  000001  002354  36$:    MOV     #1,TEMP3        ;INDICATE NOW DOING FIELD BSF
4219 023420  000627                          BR      4$              ;PROCESS THE FIELD BSF
4220
4221                                 ;HERE TO DROP THE DRIVE IF MORE THAN 16. ENTRYS OR IF CAN'T FIND A BSF
4222
4223 023422  004537  023450          6$:     JSR     R5,DRDRV        ;DROP THE DRIVE
4224
4225                                 ;HERE TO PUT HEADS 'HOME' AND TO EXIT
4226
4227 023426  004537  025620          7$:     JSR     R5,HOMOME       ;BRINGS HEADS HOME
4228 023432  012603                          MOV     (SP)+,R3
4229 023434  012602                          MOV     (SP)+,R2
4230 023436  012601                          MOV     (SP)+,R1
4231 023440  012600                          MOV     (SP)+,R0
4232 023442  000205                          RTS     R5
4233
4234                                 .SBTTL  ROUTINE TO DROP DRIVE
4235 023444                          STARS
                                     ;;*****************************************************************
4236                                 ;DRDRV  - ROUTINE TO DROP A DRIVE FROM RUNNING
4237                                 ;       R4 HAS BUFFER POINTER OF DRIVE TO DROP
4238                                 ;       WE CLEAR BIT IN "DRUT", NOT "DRPRS"
4239 023444                          STARS
```

ROUTINE TO DROP DRIVE

```
                                        ;;*****************************************************************
4240
4241 023444  005237  002474      ODRDRV: INC     OPCALL
4242 023450  010146              DRDRV:  MOV     R1,-(SP)
4243 023452  010246                      MOV     R2,-(SP)        ;SAVE REGISTERS
4244 023454  010346                      MOV     R3,-(SP)
4245 023456  005237  002476              INC     INCALL
4246 023462  005003                      CLR     R3
4247 023464  012702  030362              MOV     #DRBUF,R2       ;START OF DRIVE BUFFERS
4248 023470  012701  000001              MOV     #1,R1           ;MASK
4249 023474  020402              1$:     CMP     R4,R2           ;IS THIS THE DRIVE?
4250 023476  001405                      BEQ     2$              ;YES GO DROP IT
4251 023500  005203                      INC     R3
4252 023502  006301                      ASL     R1              ;NO SHIFT MASK
4253 023504  062702  000126              ADD     #PRPOS*2,R2     ;NEXT BUFFER
4254 023510  000771                      BR      1$              ;GO BACK
4255
4256 023512  005737  002474      2$:     TST     OPCALL          ;CALLED VIA OPERATOR?
4257 023516  001002                      BNE     6$              ;YES   SKIP CODE
4258 023520                              DODU    R3              ;NO - CALLED BY DIAGNOSTIC
     023520  010300                      MOV     R3,R0
     023522  104451                      TRAP    C$DODU
4259 023524  005037  002476      6$:     CLR     INCALL
4260 023530  005037  002474              CLR     OPCALL
4261 023534  113764  002416  000070      MOVB    HOUR,DPHOUR(R4) ;TIME AT WHICH IT WAS DROPPED
4262 023542  113764  002414  000071      MOVB    MINUTE,DPMIN(R4)  ;HOUR/MINUTE
4263 023550  001002                      BNE     3$              ;IF MINUTE 0,
4264 023552  105264  000071              INCB    DPMIN(R4)       ;MAKE 1.
4265 023556  140137  002252      3$:     BICB    R1,DRUT         ;CLEAR THE DRIVE FROM BIT MAP
4266 023562                              PRINTF  #FMT14C         ;PRINT A <CR> & <LF>
     023562  012746  007501              MOV     #FMT14C,-(SP)
     023566  012746  000001              MOV     #1,-(SP)
     023572  010600                      MOV     SP,R0
     023574  104417                      TRAP    C$PNTF
     023576  062706  000004              ADD     #4,SP
4267 023602  004737  006200              JSR     PC,LINE2
4268 023606                              PRINTF  #FMT7,#DROP,WHY
     023606  013746  002246              MOV     WHY,-(SP)
     023612  012746  004276              MOV     #DROP,-(SP)
     023616  012746  007141              MOV     #FMT7,-(SP)
     023622  012746  000003              MOV     #3,-(SP)
     023626  010600                      MOV     SP,R0
     023630  104417                      TRAP    C$PNTF
     023632  062706  000010              ADD     #10,SP
4269 023636                              PRINTF  #FMTS1
     023636  012746  007732              MOV     #FMTS1,-(SP)
     023642  012746  000001              MOV     #1,-(SP)
     023646  010600                      MOV     SP,R0
     023650  104417                      TRAP    C$PNTF
     023652  062706  000004              ADD     #4,SP
4270
4271 023656  004737  013772              JSR     PC,REPORT
4272
4273 023662  012603                      MOV     (SP)+,R3
4274 023664  012602                      MOV     (SP)+,R2        ;RESTORE REGISTERS
4275 023666  012601                      MOV     (SP)+,R1
4276
```

ROUTINE TO DROP DRIVE

```
4277 023670  000205                        RTS     R5
4278
4279                                .SBTTL  ROUTINE TO CHECK DATA
4280
4281                                ;ROUTINE TO CHECK DATA ON READ
4282
4283 023672  005037  002306        CKDATA: CLR     REGEN           ;CLEAR THE REGENERATE DATA FLAG
4284 023676  005737  010612                TST     CMRD            ;DO WE WANT TO CHECK ANY?
4285 023702  001001                        BNE     10$             ;YES - SEE IF FORCED EXIT
4286 023704  000205                        RTS     R5              ;NO - EXIT NOW
4287 023706  005737  002310        10$:    TST     KILLDC          ;INHIBIT FLAG SET?
4288 023712  001401                        BEQ     97$             ;NOPE   OK TO PROCEED
4289 023714  000205                        RTS     R5              ;NO, EXIT
4290
4291                                ;97$:   SETPRI  #340                                    ;JSD REV A
4292 023716                         97$:    SETPRI  #300                                    ;JSD REV A
     023716  012700  000300                MOV     #300,R0
     023722  104441                        TRAP    C$SPRI
4293 023724  017402  000110                MOV     @BBA(R4),R2     ;BUFFER START
4294 023730  016437  000042  002350        MOV     BMP(R4),TEMP1   ;WORDS READ IN
4295 023736  005437  002350                NEG     TEMP1           ;MAKE POSITIVE
4296 023742  013737  010614  002352        MOV     DELMT,TEMP2     ;# ERRORS TO BE PRINTED
4297 023750  005037  002344                CLR     DECNT           ;INIT ERROR COUNT
4298 023754  013737  010612  002354        MOV     CMRD,TEMP3      ;# WORDS TO BE COMPARED
4299 023762  012737  000176  002346  96$:  MOV     #126.,TEMP0     ;126 WORDS
4300 023770  012201                        MOV     (R2)+,R1        ;NON-ZERO WORDS
4301 023772  005337  002350                DEC     TEMP1
4302 023776  001522                        BEQ     CEND
4303 024000  005301                        DEC     R1
4304 024002  012237  002356                MOV     (R2)+,TEMP4     ;PATTERN ADDRESS
4305
4306                                ;MAKE SURE PATTERN ADDRESS IS LEGAL
4307
4308 024006  012700  027734                MOV     #PATLST,R0      ;GET LIST OF PATTERNS
4309 024012  012703  000010                MOV     #8.,R3          ;ONLY EIGHT
4310 024016  022037  002356        98$:    CMP     (R0)+,TEMP4     ;FOUND IT YET
4311 024022  001414                        BEQ     99$             ;YES, CONTINUE
4312 024024  005303                        DEC     R3              ;NO, EXHAUST LIST YET
4313 024026  001373                        BNE     98$             ;NO, GO BACK
4314
4315 024030  005237  002306                INC     REGEN           ;SET THE DATA REGENERATE FLAG
4316 024034  024242                        CMP     -(R2),-(R2)
4317 024036                                ERRHRD  180.,NOREV,ERR12
     024036  104456                        TRAP    C$ERHRD
     024040  000264                        .WORD   180
     024042  003631                        .WORD   NOREV
     024044  005716                        .WORD   ERR12
4318 024046  004537  026736                JSR     R5,STDMP
4319 024052  000205                        RTS     R5
4320
4321 024054  005301                99$:    DEC     R1              ;ACCOUNT FOR PATTERN ADDRESS
4322 024056  013703  002356                MOV     TEMP4,R3        ;GET ADDRESS
4323 024062  005337  002350                DEC     TEMP1           ;ACCOUNT ONCE AGAIN
4324 024066  012737  000020  002360        MOV     #16.,TEMP5      ;16 ENTRIES TO PATTERN
4325 024074  005737  002350          1$:   TST     TEMP1           ;ANY WORDS READIN LEFT?
4326 024100  001461                        BEQ     CEND            ;NO, GO TO END
4327 024102  005737  002354                TST     TEMP3           ;HAVE WE EXHAUSTED COMPARE LIMIT?
```

```
ROUTINE TO CHECK DATA

 4328 024106  001456                        BEQ     CEND            ;YES GO TO END
 4329 024110  005701                        TST     R1              ;WE CHECKING PATTERN OR ZERO FILL?
 4330 024112  001416                        BEQ     3$              ;ZERO FILL SKIP
 4331 024114  005301                        DEC     R1              ;PATTERN
 4332 024116  005737  002360                TST     TEMP5           ;WITHIN PATTERN
 4333 024122  001005                        BNE     2$              ;YES SKIP
 4334 024124  013703  002356                MOV     TEMP4,R3        ;NO, START OVER
 4335 024130  012737  000020  002360        MOV     #16.,TEMP5      ;16 ENTRIES
 4336 024136  012337  002402          2$:   MOV     (R3)+,GDDAT     ;GET PATTERN
 4337 024142  005337  002360                DEC     TEMP5           ;DOWN COUNT
 4338 024146  000402                        BR      4$
 4339 024150  005037  002402          3$:   CLR     GDDAT           ;ZERO FILL
 4340 024154  023712  002402          4$:   CMP     GDDAT,(R2)      ;CORRECT DATA
 4341 024160  001417                        BEQ     5$              ;YES YES NEXT
 4342 024162  005237  002306                INC     REGEN           ;NO - SET REGENERATE FLAG FOR WRT OPERATION
 4343 024166  005237  002344                INC     DECNT           ;COUNT THE DATA ERROR
 4344 024172  005264  000074                INC     DATCER(R4)      ;COUNT ERROR FOR THIS DRIVE
 4345 024176  005737  002352                TST     TEMP2           ;DO WE WANT TO PRINT IT
 4346 024202  001406                        BEQ     5$              ;NO,SKIP
 4347
 4348 024204                                ERRHRD  185.,MDCER,ERR8
      024204  104456                        TRAP    C$ERHRD
      024206  000271                        .WORD   185
      024210  003235                        .WORD   MDCER
      024212  005462                        .WORD   ERR8
 4349 024214  005337  002352                DEC     TEMP2           ;ACCOUNT FOR PRINT
 4350
 4351 024220  005337  002350          5$:   DEC     TEMP1           ;WORDS READ IN
 4352 024224  001407                        BEQ     CEND
 4353 024226  005722                        TST     (R2)+           ;NEXT WORD
 4354 024230  005337  002346                DEC     TEMP0
 4355 024234  001652                        BEQ     96$
 4356 024236  005337  002354                DEC     TEMP3           ;WORDS TO CHECK
 4357 024242  000714                        BR      1$
 4358
 4359 024244  005737  002344          CEND: TST     DECNT           ;DO WE WANT TO PRINT SUMMARY
 4360 024250  001406                        BEQ     1$              ;NO,EXIT
 4361 024252  005464  000042                NEG     BMP(R4)         ;MAKE POSITIVE WORD COUNT
 4362 024256                                ERRHRD  190.,MDCER,ERR6 ;DATA ERROR SUMMARY
      024256  104456                        TRAP    C$ERHRD
      024260  000276                        .WORD   190
      024262  003235                        .WORD   MDCER
      024264  005364                        .WORD   ERR6
 4363
 4364 024266  000205                  1$:   RTS     R5
 4365
 4366                                       .SBTTL  ROUTINE TO WAIT FOR CONTROLLER READY
 4367
 4368                                       ;
 4369                                       ;ROUTINE TO WAIT FOR CONTROLLER READY UNDER FLAG
 4370                                       ;MODE.  USED IN INITIALIZE PORTION OF PROGRAM, I.E..
 4371                                       ;GETTING BAD SECTOR FILE, WRITING PACK INITIALLY.
 4372
 4373 024270  010046                  WTRDY: MOV     R0,-(SP)        ;SAVE REGISTERS
 4374 024272  010146                        MOV     R1,-(SP)
 4375 024274  012701  001750                MOV     #1000.,R1       ;WAIT A WHILE
 4376                                 ;1$:   WAITUS  #2.                                      ;JSD REV A
```

ROUTINE TO WAIT FOR CONTROLLER READY

```
4377 024300                          1$:    WAITUS  2.                              ;JSD REV A
     024300  012727  000002                 MOV     #2.,(PC)+
     024304  000000                         .WORD   0
     024306  013727  002116                 MOV     L$DLY,(PC)+
     024312  000000                         .WORD   0
     024314  005367  177772                 DEC      6(PC)
     024320  001375                         BNE     .-4
     024322  005367  177756                 DEC     -22(PC)
     024326  001367                         BNE     .-20
4378 024330  032774  000200  000104        BIT     #CRDY,@DCS(R4)  ;READY SET?
4379 024336  001006                         BNE     2$              ;YES, EXIT
4380 024340  005301                         DEC     R1              ;TIMED OUT?
4381 024342  001356                         BNE     1$              ;NO GO BACK
4382
4383 024344                                 ERRDF   1002.,NOCRDY,ERR12
     024344  104455                         TRAP    C$ERDF
     024346  001752                         .WORD   1002
     024350  002654                         .WORD   NOCRDY
     024352  005716                         .WORD   ERR12
4384
4385 024354  012601                 2$:    MOV     (SP)+,R1        ;RESTORE REGISTERS
4386 024356  012600                         MOV     (SP)+,R0
4387 024360  000205                         RTS     R5
4388
4389                                 .SBTTL  GET STATUS/DRIVE RESET ROUTINE
4390
4391                                 ;ROUTINE TO ISSUE DRIVE RESET
4392                                 ;ALSO GET STATUS, R1 HAS STATUS IF GS
4393                                 ;USES R3, DOES NOT SAVE IT
4394
4395 024362  016403  000104         GETDST: MOV     DCS(R4),R3
4396 024366  012763  000003  000004         MOV     #GSBIT,DA(R3)
4397 024374  000405                         BR      CSTUFF
4398 024376  016403  000104         ISDRST: MOV     DCS(R4),R3
4399 024402  012763  00001*  000004         MOV     #DRST,DA(R3)
4400 024410  012763  000204  000000 CSTUFF: MOV     #CRDY!GSTAT,CS(R3)
4401 024416  056463  000106  000000         BIS     DRSEL(R4),CS(R3)
4402 024424  042763  000200  000000         BIC     #CRDY,CS(R3)
4403 024432  004537  024270                 JSR     R5,WTRDY
4404 024436  022763  000013  000004         CMP     #DRST,DA(R3)
4405 024444  001402                         BEQ     1$
4406 024446  016301  000006                 MOV     MP(R3),R1
4407 024452  000205                 1$:     RTS     R5
4408
4409 024454                                 STARS
                                            ;;*****************************************************************
4410                                        ;RAND --  ROUTINE TO GENERATE A RANDOM NUMBER
4411 024454                                 STARS
                                            ;;*****************************************************************
4412
4413 024454  010146                 RAND:   MOV     R1,-(SP)
4414 024456  010246                         MOV     R2,-(SP)
4415 024460  010346                         MOV     R3,-(SP)
4416
4417 024462  013703  002262                 MOV     LONUM,R3
4418 024466  013701  002260                 MOV     HINUM,R1
4419 024472  012702  177771                 MOV     #-7,R2
```

```
4420 024476  006303                    1$:     ASL     R3
4421 024500  006101                            ROL     R1
4422 024502  005202                            INC     R2
4423 024504  001374                            BNE     1$
4424 024506  063703  002262                    ADD     LONUM,R3
4425 024512  005501                            ADC     R1
4426 024514  063701  002260                    ADD     HINUM,R1
4427 024520  062703  001057                    ADD     #1057,R3
4428 024524  005501                            ADC     R1
4429 024526  062701  047401                    ADD     #47401,R1
4430 024532  010337  002260                    MOV     R3,HINUM
4431 024536  010137  002262                    MOV     R1,LONUM
4432 024542  012603                            MOV     (SP)+,R3
4433 024544  012602                            MOV     (SP)+,R2
4434 024546  012601                            MOV     (SP)+,R1
4435 024550  000205                            RTS     R5
4436
4437                                    .SBTTL  ROUTINE TO WRITE PACKS INITIALLY
4438 024552                             STARS
                                        ;;*****************************************************************
4439                                    ;WRPACK -- ROUTINE TO WRITE PACK WITH PATTERN, ALL TRACKS WILL BE
4440                                    ;       WRITTEN (EXCEPT BAD SECTOR TRACK)
4441                                    ;       FORMAT IS # OF WORDS (WORD 1), PATTERN ADDRESS (WORD 2)
4442                                    ;       PATTERN (WORDS 3 - 128)
4443                                    ;       WE WILL ATTEMPT TO WRITE MULTIPLE SECTORS AT A TIME
4444                                    ;       (MINIMUM 10 SECTORS) IF AN ERROR OCCURS WE WILL THEN
4445                                    ;       WRITE INDIVIDUAL SECTORS FOR THAT TRACK.  WE DO WRITES,
4446                                    ;       READS AND INCORE COMPARISONS TO VERIFY.
4447                                    ;
4448                                    ;       CALL:   JSR     R5,WRPACK       ;WRITE THE PACK SELECTED
4449 024552                             STARS
                                        ;;*****************************************************************
4450
4451 024552  010046                    WRPACK: MOV     R0,-(SP)        ;SAVE REGISTERS
4452 024554  010146                            MOV     R1,-(SP)
4453 024556  010246                            MOV     R2,-(SP)
4454 024560  010346                            MOV     R3,-(SP)
4455 024562  016446  000110                    MOV     BBA(R4),-(SP)
4456 024566  005764  000122                    TST     WRIPG(R4)       ;SEE IF WRITE IN PROGRESS
4457 024572  001016                            BNE     1$              ;JUMP IF DON'T WANT MESSAGE ON RECOVERY
4458 024574                                    PRINTF  #FMT18,#MSWRPK  ;MSG. "WRITING PACK"
     024574  012746  004400                    MOV     #MSWRPK,-(SP)
     024600  012746  007666                    MOV     #FMT18,-(SP)
     024604  012746  000002                    MOV     #2,-(SP)
     024610  010600                            MOV     SP,R0
     024612  104417                            TRAP    C$PNTF
     024614  062706  000006                    ADD     #6,SP
4459 024620  000240                            NOP
4460 024622  000240                            NOP
4461 024624  004737  006200                    JSR     PC,LINE2        ;PRINT TIME-RCLS & DRIVE ID
4462 024630  004537  025620            1$:     JSR     R5,HDHOME       ;HEADS HOME
4463
4464                                    ;
4465                                    ;NOW ACTUALLY WRITE DATA OUT ON PACK, WILL NOT WRITE LAST
4466                                    ;TRACK
4467                                    ;
4468
```

ROUTINE TO WRITE PACKS INITIALLY

```
4469 024634  005037  002350                    CLR     TEMP1           ;TEMP1=HEAD
4470 024640  005001                            CLR     R1              ;R1=CYL
4471 024642  022764  000001  000120  CONWR:    CMP     #1,TDR(R4)
4472 024650  001007                            BNE     45$
4473 024652  022701  077600                    CMP     #077600,R1
4474 024656  001023                            BNE     STWRT
4475 024660  005737  002350                    TST     TEMP1
4476 024664  001420                            BEQ     STWRT
4477 024666  000406                            BR      ENDWR
4478 024670  022701  177600          45$:      CMP     #177600,R1
4479 024674  001014                            BNE     STWRT           ;NO GO WRITE TRACK
4480 024676  005737  002350                    TST     TEMP1           ;YES, CHECK IF HEAD = 1?
4481 024702  001411                            BEQ     STWRT           ;HEAD = 0 GO WRITE
4482 024704  004537  025620          ENDWR:    JSR     R5,HDHOME       ;HEADS HOME
4483 024710  012664  000110                    MOV     (SP)+,BBA(R4)
4484 024714  012603                            MOV     (SP)+,R3
4485 024716  012602                            MOV     (SP)+,R2
4486 024720  012601                            MOV     (SP)+,R1
4487 024722  012600                            MOV     (SP)+,R0
4488 024724  000205                            RTS     R5              ;END EXIT
4489
4490                                    ;THIS PORTION WILL WRITE THE PACK USING MULTIPLE SECTORS IF A
4491                                    ;ERROR OCCURS WE WILL GO TO 2$ AND INDIVIDUAL SECTORS.
4492
4493 024726  005002                    STWRT:  CLR     R2              ;INITIAL SECTOR 0
4494 024730  012764  002436  000110            MOV     #BUF1,BBA(R4)   ;BUFFER START
4495 024736  012764  175400  000042            MOV     #-1280.,BMP(R4) ;10 SECTORS
4496 024744  005237  002306                    INC     REGEN           ;SET THE GENERATE BUFFER FLAG
4497 024750  004537  022320                    JSR     R5,WRBUF        ;WRITE BUFFER INTO MEMORY
4498 024754  010164  000040          201$:     MOV     R1,BDA(R4)      ;SET UP SECTOR
4499 024760  053764  002350  000040            BIS     TEMP1,BDA(R4)
4500 024766  005764  000122                    TST     WRIPG(R4)       ;WRITE IN PROGRESS?
4501 024772  001406                            BEQ     762$            ;NO - JUMP OVER
4502 024774  026464  000124  000040            CMP     PRPOS(R4),BDA(R4) ;YUP - ON CYLINDER NOW?
4503 025002  001402                            BEQ     762$            ;YUP - WRITE THIS AREA
4504 025004  000137  025414                    JMP     952$            ;NO - LOOK AT NEXT AREA ON DRIVE
4505 025010  050264  000040          762$:     BIS     R2,BDA(R4)
4506 025014  012764  002436  000110            MOV     #BUF1,BBA(R4)   ;SET UP TO WRITE
4507 025022  012764  000012  000044            MOV     #WRITE,FUNC(R4) ;WRITE
4508 025030  004537  016524                    JSR     R5,LDFUNC
4509 025034  004537  024270                    JSR     R5,WTRDY        ;WAIT FOR READY
4510 025040  005774  000104                    TST     @DCS(R4)        ;ERROR
4511 025044  100003                            BPL     203$
4512 025046  004537  024376          205$:     JSR     R5,ISDRST
4513 025052  000421                            BR      2$
4514
4515 025054  012764  000002  000044  203$:     MOV     #WRCHK,FUNC(R4)
4516 025062  004537  016524                    JSR     R5,LDFUNC
4517 025066  004537  024270                    JSR     R5,WTRDY
4518 025072  005774  000104                    TST     @DCS(R4)        ;ERROR
4519 025076  100763                            BMI     205$            ;YES GO DO SECTORS INDIVIDUALLY
4520
4521 025100  062702  000012                    ADD     #10.,R2         ;NEXT GROUP
4522 025104  022702  000050                    CMP     #40.,R2         ;DONE?
4523 025110  001321                            BNE     201$            ;NO, GO BACK
4524 025112  000137  025414                    JMP     952$            ;YES NEXT TRACK
4525
```

ROUTINE TO WRITE PACKS INITIALLY

```
                                        ;IF AN ERROR OCCURS THEN WE COME HERE AND DO THE TRACK SECTOR
                                        ;BY SECTOR.
                                
025116  005002                  2$:     CLR     R2              ;R2 = SECTOR
                                
025120  012764  177600  000042          MOV     #-128.,BMP(R4)  ;LOAD WORD COUNT
025126  010164  000040          3$:     MOV     R1,BDA(R4)      ;SETUP DISK ADDRESS
025132  053764  002350  000040          BIS     TEMP1,BDA(R4)
025140  050264  000040                  BIS     R2,BDA(R4)
                                
025144  012764  002436  000110          MOV     #BUF1,BBA(R4)
025152  004537  022320                  JSR     R5,WRBUF        ;WRITE A BUFFER
025156  005037  002244          91$:    CLR     RWCNT           ;CLEAR RETRYS OUT
025162  005037  002344          98$:    CLR     DECNT           ;
025166  012764  000012  000044  96$:    MOV     #WRITE,FUNC(R4) ;WRITE FUNCTION
025174  004537  016524                  JSR     R5,LDFUNC
025200  004537  024270                  JSR     R5,WTRDY        ;WAIT FOR WRITE TO FINISH
                                
025204  005774  000104                  TST     @DCS(R4)        ;ERROR ON WRITE?
025210  100021                          BPL     85$             ;NO, GO READ
                                
025212  016437  000040  002342          MOV     BDA(R4),CHKSEC  ;YES, CHECK IF SECTOR IS IN
025220  004537  027146                  JSR     R5,CKBDSC       ;BAD SECTOR FILE
025224  005737  002340                  TST     HDRFND          ;IF SET, IT WAS
025230  001050                          BNE     802$            ;YES GO TO NEXT SECTOR
                                
025232  005237  002344                  INC     DECNT           ;NO, GIVE IT 3 TRYS TOTAL
025236  023727  002344  000003          CMP     DECNT,#3.       ;IT MAY HAVE BEEN NOISE.
025244  001440                          BEQ     801$            ;BR IF AT RETRY LIMIT - BAD SECTOR
025246  004537  024376                  JSR     R5,ISDRST       ;RESET THE DRIVE & TRY AGAIN
025252  000745                          BR      96$             ;TRY RECOVERY AGAIN
                                
025254  005037  002242          85$:    CLR     RECNT           ;CLEAR RETRY COUNT
025260  012764  000002  000044  80$:    MOV     #WRCHK,FUNC(R4) ;READ/VERIFY THE 1 SECTOR WRITTEN
025266  004537  016524                  JSR     R5,LDFUNC       ;ISSUE A WRITE-CHECK FUNCTION
025272  004537  024270                  JSR     R5,WTRDY        ;WAIT FOR DRIVE READY
                                
025276  005774  000104                  TST     @DCS(R4)        ;ERROR ON READ?
025302  100025                          BPL     95$             ;BR IF OK ... GET THE NEXT SECTOR
                                
025304  016437  000040  002342          MOV     BDA(R4),CHKSEC  ;CHECK IF SECTOR IS
025312  004537  027146                  JSR     R5,CKBDSC       ;A KNOWN BAD SECTOR
025316  005737  002340                  TST     HDRFND          ;IT WAS THEN
025322  001013                          BNE     802$            ;GO TO NEXT SECTOR
                                
025324  005237  002242                  INC     RECNT           ;GIVE IT ANOTHER CHANCE
025330  023727  002242  000020          CMP     RECNT,#16.      ;16 RE-READS BEFORE HARD ERROR
025336  001403                          BEQ     801$            ;REPORT ERROR IF AT RETRY LIMIT
025340  004537  024376                  JSR     R5,ISDRST       ;RESET THE DRIVE
025344  000745                          BR      80$             ;AND RETRY AGAIN
                                
025346  004537  025500          801$:   JSR     R5,INBAD        ;REPORT THE BAD SECTOR
025352  004537  024376          802$:   JSR     R5,ISDRST       ;RESET THE DRIVE FOR THE NEXT OPERATION
                                
025356  062702  000012          95$:    ADD     #10.,R2         ;NEXT SECTOR (OFFSET BY 10)
025362  020227  000047                  CMP     R2,#39.         ;DONE WITH TRACK?
025366  003002                          BGT     951$            ;YES NEXT TRACK
```

ROUTINE TO WRITE PACKS INITIALLY

```
4583 025370  000137  025126                    JMP     3$              ;NO GO BACK FOR NEXT SECTOR
4584 025374                            951$:
4585 025374  005202                            INC     R2              ;NEXT SECTOR
4586 025376  162702  000050                    SUB     #40.,R2         ;DONE WITH TRACK?
4587 025402  020227  000012                    CMP     R2,#10.         ;
4588 025406  001402                            BEQ     952$            ;YES
4589 025410  000137  025126                    JMP     3$              ;NO
4590 025414                            952$:
4591
4592 025414  005737  002350                    TST     TEMP1           ;WHICH SURFACE?
4593 025420  001420                            BEQ     5$              ;TOP (0), BRANCH
4594
4595 025422  005037  002350                    CLR     TEMP1           ;BOTTOM, SWITCH TO TOP WITH
4596 025426  062701  000200                    ADD     #200,R1
4597 025432  012764  000205  000040            MOV     #205,BDA(R4)    ;SEEK, GO IN ALSO
4598 025440  012764  000006  000044  4$:       MOV     #SEEK,FUNC(R4)  ;GO SEEK
4599 025446  004537  016524                    JSR     R5,LDFUNC
4600 025452  004537  024270                    JSR     R5,WTRDY
4601
4602 025456  000137  024642                    JMP     CONWR
4603
4604 025462  012737  000100  002350  5$:       MOV     #HEAD,TEMP1     ;WAS TOP, MAKE BOTTOM.
4605 025470  012764  000021  000040            MOV     #21,BDA(R4)
4606 025476  000760                            BR      4$
4607
4608 025500  010146                    INBAD:  MOV     R1,-(SP)        ;SAVE R1
4609 025502  016403  000104                    MOV     DCS(R4),R3      ;GET THE CSR ADDRESS
4610 025506  016337  000000  002420            MOV     CS(R3),E.CS     ;GET THE ERROR INFO FROM CSR
4611 025514  016337  000002  002422            MOV     BA(R3),E.BA
4612 025522  016337  000004  002424            MOV     DA(R3),E.DA
4613 025530  000240                            NOP
4614 025532  000240                            NOP
4615 025534  004537  024362                    JSR     R5,GETDST       ;GET THE CURRENT DRIVE STATUS
4616 025540  010137  002426                    MOV     R1,E.MP         ;SAVE IT AS "(RLMP)" DATA
4617 025544                                    ERRHRD  199.,NWRTS,ERR12
     025544  104456                            TRAP    C$ERHRD
     025546  000307                            .WORD   199
     025550  002736                            .WORD   NWRTS
     025552  005716                            .WORD   ERR12
4618 025554  005264  000012                    INC     ERRCNT(R4)
4619 025560  005737  010640                    TST     T.DRP           ;ARE WE COUNTING ERRORS
4620 025564  001413                            BEQ     2$              ;NO
4621 025566  026437  000012  010600            CMP     ERRCNT(R4),ERLMT;PAST IT
4622 025574  103407                            BLO     2$              ;NO
4623 025576  012737  003322  002246            MOV     #ERLMTM,WHY
4624 025604  004537  023450                    JSR     R5,DRDRV
4625 025610  012705  024704                    MOV     #ENDWR,R5
4626
4627 025614  012601                    2$:     MOV     (SP)+,R1        ;RESET R1
4628 025616  000205                            RTS     R5
4629
4630                                   .SBTTL  HEADS HOME ROUTINE
4631 025620                            STARS
                                       ;;*****************************************************************
4632                                   ;HOHOME -- ROUTINE TO BRING HEADS OVER TRACK 0
4633 025620                            STARS
                                       ;;*****************************************************************
```

HEADS HOME ROUTINE

```
4634
4635 025620  010046                    HDHOME: MOV     R0, (SP)          ;SAVE R0
4636 025622  012764  000010  000044             MOV     #RDHDR,FUNC(R4) ;READ HEADER
4637 025630  004537  016524                     JSR     R5,LDFUNC         ;GO DO IT.
4638 025634  004537  024270                     JSR     R5,WTRDY
4639
4640 025640  016300  000006                     MOV     MP(R3),R0         ;GET HEADER
4641 025644  042700  000177                     BIC     #177,R0           ;ONLY CYLINDER
4642 025650  010064  000050                     MOV     R0,LSTHDR(R4)     ;SAVE THIS CYL # AS THE LAST POSITION
4643 025654  010064  000040                     MOV     R0,BDA(R4)        ;MOVE IT TO BUFFERED DA
4644 025660  052764  000001  000040             BIS     #MK,BDA(R4)       ;SET MARKER FOR SEEK TO 000
4645 025666  012764  000006  000044             MOV     #SEEK,FUNC(R4)  ;LOAD SEEK
4646 025674  004537  016524                     JSR     R5,LDFUNC         ;SEEK!
4647 025700  004537  024270                     JSR     R5,WTRDY          ;WAIT.
4648 025704  005064  000124                     CLR     PRPOS(R4)         ;SET BUFFER TO HOME CYLINDER (000)
4649 025710  012600                             MOV     (SP)+,R0
4650 025712  000205                             RTS     R5
4651
4652                                    .SBTTL  RANDOM WC AND DA ROUTINE
4653 025714                             STARS
                                        ;;*****************************************************************
4654                                    ;GWCDA -- ROUTINE TO GET RANDOM SECTOR AND WORD COUNT FOR R/W TRANSFER.
4655                                    ;       SECTOR IS CHOSEN BETWEEN MIN/MAX LIMITS, WORD COUNT IS BETWEEN
4656                                    ;       MIN/MAX WORD COUNT. WORD COUNT WILL BE ADJUSTED NOT TO CAUSE
4657                                    ;       TRACK OVERFLOW IF HIGH SECTORS ARE CHOSEN....
4658                                    ;       R4 HAS BUFFER OF DRIVE WE'RE WORKING WITH
4659                                    ;       ON EXIT - BMP(R4) HAS WORD COUNT
4660                                    ;                 - BDA(R4) HAS DISK ADDRESS
4661 025714                             STARS
                                        ;;*****************************************************************
4662
4663 025714  023737  010632  010634  GWCDA:  CMP     T.MXS,T.MNS       ;MIN MAX SECTORS EQUAL
4664 025722  001003                             BNE     99$               ;NO, CALCULATE ONE
4665 025724  013702  010632                     MOV     T.MXS,R2          ;LOAD SECTOR
4666 025730  000421                             BR      5$                ;GO GET WC
4667 025732  004537  024454          99$:       JSR     R5,RAND           ;GET RANDOM # FOR SECTOR
4668 025736  013702  002262                     MOV     LONUM,R2
4669 025742  042702  177700          1$:        BIC     #177700,R2        ;0-77 ONLY
4670 025746  023702  010632                     CMP     T.MXS,R2          ;R2 LOWER THAN MAX
4671 025752  103003                             BHIS    3$                ;BRANCH IF YES
4672 025754  006202                             ASR     R2                ;HALF IT
4673 025756  005202                             INC     R2                ;INC SO NOT 0
4674 025760  000770                             BR      1$
4675 025762  020237  010634          3$:        CMP     R2,T.MNS          ;MIN OKAY
4676 025766  103002                             BHIS    5$
4677 025770  006102                             ROL     R2
4678 025772  000763                             BR      1$
4679
4680                                    ;NOW GET WORD COUNT
4681
4682 025774  005737  010664          5$:        TST     T.STIP            ;RESTRICT THE XFER SIZE?
4683 026000  001003                             BNE     95$               ;BR IF YES
4684 026002  013737  002442  010620             MOV     MAXWC,T.MXB       ;NO - MAKE MAXWC = BIGEST XFER SIZE AVAIL.
4685 026010  023737  002442  010620  95$:       CMP     MAXWC,T.MXB
4686 026016  103021                             BHIS    97$
4687
4688 026020                             PRINTF  #FMT13D,#OVER,T.MXB,MAXWC
```

```
        026020  013746  002442                  MOV     MAXWC, (SP)
        026024  013746  010620                  MOV     T.MXB, (SP)
        026030  012746  003453                  MOV     #OVER,-(SP)
        026034  012746  007436                  MOV     #FMT13D,-(SP)
        026040  012746  000004                  MOV     #4,-(SP)
        026044  010600                          MOV     SP,R0
        026046  104417                          TRAP    C$PNTF
        026050  062706  000012                  ADD     #12,SP
  4689  026054  013737  002442  010620          MOV     MAXWC,T.MXB
  4690
  4691  026062  023737  010620  010642  97$:    CMP     T.MXB,T.MNB     ;MIN MAX EQUAL
  4692  026070  003006                          BGT     6$
  4693  026072  013737  010620  010642          MOV     T.MXB,T.MNB
  4694
  4695  026100  013703  010620                  MOV     T.MXB,R3        ;YES SET WC
  4696  026104  000421                          BR      9$
  4697  026106  004537  024454          6$:     JSR     R5,RAND         ;GET RANDOM WORD COUNT
  4698  026112  013703  002262                  MOV     LONUM,R3
  4699  026116  042703  160000          7$:     BIC     #160000,R3      ;MAX!!!!!
  4700  026122  023703  010620                  CMP     T.MXB,R3
  4701  026126  103003                          BHIS    8$
  4702  026130  006203                          ASR     R3
  4703  026132  005203                          INC     R3
  4704  026134  000770                          BR      7$
  4705  026136  020337  010642          8$:     CMP     R3,T.MNB
  4706  026142  103002                          BHIS    9$
  4707  026144  006103                          ROL     R3
  4708  026146  000763                          BR      7$
  4709
  4710                                  ;NOW WE HAVE SECTOR AND WORD COUNT. CHECK THAT WORD COUNT WILL FIT ON SECTOR
  4711                                  ;IF NOT LOWER SECTOR START
  4712
  4713  026150  012701  000050          9$:     MOV     #40.,R1         ;SETUP FOR FOURTY SECTORS
  4714  026154  005403                          NEG     R3              ;MAKE WORD COUNT NEGATIVE
  4715  026156  010364  000042                  MOV     R3,BMP(R4)      ;LOAD WORD COUNT
  4716  026162  005301                  11$:    DEC     R1              ;DOWN COUNT MINIMUM START SECT NEEDED
  4717  026164  062703  000200                  ADD     #128.,R3        ;ONE SECTOR'S WORTH
  4718  026170  100774                          BMI     11$             ;STILL NEED ANOTHER SECTOR
  4719  026172  020201                          CMP     R2,R1           ;DID RANDOM SECTOR SUFFICE
  4720  026174  101401                          BLOS    12$             ;BRANCH IF SUFFICED
  4721  026176  010102                          MOV     R1,R2           ;NO, THEN MAKE IT FIT
  4722  026200  016464  000124  000040  12$:    MOV     PRPOS(R4),BDA(R4)
  4723  026206  042764  000077  000040          BIC     #77,BDA(R4)
  4724  026214  050264  000040                  BIS     R2,BDA(R4)
  4725  026220  000205                          RTS     R5
  4726
  4727                                  .SBTTL  ROUTINE TO DUMP BUFFER ON DCK
  4728  026222                          STARS
                                        ;;*****************************************************************
  4729                                  ;DMPBUF -- ROUTINE TO DUMP BUFFER ON DCK ERROR, TWO DUMPS ARE POSSIBLE
  4730                                  ;       ONE WHERE WE CAN COMPARE WHAT IT SHOULD BE AND THE OTHER
  4731                                  ;       WHEN WE CAN'T.
  4732  026222                          STARS
                                        ;;*****************************************************************
  4733
  4734  026222  004737  006270          DMPBUF: JSR     PC,LINE3
  4735
```

ROUTINE TO DUMP BUFFER ON DCK

```
4736                                    ;CALCULATE THE STARTING BUS ADDRESS FOR THE COMPARE
4737
4738 026226  012737  000200  002464      MOV     #128.,DWCNT1
4739 026234  016400  000040              MOV     BDA(R4),R0              ;GET STARTING BUS ADDRESS
4740
4741 026240  013701  002424              MOV     E.DA,R1                 ;GET PRESENT DISK ADDRESS
4742 026244  042700  177700              BIC     #177700,R0              ;SAVE SECTOR BITS
4743 026250  042701  177700              BIC     #177700,R1              ;
4744 026254  010002                      MOV     R0,R2                   ;SAVE A COPY
4745 026256  010103                      MOV     R1,R3                   ;SAVE ANOTHER
4746 026260  160203                      SUB     R2,R3                   ;GET DIFF OF SECTORS
4747 026262  005002                      CLR     R2                      ;CALCULATE WORD COUNT
4748 026264  062702  000200      93$:    ADD     #128.,R2                ;ONE SECTORS WORTH
4749 026270  005303                      DEC     R3                      ;DONE
4750 026272  001374                      BNE     93$                     ;NO
4751 026274  016403  000042              MOV     BMP(R4),R3              ;GET WORD COUNT
4752 026300  005403                      NEG     R3                      ;MAKE IT POSITIVE
4753 026302  020203                      CMP     R2,R3                   ;WORKING WITH FULL SECTOR
4754 026304  003005                      BGT     94$                     ;NO, GO CALC PARTIAL SECTOR
4755 026306  013702  002422              MOV     E.BA,R2                 ;PRESENT BUS ADDRESS
4756 026312  162702  000400              SUB     #400,R2                 ;START OF COMPARE
4757 026316  000412                      BR      96$                     ;GO COMPARE BUFFER
4758 026320  160302      94$:    SUB     R3,R2                   ;GET SECTOR DIFF
4759 026322  012700  000200              MOV     #128.,R0
4760 026326  160200                      SUB     R2,R0
4761 026330  010037  002464              MOV     R0,DWCNT1
4762 026334  006300                      ASL     R0
4763 026336  013702  002422              MOV     E.BA,R2
4764 026342  160002                      SUB     R0,R2
4765 026344              96$:    PRINTB  #FMT13,#BUSAD,R2,#CRLDA,CHKSEC
     026344  013746  002342              MOV     CHKSEC,-(SP)
     026350  012746  002577              MOV     #CRLDA,-(SP)
     026354  010246                      MOV     R2,-(SP)
     026356  012746  004132              MOV     #BUSAD,-(SP)
     026362  012746  007421              MOV     #FMT13,-(SP)
     026366  012746  000005              MOV     #5,-(SP)
     026372  010600                      MOV     SP,R0
     026374  104414                      TRAP    C$PNTB
     026376  062706  000014              ADD     #14,SP
4766 026402  012700  027734              MOV     #PATLST,R0              ;CHECK PATTERN LIST
4767 026406  012701  000010              MOV     #8.,R1
4768 026412  022062  000002      1$:     CMP     (R0)+,2(R2)
4769 026416  001415                      BEQ     2$
4770 026420  005301                      DEC     R1
4771 026422  001373                      BNE     1$
4772
4773 026424              3$:     PRINTB  #FMT14,#NOREV
     026424  012746  003631              MOV     #NOREV,-(SP)
     026430  012746  007463              MOV     #FMT14,-(SP)
     026434  012746  000002              MOV     #2,-(SP)
     026440  010600                      MOV     SP,R0
     026442  104414                      TRAP    C$PNTB
     026444  062706  000006              ADD     #6,SP
4774 026450  000532                      BR      STDMP
4775
4776 026452  021227  000200      2$:     CMP     (R2),#128.
4777 026456  101362                      BHI     3$
```

ROUTINE TO DUMP BUFFER ON DCK

```
4778 026460  005037  002344                    CLR     DECNT
4779 026464  013701  010662                    MOV     T.CLT,R1
4780
4781 026470  012237  002346                    MOV     (R2)+,TEMP0      ;NONZERO WORD COUNT
4782 026474  013737  002346  002462            MOV     TEMP0,DWCNT
4783 026502  005437  002462                    NEG     DWCNT
4784 026506  012237  002350                    MOV     (R2)+,TEMP1
4785 026512  162737  000002  002346            SUB     #2,TEMP0
4786 026520  012737  000002  002352            MOV     #2,TEMP2         ;WORD
4787 026526  013703  002350                    MOV     TEMP1,R3         ;PATTERN ADDRESS
4788 026532  012737  000020  002360            MOV     #16.,TEMP5       ;16 ENTRIES
4789 026540  005737  002346          4$:       TST     TEMP0            ;ZERO OR PATTERN
4790 026544  001417                            BEQ     6$               ;ZERO BRANCH
4791 026546  005337  002346                    DEC     TEMP0
4792 026552  005737  002360                    TST     TEMP5            ;WITHIN LIST
4793 026556  001005                            BNE     5$
4794 026560  012737  000020  002360            MOV     #16.,TEMP5
4795 026566  013703  002350                    MOV     TEMP1,R3
4796 026572  012337  002402          5$:       MOV     (R3)+,GDDAT
4797 026576  005337  002360                    DEC     TEMP5
4798 026602  000402                            BR      7$
4799 026604  005037  002402          6$:       CLR     GDDAT
4800 026610  005237  002462          7$:       INC     DWCNT
4801 026614  021237  002402                    CMP     (R2),GDDAT
4802 026620  001422                            BEQ     8$
4803
4804 026622  005237  002344                    INC     DECNT
4805 026626  005701                            TST     R1
4806 026630  001416                            BEQ     8$
4807 026632  005301                            DEC     R1
4808 026634                                    PRINTB  #FMT14B,TEMP2,GDDAT,(R2)
     026634  011246                            MOV     (R2),-(SP)
     026636  013746  002402                    MOV     GDDAT,-(SP)
     026642  013746  002352                    MOV     TEMP2,-(SP)
     026646  012746  007504                    MOV     #FMT14B,-(SP)
     026652  012746  000004                    MOV     #4,-(SP)
     026656  010600                            MOV     SP,R0
     026660  104414                            TRAP    C$PNTB
     026662  062706  000012                    ADD     #12,SP
4809
4810 026666  005237  002352          8$:       INC     TEMP2
4811 026672  005722                            TST     (R2)+
4812 026674  023737  002352  002464            CMP     TEMP2,DWCNT1
4813 026702  003716                            BLE     4$
4814 026704                                    PRINTB  #FMT9A,DECNT,TEMP2
     026704  013746  002352                    MOV     TEMP2,-(SP)
     026710  013746  002344                    MOV     DECNT,-(SP)
     026714  012746  007223                    MOV     #FMT9A,-(SP)
     026720  012746  000003                    MOV     #3,-(SP)
     026724  010600                            MOV     SP,R0
     026726  104414                            TRAP    C$PNTB
     026730  062706  000010                    ADD     #10,SP
4815
4816 026734  000205                            RTS     R5
4817
4818                                   ;ROUTINE TO DUMP THE CONTENTS OF THE READ BUFFER ON ERROR DETECTED
4819                                   ;WILL ALSO TELL HOW MANY WORDS WERE IN THE XFER
```

ROUTINE TO DUMP BUFFER ON DCK

```
4820
4821 026736 016437 000042 002346 STDMP:  MOV     BMP(R4),TEMPO          ;GET NEGATIVE WORD COUNT
4822 026744 005437 002346                NEG     TEMPO                  ;MAKE THE # POSITIVE
4823 026750 012737 000200 002464         MOV     #128.,DWCNT1           ;SET THE SIZE OF SECTOR
4824 026756                              PRINTB  #FMTXS,TEMPO           ;TELL TRANSFER SIZE
     026756 013746 002346                MOV     TEMPO,-(SP)
     026762 012746 007673                MOV     #FMTXS,-(SP)
     026766 012746 000002                MOV     #2,-(SP)
     026772 010600                       MOV     SP,R0
     026774 104414                       TRAP    C$PNTB
     026776 062706 000006                ADD     #6,SP
4825 027002 013701 010662                MOV     T.CLT,R1               ;GET THE PRINT LIMIT
4826 027006 012703 000012                MOV     #10.,R3                ;SETUP LINE LIMIT
4827 027012                      1$:     PRINTB  #FMT14A,(R2)           ;PRINT A DATA WORD
     027012 011246                       MOV     (R2),-(SP)
     027014 012746 007472                MOV     #FMT14A,-(SP)
     027020 012746 000002                MOV     #2,-(SP)
     027024 010600                       MOV     SP,R0
     027026 104414                       TRAP    C$PNTB
     027030 062706 000006                ADD     #6,SP
4828 027034 005722                       TST     (R2)+                  ;POINT TO THE NEXT DATA WORD
4829 027036 005303                       DEC     R3                     ;DONE WITH THE LINE?
4830 027040 001012                       BNE     2$                     ;BR IF NO
4831 027042                              PRINTB  #FMT14C                ;YES - PRINT <CR>
     027042 012746 007501                MOV     #FMT14C,-(SP)
     027046 012746 000001                MOV     #1,-(SP)
     027052 010600                       MOV     SP,R0
     027054 104414                       TRAP    C$PNTB
     027056 062706 000004                ADD     #4,SP
4832 027062 012703 000012                MOV     #10.,R3                ;RESET THE LINE LIMIT
4833 027066 005337 002464        2$:     DEC     DWCNT1                 ;END OF SECTOR?
4834 027072 001001                       BNE     3$                     ;BR IF NO
4835 027074 000402                       BR      4$                     ;YES - EXIT
4836 027076 005301               3$:     DEC     R1                     ;AT PRINT LIMIT?
4837 027100 001344                       BNE     1$                     ;BR IF NO
4838 027102                      4$:     PRINTB  #FMT14C                ;PRINT <CR>
     027102 012746 007501                MOV     #FMT14C,-(SP)
     027106 012746 000001                MOV     #1,-(SP)
     027112 010600                       MOV     SP,R0
     027114 104414                       TRAP    C$PNTB
     027116 062706 000004                ADD     #4,SP
4839 027122 000205                       RTS     R5                     ;EXIT
4840
4841                                     ;
4842                                     ;ROUTINE TO CLEAR ALL DRIVE INFO. USED ON START OR
4843                                     ;RESTART IF CALLED.  CAN BE USED TO CLEAR INDIVIDUAL DRIVE
4844                                     ;INFO BY BITMAP FOLLOWING CALL
4845                                     ;CALL   JSR     R5,CLEAR
4846                                     ;
4847
4848 027124 010446               CLEAR:  MOV     R4,-(SP)               ;SAVE R4
4849 027126 012704 030362                MOV     #DRBUF,R4              ;GET BUFFER STARTS
4850 027132 005024               2$:     CLR     (R4)+                  ;CLEAR
4851 027134 020427 031642                CMP     R4,#ENDBUF             ;AT END OF BUFFERS
4852 027140 001374                       BNE     2$                     ;NO, GO TO 2$
4853 027142 012604               4$:     MOV     (SP)+,R4               ;RESTORE CURRENT BUFFER POINTER
4854 027144 000205                       RTS     R5                     ;EXIT
```

ROUTINE TO DUMP BUFFER ON DCK

```
4855
4856                                    .SBTTL  ROUTINE TO CHECK FOR BAD SECTOR
4857 027146                             STARS
                                        ;;*****************************************************************
4858                                    ;CKBDSC    POUTINE TO MATCH BAD SECTOR.....BDA(R4) IS SECTOR WE ARE LOOKING
4859                                    ;       FOR IN LIST POINTED TO BY BSECPT(R4)......HDRFND IS SET IF WE FIND IT.
4860 027146                             STARS
                                        ;;*****************************************************************
4861
4862 027146  005037  002340            CKBDSC: CLR     HDRFND          ;CLEAR FLAG
4863 027152  010046                             MOV     R0,-(SP)        ;SAVE R0
4864 027154  010246                             MOV     R2,-(SP)        ;SAVE   R2
4865 027156  012700  000021                     MOV     #17.,R0         ;16 ENTRIES + BSF POINTER
4866 027162  016402  000112            1$:     MOV     BSECPT(R4),R2   ;GET WHERE WE'RE LOOKING
4867 027166  022712  177777            2$:     CMP     #-1,(R2)        ;END OF ENTRY LIST?
4868 027172  001411                             BEQ     4$              ;BRANCH IF END
4869 027174  023712  002342                     CMP     CHKSEC,(R2)     ;HAVE WE GOT A MATCH
4870 027200  001404                             BEQ     3$              ;THEN GO SET INDICATOR, ELSE
4871 027202  005722                             TST     (R2)+
4872 027204  005300                             DEC     R0
4873 027206  001367                             BNE     2$
4874 027210  000402                             BR      4$
4875 027212  005237  002340            3$:     INC     HDRFND          ;SET FLAG FOUND
4876 027216  012602                    4$:     MOV     (SP)+,R2
4877 027220  012600                             MOV     (SP)+,R0
4878 027222  000205                             RTS     R5
4879
4880 027224                             STARS
                                        ;;*****************************************************************
4881                                    ;CKBDTK -- HERE TO CHECK IF CYLINDER & HEAD SELECTED IS IN THE BAD SECTOR FILE
4882 027224                             STARS
                                        ;;*****************************************************************
4883
4884 027224  005037  002340            CKBDTK: CLR     HDRFND          ;CLEAR FLAG
4885 027230  010046                             MOV     R0,-(SP)        ;SAVE R0
4886 027232  010146                             MOV     R1,-(SP)        ;SAVE R1
4887 027234  010246                             MOV     R2,-(SP)        ;SAVE R2
4888 027236  012700  000021                     MOV     #17.,R0         ;16 ENTRIES + BSF POINTER
4889 027242  016402  000112            1$:     MOV     BSECPT(R4),R2   ;GET WHERE WE'RE LOOKING
4890 027246  022712  177777            2$:     CMP     #-1,(R2)        ;END OF LIST?
4891 027252  001414                             BEQ     4$              ;BRANCH IF END
4892 027254  011201                             MOV     (R2),R1         ;GET THE ENTRY FROM BAD SECT FILE
4893 027256  043701  002272                     BIC     SMSK,R1         ;LEAVE ONLY CYL # & HEAD
4894 027262  023701  002342                     CMP     CHKSEC,R1       ;HAVE WE GOT A MATCH
4895 027266  001404                             BEQ     3$              ;THEN GO SET INDICATOR, ELSE
4896 027270  005722                             TST     (R2)+
4897 027272  005300                             DEC     R0
4898 027274  001364                             BNE     2$
4899 027276  000402                             BR      4$
4900 027300  005237  002340            3$:     INC     HDRFND          ;SET FLAG FOUND
4901 027304  012602                    4$:     MOV     (SP)+,R2
4902 027306  012601                             MOV     (SP)+,R1
4903 027310  012600                             MOV     (SP)+,R0
4904 027312  000205                             RTS     R5
4905
4906 027314                             STARS
                                        ;;*****************************************************************
```

ROUTINE TO CHECK FOR BAD SECTOR

```
                                        STARS
                                        ;;*****************************************************************
                                        ;BUFFER TO STORE BAD SECTOR LISTS

                                        BSEC0:  .BLKW   17.
                                        BSEC1:  .BLKW   17.
                                        BSEC2:  .BLKW   17.
                                        BSEC3:  .BLKW   17.
                                        BSEC4:  .BLKW   17.
                                        BSEC5:  .BLKW   17.
                                        BSEC6:  .BLKW   17.
                                        BSEC7:  .BLKW   17.
                                        STARS
                                        ;;*****************************************************************
                                        STARS
                                        ;;*****************************************************************

                                        ;LIST OF PATTERNS USED IN WRITING

027734  027754                  PATLST: PAT0    ;ALL 0'S
027736  030014                          PAT1    ;-1'S TO ALT BITS
027740  030054                          PAT2    ;0'S TO ALT BITS
027742  030114                          PAT3    ;SHIFTING ALT BITS
027744  030154                          PAT4    ;WORST CASE DATA
027746  030214                          PAT5    ;STRANGE DATA
027750  030254                          PAT6    ;ALL 1'S
027752  030314                          PAT7    ;STRANGE DATA

027754  000000                  PAT0:   .WORD   0
027756  000000                          .WORD   0
027760  000000                          .WORD   0
027762  000000                          .WORD   0
027764  000000                          .WORD   0
027766  000000                          .WORD   0
027770  000000                          .WORD   0
027772  000000                          .WORD   0
027774  000000                          .WORD   0
027776  000000                          .WORD   0
030000  000000                          .WORD   0
030002  000000                          .WORD   0
030004  000000                          .WORD   0
030006  000000                          .WORD   0
030010  000000                          .WORD   0
030012  000000                          .WORD   0

030014  177777                  PAT1:   .WORD   177777
030016  177777                          .WORD   177777
030020  177777                          .WORD   177777
030022  052525                          .WORD   052525
030024  052525                          .WORD   052525
030026  052525                          .WORD   052525
030030  177777                          .WORD   177777
030032  177777                          .WORD   177777
030034  052525                          .WORD   052525
030036  052525                          .WORD   052525
030040  177777                          .WORD   177777
030042  052525                          .WORD   052525
```

ROUTINE TO CHECK FOR BAD SECTOR

```
4961 030044  177252                        .WORD   177252
4962 030046  177252                        .WORD   177252
4963 030050  172765                        .WORD   172765
4964 030052  172765                        .WORD   172765
4965
4966 030054  000000                PAT2:   .WORD   0
4967 030056  000000                        .WORD   0
4968 030060  000000                        .WORD   0
4969 030062  177777                        .WORD   177777
4970 030064  177777                        .WORD   177777
4971 030066  177777                        .WORD   177777
4972 030070  000000                        .WORD   0
4973 030072  000000                        .WORD   0
4974 030074  177777                        .WORD   177777
4975 030076  177777                        .WORD   177777
4976 030100  000000                        .WORD   0
4977 030102  177777                        .WORD   177777
4978 030104  000000                        .WORD   0
4979 030106  177777                        .WORD   177777
4980 030110  000000                        .WORD   0
4981 030112  177777                        .WORD   177777
4982
4983 030114  025252                PAT3:   .WORD   25252
4984 030116  052525                        .WORD   52525
4985 030120  052525                        .WORD   52525
4986 030122  125252                        .WORD   125252
4987 030124  125252                        .WORD   125252
4988 030126  125252                        .WORD   125252
4989 030130  052525                        .WORD   52525
4990 030132  052525                        .WORD   52525
4991 030134  125252                        .WORD   125252
4992 030136  125252                        .WORD   125252
4993 030140  052525                        .WORD   52525
4994 030142  125252                        .WORD   125252
4995 030144  052525                        .WORD   52525
4996 030146  125252                        .WORD   125252
4997 030150  052525                        .WORD   52525
4998 030152  125252                        .WORD   125252
4999
5000 030154  155555                PAT4:   .WORD   155555
5001 030156  066666                        .WORD   066666
5002 030160  133333                        .WORD   133333
5003 030162  155555                        .WORD   155555
5004 030164  066666                        .WORD   066666
5005 030166  133333                        .WORD   133333
5006 030170  155555                        .WORD   155555
5007 030172  066666                        .WORD   066666
5008 030174  133333                        .WORD   133333
5009 030176  155555                        .WORD   155555
5010 030200  066666                        .WORD   066666
5011 030202  133333                        .WORD   133333
5012 030204  155555                        .WORD   155555
5013 030206  066666                        .WORD   066666
5014 030210  133333                        .WORD   133333
5015 030212  155555                        .WORD   155555
5016
5017 030214  121105                PAT5:   .WORD   121105
```

ROUTINE TO CHECK FOR BAD SECTOR

```
5018 030216  150442                        .WORD   150442
5019 030220  064221                        .WORD   64221
5020 030222  132110                        .WORD   132110
5021 030224  055044                        .WORD   55044
5022 030226  026422                        .WORD   26422
5023 030230  013211                        .WORD   13211
5024 030232  105504                        .WORD   105504
5025 030234  042642                        .WORD   42642
5026 030236  021321                        .WORD   21321
5027 030240  110550                        .WORD   110550
5028 030242  044264                        .WORD   44264
5029 030244  022132                        .WORD   22132
5030 030246  011055                        .WORD   11055
5031 030250  104426                        .WORD   104426
5032 030252  042213                        .WORD   42213
5033
5034 030254  177777                PAT6:   .WORD   177777
5035 030256  177777                        .WORD   177777
5036 030260  177777                        .WORD   177777
5037 030262  177777                        .WORD   177777
5038 030264  177777                        .WORD   177777
5039 030266  177777                        .WORD   177777
5040 030270  177777                        .WORD   177777
5041 030272  177777                        .WORD   177777
5042 030274  177777                        .WORD   177777
5043 030276  177777                        .WORD   177777
5044 030300  177777                        .WORD   177777
5045 030302  177777                        .WORD   177777
5046 030304  177777                        .WORD   177777
5047 030306  177777                        .WORD   177777
5048 030310  177777                        .WORD   177777
5049 030312  177777                        .WORD   177777
5050
5051 030314  045513                PAT7:   .WORD   45513
5052 030316  122645                        .WORD   122645
5053 030320  151322                        .WORD   151322
5054 030322  064551                        .WORD   64551
5055 030324  132264                        .WORD   132264
5056 030326  055132                        .WORD   55132
5057 030330  026455                        .WORD   26455
5058 030332  113226                        .WORD   113226
5059 030334  045513                        .WORD   45513
5060 030336  122645                        .WORD   122645
5061 030340  151322                        .WORD   151322
5062 030342  064551                        .WORD   64551
5063 030344  132264                        .WORD   132264
5064 030346  055132                        .WORD   55132
5065 030350  026455                        .WORD   26455
5066 030352  113226                        .WORD   113226
5067
5068 030354  000240                ENDOFPROGRAM:   NOP
5069 030356                        ENDTST
     030356                        L10024:
     030356  104401                        TRAP    C$ETST
5070 030360  000000                        HALT
5071
5072                               .SBTTL  DRIVE INFORMATION BUFFERS
```

DRIVE INFORMATION BUFFERS

```
5073
5074                                    ;DRIVE INFORMATION BUFFER
5075
5076                                    .LIST   ME
5077
5078 030362                             DRBUF:
5079            000010                  .REPT   8.
5124
     030362  000000                             SKCNT                   ;SEEK OPERATION COUNT
     030364  000002                             RXFR1                   ;READ OPERATION COUNT (BITS) LOW ORDER
     030366  000004                             RXFR2                   ; "         "         "     "   HIGH ORDER
     030370  000006                             WXFR1                   ;WRITE OPERATION COUNT (BITS) LOW ORDER
     030372  000010                             WXFR2                   ; "         "         "     "   HIGH ORDER
     030374  000012                             ERRCNT                  ;ERROR COUNT - HARD
     030376  000014                             SFTCNT                  ;ERROR COUNT - SOFT
     030400  000016                             SKECNT                  ;SEEK ERROR COUNT
     030402  000020                             DERCNT                  ;DRIVE ERROR COUNT
     030404  000022                             DCRCER                  ;DATA CRC ERROR COUNT
     030406  000024                             HCRCER                  ;HEADER CRC ERROR COUNT
     030410  000026                             DLTCNT                  ;DATA LATE ERROR COUNT
     030412  000030                             OPICNT                  ;OPERATION INCOMPLETE ERROR COUNT
     030414  000032                             HNFERR                  ;HEADER NOT FOUND ERROR COUNT
     030416  000034                             NXMCNT                  ;NON EXISTANT MEMORY ERROR COUNT
     030420  000036                             RETRY                   ;PRESENT RETRY NUMBER
     030422  000040                             BDA                     ;   "     DISK ADDRESS CONTENTS
     030424  000042                             BMP                     ;PRESENT MULTIPURPOSE CONTENTS
     030426  000044                             FUNC                    ;LAST FUNCTION LOADED
     030430  000046                             BCSADR                  ;CSR IMAGE OF LAST COMMAND
     030432  000050                             LSTHDR                  ;LAST POSITION ON DISK
     030434  000052                             RTYPE                   ;ERROR ON WHICH RECOVERY IS IN PROGRESS
     030436  000054                             SKCNT1                  ;SEEK COUNT LOW ORDER
     030440  000056                             PRFLGS                  ;PROGRAM INTERNAL FLAGS
     030442  000060                             RXFR3                   ;READ COUNT THIRD
     030444  000062                             WXFR3                   ;WRITE COUNT THIRD
     030446  000064                             LSTDA                   ;DISK ADDRESS OF SOFT ERROR
     030450  000066                             DIFWD                   ;LAST DIFFERENCE WORD OF SEEK
     030452  000070                             DPHOUR                  ;TIME DRIVE WAS DROPPED
     030454  000072                             TRERR                   ;TRACKING ERROR COUNT
     030456  000074                             DATCER
     030460  000076                             DOWCK                   ;WRITE CHECK NECESARY
     030462  000100                             SERNM1                  ;SERIAL NUMBER OF CARTRIDGE
     030464  000102                             SERNM2                  ;SERIAL NUMBER OF CARTRIDGE
     030466  000104                             DCS                     ;CSR ADDRESS
     030470  000106                             DRSEL                   ;DRIVE SELECT BITS(8,9,10)
     030472  000110                             BBA                     ;PRESENT BUS ADDRESS CONTENTS
     030474  000112                             BSECPT                  ;POINTER TO BAD SECTOR FILE
     030476  000114                             RSEEK
     030500  000116                             SOFTCS                  ;CSR AT TIME OF SOFT ERROR
     030502  000120                             TDR                     ;DRIVE TYPE FLAG (RL01 =1)
     030504  000122                             WRIPG                   ;WRITE IN PROGRESS FLAG
     030506  000124                             PRPOS                   ;PRESENT POSITION ON DISK

     030510  000000                             SKCNT                   ;SEEK OPERATION COUNT
     030512  000002                             RXFR1                   ;READ OPERATION COUNT (BITS) LOW ORDER
     030514  000004                             RXFR2                   ; "         "         "     "   HIGH ORDER
     030516  000006                             WXFR1                   ;WRITE OPERATION COUNT (BITS) LOW ORDER
     030520  000010                             WXFR2                   ; "         "         "     "   HIGH ORDER
```

DRIVE INFORMATION BUFFERS

```
030522  000012                  ERRCNT          ;ERROR COUNT    HARD
030524  000014                  SFTCNT          ;ERROR COUNT    SOFT
030526  000016                  SKECNT          ;SEEK ERROR COUNT
030530  000020                  DERCNT          ;DRIVE ERROR COUNT
030532  000022                  DCRCER          ;DATA CRC ERROR COUNT
030534  000024                  HCRCER          ;HEADER CRC ERROR COUNT
030536  000026                  DLTCNT          :DATA LATE ERROR COUNT
030540  000030                  OPICNT          ;OPERATION INCOMPLETE ERROR COUNT
030542  000032                  HNFERR          ;HEADER NOT FOUND ERROR COUNT
030544  000034                  NXMCNT          ;NON EXISTANT MEMORY ERROR COUNT
030546  000036                  RETRY           ;PRESENT RETRY NUMBER
030550  000040                  BDA             ;    "     DISK ADDRESS CONTENTS
030552  000042                  BMP             ;PRESENT MULTIPURPOSE CONTENTS
030554  000044                  FUNC            ;LAST FUNCTION LOADED
030556  000046                  BCSADR          ;CSR IMAGE OF LAST COMMAND
030560  000050                  LSTHDR          ;LAST POSITION ON DISK
030562  000052                  RTYPE           ;ERROR ON WHICH RECOVERY IS IN PROGRESS
030564  000054                  SKCNT1          ;SEEK COUNT LOW ORDER
030566  000056                  PRFLGS          ;PROGRAM INTERNAL FLAGS
030570  000060                  RXFR3           ;READ COUNT THIRD
030572  000062                  WXFR3           ;WRITE COUNT THIRD
030574  000064                  LSTDA           ;DISK ADDRESS OF SOFT ERROR
030576  000066                  DIFWD           ;LAST DIFFERENCE WORD OF SEEK
030600  000070                  DPHOUR          ;TIME DRIVE WAS DROPPED
030602  000072                  TRERR           ;TRACKING ERROR COUNT
030604  000074                  DATCER
030606  000076                  DOWCK           ;WRITE CHECK NECESARY
030610  000100                  SERNM1          ;SERIAL NUMBER OF CARTRIDGE
030612  000102                  SERNM2          ;SERIAL NUMBER OF CARTRIDGE
030614  000104                  DCS             ;CSR ADDRESS
030616  000106                  DRSEL           ;DRIVE SELECT BITS(8,9,10)
030620  000110                  BBA             ;PRESENT BUS ADDRESS CONTENTS
030622  000112                  BSECPT          ;POINTER TO BAD SECTOR FILE
030624  000114                  RSEEK
030626  000116                  SOFTCS          ;CSR AT TIME OF SOFT ERROR
030630  000120                  TDR             ;DRIVE TYPE FLAG (RL01 -1)
030632  000122                  WRIPG           ;WRITE IN PROGRESS FLAG
030634  000124                  PRPOS           ;PRESENT POSITION ON DISK

030636  000000                  SKCNT           ;SEEK OPERATION COUNT
030640  000002                  RXFR1           ;READ OPERATION COUNT (BITS) LOW ORDER
030642  000004                  RXFR2           ;  "       "       "     "   HIGH ORDER
030644  000006                  WXFR1           ;WRITE OPERATION COUNT (BITS) LOW ORDER
030646  000010                  WXFR2           ;  "       "       "     "   HIGH ORDER
030650  000012                  ERRCNT          ;ERROR COUNT - HARD
030652  000014                  SFTCNT          ;ERROR COUNT - SOFT
030654  000016                  SKECNT          ;SEEK ERROR COUNT
030656  000020                  DERCNT          ;DRIVE ERROR COUNT
030660  000022                  DCRCER          ;DATA CRC ERROR COUNT
030662  000024                  HCRCER          ;HEADER CRC ERROR COUNT
030664  000026                  DLTCNT          ;DATA LATE ERROR COUNT
030666  000030                  OPICNT          ;OPERATION INCOMPLETE ERROR COUNT
030670  000032                  HNFERR          ;HEADER NOT FOUND ERROR COUNT
030672  000034                  NXMCNT          ;NON EXISTANT MEMORY ERROR COUNT
030674  000036                  RETRY           ;PRESENT RETRY NUMBER
030676  000040                  BDA             ;    "     DISK ADDRESS CONTENTS
030700  000042                  BMP             ;PRESENT MULTIPURPOSE CONTENTS
```

DRIVE INFORMATION BUFFERS

```
030702  000044                  FUNC            ;LAST FUNCTION LOADED
030704  000046                  BCSADR          ;CSR IMAGE OF LAST COMMAND
030706  000050                  LSTHDR          ;LAST POSITION ON DISK
030710  000052                  RTYPE           ;ERROR ON WHICH RECOVERY IS IN PROGRESS
030712  000054                  SKCNT1          ;SEEK COUNT LOW ORDER
030714  000056                  PRFLGS          ;PROGRAM INTERNAL FLAGS
030716  000060                  RXFR3           ;READ COUNT THIRD
030720  000062                  WXFR3           ;WRITE COUNT THIRD
030722  000064                  LSTDA           ;DISK ADDRESS OF SOFT ERROR
030724  000066                  DIFWD           ;LAST DIFFERENCE WORD OF SEEK
030726  000070                  DPHOUR          ;TIME DRIVE WAS DROPPED
030730  000072                  TRERR           ;TRACKING ERROR COUNT
030732  000074                  DATCER
030734  000076                  DOWCK           ;WRITE CHECK NECESARY
030736  000100                  SERNM1          ;SERIAL NUMBER OF CARTRIDGE
030740  000102                  SERNM2          ;SERIAL NUMBER OF CARTRIDGE
030742  000104                  DCS             ;CSR ADDRESS
030744  000106                  DRSEL           ;DRIVE SELECT BITS(8,9,10)
030746  000110                  BBA             ;PRESENT BUS ADDRESS CONTENTS
030750  000112                  BSECPT          ;POINTER TO BAD SECTOR FILE
030752  000114                  RSEEK
030754  000116                  SOFTCS          ;CSR AT TIME OF SOFT ERROR
030756  000120                  TDR             ;DRIVE TYPE FLAG (RL01 =1)
030760  000122                  WRIPG           ;WRITE IN PROGRESS FLAG
030762  000124                  PRPOS           ;PRESENT POSITION ON DISK

030764  000000                  SKCNT           ;SEEK OPERATION COUNT
030766  000002                  RXFR1           ;READ OPERATION COUNT (BITS) LOW ORDER
030770  000004                  RXFR2           ; "      "      "     "   HIGH ORDER
030772  000006                  WXFR1           ;WRITE OPERATION COUNT (BITS) LOW ORDER
030774  000010                  WXFR2           ; "       "       "     "   HIGH ORDER
030776  000012                  ERRCNT          ;ERROR COUNT - HARD
031000  000014                  SFTCNT          ;ERROR COUNT - SOFT
031002  000016                  SKECNT          ;SEEK ERROR COUNT
031004  000020                  DERCNT          ;DRIVE ERROR COUNT
031006  000022                  DCRCER          ;DATA CRC ERROR COUNT
031010  000024                  HCRCER          ;HEADER CRC ERROR COUNT
031012  000026                  DLTCNT          ;DATA LATE ERROR COUNT
031014  000030                  OPICNT          ;OPERATION INCOMPLETE ERROR COUNT
031016  000032                  HNFERR          ;HEADER NOT FOUND ERROR COUNT
031020  000034                  NXMCNT          ;NON EXISTANT MEMORY ERROR COUNT
031022  000036                  RETRY           ;PRESENT RETRY NUMBER
031024  000040                  BDA             ;   "    DISK ADDRESS CONTENTS
031026  000042                  BMP             ;PRESENT MULTIPURPOSE CONTENTS
031030  000044                  FUNC            ;LAST FUNCTION LOADED
031032  000046                  BCSADR          ;CSR IMAGE OF LAST COMMAND
031034  000050                  LSTHDR          ;LAST POSITION ON DISK
031036  000052                  RTYPE           ;ERROR ON WHICH RECOVERY IS IN PROGRESS
031040  000054                  SKCNT1          ;SEEK COUNT LOW ORDER
031042  000056                  PRFLGS          ;PROGRAM INTERNAL FLAGS
031044  000060                  RXFR3           ;READ COUNT THIRD
031046  000062                  WXFR3           ;WRITE COUNT THIRD
031050  000064                  LSTDA           ;DISK ADDRESS OF SOFT ERROR
031052  000066                  DIFWD           ;LAST DIFFERENCE WORD OF SEEK
031054  000070                  DPHOUR          ;TIME DRIVE WAS DROPPED
031056  000072                  TRERR           ;TRACKING ERROR COUNT
031060  000074                  DATCER
```

DRIVE INFORMATION BUFFERS

```
031062  000076                  DOWCK           ;WRITE CHECK NECESARY
031064  000100                  SERNM1          ;SERIAL NUMBER OF CARTRIDGE
031066  000102                  SERNM2          ;SERIAL NUMBER OF CARTRIDGE
031070  000104                  DCS             ;CSR ADDRESS
031072  000106                  DRSEL           ;DRIVE SELECT BITS(8,9,10)
031074  000110                  BBA             ;PRESENT BUS ADDRESS CONTENTS
031076  000112                  BSECPT          ;POINTER TO BAD SECTOR FILE
031100  000114                  RSEEK
031102  000116                  SOFTCS          ;CSR AT TIME OF SOFT ERROR
031104  000120                  TDR             ;DRIVE TYPE FLAG (RL01 =1)
031106  000122                  WRIPG           ;WRITE IN PROGRESS FLAG
031110  000124                  PRPOS           ;PRESENT POSITION ON DISK

031112  000000                  SKCNT           ;SEEK OPERATION COUNT
031114  000002                  RXFR1           ;READ OPERATION COUNT (BITS) LOW ORDER
031116  000004                  RXFR2           ;  "         "       "     "   HIGH ORDER
031120  000006                  WXFR1           ;WRITE OPERATION COUNT (BITS) LOW ORDER
031122  000010                  WXFR2           ;  "         "       "     "   HIGH ORDER
031124  000012                  ERRCNT          ;ERROR COUNT - HARD
031126  000014                  SFTCNT          ;ERROR COUNT - SOFT
031130  000016                  SKECNT          ;SEEK ERROR COUNT
031132  000020                  DERCNT          ;DRIVE ERROR COUNT
031134  000022                  DCRCER          ;DATA CRC ERROR COUNT
031136  000024                  HCRCER          ;HEADER CRC ERROR COUNT
031140  000026                  DLTCNT          ;DATA LATE ERROR COUNT
031142  000030                  OPICNT          ;OPERATION INCOMPLETE ERROR COUNT
031144  000032                  HNFERR          ;HEADER NOT FOUND ERROR COUNT
031146  000034                  NXMCNT          ;NON EXISTANT MEMORY ERROR COUNT
031150  000036                  RETRY           ;PRESENT RETRY NUMBER
031152  000040                  BDA             ;   "    DISK ADDRESS CONTENTS
031154  000042                  BMP             ;PRESENT MULTIPURPOSE CONTENTS
031156  000044                  FUNC            ;LAST FUNCTION LOADED
031160  000046                  BCSADR          ;CSR IMAGE OF LAST COMMAND
031162  000050                  LSTHDR          ;LAST POSITION ON DISK
031164  000052                  RTYPE           ;ERROR ON WHICH RECOVERY IS IN PROGRESS
031166  000054                  SKCNT1          ;SEEK COUNT LOW ORDER
031170  000056                  PRFLGS          ;PROGRAM INTERNAL FLAGS
031172  000060                  RXFR3           ;READ COUNT THIRD
031174  000062                  WXFR3           ;WRITE COUNT THIRD
031176  000064                  LSTDA           ;DISK ADDRESS OF SOFT ERROR
031200  000066                  DIFWD           ;LAST DIFFERENCE WORD OF SEEK
031202  000070                  DPHOUR          ;TIME DRIVE WAS DROPPED
031204  000072                  TRERR           ;TRACKING ERROR COUNT
031206  000074                  DATCER
031210  000076                  DOWCK           ;WRITE CHECK NECESARY
031212  000100                  SERNM1          ;SERIAL NUMBER OF CARTRIDGE
031214  000102                  SERNM2          ;SERIAL NUMBER OF CARTRIDGE
031216  000104                  DCS             ;CSR ADDRESS
031220  000106                  DRSEL           ;DRIVE SELECT BITS(8,9,10)
031222  000110                  BBA             ;PRESENT BUS ADDRESS CONTENTS
031224  000112                  BSECPT          ;POINTER TO BAD SECTOR FILE
031226  000114                  RSEEK
031230  000116                  SOFTCS          ;CSR AT TIME OF SOFT ERROR
031232  000120                  TDR             ;DRIVE TYPE FLAG (RL01 =1)
031234  000122                  WRIPG           ;WRITE IN PROGRESS FLAG
031236  000124                  PRPOS           ;PRESENT POSITION ON DISK
```

DRIVE INFORMATION BUFFERS

```
031240  000000  SKCNT   ;SEEK OPERATION COUNT
031242  000002  RXFR1   ;READ OPERATION COUNT (BITS) LOW ORDER
031244  000004  RXFR2   ;                          HIGH ORDER
031246  000006  WXFR1   ;WRITE OPERATION COUNT (BITS) LOW ORDER
031250  000010  WXFR2   ;                          HIGH ORDER
031252  000012  ERRCNT  ;ERROR COUNT   HARD
031254  000014  SFTCNT  ;ERROR COUNT - SOFT
031256  000016  SKECNT  ;SEEK ERROR COUNT
031260  000020  DERCNT  ;DRIVE ERROR COUNT
031262  000022  DCRCER  ;DATA CRC ERROR COUNT
031264  000024  HCRCER  ;HEADER CRC ERROR COUNT
031266  000026  DLTCNT  ;DATA LATE ERROR COUNT
031270  000030  OPICNT  ;OPERATION INCOMPLETE ERROR COUNT
031272  000032  HNFERR  ;HEADER NOT FOUND ERROR COUNT
031274  000034  NXMCNT  ;NON EXISTANT MEMORY ERROR COUNT
031276  000036  RETRY   ;PRESENT RETRY NUMBER
031300  000040  BDA     ;   "     DISK ADDRESS CONTENTS
031302  000042  BMP     ;PRESENT MULTIPURPOSE CONTENTS
031304  000044  FUNC    ;LAST FUNCTION LOADED
031306  000046  BCSADR  ;CSR IMAGE OF LAST COMMAND
031310  000050  LSTHDR  ;LAST POSITION ON DISK
031312  000052  RTYPE   ;ERROR ON WHICH RECOVERY IS IN PROGRESS
031314  000054  SKCNT1  ;SEEK COUNT LOW ORDER
031316  000056  PRFLGS  ;PROGRAM INTERNAL FLAGS
031320  000060  RXFR3   ;READ COUNT THIRD
031322  000062  WXFR3   ;WRITE COUNT THIRD
031324  000064  LSTDA   ;DISK ADDRESS OF SOFT ERROR
031326  000066  DIFWD   ;LAST DIFFERENCE WORD OF SEEK
031330  000070  DPHOUR  ;TIME DRIVE WAS DROPPED
031332  000072  TRERR   ;TRACKING ERROR COUNT
031334  000074  DATCER
031336  000076  DOWCK   ;WRITE CHECK NECESARY
031340  000100  SERNM1  ;SERIAL NUMBER OF CARTRIDGE
031342  000102  SERNM2  ;SERIAL NUMBER OF CARTRIDGE
031344  000104  DCS     ;CSR ADDRESS
031346  000106  DRSEL   ;DRIVE SELECT BITS(8,9,10)
031350  000110  BBA     ;PRESENT BUS ADDRESS CONTENTS
031352  000112  BSECPT  ;POINTER TO BAD SECTOR FILE
031354  000114  RSEEK
031356  000116  SOFTCS  ;CSR AT TIME OF SOFT ERROR
031360  000120  TDR     ;DRIVE TYPE FLAG (RL01 =1)
031362  000122  WRIPG   ;WRITE IN PROGRESS FLAG
031364  000124  PRPOS   ;PRESENT POSITION ON DISK

031366  000000  SKCNT   ;SEEK OPERATION COUNT
031370  000002  RXFR1   ;READ OPERATION COUNT (BITS) LOW ORDER
031372  000004  RXFR2   ;  "       "       "     "   HIGH ORDER
031374  000006  WXFR1   ;WRITE OPERATION COUNT (BITS) LOW ORDER
031376  000010  WXFR2   ;  "       "       "     "   HIGH ORDER
031400  000012  ERRCNT  ;ERROR COUNT - HARD
031402  000014  SFTCNT  ;ERROR COUNT - SOFT
031404  000016  SKECNT  ;SEEK ERROR COUNT
031406  000020  DERCNT  ;DRIVE ERROR COUNT
031410  000022  DCRCER  ;DATA CRC ERROR COUNT
031412  000024  HCRCER  ;HEADER CRC ERROR COUNT
031414  000026  DLTCNT  ;DATA LATE ERROR COUNT
031416  000030  OPICNT  ;OPERATION INCOMPLETE ERROR COUNT
```

DRIVE INFORMATION BUFFERS

```
031420  000032                  HNFERR          ;HEADER NOT FOUND ERROR COUNT
031422  000034                  NXMCNT          ;NON EXISTANT MEMORY ERROR COUNT
031424  000036                  RETRY           ;PRESENT RETRY NUMBER
031426  000040                  BDA             ;    "     DISK ADDRESS CONTENTS
031430  000042                  BMP             ;PRESENT MULTIPURPOSE CONTENTS
031432  000044                  FUNC            ;LAST FUNCTION LOADED
031434  000046                  BCSADR          ;CSR IMAGE OF LAST COMMAND
031436  000050                  LSTHDR          ;LAST POSITION ON DISK
031440  000052                  RTYPE           ;ERROR ON WHICH RECOVERY IS IN PROGRESS
031442  000054                  SKCNT1          ;SEEK COUNT LOW ORDER
031444  000056                  PRFLGS          ;PROGRAM INTERNAL FLAGS
031446  000060                  RXFR3           ;READ COUNT THIRD
031450  000062                  WXFR3           ;WRITE COUNT THIRD
031452  000064                  LSTDA           ;DISK ADDRESS OF SOFT ERROR
031454  000066                  DIFWD           ;LAST DIFFERENCE WORD OF SEEK
031456  000070                  DPHOUR          ;TIME DRIVE WAS DROPPED
031460  000072                  TRERR           ;TRACKING ERROR COUNT
031462  000074                  DATCER
031464  000076                  DOWCK           ;WRITE CHECK NECESARY
031466  000100                  SERNM1          ;SERIAL NUMBER OF CARTRIDGE
031470  000102                  SERNM2          ;SERIAL NUMBER OF CARTRIDGE
031472  000104                  DCS             ;CSR ADDRESS
031474  000106                  DRSEL           ;DRIVE SELECT BITS(8,9,10)
031476  000110                  BBA             ;PRESENT BUS ADDRESS CONTENTS
031500  000112                  BSECPT          ;POINTER TO BAD SECTOR FILE
031502  000114                  RSEEK
031504  000116                  SOFTCS          ;CSR AT TIME OF SOFT ERROR
031506  000120                  TDR             ;DRIVE TYPE FLAG (RL01 =1)
031510  000122                  WRIPG           ;WRITE IN PROGRESS FLAG
031512  000124                  PRPOS           ;PRESENT POSITION ON DISK

031514  000000                  SKCNT           ;SEEK OPERATION COUNT
031516  000002                  RXFR1           ;READ OPERATION COUNT (BITS) LOW ORDER
031520  000004                  RXFR2           ;  "       "       "     "   HIGH ORDER
031522  000006                  WXFR1           ;WRITE OPERATION COUNT (BITS) LOW ORDER
031524  000010                  WXFR2           ;  "        "        "    "   HIGH ORDER
031526  000012                  ERRCNT          ;ERROR COUNT - HARD
031530  000014                  SFTCNT          ;ERROR COUNT - SOFT
031532  000016                  SKECNT          ;SEEK ERROR COUNT
031534  000020                  DERCNT          ;DRIVE ERROR COUNT
031536  000022                  DCRCER          ;DATA CRC ERROR COUNT
031540  000024                  HCRCER          ;HEADER CRC ERROR COUNT
031542  000026                  DLTCNT          ;DATA LATE ERROR COUNT
031544  000030                  OPICNT          ;OPERATION INCOMPLETE ERROR COUNT
031546  000032                  HNFERR          ;HEADER NOT FOUND ERROR COUNT
031550  000034                  NXMCNT          ;NON EXISTANT MEMORY ERROR COUNT
031552  000036                  RETRY           ;PRESENT RETRY NUMBER
031554  000040                  BDA             ;    "     DISK ADDRESS CONTENTS
031556  000042                  BMP             ;PRESENT MULTIPURPOSE CONTENTS
031560  000044                  FUNC            ;LAST FUNCTION LOADED
031562  000046                  BCSADR          ;CSR IMAGE OF LAST COMMAND
031564  000050                  LSTHDR          ;LAST POSITION ON DISK
031566  000052                  RTYPE           ;ERROR ON WHICH RECOVERY IS IN PROGRESS
031570  000054                  SKCNT1          ;SEEK COUNT LOW ORDER
031572  000056                  PRFLGS          ;PROGRAM INTERNAL FLAGS
031574  000060                  RXFR3           ;READ COUNT THIRD
031576  000062                  WXFR3           ;WRITE COUNT THIRD
```

DRIVE INFORMATION BUFFERS

```
        031600  000064                  LSTDA                   ;DISK ADDRESS OF SOFT ERROR
        031602  000066                  DIFWD                   ;LAST DIFFERENCE WORD OF SEEK
        031604  000070                  DPHOUR                  ;TIME DRIVE WAS DROPPED
        031606  000072                  TRERR                   ;TRACKING ERROR COUNT
        031610  000074                  DATCER
        031612  000076                  DOWCK                   ;WRITE CHECK NECESARY
        031614  000100                  SERNM1                  ;SERIAL NUMBER OF CARTRIDGE
        031616  000102                  SERNM2                  ;SERIAL NUMBER OF CARTRIDGE
        031620  000104                  DCS                     ;CSR ADDRESS
        031622  000106                  DRSEL                   ;DRIVE SELECT BITS(8,9,10)
        031624  000110                  BBA                     ;PRESENT BUS ADDRESS CONTENTS
        031626  000112                  BSECPT                  ;POINTER TO BAD SECTOR FILE
        031630  000114                  RSEEK
        031632  000116                  SOFTCS                  ;CSR AT TIME OF SOFT ERROR
        031634  000120                  TDR                     ;DRIVE TYPE FLAG (RL01 =1)
        031636  000122                  WRIPG                   ;WRITE IN PROGRESS FLAG
        031640  000124                  PRPOS                   ;PRESENT POSITION ON DISK
 5125                           .NLIST  ME
 5126
 5127 031642  000000           ENDBUF: .WORD   0
 5128
 5129                           ;QUESTIONS TO GET PARAMETERS FOR HARDWARE P-TABLE
 5130
 5131 031644                    BGNMOD  HRDPRM
 5132 031644                            BGNHRD
        031644  000030                  .WORD L10030-L$HARD/2
 5133
 5134 031646                            GPRML   CNTYPE,CNT,1,YES
        031646  005130                  .WORD   T$CODE
        031650  031726                  .WORD   CNTYPE
        031652  000001                  .WORD   1
 5135 031654                            GPRMA   CSRMSG,CSR,0,160000,177776,YES
        031654  000031                  .WORD   T$CODE
        031656  031733                  .WORD   CSRMSG
        031660  160000                  .WORD   T$LOLIM
        031662  177776                  .WORD   T$HILIM
 5136 031664                            GPRMA   VECMSG,VECT,0,0,776,YES
        031664  001031                  .WORD   T$CODE
        031666  032002                  .WORD   VECMSG
        031670  000000                  .WORD   T$LOLIM
        031672  000776                  .WORD   T$HILIM
 5137 031674                            GPRMD   DRMSG,DRBT,0,03400,0,7,YES
        031674  004032                  .WORD   T$CODE
        031676  032011                  .WORD   DRMSG
        031700  003400                  .WORD   03400
        031702  000000                  .WORD   T$LOLIM
        031704  000007                  .WORD   T$HILIM
 5138 031706                            GPRML   DRTYPE,TYPDR,1,YES
        031706  003130                  .WORD   T$CODE
        031710  031760                  .WORD   DRTYPE
        031712  000001                  .WORD   1
 5139 031714                            GPRMD   BRMSG,PRIOR,0,340,0,7,YES
        031714  002032                  .WORD   T$CODE
        031716  031747                  .WORD   BRMSG
        031720  000340                  .WORD   340
        031722  000000                  .WORD   T$LOLIM
        031724  000007                  .WORD   T$HILIM
```

DRIVE INFORMATION BUFFERS

```
5140
5141 031726                                ENDHRD
                                           .EVEN
     031726                          L10030:
5142
5146
5147 031726  122  114  061  CNTYPE: .ASCIZ  /RL11/
5148 031733  102  125  123  CSRMSG: .ASCIZ  /BUS ADDRESS/
5149 031747  102  122  040  BRMSG:  .ASCIZ  /BR LEVEL/
5150 031760  104  122  111  DRTYPE: .ASCIZ  /DRIVE TYPE = RL01/
5151 032002  126  105  103  VECMSG: .ASCIZ  /VECTOR/
5152 032011  104  122  111  DRMSG:  .ASCIZ  /DRIVE/
5153
5157
5158                                .EVEN
5159
5160 032020                                ENDMOD
5161
5162                                ;QUESTIONS TO GET PARAMETERS FOR SOFTWARE P-TABLE
5163
5164 032020                          BGNMOD  SFTPRM
5165
5166 032020                                BGNSFT
     032020  000210                        .WORD L10031-L$SOFT/2
5167
5168 032022                                GPRMD   RTMSG,RLT,D,177777,0,177777,YES
     032022  000052                        .WORD   T$CODE
     032024  032707                        .WORD   RTMSG
     032026  177777                        .WORD   177777
     032030  000000                        .WORD   T$LOLIM
     032032  177777                        .WORD   T$HILIM
5169 032034                                GPRMD   SRTMSG,SRLT,D,177777,0,177777,YES
     032034  031052                        .WORD   T$CODE
     032036  032532                        .WORD   SRTMSG
     032040  177777                        .WORD   177777
     032042  000000                        .WORD   T$LOLIM
     032044  177777                        .WORD   T$HILIM
5170 032046                                GPRML   FDCHK,DCKFG,1,YES
     032046  020130                        .WORD   T$CODE
     032050  033175                        .WORD   FDCHK
     032052  000001                        .WORD   1
5171 032054                                XFERF   5$
     032054  006044                        .WORD   T$CODE
5172 032056                                GPRMD   CHKLMT,CLMT,D,177777,0,128..,YES
     032056  032052                        .WORD   T$CODE
     032060  032551                        .WORD   CHKLMT
     032062  177777                        .WORD   177777
     032064  000000                        .WORD   T$LOLIM
     032066  000200                        .WORD   T$HILIM
5173 032070                          5$:   ;GPRMD  INMSG,TYT,D,177777,1,177777,YES          ;JSD REV A
5174 032070                                GPRML   DRPMS,DRFLG,1,YES
     032070  021130                        .WORD   T$CODE
     032072  033256                        .WORD   DRPMS
     032074  000001                        .WORD   1
5175 032076                                XFERF   3$
     032076  032044                        .WORD   T$CODE
5176 032100                                GPRMD   ERMSG,ELT,D,177777,0,177777,YES
```

DRIVE INFORMATION BUFFERS

```
        032100  001052                          .WORD   T$CODE
        032102  032623                          .WORD   ERMSG
        032104  177777                          .WORD   177777
        032106  000000                          .WORD   T$LOLIM
        032110  177777                          .WORD   T$HILIM
   5177 032112                                  GPRMD   SFTMSG,SEL,D,177777,0,177777,YES
        032112  023052                          .WORD   T$CODE
        032114  032637                          .WORD   SFTMSG
        032116  177777                          .WORD   177777
        032120  000000                          .WORD   T$LOLIM
        032122  177777                          .WORD   T$HILIM
   5178 032124                                  GPRMD   DERPMS,DCD,D,177777,0,177777,YES
        032124  036052                          .WORD   T$CODE
        032126  033312                          .WORD   DERPMS
        032130  177777                          .WORD   177777
        032132  000000                          .WORD   T$LOLIM
        032134  177777                          .WORD   T$HILIM
   5179 032136                                  GPRMD   SEMSG,SET,D,177777,0,177777,YES
        032136  002052                          .WORD   T$CODE
        032140  032721                          .WORD   SEMSG
        032142  177777                          .WORD   177777
        032144  000000                          .WORD   T$LOLIM
        032146  177777                          .WORD   T$HILIM
   5180 032150                                  GPRMD   DREMSG,DET,D,177777,0,177777,YES
        032150  025052                          .WORD   T$CODE
        032152  032734                          .WORD   DREMSG
        032154  177777                          .WORD   177777
        032156  000000                          .WORD   T$LOLIM
        032160  177777                          .WORD   T$HILIM
   5181 032162                          3$:     GPRML   STLMT,OPFLG,1,YES
        032162  024130                          .WORD   T$CODE
        032164  033221                          .WORD   STLMT
        032166  000001                          .WORD   1
   5182 032170                                  XFERF   2$
        032170  013044                          .WORD   T$CODE
   5183 032172                                  GPRMD   DAMSG,DAT,D,177777,1,177776,YES
        032172  003052                          .WORD   T$CODE
        032174  032747                          .WORD   DAMSG
        032176  177777                          .WORD   177777
        032200  000001                          .WORD   T$LOLIM
        032202  177776                          .WORD   T$HILIM
   5184 032204                                  GPRMD   SKMSG,SKT,D,177777,1,177776,YES
        032204  004052                          .WORD   T$CODE
        032206  032777                          .WORD   SKMSG
        032210  177777                          .WORD   177777
        032212  000001                          .WORD   T$LOLIM
        032214  177776                          .WORD   T$HILIM
   5185 032216                          2$:     GPRML   CHANGE,CHFLG,1,YES
        032216  010130                          .WORD   T$CODE
        032220  033017                          .WORD   CHANGE
        032222  000001                          .WORD   1
   5186 032224                                  XFERF   1$
        032224  107044                          .WORD   T$CODE
   5187 032226                                  GPRML   STIPMS,STIP,1,YES
        032226  034130                          .WORD   T$CODE
        032230  032502                          .WORD   STIPMS
        032232  000001                          .WORD   1
```

DRIVE INFORMATION BUFFERS

```
5188 032234                                    XFERF   6$
     032234  013044                            .WORD   T$CODE
5189 032236                                    GPRMD   MXBUF,MXB,D,177777,3,5120.,YES
     032236  011052                            .WORD   T$CODE
     032240  033053                            .WORD   MXBUF
     032242  177777                            .WORD   177777
     032244  000003                            .WORD   T$LOLIM
     032246  012000                            .WORD   T$HILIM
5190 032250                                    GPRMD   MINBUF,MNB,D,177777,3.,5120.,YES
     032250  022052                            .WORD   T$CODE
     032252  033064                            .WORD   MINBUF
     032254  177777                            .WORD   177777
     032256  000003                            .WORD   T$LOLIM
     032260  012000                            .WORD   T$HILIM
5191 032262                            6$:     GPRML   RDONLY,ROF,1,YES
     032262  026130                            .WORD   T$CODE
     032264  032571                            .WORD   RDONLY
     032266  000001                            .WORD   1
5192 032270                                    GPRML   RANPAT,RAN,1,YES
     032270  027130                            .WORD   T$CODE
     032272  032601                            .WORD   RANPAT
     032274  000001                            .WORD   1
5193 032276                                    XFERT   7$
     032276  006024                            .WORD   T$CODE
5194 032300                                    GPRMD   ONLONE,PAT,0,17,0,7,YES
     032300  050032                            .WORD   T$CODE
     032302  032611                            .WORD   ONLONE
     032304  000017                            .WORD   17
     032306  000000                            .WORD   T$LOLIM
     032310  000007                            .WORD   T$HILIM
5195 032312                            7$:     GPRMD   CMMSG,RDT,D,177777,0,128.,YES
     032312  006052                            .WORD   T$CODE
     032314  033340                            .WORD   CMMSG
     032316  177777                            .WORD   177777
     032320  000000                            .WORD   T$LOLIM
     032322  000200                            .WORD   T$HILIM
5196 032324                                    GPRMD   DEMSG,DDT,D,177777,0,175,YES
     032324  007052                            .WORD   T$CODE
     032326  032653                            .WORD   DEMSG
     032330  177777                            .WORD   177777
     032332  000000                            .WORD   T$LOLIM
     032334  000175                            .WORD   T$HILIM
5197 032336                                    GPRMD   MXHD,MXH,D,100,0,1,YES
     032336  012052                            .WORD   T$CODE
     032340  033075                            .WORD   MXHD
     032342  000100                            .WORD   100
     032344  000000                            .WORD   T$LOLIM
     032346  000001                            .WORD   T$HILIM
5198 032350                                    GPRMD   MINHD,MNH,D,100,0,1,YES
     032350  013052                            .WORD   T$CODE
     032352  033104                            .WORD   MINHD
     032354  000100                            .WORD   100
     032356  000000                            .WORD   T$LOLIM
     032360  000001                            .WORD   T$HILIM
5199 032362                                    GPRML   ASK,ANS,1,YES
     032362  037130                            .WORD   T$CODE
     032364  032442                            .WORD   ASK
```

DRIVE INFORMATION BUFFERS

```
     032366  000001                          .WORD   1
5200 032370                                  XFERF   15$
     032370  013044                          .WORD   T$CODE
5201 032372                                  GPRMD   MXCYL,MXC,D,177600,0,511.,YES
     032372  014052                          .WORD   T$CODE
     032374  033113                          .WORD   MXCYL
     032376  177600                          .WORD   177600
     032400  000000                          .WORD   T$LOLIM
     032402  000777                          .WORD   T$HILIM
5202 032404                                  GPRMD   MINCYL,MNC,D,177600,0,511.,YES
     032404  015052                          .WORD   T$CODE
     032406  033123                          .WORD   MINCYL
     032410  177600                          .WORD   177600
     032412  000000                          .WORD   T$LOLIM
     032414  000777                          .WORD   T$HILIM
5203 032416                          15$:    GPRMD   MXSEC,MXS,D,77,0,39.,YES
     032416  016052                          .WORD   T$CODE
     032420  033133                          .WORD   MXSEC
     032422  000077                          .WORD   77
     032424  000000                          .WORD   T$LOLIM
     032426  000047                          .WORD   T$HILIM
5204 032430                                  GPRMD   MINSEC,MNS,D,77,0,39.,YES
     032430  017052                          .WORD   T$CODE
     032432  033154                          .WORD   MINSEC
     032434  000077                          .WORD   77
     032436  000000                          .WORD   T$LOLIM
     032440  000047                          .WORD   T$HILIM
5205 032442                          1$:
5206
5207 032442                                  ENDSFT
                                             .EVEN
     032442                          L10031:
5208
5212
5213 032442  103  110  101  ASK:     .ASCIZ  /CHANGE VALUES OF MXCYL & MINCYL/
5214 032502  123  124  111  STIPMS:  .ASCIZ  #STIPULATE R/W XFER SIZE#
5215 032532  123  105  105  SRTMSG:  .ASCIZ  /SEEK RETRY LMT/
5216 032551  043  040  117  CHKLMT:  .ASCIZ  /# OF ERR DUMPED/
5217 032571  122  104  040  RDONLY:  .ASCIZ  /RD ONLY/
5218 032601  122  101  116  RANPAT:  .ASCIZ  /RAN PAT/
5219 032611  127  110  111  ONLONE:  .ASCIZ  /WHICH ONE/
5220 032623  110  122  104  ERMSG:   .ASCIZ  /HRD ERR LMT/
5221 032637  123  106  124  SFTMSG:  .ASCIZ  /SFT ERR LMT/
5222 032653  043  040  117  DEMSG:   .ASCIZ  /# OF DATA ERR RPT'D PER BUF/
5223 032707  122  105  124  RTMSG:   .ASCIZ  /RETRY LMT/
5224 032721  123  113  040  SEMSG:   .ASCIZ  /SK ERR LMT/
5225 032734  104  122  040  DREMSG:  .ASCIZ  /DR ERR LMT/
5226 032747  104  101  124  DAMSG:   .ASCIZ  /DATA XFER LMT (*10(10))/
5227 032777  123  113  040  SKMSG:   .ASCIZ  /SK LMT (*10(3))/
5228                         ;INMSG:  .ASCIZ  /TIME BETW REPORTS (MIN)/                    ;JSD REV A
5229 033017  103  110  101  CHANGE:  .ASCIZ  #CHANGE SEEK, R/W PARAMETERS#
5230 033053  115  101  130  MXBUF:   .ASCIZ  /MAX XFER/
5231 033064  115  111  116  MINBUF:  .ASCIZ  /MIN XFER/
5232 033075  115  101  130  MXHD:    .ASCIZ  /MAX HD/
5233 033104  115  111  116  MINHD:   .ASCIZ  /MIN HD/
5234 033113  115  101  130  MXCYL:   .ASCIZ  /MAX CYL/
5235 033123  115  111  116  MINCYL:  .ASCIZ  /MIN CYL/
```

DRIVE INFORMATION BUFFERS

```
5236 033133      123     124     101  MXSEC:   .ASCIZ  /STARTING MAX SEC/
5237 033154      123     124     101  MINSEC:  .ASCIZ  /STARTING MIN SEC/
5238 033175      104     101     124  FDCHK:   .ASCIZ  /DATA DMP ON DCK ERR/
5239 033221      104     122     117  STLMT:   .ASCIZ  /DROP DR ON OPER LMTS REACHED/
5240 033256      104     122     117  DRPMS:   .ASCIZ  /DROP DR ON ERR LMTS REACHED/
5241 033312      104     101     124  DERPMS:  .ASCIZ  /DATA MISCOMPARE LIMIT/
5242 033340      127     117     122  CMMSG:   .ASCIZ  /WORDS PER SECTOR COMPARED ON READ/
5243
5244                                  .EVEN
5248
5249 033402                                    ENDMOD
5250
5251 033402                           LASTAD
                                               .EVEN
     033402  000000                            .WORD   0
     033404  000000                            .WORD   0
     033406                           L$LAST::
5252
5253         000001                   .END
```

SYMBOL TABLE

```
ADDCOD  013314 G    BSEC6   027630    C$CEFG= 000045    DCRC   = 004000    EPS     004054
ADR     = 000020 G  BSEC7   027672    C$CLCK= 000062    DCRCER= 000022    ERLMT   010600
AFREAD  020520      BUF1    002436    C$CLEA= 000012    DCS    = 000104    ERLMTM  003322
AFWRCK  020540      BUF2    002440    C$CLOS= 000035    DDT    = 000016    ERMSG   032623
AGSTAT  020762      BUSAD   004132    C$CLP1= 000006    DECNT   002344    ERR    = 100000
ANS     = 000076    BVEC    002332    C$CVEC= 000036    DELMT   010614    ERRCNT= 000012
ARDHDR  020620      CART    002634    C$DCLN= 000044    DEMSG   032653    ERREX   021210
ASEEK   020456      CEND    024244    C$DODU= 000051    DERCNT= 000020    ERRHDR  004476
ASK     032442      CHANGE  033017    C$DRPT= 000024    DERMSG  003432    ERRVEC  002466
ASSEMB= 000010      CHFLG   = 000020  C$DU   = 000053    DERPMS  033312    ERR1    005070 G
AUTO    = 000066    CHKFNC  020362    C$EDIT= 000003    DERR   = 040000    ERR10   005646 G
AWRITF  021066      CHKLMT  032551    C$ERDF= 000055    DET    = 000052    ERR12   005716 G
BA      = 000002    CHKSEC  002342    C$ERHR= 000056    DIAGMC= 000000    ERR13   005724 G
BA16    = 000020    CKBDSC  027146    C$ERRO= 000060    DIFMSG  002623    ERR2    005076 G
BA17    = 000040    CKBDTK  027224    C$ERSF= 000054    DIFWD  = 000066    ERR3    005162 G
BBA     = 000110    CKDATA  023672    C$ERSO= 000057    DLT    = 010000    ERR4    005312 G
BCSADR= 000046      CKDERR  021350    C$ESCA= 000010    DLTCNT= 000026    ERR6    005364 G
BCSR    002330      CLEAR   027124    C$ESEG= 000005    DLYCNT  002500    ERR7    005424 G
BDA     = 000040    CLKACC  002512    C$ESUB= 000003    DMPBUF  026222    ERR8    005462 G
BDRSEL  002336      CLKADR  002316    C$ETST= 000001    DMPDCK  003265    ERR9    005602 G
BIT0    = 000001 G  CLKBFR  002510    C$EXIT= 000032    DNRDY   002664    ERT     004161
BIT00   = 000001 G  CLKCNT  002506    C$GETB= 000026    DOWCK  = 000076    EVL    = 000004 G
BIT01   = 000002 G  CLKFLD  002514    C$GETW= 000027    DPHOUR= 000070    EXHAUS  002773
BIT02   = 000004 G  CLKFRQ  002312    C$GMAN= 000043    DPMIN  = 000071    EXIT    021152
BIT03   = 000010 G  CLKINI  013522    C$GPHR= 000042    DRBT   = 000010    EXIT1   021204
BIT04   = 000020 G  CLKSON  002504    C$GPLO= 000030    DRBUF   030362    EXIT2   016766
BIT05   = 000040 G  CLKST   013674    C$GPRI= 000040    DRDRV   023450    EXP     004254
BIT06   = 000100 G  CLKTIK  017040 G  C$INIT= 000011    DRDY   = 000001    E$END  = 002100
BIT07   = 000200 G  CLKTYP  002314    C$INLP= 000020    DREMSG  032734    E$LOAD= 000035
BIT08   = 000400 G  CLMT    = 000064  C$MANI= 000050    DRFLG  = 000042    E.BA    002422
BIT09   = 001000 G  CLNCOD  013116 G  C$MEM  = 000031    DRLMT   010650    E.CS    002420
BIT1    = 000002 G  CLRDAT  011662    C$MSG  = 000023    DRMSG   032011    E.DA    002424
BIT10   = 002000 G  CMMSG   033340    C$OPEN= 000034    DRNM    004030    E.MP    002426
BIT11   = 004000 G  CMRD    010612    C$PNTB= 000014    DROP    004276    E.MP1   002430
BIT12   = 010000 G  CMSK    002270    C$PNTF= 000017    DROPCO  013400 G  E.MP2   002432
BIT13   = 020000 G  CNT     = 000012  C$PNTS= 000016    DRPMS   033256    FASCII  002456
BIT14   = 040000 G  CNTFLG  002454    C$PNTX= 000015    DRPRS   002253    FASPNT  002460
BIT15   = 100000 G  CNTLR1  002320    C$QIO  = 000377    DRSEL  = 000106    FDCHK   033175
BIT2    = 000004 G  CNTLR2  002322    C$RDBU= 000007    DRST   = 000013    FILINF  013442
BIT3    = 000010 G  CNTYPE  031726    C$REFG= 000047    DRTYPE  031760    FILTD   013466
BIT4    = 000020 G  CONWR   024642    C$RESE= 000033    DRUT    002252    FINDBF  012516
BIT5    = 000040 G  CRDY    = 000200  C$REVI= 000003    DRVER   003061    FINERR  021312
BIT6    = 000100 G  CRLBA   002565    C$RFLA= 000021    DSE    = 000400    FMTDT   010103
BIT7    = 000200 G  CRLCS   002535    C$RPT  = 000025    DSPCOD  010674 G  FMTS1   007732
BIT8    = 000400 G  CRLDA   002577    C$SEFG= 000046    DWCNT   002462    FMTS1A  010015
BIT9    = 001000 G  CRLMP   002611    C$SPRI= 000041    DWCNT1  002464    FMTS1B  010035
BMP     = 000042    CS      = 000000  C$SVEC= 000037    EF.CON= 000036 G  FMTS2   010070
BOE     = 000400 G  CSR     = 000000  C$TPRI= 000013    EF.NEW= 000035 G  FMTS2A  010134
BPRIOR  002334      CSRMSG  031733    C.HDR   002434    EF.PWR= 000034 G  FMTS2B  010223
BRMSG   031747      CSTUFF  024410    DA     = 000004    EF.RES= 000037 G  FMTS3   010260
BSECPT= 000112      CYL     002300    DALMT   010604    EF.STA= 000040 G  FMTS3A  010367
BSEC0   027314      CYLMSK  002264    DAMSG   032747    ELT    = 000002    FMTS4   010424
BSEC1   027356      C$AU    = 000052  DAT    = 000006    END     012170    FMTS5   010517
BSEC2   027420      C$AUTO= 000061    DATCER= 000074    ENDBUF  031642    FMTXS   007673
BSEC3   027462      C$BRK  = 000022   DCD    = 000074    ENDINI  013670    FMT1    006610
BSEC4   027524      C$BSEG= 000004    DCDMSG  003407    ENDOFP  030354    FMT1A   006645
BSEC5   027566      C$BSUB= 000002    DCKFG  = 000040    ENDWR   024704    FMT10   007277
```

SYMBOL TABLE

```
FMT10A  007312        GLBTXT  002516 G      I$AU  = 000041        L$ETP   002102 G      MAIN    014726
FMT10B  007363        GOERRX  020352        I$AUTO= 000041        L$EXP1  002046 G      MAXWC   002442
FMT11   007371        GOFIN   020356        I$CLN = 000041        L$EXP4  002064 G      MBDMSC  003667
FMT12   007411        GSBIT = 000003        I$DU  = 000041        L$EXP5  002066 G      MDCER   003235
FMT13   007421        GSTAT = 000004        I$HRD = 000041        L$HARD  031646 G      MDERS   003070
FMT13D  007436        GSTFNC  015676        I$INIT= 000041        L$HIME  002120 G      MDHEDR  002000 G
FMT14   007463        GWCDA   025714        I$MOD = 000041        L$HPCP  002016 G      MDSER   003116
FMT14A  007472        G$CNTO= 000200        I$MSG = 000041        L$HPTP  002022 G      MFUNC   002547
FMT14B  007504        G$DELM= 000372        I$PROT= 000040        L$HW    010560 G      MHDER   003252
FMT14C  007501        G$DISP= 000003        I$PTAB= 000041        L$ICP   002104 G      MINBUF  033064
FMT15   007546        G$EXCP= 000400        I$PWR = 000041        L$INIT  010764 G      MINCYL  033123
FMT17   006632        G$HILI= 000002        I$RPT = 000041        L$LADP  002026 G      MINHD   033104
FMT18   007666        G$LOLI= 000001        I$SEG = 000041        L$LAST  033406 G      MINSEC  033154
FMT2    006654        G$NO  = 000000        I$SETU= 000041        L$LOAD  002100 G      MINUTE  002414
FMT2A   006675        G$OFFS= 000400        I$SFT = 000041        L$LUN   002074 G      MK    = 000001
FMT3    006717        G$OFSI= 000376        I$SRV = 000041        L$MREV  002050 G      MNB   = 000044
FMT3A   006760        G$PRMA= 000001        I$SUB = 000041        L$NAME  002000 G      MNC   = 000032
FMT4    007044        G$PRMD= 000002        I$TST = 000041        L$PRIO  002042 G      MNH   = 000026
FMT5    007077        G$PRML= 000000        J$JMP = 000167        L$PROT  010756 G      MNS   = 000036
FMT6    007113        G$RADA= 000140        KILLDC  002310        L$PRT   002112 G      MP    = 000006
FMT7    007141        G$RADB= 000000        LCLK    013564        L$REPP  002062 G      MPT     003756
FMT8    007155        G$RADD= 000040        LCLKCH  011100        L$REV   002010 G      MRDER   003212
FMT9    007165        G$RADL= 000120        LDFUNC  016524        L$RPT   010700 G      MRLCS   002525
FMT9A   007223        G$RADO= 000020        LIMIT   010576        L$SOFT  032022 G      MRT     004150
FRMT16  007624        G$XFER= 000004        LINE1   006014        L$SPC   002056 G      MSFER   003042
FRQCHK  013622        G$YES = 000010        LINE2   006200        L$SPCP  002020 G      MSKER   003031
FRQSO   013654        HCE   = 040000        LINE3   006270        L$SPTP  002024 G      MST     004177
FUNC  = 000044        HCRC  = 004000        LIST    022566        L$STA   002030 G      MSTART  004360
F$AU  = 000015        HCRCER= 000024        LOE   = 040000 G      L$SW    010576 G      MST1    004214
F$AUTO= 000020        HDHOME  025620        LONUM   002262        L$TEST  002114 G      MSWRPK  004400
F$BGN = 000040        HDRFND  002340        LOT   = 000010 G      L$TIML  002014 G      MTCR    004777
F$CLEA= 000007        HEAD  = 000100        LPS     004041        L$UNIT  002012 G      MTDCRC  004327
F$DU  = 000016        HINUM   002260        LSTDA = 000064        L10000  005074        MTDLT   004334
F$END = 000041        HNF   = 010000        LSTDR1  002324        L10001  005160        MTDRV   004353
F$HARD= 000004        HNFERR= 000032        LSTDR2  002326        L10002  005310        MTEST   014464
F$HW  = 000013        HOE   = 100000 G      LSTHDR= 000050        L10003  005362        MTGS    005007
F$INIT= 000006        HOUR    002416        L$ACP   002110 G      L10004  005422        MTHCRC  004321
F$JMP = 000050        HPTCOD  010556 G      L$APT   002036 G      L10005  005460        MTHNF   004314
F$MOD = 000000        HRDPRM  031644 G      L$AU    013314 G      L10006  005600        MTNXM   004346
F$MSG = 000011        HWSEC   003720        L$AUT   002070 G      L10007  005644        MTOPI   004341
F$PROT= 000021        IBE   = 010000 G      L$AUTO  012644 G      L10010  005714        MTRD    005047
F$PWR = 000017        IDU   = 000040 G      L$CCP   002106 G      L10011  005722        MTRH    005027
F$RPT = 000012        IER   = 020000 G      L$CLEA  013116 G      L10012  006012        MTRNH   005057
F$SEG = 000003        ILLEG   003770        L$CO    002032 G      L10013  010574        MTSK    005017
F$SOFT= 000005        INBAD   025500        L$DEPO  002011 G      L10014  010674        MTWR    005037
F$SRV = 000010        INCALL  002476        L$DESC  002122 G      L10015  010754        MVCER   003132
F$SUB = 000002        INIEND  012642        L$DESP  002076 G      L10017  012642        MVEC    004004
F$SW  = 000014        INITCO  010764 G      L$DEVP  002060 G      L10020  013114        MXB   = 000022
F$TEST= 000001        INSMEM  004711        L$DISP  010676 G      L10021  013312        MXBUF   033053
GDDAT   002402        INTEN = 000100        L$DLY   002116 G      L10022  013376        MXC   = 000030
GETDST  024362        INTERV  002406        L$DTP   002040 G      L10023  013440        MXCYL   033113
GETFNC  015244        INTR1   017046 G      L$DTYP  002034 G      L10024  030356        MXH   = 000024
GHDR    022246        INTR2   017056        L$DU    013400 G      L10025  017036        MXHD    033075
GLBDAT  002242 G      ISORST  024376        L$DUT   002072 G      L10026  017044        MXS   = 000034
GLBEQA  002242 G      ISR   = 000100 G      L$DVTY  002230 G      L10027  021310        MXSEC   033133
GLBERR  005070 G      ISSUE   016472        L$EF    002052 G      L10030  031726        NILCLK  011274
GLBSUB  013442 G      IXE   = 004000 G      L$ENVI  002044 G      L10031  032442        NOCLK   004530
```

SYMBOL TABLE

```
NOCRDY  002654          PRIO6 = 000300 G        SFTPRM  032020 G        TEMP8   002366          T.DRIV  002254
NOCTLR  004673          PRIO7 = 000340 G        SIGN  = 000004          TEMP9   002370          T.DRP   010640
NODRIV  004016          PRPOS = 000124          SKCNT = 000000          TICK    002410          T.MNB   010642
NOLOAD  003573          PWRCH   011344          SKCNT1= 000054          TIME    002516          T.MNC   010630
NOPWR   004101          PWRFLG  002446          SKDON = 000001          TRACK   003305          T.MNH   010624
NORDOC  004416          RAN   = 000056          SKECNT= 000016          TRERR = 000072          T.MNS   010634
NORDY   002704          RAND    024454          SKFNC   015716          TRPFLG  002450          T.MXB   010620
NOREPT  004566          RANPAT  032601          SKMS  = 000020          TRPHAN  013764          T.MXC   010626
NOREV   003631          RCD     004265          SKLMT   010606          TSTDRV  002250          T.MXH   010622
NOTRDY  004636          RCNT    022546          SKMSG   032777          TYINT   010610          T.MXS   010632
NWRTS   002736          RDBDSC  022604          SKRD    015410          TYPDR = 000006          T.PAT   010656
NXM   = 020000          RDDFNC  016430          SKRDRD  015434          TYT   = 000012          T.RAN   010654
NXMCNT= 000034          RDHDR = 000010          SKRETR= 020700          T$ARGC= 000001          T.ROF   010652
ODRDRV  023444          RDHFNC  016346          SKRH    015550          T$CODE= 017052          T.SLT   010660
ONLONE  032611          RDNHC   015634          SKT   = 000010          T$ERRN= 000307          T.STA   010646
OPCALL  002474          RDONLY  032571          SKTO  = 010000          T$EXCP= 000000          T.STIP  010664
OPFLG = 000050          RDT   = 000014          SKWRT   015354          T$GMAN= 000000          T.WCK   010666
OPI   = 002000          READ  = 000014          SMSG    002761          T$HILI= 000047          T1      014464 G
OPICNT= 000030          RECNT   002242          SMSK    002272          T$LAST= 000001          UAM   = 000200 G
OPROK   015470          REGEN   002306          SOFTCS= 000116          T$LOLI= 000000          UDERR   003007
OVER    003453          REPORT  013772          SOPLMT  003611          T$LSYM= 010000          UNLOAD  003547
O$APTS= 000000          REQ     003510          SPE   = 004000          T$LTNO= 000001          UPDATE  016636 G
O$AU  = 000001          RESTAR  012256          SPTCOD  010574 G        T$NEST= 177777          UUT     002444
O$BGNR= 000001          RETRY = 000036          SRLT  = 000062          T$NSO = 000000          VC    = 001000
O$BGNS= 000001          RLT   = 000000          SRTMSG  032532          T$NS1 = 000005          VECMSG  032002
O$DU  = 000001          RNTEMP  002404          STDMP   026736          T$PTNU= 000000          VECT  = 000002
O$ERRT= 000000          ROF   = 000054          STFLG   002452          T$SAVL= 177777          VECT1   002372
O$GNSW= 000001          RPS     004066          STIP  = 000070          T$SEGL= 177777          VECT2   002374
O$POIN= 000001          RPTCOD  010700 G        STIPMS  032502          T$SUBN= 000000          WCK   = 000072
O$SETU= 000000          RSEEK = 000114          STLMT   033221          T$TAGL= 177777          WDE   = 100000
PAT   = 000060          RTMSG   032707          STWRT   024726          T$TAGN= 010032          WGE   = 002000
PATLST  027734          RTYPE = 000052          ST1     002470          T$TEMP= 000000          WGEST   003157
PATO    027754          RT1     004231          ST2     002472          T$TEST= 000001          WHY     002246
PAT1    030014          RWCNT   002244          SUR     002302          T$TSTM= 177777          WL    = 020000
PAT2    030054          RXFR1 = 000002          SVCGBL= 000000          T$TSTS= 000001          WRBUF   022320
PAT3    030114          RXFR2 = 000004          SVCINS= 000000          T$$AU = 010022          WRCHK = 000002
PAT4    030154          RXFR3 = 000060          SVCSUB= 177777          T$$AUT= 010020          WRINIT  002274
PAT5    030214          SAVE    017064          SVCTAG= 000000          T$$CLE= 010021          WRIPG = 000122
PAT6    030254          SEC     002304          SVCTST= 177777          T$$DU = 010023          WRITE = 000012
PAT7    030314          SECMSK  002266          SWSEC   003740          T$$HAR= 010030          WRPACK  024552
PEROTH  014736          SECOND  002412          SYSCLK  002502          T$$HW = 010013          WRPOS   002276
PNT   = 001000 G        SEEK  = 000006          SYSMSK  002256          T$$INI= 010017          WRTCKT  015654
PRFLGS= 000056          SEL   = 000046          S$LSYM= 010000          T$$MSG= 010012          WRTFNC  016360
PRGER   002732          SELMT   010602          TAGX    015102          T$$PRO= 010016          WTRDY   024270
PRI   = 002000 G        SEMSG   032721          TDR   = 000120          T$$RPT= 010015          WXFR1 = 000006
PRIOR = 000004          SERLMT  003344          TELCYL  006450          T$$SOF= 010031          WXFR2 = 000010
PRIOR1  002376          SERNM1= 000100          TEMP0   002346          T$$SRV= 010027          WXFR3 = 000062
PRIOR2  002400          SERNM2= 000102          TEMP1   002350          T$$SW = 010014          XCHFLG  010616
PRIO0 = 000000 G        SET   = 000004          TEMP2   002352          T$$TES= 010024          XEXIT   020452
PRIO1 = 000040 G        SEXHAU  003523          TEMP3   002354          T.ANS   010672          X$ALWA= 000000
PRIO2 = 000100 G        SFEMSG  003365          TEMP4   002356          T.CLT   010662          X$FALS= 000040
PRIO3 = 000140 G        SFLMT   010644          TEMP5   002360          T.DCD   010670          X$OFFS= 000400
PRIO4 = 000200 G        SFTCNT= 000014          TEMP6   002362          T.DCK   010636          X$TRUE= 000020
PRIO5 = 000240 G        SFTMSG  032637          TEMP7   002364

. ABS.  033406      000
        000000      001
```

SYMBOL TABLE

ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 30584 WORDS (120 PAGES)
DYNAMIC MEMORY: 20060 WORDS (77 PAGES)
ELAPSED TIME: 00:30:10
CNRLKA.BIN,CNRLKA.LST/-SP=SVC34.MLB/ML,CNRLKA.MAC