11/Z1+
RL01/02

RL01/02 DRIVE TEST 2

CNRLJA0

AH-T750A-MC
FICHE 1 OF 1

APR 1984
digital
Made In USA

```
                        .TITLE CNRLJA RL01/2 DRIVE TEST 2
        000001          PART2==1
000000                  .ENABLE ABS
                        .ENABLE AMA
                        .NLIST TOC
                        .REM @
```

IDENTIFICATION

PRODUCT CODE: AC T749A MC

PRODUCT NAME: CNRLJA0 RL01/2 DRIVE TEST 2

PRODUCT DATE: DECEMBER 19, 1983

MAINTAINER: ISS DIAGNOSTIC SERVICES

AUTHOR: JAMES S. DOUCETTE

REVISION HISTORY

CHANGES MADE TO CZRLJBO IN PRODUCING CNRLJAO FOR THE SBC-11/21. (FALCON-PLUS), DEC. 19, 1983. CHANGES ARE IDENTIFIED BY ";JSD REV A'.

1. CHANGED THE FORM OF THE ARGUMENT TO ALL "DELAY" MACRO CALLS FROM #<VALUE> TO <VALUE>. THE FORMER GAVE ASSEMBLY ERRORS UNDER THE VAX/VMS DEVELOPMENT ENVIRONMENT (MCR MAC).

2. CHANGED THE GENERAL OPERATING PRIORITY OF THE PROGRAM FROM LEVEL 7 TO LEVEL 6 TO ALLOW THE "BREAK' KEY TO INVOKE ODT. (THE TRAP HANDLER AND DEVICE INTERRUPT SERVICE ROUTINES STILL RUN BRIEFLY AT LEVEL 7).

3. SET VECTOR 140 WITH THE ADDRESS OF ODT IN ROM (170000).

TABLE OF CONTENTS

1.0 GENERAL INFORMATION

1.1 PROGRAM ABSTRACT

1.1.1 STRUCTURE OF PROGRAM

THIS DIAGNOSTIC 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 DIAGNOSTICS PROGRAM INTERFACES TO THE ENVIRONMENT AS IT EXECUTES. (IN THIS DOCUMENT, "CNDP+" REFERS TO THE FALCON SPECIFIC XXDP+ SYSTEM).

WHEN THIS DIAGNOSTIC IS STARTED AT ADDRESS 200, 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

THIS PROGRAM TESTS AND EXERCISES RL01/02 DISK DRIVES RL11/RLV11 CONTROLLERS (4 DRIVES PER CONTROLLER). THE ENTIRE PROGRAM IS RUN ON THE FIRST DRIVE BEFORE STARTING ON THE SECOND. THE PROGRAM STARTS BY TESTING THE SIMPLEST FUNCTIONS FIRST USING THE LOGIC TESTED IN EARLIER TESTS TO TEST MORE COMPLEX FUNCTIONS.

THIS PROGRAM TESTS THE RL01/02 OUTER AND INNER GUARD BAND DETECTION. SEEK OPERATIONS UNDERGO A BROAD RANGE OF TESTING USING SINGLE DIFFERENCES, PROCEEDING TO SEEKS OF GREATER DIFFERENCES.

1.2 SYSTEM REQUIREMENTS

1.2.1 HARDWARE REQUIREMENTS

- SBC-11/21+ PROCESSOR, 28KW MEMORY, JUMPERED FOR MEMORY MAP C
- CONSOLE DEVICE (LA30,LA36,VT50,ETC.)
- 1 OR 2 RL11/RLV11 CONTROLLER(S) WITH:

  1 - 8 RL01 DRIVES WITH RL01K CARTRIDGES CONTAINING A 'BAD SECTOR FILE'
  1 8 RL02 DRIVES WITH RL02K CARTRIDGES CONTAINING A 'BAD SECTOR FILE'

- CNDP+ (XXDP+) LOAD DEVICE (RL02, RX02, ETC.)
- LINE PRINTER (OPTIONAL)

1.2.2 SOFTWARE REQUIREMENTS

CNRLJA0 RL01/02 DRIVE TEST PART 2 (FORMERLY CZRLDB0)

1.3 RELATED DOCUMENTS AND STANDARDS

RL01/02 DISK SUBSYSTEM USER'S GUIDE (EK-RL01 UG 002)
XXDP+/SUPERVISOR USER'S MANUAL

1.4 DIAGNOSTIC HIERARCHY PREREQUISITES

THE RL01/02 SUBSYSTEM SHOULD HAVE SUCCESSFULLY RUN THE FOLLOWING PROGRAMS:

CVRLAB0	RLV11 RL01 DISKLESS TEST (RLV11 ONLY)
CNRLGA0	RL11/RLV11 RL01/02 CONTROLLER TEST (PART 1)
CNRLHA0	RL11/RLV11 RL01/02 CONTROLLER TEST (PART 2)
CNRLIA0	RL01/02 DRIVE TEST (PART 1)

1.5 ASSUMPTIONS

THE HARDWARE OTHER THAN THE RL01/02 SUBSYSTEM IS ASSUMED TO WORK PROPERLY. FALSE ERRORS MAY BE REPORTED IF THE PROCESSOR, ETC., DO NOT FUNCTION PROPERLY.

2.0 OPERATING INSTRUCTIONS

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:

```
CNMDYAO 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.

THE FULL PRINT OUT FROM THE ABOVE DIALOGUE MIGHT LOOK LIKE THIS
(O=OPERATOR, D=DIAGNOSTIC):

```
                                                    BY
                                                    WHOM
                                                    ENTERED:
                                                     - -

.R NRLJAO                                           O
DRS LOADED                                          D
DIAG. RUN-TIME SERVICES REV. D APR 79               D
CNRLJ-A-0                                           D
CNRLJ TESTS OUTER & INNER GUARD                     D
BAND DETECTION AND SEEK OPERATIONS
UNIT IS RL01, RL02                                  D
DR>STA/PASS:1/FLAGS:HOE                             D.0

# UNITS (D) ? 2                                     D.0

UNIT 0                                              D
RL11 (L) Y ?                                        D.0
BUS ADDRESS (O) 174400 ?                            D.0
VECTOR (O) 160 ?                                    D.0
DRIVE (O) 0 ?                                       D.0
DRIVE TYPE = RL01 (L) Y ?                           D.0
BR LEVEL (O) 5 ?                                    D.0

UNIT 1                                              D
RL11 (L) Y ?                                        D.0
BUS ADDRESS (O) 174400 ?                            D.0
VECTOR (O) 160 ?                                    D.0
DRIVE (O) 0 ? 1                                     D.0
DRIVE TYPE = RL01 (L) ? N                           D.0 (N=RL02)
BR LEVEL (O) 5 ?                                    D.0

CHANGE SW (L) ? Y                                   D.0

USE ALL CYL (L) N ?                                 D.0
USE ALL SECT (L) N ?                                D.0
LOW SEEK LIMIT (L) N ?                              D.0
UPPER SEEK LIMIT (L) N ?                            D.0
USE ONLY ONE SURF (L) N ?                           D.0
INPUT ERROR LIMIT (D) 20 ?                          D.0
DATA CMP ERR  1T (D) 10 ?                           D.0


CNRLJ HRD ERR 00004 TST 003 SUB 002 PC:004130
ERR HLT

DR>PRO/FLAGS:IER:LOE:HOE=0                          D.0
```

```
**********************************************************
AT THIS POINT THE DIAGNOSTIC IS LOOPING ON THE
ERROR WITHOUT PRINTING ANYTHING.  YOU CAN SCOPE
THE ERROR UNTIL YOU HAVE LOCATED IT, THEN ↑C OUT.
**********************************************************

↑C                                                  0

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

CHANGE SW (L) ? N                                   D.0

CNRLJ 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 SEQUENTIAL 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.

TO EXECUTE 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 THE 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	START 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

```
694
695                     IBE INHIBIT BASIC ERROR REPORTS
696
697                     IXE INHIBIT EXTENDED ERROR REPORTS
698
699                     PRI DIRECT ALL MESSAGES TO A LINE PRINTER
700
701                     PNT PRINT NUMBER OF TEST BEING EXECUTED
702
703                     BOE BELL ON ERROR
704
705                     UAM RUN IN UNATTENDED MODE, BYPASSING MANUAL INTERVENTION TESTS
706
707                     ISR INHIBIT STATISTICAL REPORTS
708
709                     IDU INHIBIT DROPPING OF UNITS BY DIAGNOSTIC
710
711                     ADR EXECUTE AUTODROP CODE
712
713                     LOT LOOP ON TEST
714
715                     EVL EVALUATE
716
717                 THE FLAGS NAMED OR EQUATED TO 1 ARE SET, THOSE EQUATED TO 0 ARE
718                 CLEARED.  A FLAG NOT SPECIFIED IS CLEARED.  IF THE FLAGS SWITCH IS
719                 NOT GIVEN ALL FLAGS ARE CLEARED.
720
721                 "EOP-INCR" IS A DECIMAL NUMBER INDICATING HOW OFTEN (IN TERMS OF
722                 PASSES) IT IS DESIRED THAT THE END OF PASS MESSAGE BE PRINTED.  THE
723                 DEFAULT IS AT THE END OF EVERY PASS.
724
725                 ********************************************************************
726                 RES(TART)/TEST:TEST-LIST/PASS:PASS-CNT/FLAGS:FLAG-LIST/EOP:EOP-INCR/UNITS:UNIT LIST
727                 ********************************************************************
728
729                 THE DIAGNOSTIC IN CORE IS EXECUTED IN ACCORDANCE WITH THE SWITCHES
730                 SPECIFIED.  HOWEVER, NEW "P-TABLES" ARE NOT BUILT.  INSTEAD, THE
731                 ONES IN CORE ARE USED.
732
733                 THE QUESTION "CHANGE SW?" IS ASKED AND THE ANSWERS GIVEN BECOME THE
734                 NEW DEFAULTS.  THE COMMAND MAY BE ISSUED WHEN COMAND MODE HAS BEEN
735                 ENTERED VIA A) DIAGNOSTIC IS FINISHED B) HALT ON ERROR C)
736                 CONTROL/C.
737
738                 THE SWITCH ARGUMENTS ARE AS IN THE START COMMAND EXCEPT:
739
740                     1.  "UNIT-LIST" IS A SEQUENCE OF LOGICAL UNIT NUMBERS RANGING
741                         FROM 1 THRU N (N = NUMBER OF UNITS BEING TESTED) SPECIFYING
742                         WHICH UNITS ARE TO BE TESTED.  THE LOGICAL UNIT NUMBER
743                         DESIGNATES THE POSITION OF THE P-TABLE IN CORE, ACCORDING
744                         TO THE ORDER IN WHICH THEY WERE BUILT.  THE UNITS SPECIFIED
745                         MUST NOT HAVE BEEN DROPPED BY THE OPERATOR DROP COMMAND.
746                         THE UNIT-LIST DEFAULTS TO "ALL THAT HAVE NOT BEEN DROPPED
747                         BY OPERATOR COMMAND".  THE EFFECT OF THE UNIT-LIST LASTS
748                         UNTIL THE NEXT START (WHERE IT IS AUTOMATICALLY RESET TO
749                         'ALL") OR THE NEXT RESTART.
```

```
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
```

2. ALL UNSPECIFIED FLAG SETTINGS ARE UNCHANGED.

****
CON(TINUE)/PASS:<PASS-CNT/FLAGS:<FLAG-LIST>
****

COMMAND MODE MUST HAVE BEEN ENTERED DUE TO A HALT ON ERROR OR A CONTROL/C. THE EFFECT OF THE COMMAND IS TO GO TO THE BEGINNING OF THE TEST THAT WAS BEING EXECUTED WHEN THE HALT OR CONTROL/C TOOK PLACE. SOFTWARE DIALOGUE MAY OPTIONALLY BE RE-EXECUTED. HARDWARE PARAMETERS MAY NOT BE CHANGED.

THE SWITCH ARGUMENTS ARE AS IN THE START COMMAND EXCEPT:

1. DEFAULT FOR PASS-CNT IS THE UNSATISFIED PASS-CNT FROM THE PREVIOUS START OR RESTART

2. UNSPECIFIED FLAG SETTINGS ARE UNCHANGED

****
PRO(CEED)/FLAGS:<FLAG-LIST>
****

COMMAND MODE MUST HAVE BEEN ENTERED VIA A HALT ON ERROR. THE EFFECT OF THE COMMAND IS TO BEGIN EXECUTION AT THE LOCATION FOLLOWING THE ERROR CALL. NEITHER HARDWARE NOR SOFTWARE PARAMETERS MAY BE ALTERED.

THE SWITCH ARGUMENTS ARE THE SAME AS THE START COMMAND EXCEPT:

1. UNSPECIFIED FLAG SETTINGS ARE UNCHANGED

****
EXIT
****

RETURN TO CNDP> PROMPT MODE.

****
DRO(P)/UNITS:UNIT-LIST
****

THE UNITS SPECIFIED ARE DROPPED FROM TESTING UNTIL THEY ARE ADDED BACK OR UNTIL A START COMMAND IS GIVEN. A DROP CANNOT BE FOLLOWED BY A PROCEED.

THERE IS ALSO A "DROP" MACRO INTERNAL TO THE DIAGNOSTIC, WHICH GIVES THE FACILITY OF AUTO-DROPPING. THE DURATION OF A PROGRAM DROP, HOWEVER, IS ONLY UNTIL THE NEXT START OR RESTART.

```
808
809                     **********
810                     ADD/UNITS:UNIT-LIST
811                     **********
812
813                     THE UNITS SPECIFIED ARE ADDED BACK (THEY MUST HAVE BEEN  PREVIOUSLY
814                     DROPPED  BY  THE DROP COMMAND) TO THE TEST SEQUENCE.  AN ADD CANNOT
815                     BE FOLLOWED BY A PROCEED.
816
817
818                     *******
819                     PRI(NT)
820                     *******
821
822                     ALL STATISTICS TABLES ACCUMULATED BY THE  DIAGNOSTIC  ARE  PRINTED.
823                     THE ISR (INHIBIT STATISTICAL REPORTING) FLAG IS CLEARED.
824
825
826                     ****************************
827                     DIS(PLAY)/UNITS:<UNIT-LIST>
828                     ****************************
829
830                     THE HARDWARE P-TABLES FOR ALL UNITS UNDER TEST ARE PRINTED  OUT  IN
831                     THE FORMAT IN WHICH THEY WERE ENTERED.  ANY UNITS THAT WERE DROPPED
832                     BY THE OPERATOR "DROP" COMMAND ARE SO DESIGNATED.
833
834
835                     *******
836                     FLA(GS)
837                     *******
838
839                     THE CURRENT SETTINGS OF ALL FLAGS ARE PRINTED.
840
841
842                     ********
843                     ZFL(AGS)
844                     ********
845
846                     ALL FLAGS ARE CLEARED.
847
848
849           2.4             EXTENDED P-TABLE DIALOGUE
850                           -------- ------ --------
851
852                     THE FULL CAPABILITY OF THE HARDWARE DIALOGUE  IS  REVEALED  BY  THE
853                     FOLLOWING DISCUSSION OF WHAT HAPPENS INTERNALLY.
854
855                     AS SOON AS THE QUESTION "# UNITS?" IS ANSWERED (WITH THE NUMBER N),
856                     SPACE  IN  CORE IS ALLOCATED FOR "N" P-TABLES.  ALL OF THE P-TABLES
857                     ARE OF THE SAME FORMAT, AND THERE IS  A  ONE-TO-ONE  CORRESPONDENCE
858                     BETWEEN  THE  HARDWARE  PARAMETER  QUESTIONS  AND  THE SLOTS IN THE
859                     P-TABLE FORMAT.
860
861                     IN GIVING A STRING OF VALUES, COMMAS WITHOUT INTERVENING VALUES MAY
862                     BE USED TO INDICATE A REPETITION OF THE LAST NAMED VALUE.
```

A STRING OF VALUES MAY BE GIVEN AS A RANGE (6 10 FOR EXAMPLE). IF THE VALUES REPRESENT PURE NUMERICAL DATA, THIS SAMPLE RANGE TRANSLATES TO THE STRING 6,7,8,9,10 (AN INCREMENT OF 1). IF THE VALUES ARE ADDRESSES, THE SAMPLE RANGE TRANSLATES TO THE STRING 6,8,10 (AN INCREMENT OF 2).

NOW LET US SEE HOW WE COULD USE THESE CAPABILITIES TO CONSTRUCT A SET OF P-TABLES. ASSUME THAT WE HAVE 8 RL UNITS, AND THAT THERE ARE FIVE (5) HARDWARE PARAMETERS FOR EACH (5 SLOTS IN THE P-TABLE, 5 HARDWARE QUESTIONS IN THE DIALOGUE).

FOLLOWING IS THE DIALOGUE FOR THIS 8 RLOX DRIVE SYSTEM. THIS SYSTEM HAS TWO (2) RL11 TYPE CONTROLLERS ALL TO BE SET AT "BR LEVEL' 5. THE FIRST 4 DRIVES ARE RL01'S AND THE LAST 4 DRIVES ARE RL02'S (ON THE SECOND CONTROLLER):

```
# UNITS (D) ? 8

UNIT 0
RL11 (L) Y ?
BUS ADDDRESS (O) 174400 ?
VECTOR (O) 160 ?
DRIVE (O) 0 ? 0-3
DRIVE TYPE = RL01 (L) Y ?
BR LEVEL (O) 5 ?

UNIT 4
RL11 (L) Y ?
BUS ADDRESS (O) 174400 ? 175400
VECTOR (O) 160 ? 164
DRIVE (O) 0 ? 0-3
DRIVE TYPE = RL01 (L) Y ? N
BR LEVEL (O) 5 ?
```

THE FIRST TIME THRU THE P-TABLE QUESTIONS THE DEFAULT VALUES ARE USED FOR THE CONTROLLER TYPE (QUESTION #1), CSR ADDRESS OF THE CONTROLLER (QUESTION #2), THE CONTROLLER VECTOR ASSIGNMENT (QUESTION #3), THE DRIVE TYPE (QUESTION #5), AND THE "BR LEVEL" (QUESTION #6). THE ACTUAL UNIT NUMBERS OF THE RL01'S FOR QUESTION #4 WAS ASSIGNED 0 THRU 3 FOR THE FIRST 4 P-TABLE SLOTS.

THE SECOND TIME THRU THE P-TABLE QUESTIONS (FOR THE RL02 ASSIGNMENT ON THE SECOND CONTROLLER), THE FIRST QUESTION DEFAULTED TO "RL11" TYPE CONTROLLER. THE SECOND QUESTION WAS ANSWERED TO REFLECT THE CHANGE IN CSR ADDRESS FOR THE RL02 CONTROLLER (175400). THE SECOND CONTROLLER'S VECTOR WAS ALSO CHANGED TO 164 IN QUESTION #3. THE RL02 TEST UNIT NUMBERS WERE ASSIGNED VALUES 0 TO 3 IN QUESTION #4 AND THE DRIVE TYPE WAS SET FOR RL02'S FOR THE REMAINING 4 UNITS IN QUESTION #5. THE LAST QUESTION WAS DEFAULTED USING THE "BR LEVEL' 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) O?

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.

USE ALL CYLINDERS (N)?

IF "YES", THOSE TESTS THAT NORMALLY USE A SELECTED SET OF CYLINDERS WILL TEST EVERY CYLINDER ON THE CARTRIDGE.

USE ALL SECTORS (N)?

IF "YES", THOSE TESTS THAT NORMALLY USE A SINGLE SECTOR TO TEST A GIVEN OPERATION (SUCH AS SEEK DESTINATION) WILL READ AND VERIFY EVERY SECTOR HEADER.

LOWER SEEK LIMIT (N)?

IF "YES", THE NEXT PARAMETER IS REQUESTED.

ENTER VALUE (DECIMAL) (0)?

THIS LIMIT IS IMPOSED ON ALL SEEK OPERATIONS SUCH THAT TESTING IS NOT DONE BELOW THAT LIMIT. IN ADDITION, SETTING THIS LIMIT (OR THE UPPER LIMIT, SEE BELOW) CAUSES THE FORWORD AND REVERSE OSCILLATING SEEK TESTS TO PERFORM DIFFERENTLY (SEE TEST DESCRIPTION). TESTS THAT REQUIRE ACCESS TO A SPECIFIC CYLINDER THAT FALLS BELOW THE SPECIFIED LIMIT WILL IGNORE THE LIMIT (SEE WRITE/READ TEST PART 1).

UPPER SEEK LIMIT (N)?

IF "YES", AN UPPER CYLINDER LIMIT IS IMPOSED IN THE SAME MANNER AS THE LOWER SEEK LIMIT. A "YES" RESPONSE WILL CAUSE THE FOLLOWING PARAMETER REQUEST.

ENTER VALUE (DECIMAL) (255)?

USE ONLY ONE SURFACE (N)?

IF "YES", THE NEXT PARAMETER IS REQUESTED.

SPECIFY SURFACE (0 OR 1) (DECIMAL) (0)?

WHICHEVER SURFACE IS SPECIFIED IS THE ONLY SURFACE TESTED IN THE ENTIRE PROGRAM. ANY TEST THAT IS DESIGNED TO TEST THE OTHER SURFACE IS AUTOMATICALLY BYPASSED. THE PROGRAM DOES NOT PRINT ANY INDICATION THAT A TEST IS BYPASSED IN THIS CASE.

SPECIFY ERROR LIMIT (DECIMAL) (20)?

THIS PARAMETER SPECIFIES THE MAXIMUM NUMBER OF ERRORS ALLOWED. THIS LIMIT IS ON A PER DRIVE BASIS IN A SINGLE PASS. IF THE ERROR LIMIT IS EXCEEDED, THE DRIVE IS DROPPED FROM FURTHUR TESTING.

DATA COMPARE ERROR LIMIT (DECIMAL) (20)?

THIS PARAMETER SPECIFIES THE NUMBER OF DATA COMPARE ERRORS THAT WILL BE LISTED FOR A GIVEN COMPARE OPERATION. AFTER THE LIMIT IS REACHED, THE DATA ERRORS ARE NOT PRINTED BUT THE COMPARE CONTINUES UNTIL THE END OF THE DATA FIELD. A TOTAL IS REPORTED AT THE END OF THE COMPARE.

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 OPERATION MESSAGE (LINE 4) IS GENERATED IN A DYNAMIC MANNER BASED ON THE SUBSYSTEM FUNCTION BEING EXECUTED AT THE TIME OF THE ERROR AND THE STATE OF THE FLAGS IN THE LOCATION TAGGED 'OPFLAGS". THE POSSIBLE OPERATION MESSAGES ARE GIVEN BELOW.

SEEK - FROM (CYL NUM) DIFF (CYL DIFF) SGN (0 OR 1) HD (0 OR 1) WHERE THE VALUES ARE GIVEN IN OCTAL. THIS MESSAGE IS THE RESULT OF A SEEK OPERATION THAT WAS VERIFIED BY A READ HEADER AND THE HEAD POSITION AFTER A SEEK IS IN ERROR. (THE ACTUAL HEAD POSITION IN THIS ERROR SITUATION IS GIVEN IN THE RESULT LINE, LINE 5.)

READ DATA - IS A READ DATA OPERATION WHERE SOME FORM OF ERROR WAS DETECTED IN THE ACTUAL READ OPERATION. THIS ERROR COULD BE HARDWARE DETECTED SUCH AS DATA CRC, HEADER CRC, HEADER NOT FOUND, ETC., OR A SOFTWARE DETECTED ERROR SUCH AS DRIVE READY RESET AFTER A READ DATA COMPLETED.

READ DATA WITH DATA COMPARE - IS AN ERROR THAT WAS DETECTED AS BAD DATA IN THE BUFFER AFTER

A READ DATA OPERATION. WHEN THIS OPERATION IS REPORTED IT INDICATES THE ACTUAL READ DATA OPERATION COMPLETED WITH NO DETECTED ERRORS BUT THE DATA WAS WRONG.

READ HEADER - READ HEADER FOR 40 HEADERS - READ HEADER FOR 40 HEADERS WITH HEADER COMPARE - HAVE THE SAME GENERAL MEANING AS THE READ DATA AND READ DATA WITH DATA COMPARE. MESSAGES HAVING THE OPERATION OF READ HEADER OR READ HEADER FOR 40 HEADERS ARE THE RESULT OF ERRORS DETECTED IN THE ACTUAL OPERATION WHILE THE READ HEADER FOR 40 HEADERS WITH HEADER COMPARE INDICATES NO ERROR IN THE ACTUAL OPERATION BUT THE HEADER DATA ITSELF WAS IN ERROR.

WRITE DATA - RESET - GET STATUS - GET STATUS WITH RESET - ARE ALL BASIC OPERATIONS. AS BEFORE, THE ERROR DETECTION CAN BE EITHER HARDWARE OR SOFTWARE. THE RESULT LINE (LINE 5) WILL DEFINE THE REASON FOR THE REPORT.

LD DRV - UNLD DRV - ARE OPERATION MESSAGES THAT WILL APPEAR IN THE REPORT WHEN THE DRIVE LOAD AND UNLOAD SEQUENCE IS BEING TESTED.

ANOTHER GROUP OF OPERATION QUALIFIERS WILL BE REPORTED FOR OPERATIONS THAT FAIL IN SPECIFIC TESTS. THESE TESTS ARE THE WRITE/READ TEST PART 2, OVERWRITE TEST, AND THE ADJACENT CYLINDER INTERFERENCE TEST.

OPERATION	QUALIFIER
READ DATA WITH DATA COMPARE	FOL 0 TO CC SEEK
READ DATA	FOL 255 TO CC SEEK
WRITE DATA	FOL WRITE (NO SEEK)
READ HEADER	ADJ. CYL WRITTEN AFTER FWD SK
	ADJ. CYL WRITTEN AFTER REV SK
	SK FWD, WRT-SK REV, OVERWRT
	SK REV, WRT-SK FWD, OVERWRT

THE ABOVE OPERATIONS CAN BE REPORTED WITH ANY OF THE QUALIFIERS. THE QUALIFIERS IN THESE TESTS ARE AN ATTEMPT TO MAKE THE REPORT MORE MEANINGFUL BY PROVIDING INFORMATION ABOUT THE SEQUENCE OF OPERATIONS BEING DONE.

THE QUALIFIERS "FOL 0 TO CC SEEK" AND "FOL 255 TO CC SEEK" INDICATE THAT THE SEQUENCE OF OPERATIONS INCLUDED A SEEK OF A GIVEN DIRECTION TO THE CYLINDER WHERE THE TEST IS BEING PERFORMED.

THE "FOL WRITE (NO SEEK)" QUALIFIER MEANS THAT THE OPERATION WAS DONE AFTER A WRITE WITH NO HEAD MOVEMENT BETWEEN THE WRITE AND READ.

THE QUALIFIER "ADJ CYL WRITTEN AFTER FWD SK" AND "ADJ CYL WRITTEN AFTER REV SK" WILL BE REPORTED ONLY IN THE ADJACENT CYLINDER INTERFERENCE TEST. THESE QUALIFIERS ARE USED WHEN THE ERROR OCCURS ON THE CYLINDER UNDER TEST AND DEFINE THE DIRECTION THE HEADS WERE MOVED WHEN THE ADJACENT CYLINDER WAS WRITTEN.

THE QUALIFIERS "SK FWD, WRT-SK REV, OVERWRT" AND "SK REV, WRT-SK FWD, OVERWRT" WILL BE REPORTED ONLY IN THE OVERWRITE TEST. THESE QUALIFIERS DEFINE THE DIRECTION OF HEAD MOTION BEFORE THE INITIAL WRITE AND THE OVERWRITE.

THE QUALIFIER "ON BAD SEC FILES" WILL BE REPORTED WITH THE WRITE DATA COMMAND IF THE PROGRAM ABORTS THAT COMMAND BECAUSE THE WRITE WOULD BE ON THE BAD SECTOR FILES.

3.1.2 SPECIFIC RESULT MESSAGES

THE RESULT MESSAGE (LINE 5) IS GENERATED DYNAMICALLY BASED ON THE EXPECTED RESULT OF THE OPERATION BEING TESTED. SINCE OPERATIONS ARE MONITORED DURING EXECUTION THE RESULT MESSAGE MAY REPORT AN ERROR DETECTED DURING THE OPERATION AS WELL AS THE ERRORS SEEN AT THE END OF THE OPERATION. ONLY THE FIRST ERROR SEEN IS REPORTED IN ALL CASES.

THE GENERAL FORMAT FOR THE RESULT LINE IS:

```
1136
1137                    RESULT:(VAR 1) IS (VAR 2) SB (VAR 3) (OPTIONAL QUALIFIER)
1138                    WHERE VARIABLE 1 CAN BE ONE OF THE FOLLOWING:
1139
1140                          CONT ERR                        (CONTROLLER ERROR)
1141                          DRV ERR                         (DRIVE ERROR)
1142                          NON-EXSTNT MEM                  (NON EXISTANT MEMORY)
1143                          HDR CRC                         (HEADER CRC ERROR)
1144                          DATA CRC
1145                          HDR NOT FND                     (HEADER NOT FOUND)
1146                          DATA LATE
1147                          HDR NOT FND/HDR CRC/OPI         (ALL 3 BITS SET)
1148                          DRV RDY                         (DRIVE READY)
1149                          SELECTED HEAD
1150                          VOL CHK                         (VOLUME CHECK)
1151                          COVER OPEN
1152                          BRUSH HME                       (BRUCH HOME)
1153                          WRT LCK                         (WRITE LOCK)
1154
1155                          HDS OUT                         (HEADER OUT)
1156                          DRV SEL ERR                     (DRIVE SELECT ERROR)
1157                          DRV STATE                       (DRIVE STATE)
1158                          SPIN TIMEOUT                    (SPINDLE TIMEOUT SPD ERROR)
1159                          WRT GAT ERR                     (WRITE GATE ERROR)
1160                          SEEK TIMEOUT                    (SKTO ERROR)
1161                          CUR HEAD ERR                    (CURRENT IN HEAD ERROR)
1162                          WRT DAT ERR                     (WRITE DATA ERROR)
1163                          OP INCOMPLETE                   (OPI ERROR)
1164                          HDR/DAT ERR                     (HDR CRC OR DATA CRC ERROR
1165                                                           BIT 11 OF CS REGISTER)
1166                          HDR NOT FND/DAT LATE            (HDR NOT FOUND OR DATA LATE
1167                                                           ERROR BIT 12 OF CS REGISTER)
1168                          CYL                             (CYLINDER WHEN REPORTING A
1169                                                           SEEK ERROR)
1170
1171                    VARIABLE 2 WILL BE A VALUE THAT DEFINES WHAT  THE  RESULT  ACTUALLY
1172                    IS.  THIS CAN BE A 1 OR 0 TO INDICATE A SET OF RESULT CONDITIONS. A
1173                    NUMBER 0 TO 7 TO INDICATE THE DRIVE STATE, OR A  NUMBER  0  TO  377
1174                    (OCTAL) TO IDENTIFY A CYLINDER NUMBER.
1175
1176                    VARIABLE 3 DEFINES THAT THE VALUE GIVEN IS VARIABLE  2  SHOULD  BE.
1177                    THE  OPTIONAL  QUALIFIER IS PROVIDED WHEN IT IS USEFUL TO KNOW WHEN
1178                    THE ERROR WAS DETECTED IN  THE  OPERATION  BEING  PERFORMED.   THIS
1179                    QUALIFIER IS USED TO REPORT RESULTS SUCH AS:
1180
1181                          BRUSH HME IS 1 SB 0 IN STATE 2
1182                          HEADS OUT IS 0 SB 1 IN STATE 3
1183                          DRV RDY   IS 0 SB 1 IN DATA XFER
1184                          SELECTED HEAD IS 1 SB 0 IN CYCLE UP
1185                          DRV RDY   IS 0 SB 1 IN STATE 5
1186                          DRV RDY   IS 1 SB 0 IN SEEK W/O MOTION
1187                          DRV RDY   IS 0 SB 1 IN 10MS
1188                          DRV RDY   IS 0 SB 1 IN 500MS
1189                          DRV RDY   IS 0 SB 1 IN 5SECONDS
```

THESE RESULTS, WHEN SEEN WITH THE OPERATION MESSAGE, WILL BE SELF EXPLANATORY

OTHER RESULT MESSAGES THAT CAN BE PART OF AN ERROR REPORT ARE:

"INTERRUPT TOO LATE"

WHICH INDICATES THAT THE OPERATION BEING PERFORMED DID NOT COMPLETE IN THE EXPECTED AMOUNT OF TIME. THIS RESULT CAN BE CAUSED BY THE DRIVE LOOSING READY BEFORE STARTING A READ HEADER AND THEREFORE NOT COMPLETING THE READ HEADER IN 1MS.

"FAIL TO RELOAD HEADS AFTER ERR CLEAR"

THIS IS REPORTED WHEN AN ERROR CAUSES HEADS TO UNLOAD AND AFTER THE ERROR IS CLEARED THE HEADS DO NOT RELOAD.

"UNKN DRV STATE-NO RDY, NO ERR, HDS OUT'

THIS IS REPORTED WHEN THE PROGRAM CANNOT DETERMINE THE DRIVE STATE OR STATUS.

"WRITE ABORTED"

THIS IS REPORTED WHEN THE PROGRAM ABORTS A WRITE TO PROTECT THE BAD SECTOR FILES.

"COULD NOT RETRIEVE DRIVE STATUS"

THIS IS REPORTED IF THE GET STATUS COMMAND DOES NOT COMPLETE SUCCESSFULLY WHEN THE STATUS IS REQUIRED TO REPORT AN ERROR.

"OPI SET-NO DRIVE RESPONSE"

THIS IS REPORTED AS THE RESULT WHEN THE GET STATUS COMMAND IS TIMED OUT (OPI SETS) WHEN THAT COMMAND IS BEING USED IN THE EARLY TESTS TO CHECK THE DRIVE INTERFACE.

"NO INTERRUPT ON CMND COMPLETE"

THIS IS REPORTED WHEN THE COMMAND SUCCESSFULLY COMPLETES BUT THE CONTROLLER HAS NOT GENERATED AN INTERRUPT.

"ERR DID NOT CLEAR"

THIS IS REPORTED WHEN THE RESET COMMAND DOES NOT CLEAR THE CONTROLLER ERRORS THIS IS A CONTROLER RELATED PROBLEM BUT IS REPORTED IF SEEN IN THE DRIVE TEST PROGRAMS.

1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302

DRV ERR IS NOT CLEARED"

THIS IS REPORTED WHEN THE GET STATUS W/RESET COMMAND DOES NOT CLEAR ALL DRIVE ERRORS.

"UNEXPECTED ERR"

THIS IS REPORTED WHEN THE CONTROLLER SENSES AN ERROR BUT NO ERROR BITS ARE SET.

"BAD SEC FILE FMT ERR'

THIS IS REPORTED IF THE CONTENTS OF THE FILES DO NO CORRESPOND TO THE EXPECTED FORMAT. (REFER TO DEC STANDARD 144 FOR FORMAT SPECIFICS.)

3.1.3 OTHER MESSAGES

OTHER INFORMATION IS REPORTED UNDER VARIOUS CIRCUMSTANCES. THESE ARE:

"BAD SEC FILES NOT STRD. ALL SEC ASSUMED GOOD."

THIS MESSAGE IS PRINTED WHEN A PARTICULAR TEST REQUIRES THE BAD SECTOR FILES BUT THEY HAVE NOT BEEN STORED. THIS SITUATION WILL OCCUR IF THIS TEST IS STARTED OUT OF THE NORMAL PROGRAM SEQUENCE OR IF THE BAD SECTOR FILES COULD NOT BE READ.

"ERROR LIMIT EXCEEDED-UNIT DROPPED"

THIS IS REPORTED (WITH THE UNIT NUMBER) WHEN MORE THAN THE SPECIFIED NUMBER OF ERRORS (DEFAULT 20) HAVE OCCURED IN ANY SINGLE PASS.

MOST ERROR REPORTS HAVE THE FOLLOWING FORMAT.

(1) PROG NAME ERR NUM TEST NUM SUBTEST NUM ERR PC
(2) ROUTINE TRACE SEQ (IN SEQ CALLED)
(ADDRESS)
(ADDRESS)
.
.
(ADDRESS)
(3) TEST DESCRIPTION
(4) OPERATION:
(5) RESULT:
(6) ADDRESS OF UNIT UNDER TEST

```
1304
1305                (7)                 RLCS    RLDA    RLBA    RLMP    CYL     HD
1306                (8)   OP INIT
1307                (9)   OP DONE
1308                (10)        DRIVE STATUS
1309                (11)              WORD NUM IS (XXXXXX) SB (YYYYYY)
1310                (12)   TOTAL COMPARE ERRS: (ZZZ) OF (128)
1311
1312            THE ONLY EXCEPTION TO THE ABOVE FORMAT IS PURE DATA COMPARE  ERRORS
1313            (NOT  DETECTED  BY  READ  ERROR).  THEN THE FORMAT DOES NOT INCLUDE
1314            LINES 5 THROUGH 10.
1315
1316            LINE 1 IS THE ERROR HEADER AND IS PROVIDED BY THE SUPERVISOR.   THE
1317            PROGRAM  IS  IDENTIFIED BY NAME WITH THE NUMBER OF TEST AND SUBTEST
1318            PRESENTLY BEING EXECUTED.
1319
1320            THE SUBTEST NUMBER IS UNIQUE IN THIS PROGRAM IN THAT  IT  DOES  NOT
1321            REFER  TO  A  PHYSICAL  SUBTEST  WITHIN  A  GIVEN  TEST.  RATHER IT
1322            REFLECTS THE NUMBER OF TIMES A SUBTEST HAS BEEN EXECUTED  WITHIN  A
1323            TEST.   CONSEQUENTLY,  ON  A TEST THAT TESTS AN INCREMENTAL TYPE OF
1324            OPERATION (SUCH A INCREMENTAL SEEKS, READ  ALL  HEADERS  FROM  BOTH
1325            SURFACES,  ETC.)  THE  SUBTEST  WILL BE DESCRIPTIVE OF WHERE IN THE
1326            TEST THE ERROR OCCURRED.
1327
1328            THE ERROR P.C.  IS THE PHYSICAL MEMORY  LOCATION  WHERE  THE  ERROR
1329            REPORT  WAS  INITIATED.   SINCE MANY FUNCTIONS ARE SUBROUTINED, AND
1330            ERRORS ARE REPORTED  FROM  SUBROUTINES,  THE  ERROR  P.C.   IS  NOT
1331            SUFFICIENT  TO  IDENTIFY  THE  LOCATION  OF  THE ERROR CALL AND THE
1332            ROUTINE TRACE SEQUENCE IS PROVIDED.
1333
1334            LINE 2 IS THE  ROUTINE  TRACE  SEQUENCE.   IF  THE  ERROR  CALL  IS
1335            INITIATED  FROM  WITHIN  THE TEST (AS OPPOSED TO WITHIN A ROUTINE),
1336            THIS PORTION OF THE REPORT IS OMITTED.  IF THE  CALL  IS  INITIATED
1337            FROM  A  ROUTINE (WHICH MAY BE CALLED BY ANOTHER ROUTINE, WHICH MAY
1338            BE CALLED BY  ANOTHER  ROUTINE,  ETC.   SEVERAL  LEVELS  DEEP)  THE
1339            ROUTINE  TRACE  SEQUENCE  PROVIDES  A  TRAIL TO THE ACTUAL LOCATION
1340            WITHIN THE TEST THAT CALLED THE FIRST  ROUTINE.   THE  FIRST  ENTRY
1341            LISTED IS THE LOCATION WHERE THE FIRST ROUTINE WAS CALLED.
1342
1343            LINE 3 IS THE TEST DESCRIPTION AND IS ROUGHLY IDENTICAL TO THE NAME
1344            OF THE TEST BEING PERFORMED.
1345
1346            LINE 4 IDENTIFIES  THE  ACTUAL  HARDWARE  FUNCTION  THAT  IS  BEING
1347            PERFORMED.   ADDITIONAL  INFORMATION ON THIS LINE IS DESCRIPTIVE OF
1348            SPECIFIC USE OF THE FUNCTION.  FOR EXAMPLE, THE OPERATION LINE WILL
1349            READ  "READ HEADERS FOR 40 HEADERS" WHEN ALL HEADERS ARE BEING READ
1350            FROM A TRACK.
1351
1352            LINE 5 IDENTIFIES THE ERROR THAT HAS BEEN DETECTED.  THE CONTENT OF
1353            LINE  5  IDENTIFIES  WHAT  WAS  BEING  TESTED  (SUCH AS DRIVE READY,
1354            CONTROLLER ERROR, DRIVE STATE, ETC.), WHAT IT IS AND WHAT IT SHOULD
1355            BE.   LINE  5 MAY BE REPEATED IF MORE THAN ONE TESTED ITEM IS FOUND
1356            IN ERROR.
```

IN ADDITION LINE 5 WILL REPORT ANY HARDWARE DETECTED ERRORS SUCH AS OPERATION INCOMPLETE, HEADER CRC, ETC. IN THIS CASE THE FIRST LINE PRINTED AS RESULT WILL BE DETERMINED BY THE THREE ERROR BITS OPI, HNF/DLT, AND HCRC/DCRC. THE LINE WILL BE DETERMINED AS IN THE FOLLOWING TRUTH TABLE:

HNF/DLT	DCRC/HCRC	OPI	MESSAGE
1	1	1	HDR NOT FND/HDR CRC/OPI ERROR
0	1	1	HDR CRC ERROR
1	0	1	HDR NOT FND ERROR
0	1	0	DATA CRC ERROR
1	0	0	DATA LATE ERROR

LINE 6 IDENTIFIES THE PHYSICAL ADDRESS OF THE UNIT UNDER TEST. THIS ADDRESS IS BY UNIBUS ADDRESS OF THE CONTROLLER AND DRIVE NUMBER.

LINE 7 NAMES THE CONTROLLER REGISTERS (AND CYLINDER AND HEAD WHERE THESE ARE APPLICABLE IN THE REPORT) TO BE REPORTED.

LINE 8 PROVIDES THE CONTENTES OF CONTROLLER REGISTERS WHEN THE OPERATION WAS INITIATED.

LINE 9 PROVIDES THE CONTENTS OF THE CONTROLLER REGISTERS WHEN THE ERROR BEING REPORTED WAS DETECTED. FREQUENTLY THE REGISTER CONTENTS OF OP INIT AND OP DONE WILL BE DIFFERENT. OP INIT MAY INDICATE A SEEK WAS BEING PERFORMED BUT OP DONE MAY INDICATE THE ERROR WAS DETECTED BY A READ HEADER. THE REASON IS THAT A SEEK WAS EXECUTED AND DID NOT PROPERLY POSITION HEADS AND WHEN THE READ HEADER WAS DONE THE HEADS WERE ON THE WRONG CYLINDER.

LINE 10 IS THE DRIVE STATUS. THIS LINE IS ONLY REPORTED IF THE RLMP REGISTER DOES NOT CONTAIN THE ACTUAL DRIVE STATUS.

LINE 11 AND LINE 12 ARE REPORTED IF THE ERROR WAS DETECTED AS A COMPARE OPERATION, EITHER DATA OR HEADERS. IN ADDITION, GOOD AND BAD DATA IS REPORTED FOR ALL READ ERRORS.

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

THIS PROGRAM WILL NOT GIVE ANY PERFORMANCE REPORTS.

4.2 PROGRESS REPORTS

THIS PROGRAM WILL NOT GIVE ANY PROGRESS REPORTS.

5.0 DEVICE INFORMATION TABLES

THE RL11/RLV11 CONTROLLER HAS THE FOLLOWING FOUR(4) REGISTERS FOR CONTROL OF THE SUBSYSTEM.

RLCS CONTROL AND STATUS REGISTER (XXXXX0)

BIT 15 COMPOSITE ERROR
BIT 14 DRIVE ERROR
BIT 13 - NON EXISTANT MEMORY ERROR
BIT 12 HEADER NOT FOUND (WITH BIT 10 SET)
- DATA LATE (WITH BIT 10 CLEAR)
BIT 11 - HEADER CRC (WITH BIT 10 SET)
DATA CRC (WITH BIT 10 CLEAR)
BIT 10 - OPERATION INCOMPLETE
BIT 9/8 - DRIVE SELECT (0-3)
BIT 7 - CONTROLLER READY
BIT 6 - INTERRUPT ENABLE
BIT 5 EXTENDED BUS ADDRESS (BIT 17)
BIT 4 - EXTENDED BUS ADDRESS (BIT 16)
BIT 3-1 - FUNCTION CODE
0 - NOP (PDP-11) MAINT (LSI-11)
1 - WRITE CHECK
2 - GET DRIVE STATUS
3 - SEEK
4 - READ HEADER
5 - WRITE DATA
6 - READ DATA
7 - READ WITHOUT HEADER COMPARE

BIT 0 - DRIVE READY

RLBA - BUS ADDRESS REGISTER (XXXXX2)

BITS 15-1 BUS ADDRESS OF DATA TRANSFER
BIT 0 SHOULD BE 0

RLDA - DISK ADDRESS REGISTER (XXXXX4)

```
1467
1468                    FOR READ/WRITE FUNCTIONS
1469                        - -     - --
1470
1471                    BIT 15 7   CYLINDER ADDRESS FOR TRANSFER
1472                    BIT 6    SURFACE FOR TRANSFER
1473                    BIT 5-0    SECTOR FOR TRANSFER (1-40.)
1474
1475                    FOR SEEK FUNCTION
1476                        - -- - ---
1477
1478                    BIT 15 7   DIFFERENCE TO NEW CYLINDER
1479                    BIT 6-5 - MUST BE ZERO (0)
1480                    BIT 4 - SURFACE (0=UPPER, 1=LOWER)
1481                    BIT 3    MUST BE ZERO (0)
1482                    BIT 2 - SEEK DIRECTION( 1=IN / 0=OUT )
1483                    BIT 1    MUST BE ZERO (0)
1484                    BIT 0 - MUST BE ONE (1)
1485
1486                    FOR GET STATUS FUNCTION
1487                    - - ---- ------ ------
1488
1489                    BIT 15-4 - IGNORED SHOULD BE ZERO (0)
1490                    BIT 3 - DRIVE RESET
1491                    BIT 2 - MUST BE ZERO (0)
1492                    BIT 1 - MUST BE ONE (1)
1493                    BIT 0    MUST BE ONE (1)
1494
1495                    RLMP - MULTIPURPOSE REGISTER
1496                    -----  --------------------
1497
1498                    FOR READ/WRITE FUNCTION
1499                    - ------------- -----
1500
1501                    BIT 15 - 0 - WORD COUNT (TWO'S COMPLIMENT)
1502
1503                    FOR READ HEADER FUNCTION
1504                    ------ ----------------
1505
1506                    BIT 15-0 - DISK HEADER OF SECTOR (FIRST READ)
1507                             - ZERO WORD (SECOND READ)
1508                             - HEADER CRC (THIRD READ)
1509
1510                    FOR GET STATUS FUNCTION
1511                    --------------------
1512
1513                    HAS DRIVE STATUS
1514
1515                    BIT 15 - WRITE DATA ERROR
1516                    BIT 14 - CURRENT HEAD ERROR (CHE)
1517                    BIT 13 - WRITE LOCK STATUS (WL)
1518                    BIT 12 - SEEK TIME OUT (SKTO)
1519                    BIT 11 - SPIN ERROR (SPE)
1520                    BIT 10 - WRITE GATE ERROR (WGE)
1521                    BIT  9 - VOLUME CHECK (VC)
```

```
1523
1524                                    BIT  8   DRIVE SELECT ERROR (DSE)
1525                                    BIT  7   DRIVE TYPE IS RL02 IF SET
1526                                    BIT  6   SURFACE (0=UPPPER, 1=LOWER)
1527                                    BIT  5   COVER OPEN
1528                                    BIT  4   HEADS HOME
1529                                    BIT  3   BRUSHES HOME
1530                                    BIT 2 0 -STATE BITS
1531                                              0   LOAD STATE
1532                                              1 - SPIN UP
1533                                              2 - BRUSH CYCLE
1534                                              3   LOAD HEADS
1535                                              4 - SEEK - TRACK COUNTING
1536                                              5   SEEK   LINEAR MODE
1537                                              6 - UNLOAD HEADS
1538                                              7   SPIN DOWN
1539
1540
1541                        6.0         TEST SUMMARIES
1542                                     --- ---------
1543
1544                            TEST 1  OUTER GUARD BAND DETECTION TEST
1545                            ******
1546
1547                                DO READ HEADER, WAIT FOR INTERRUPT.  CHECK IF AT  CYLINDER  0.
1548                                IF  NOT, SEEK REVERSE 1 CYLINDER AT A TIME UNTIL CYLINDER 0 IS
1549                                REACHED.  IF ANY REVERSE SEEK FAILS TO MOVE THE  HEADS  IN  10
1550                                TRIES:
1551
1552                                          DETECTION OF GUARD BAND PREMATURE.
1553
1554                                WHEN AT CYLINDER 0, DO SEEK DIFFERENCE OF 1, SIGN 0,  HEAD  0.
1555                                WAIT  FOR  INTERRUPT,  WAIT  FOR  READY.   READY SHOULD SET IN
1556                                20MS>T>15MS.  IF NOT:
1557
1558                                          FAILED TO DETECT GUARD BAND
1559
1560                                DO READ HEADER.  WAIT FOR INTERRUPT.  CHECK  FOR  CYLINDER  0.
1561                                IF NOT:
1562
1563                                          FAILED TO SEEK BACK TO ZERO
1564
1565                                DO SEEK WITH DIFFERENCE OF 1, SIGN 0, HEAD 1.  DO  SAME  TESTS
1566                                AS  ABOVE  WITH  REGARD TO READY VS TIME AND CYLINDER FOUND IN
1567
1568                                HEADER.
1569
1570                                NOTE:   CHOOSING A SINGLE SURFACE WILL LIMIT  THE  TESTING  TO
1571                                        THAT SURFACE.
```

```
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
```

TEST 2 INCREMENTAL FORWARD SEEK HEAD 0 TEST
******

POSITION HEADS AT CYLINDER "LOLIMIT" USING SEEKS WITH DIFFERENCE OF ONE, HEAD 0.

DO SEEK WITH DIFFERENCE OF 1, SIGN 1, HEAD 0. WAIT FOR INTERRUPT, WAIT FOR DRIVE READY. CHECK READY IS SET IN 15 MS. IF NOT:

POSITIONING PROBLEM AT A SPECIFIC CYLINDER
MECHANICAL OBSTRUCTION

CHECK THAT THIS CYLINDER IS OLD CYLINDER + 1. IF NOT:

DIFFERENCE REGISTER OR COUNT LOGIC FAILURE
TRACK CROSSING DETECTION FAILURE

REPEAT SEEKS AND READS UNTIL CYLINDER READ IS "HILIMIT".

NOTE 1: IF THE "USE ALL SECTORS" PARAMETER IS SPECIFIED AS "Y", THE TEST WILL READ AND TEST ALL 40 HEADERS (CARTRIDGE VERIFY).

NOTE 2: TESTING WILL BE DONE BETWEEN UPPER AND LOWER CYLINDER LIMITS. CHOOSING A SINGLE SURFACE WILL LIMIT TESTING TO THAT SURFACE. THIS TEST WILL BE BYPASSED IF SURFACE 1 IS CHOSEN.

TEST 3 INCREMENTAL REVERSE SEEK HEAD 0 TEST
******

POSITION HEADS AT CYLINDER "HILIMIT" USING SEEKS WITH DIFFERENCE OF 1, HEAD 0.

DO SEEK WITH DIFFERENCE OF 1, SIGN 0, HEAD 0. WAIT FOR INTERRUPT, WAIT FOR DRIVE READY. CHECK READY SET IN 15 MS:

POSITIONING PROBLEM AT A SPECIFIC CYLINDER
DO READ HEADER, WAIT FOR INTERRUPT. CHECK THAT THIS CYLINDER IS OLD CYLINDER - 1. IF NOT:

DIFFERENCE REGISTER OR COUNT LOGIC FAILURE
TRACK CROSSING DETECTION FAILURE

REPEAT SEEK AND CHECKS UNTIL CYLINDER IS "LOLIMIT".

NOTE: IF THE "USE ALL SECTORS" PARAMETER IS SPECIFIED AS "Y", THE TEST WILL READ AND TEST ALL 40 HEADERS (CARTRIDGE VERIFY).

```
1626
1627                    NOTE:   TESTING WILL BE DONE BETWEEN JPPER AND LOWER  CYLINDER
1628                            LIMITS.   CHOOSING A SINGLE SURFACE WILL LIMIT TESTING
1629                            TO THAT  SURFACE.   THIS  TEST  WILL  BE  BYPASSED  IF
1630                            SURFACE 1 IS CHOSEN.
1631
1632                TEST 4  INCREMENTAL FORWARD SEEK HEAD 1 TEST
1633                ******
1634
1635                    POSITION  HEADS  AT  CYLINDER  "HILIMIT"  USING   SEEKS   WITH
1636                    DIFFERENCE OF ONE, HEAD 0.
1637
1638                    DO SEEK WITH DIFFERENCE OF  1,  SIGN  1,  HEAD  1.   WAIT  FOR
1639                    INTERRUPT, WAIT FOR DRIVE READY.  CHECK READY IS SET IN 15 MS.
1640                    IF NOT:
1641
1642                                POSITIONING PROBLEM AT A SPECIFIC CYLINDER
1643
1644                    DO READ HEADER, WAIT FOR INTERRUPT.  CHECK THAT THIS  CYLINDER
1645                    IS OLD CYLINDER + 1.  IF NOT:
1646
1647                                DIFFERENCE REGISTER OR COUNT LOGIC FAILURE
1648                                TRACK CROSSING DETECTION FAILURE
1649
1650                    REPEAT SEEKS AND READS UNTIL CYLINDER READ IS "HILIMIT".
1651
1652                    NOTE 1: IF THE "USE ALL SECTORS"  PARAMETER  IS  SPECIFIED  AS
1653                            "Y".  THE  TEST  WILL  READ  AND  TEST  ALL 40 HEADERS
1654                            (CARTRIDGE VERIFY).
1655
1656                    NOTE 2: TESTING WILL BE DONE BETWEEN UPPER AND LOWER  CYLINDER
1657                            LIMITS.   CHOOSING A SINGLE SURFACE WILL LIMIT TESTING
1658                            TO THAT  SURFACE.   THIS  TEST  WILL  BE  BYPASSED  IF
1659                            SURFACE 0 IS CHOSEN.
1660
1661                TEST 5  INNER GUARD BAND DETECTION TEST
1662                ******
1663
1664                    POSITION HEADS AT CYLINDER "HILIMIT" USING SEEK WITH DIFFERENCE
1665                    OF 1, HEAD 0.
1666
1667                    WHEN AT MAX CYLINDER, DO SEEK WITH DIFFERENCE OF  1,  SIGN  1,
1668                    HEAD  0.   WAIT  FOR  INTERRUPT,  WAIT FOR DRIVE READY.  READY
1669                    SHOULD SET IN 20MS>T>15MS.  IF NOT:
1670
1671                                FAILED TO DETECT GUARD BAND
1672
1673                    DO READ HEADER. WAIT FOR INTERRUPT. CHECK FOR MAX. CYLINDER
1674                    IF NOT:
1675
1676                                FAILED TO SEEK BACK TO MAX CYLINDER
1677
1678                    DO SEEK WITH DIFFERENCE OF 1, SIGN 1, HEAD 1.  DO  SAME  TESTS
1679                    AS ABOVE.
1680
1681                    NOTE:   CHOOSING A SINGLE SURFACE WILL LIMIT  THE  TESTING  TC
```

```
1683
1684                    THAT SURFACE.
1685
1686            TEST 6  INCREMENTAL REVERSE SEEK HEAD 1 TEST
1687            ******
1688
1689                POSITION HEADS AT CYLINDER "HILIMIT" USING SEEKS WITH
1690                DIFFERENCE OF 1, HEAD 0.
1691
1692                DO SEEK WITH DIFFERENCE OF 1, SIGN 0, HEAD 1. WAIT FOR
1693                INTERRUPT, WAIT FOR DRIVE READY. CHECK READY SET IN 15 MS:
1694
1695                        POSITIONING PROBLEM AT A SPECIFIC CYLINDER
1696
1697                DO READ HEADER, WAIT FOR INTERRUPT. CHECK THAT THIS CYLINDER
1698                IS OLD CYLINDER - 1. IF NOT:
1699
1700                        DIFFERENCE REGISTER OR COUNT LOGIC FAILURE
1701                        TRACK CROSSING DETECTION FAILURE
1702
1703                REPEAT SEEK AND CHECKS UNTIL CYLINDER IS "LOLIMIT".
1704
1705                NOTE 1: IF PROGRAM MODE 2 IS USED AND THE "USE ALL SECTORS"
1706                        PARAMETER IS SPECIFIED AS "Y", THE TEST WILL READ AND
1707                        TEST ALL 40 HEADERS (CARTRIDGE VERIFY).
1708
1709                NOTE 2: TESTING WILL BE DONE BETWEEN UPPER AND LOWER CYLINDER
1710                        LIMITS. CHOOSING A SINGLE SURFACE WILL LIMIT TESTING
1711                        TO THAT SURFACE. THIS TEST WILL BE BYPASSED IF
1712                        SURFACE 0 IS CHOSEN.
1713
1714            TEST 7  SEEK TESTS
1715            ******
1716
1717                POSITION HEADS AT CYLINDER "LOLIMIT" USING SEEKS WITH
1718                DIFFERENCE OF 1, HEAD 0.
1719
1720                DO READ HEADER, RECORD POSITION. DO SEEK WITH DIFFERENCE OF 2
1721                (MAX DISTANCE AT 3 IPS), SIGN 1, HEAD 0. DO READ HEADER,
1722                CHECK NEW CYLINDER IS OLD CYLINDER + DISTANCE. IF NOT:
1723
1724                        TRACK CROSSING DETECTION FAILURE
1725                        DIFFERENCE COUNTER FAILURE
1726                        COUNT PULSE GENERATION FAILURE
1727                        VELOCITY ROM FAILURE
1728
1729                REPEAT ABOVE UNTIL OLD CYLINDER + DISTANCE > MAX  POSITION AT
1730                MAX.
1731
1732
1733                DO READ HEADER, RECORD POSITION. DO SEEK WITH DIFFERENCE OF 2
1734                (MAX DISTANCE AT 3 IPS), SIGN 0, HEAD 0. DO READ HEADER,
1735                CHECK NEW CYLINDER IS OLD CYLINDER - DISTANCE. IF NOT:
1736
1737                        TRACK CROSSING DETECTION FAILURE
1738
```

```
1740
1741                    REPEAT UNTIL OLD CYLINDER   DISTANCE < 0.  REPEAT ALL  OF  THE
1742                    ABOVE USING HEAD 1.
1743
1744                    REPEAT ALL OF THE ABOVE TESTS USING  THE  FOLLOWING  DISTANCES:
1745                    2, 6, 9, 12, 17, 22, 27, 34, 41, 128, 256 FOR RL01 OR 4, 12, 18,
1746                    24, 34, 44, 54, 68, 82, 256, 512 FOR RL02.  THESE DISTANCES ARE
1747                    SPECIFIED BECAUSE THEY REPRESENT THE MAXIMUM DISTANCE FOR EACH
1748                    VELOCITY LEVEL USED IN THE DRIVE.
1749
1750                    NOTE:   TESTING WILL BE DONE BETWEEN UPPER AND LOWER  CYLINDER
1751                            LIMITS.   CHOOSING A SINGLE SURFACE WILL LIMIT TESTING
1752                            TO THAT SURFACE.
1753
1754                TEST 8 FORWARD OSCILLATING SEEK TEST
1755                ******
1756
1757                    POSITION HEADS AT CYLINDER 0.
1758
1759                    DO OSCILLATING SEEK USING HEAD 0 (SEEK FROM 0 TO 1 TO 0, 0  TO
1760                    2  TO  0,  0  TO 3 TO 0, ...0 TO MAX CYL TO 0).  AFTER EACH SEEK
1761                    READ HEADER AND VERIFY POSITION.
1762
1763                    REPEAT TEST USING HEAD 1.
1764
1765                    NOTE:   IF EITHER CYLINDER LIMIT IS SPECIFIED, THE  TEST  WILL
1766                            SEEK  BETWEEN UPPER AND LOWER LIMITS FOR EACH SURFACE.
1767                            CHOOSING A SINGLE SURFACE WILL LIMIT TESTING  TO  THAT
1768                            SURFACE.   NOTE  THAT  LOOPING ON TEST THEN PROVIDES A
1769                            FIXED DISTANCE SEEK LOOP.
1770
1771                TEST 9 REVERSE OSCILLATING SEEK TEST
1772                ******
1773
1774                    POSITION HEADS AT MAX CYLINDER.   DO  OSCILLATING  SEEK  USING
1775                    HEAD 0. (IF RL01 SEEK FROM 255 TO 254 TO 255, 255 TO  253  TO
1776                    255.....255 TO 0 TO 255.) AFTER  EACH  SEEK  READ  HEADER  AND
1777                    VERIFY POSITION.
1778
1779                    REPEAT TEST USING HEAD 1.
1780
1781                    NOTE:   IF EITHER CYLINDER LIMIT IS SPECIFIED, THE  TEST  WILL
1782                            SEEK  BETWEEN UPPER AND LOWER LIMITS FOR EACH SURFACE.
1783                            CHOOSING A SINGLE SURFACE WILL LIMIT TESTING  TO  THAT
1784                            SURFACE.   NOTE  THAT  LOOPING ON TEST THEN PROVIDES A
1785                            FIXED DISTANCE SEEK LOOP.
1786
1787                @
```

```
1789          002000                          .=2000
1790                                          .MCALL   SVC
1791
1792 002000                                   SVC
1793          000001                          SVCTST=1
1794          000001                          SVCSUB=1
1795          000001                          SVCBGL=1
1796          000000                          SVCINS=0
1797          000000                          SVCTAG=0
1798
1799                                .SBTTL  MACRO DEFINITIONS
1800
1801                                .MACRO  WAITUS  ARG             ;MACRO MICRO-SEC WAIT
1802                                        MOV     ARG,XDELAY      ;SAVE ARGUMENT
1803                                        JSR     PC,TIME         ;CALL TIMING ROUTINE
1804                                .ENDM
1805
1806                                .MACRO  WAITMS  ARG             ;MACRO MILLI-SEC WAIT
1807                                        MOV     ARG,YDELAY      ;SAVE ARGUMENT
1808                                        JSR     PC,XTIME        ;CALL TIMING ROUTINE
1809                                .ENDM
1810
1811                                .MACRO  ABORTWAIT               ;MACRO CLEAR UNELAPSED TIME
1812                                        MOV     XDELAY,TEMPO    ;SAVE MICRO-SEC RUN TIME
1813                                        MOV     YDELAY,TEMP     ;SAVE MILLI-SEC RUN TIME
1814                                        CLR     XDELAY          ;ABORT MICRO-SEC WAIT
1815                                        CLR     YDELAY          ;ABORT MILLI-SEC WAIT
1816                                .ENDM
1817
1818                                .MACRO  GETTIM  ARG             ;MACRO GET ELAPSED TIME
1819                                        MOV     @@CLKCTR,ARG    ;STORE CLOCK COUNTER CONTENTS
1820                                        CLR     @@CLKCSR        ;EVENT FINISHED, STOP CLOCK
1821                                .ENDM
1822
1823                                .MACRO  STCLK                   ;MACRO START P-CLOCK
1824                                        CLR     @@CLKCSB        ;CLEAR CLOCK COUNT SET BUFFER
1825                                        CLR     @@CLKCTR        ;CLEAR CLOCK COUNTER
1826                                        MOV     #23,@@CLKCSR    ;INITIALIZE CLOCK FOR COUNT-UP MODE.
1827                                                                ;/10 KHZ RATE, AND START CLOCK
1828                                .ENDM
1829
1830                                .NLIST  CND,MD,ME
1831
1832 002000                                   POINTER BGNSW,BGNSFT,BGNDU
1833
1834 002000                          BGNMOD   MDHEDR
1839 002000                          HEADER   CNRLJ,A,0,30000,0,PRI06                ;JSD REV A - ADDED PRI06
     002000     103                          .ASCII /C/
     002001     116                          .ASCII /N/
     002002     122                          .ASCII /R/
     002003     114                          .ASCII /L/
     002004     112                          .ASCII /J/
     002005     000                          .BYTE   0
     002006     000                          .BYTE   0
     002007     000                          .BYTE   0
     002010     101                          .ASCII  /A/
     002011     060                          .ASCII  /0/
```

MACRO DEFINITIONS

```
002012  000000                          .WORD   0
002014  030000                          .WORD   30000
002016  030652                           WORD   L$HARD
002020  031026                          .WORD   L$SOFT
002022  013540                          .WORD   L$HW
002024  013556                          .WORD   L$SW
002026  031370                          .WORD   L$LAST
002030  000000                          .WORD   0
002032  000000                          .WORD   0
002034  000000                          .WORD   0
002036  000000                          .WORD   0
002040  013574                          .WORD   L$DISPATCH
002042  000300                          .WORD   PRI06
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  002226                          .WORD   L$DVTYP
002062  000000                          .WORD   0
002064  000000                          .WORD   0
002066  000000                          .WORD   0
002070  000000                          .WORD   0
002072  015300                          .WORD   L$DU
002074  000000                          .WORD   0
002076  002122                          .WORD   L$DESC
002100  104035                         EMT     E$LOAD
002102  000000                          .WORD   0
002104  013616                          .WORD   L$INIT
002106  015152                          .WORD   L$CLEAN
002110  014614                          .WORD   L$AUTO
002112  013530                          .WORD   L$PROT
002114  000000                          .WORD   0
002116  000000                          .WORD   0
002120  000000                          .WORD   0
1841 002122                             ENDMOD
1842
1843 002122                             DESCRIPT <CNRLJ TESTS OUTER & INNER GUARD BAND DETECTION AND SEEK OPERATIONS>
     002122     103     116     122             .ASCIZ  /CNRLJ TESTS OUTER & INNER GUARD BAND DETECTION AND SEEK OPERATIONS/
     002125     114     112     040
     002130     124     105     123
     002133     124     123     040
     002136     117     125     124
     002141     105     122     040
     002144     046     040     111
     002147     116     116     105
     002152     122     040     107
     002155     125     101     122
     002160     104     040     102
     002163     101     116     104
     002166     040     104     105
     002171     124     105     103
     002174     124     111     117
     002177     116     040     101
     002202     116     104     040
```

MACRO DEFINITIONS

```
        002205  123  105  105
        002210  113  040  117
        002213  120  105  122
        002216  101  124  111
        002221  117  116  123
        002224  000
                                        .EVEN

1844
1845 002226                         DEVTYP  <RL01,RL02>
        002226  122  114  060              .ASCIZ  #RL01,RL02#
        002231  061  054  122
        002234  114  060  062
        002237  000
                                        .EVEN

1846
1847                                .SBTTL  GLOBAL DATA SECTION
1848
1849 002240                         BGNMOD  GLBEQAT
1850
1851 002240                         EQUALS
                                    ;
                                    ; BIT DIFINITIONS
                                    ;
            100000                  BIT15== 100000
            040000                  BIT14== 40000
            020000                  BIT13== 20000
            010000                  BIT12== 10000
            004000                  BIT11== 4000
            002000                  BIT10== 2000
            001000                  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
```

GLOBAL DATA SECTION

```
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
020000                  IER== 20000
040000                  LOE== 40000
100000                  HOE== 100000
1852                    ;       OFFSETS FOR HARDWARE P-TABLE
1853    000000          CSR     =0                      ;BUS ADDRESS
1854    000002          VECT    =2                      ;VECTOR ADDRESS
1855    000004          PRIOR   =4                      ;PRIORITY
1856    000006          TYPDR=6
1857    000010          DRSB    =10                     ;DRIVE SELECT BIT
1858    000012          CNT     =12                     ;CONTROLLER TYPE
1859
1860                    ;       OFFSET FOR SOFTWARE P-TABLE
1861    000000          MISWI   =0                      ;SOFTWARE PARAMETERS SWITCHES
1862    000002          LOLIM   =2                      ;CYLINDER LOWER LIMIT
1863    000004          HILIM   =4                      ;CYLINDER HIGH LIMIT
1864    000006          HEAD    =6                      ;SELECTED HEAD FOR RUNNING TESTS
1865    000010          ERLIM   =10                     ;ERROR LIMIT
1866    000012          DCLIM   =12                     ;DATA COMPARE ERROR LIMIT
1867
1868                    ;       BIT ASSIGNMENT FOR SOFTWARE P-TABLE SWITCHES
1869    000001          ALLCYL  =BIT00                  ;USE ALL CYLINDERS
1870    000002          ALLSEC  =BIT01                  ;USE ALL SECTORS
1871    000004          DRSELT  =BIT02                  ;EXECUTE DRIVE SELECT TEST
1872    000010          HDALIGN =BIT03                  ;EXECUTE HEAD ALIGNMENT TEST
1873    010000          HEADLM  =BIT12                  ;HEAD LIMIT SPECIFIED FLAG
1874    020000          HICYL   =BIT13                  ;HI LIMIT SPECIFIED FLAG
1875    040000          LOCYL   =BIT14                  ;LO LIMIT SPECIFIED
1876    100000          MITEST  =BIT15                  ;EXECUTE MANUAL INTERVENTION TESTS
```

GLOBAL DATA SECTION

```
1877
1878                                    ;       SUBSYSTEM FUNCTIONS
1879    000102                          CKDATA  =102                    ;WRITE CHECK
1880    000104                          GTSTAT  =104                    ;GET STATUS
1881    000106                          SEEK    =106                    ;SEEK
1882    000110                          RDHEAD  =110                    ;READ HEADER
1883    000112                          WTDATA  =112                    ;WRITE DATA
1884    000114                          RDDATA  =114                    ;READ DATA
1885    000116                          RDNOHR  =116                    ;READ DATA, IGNORE HEADERS
1886    000100                          NOOP    =100                    ;NO OPERATION
1887
1888                                    ;       OPERATION FLAGS
1889    007777                          COMPOP  =7777                   ;COMPOSITE OPERATION FLAGS
1890    000002                          HDRCMP  =BIT01                  ;HEADER COMPARE OPERATION
1891    000001                          DATACMP =BIT00                  ;DATA COMPARE OPERATION
1892    000004                          CYLUP   =BIT02                  ;CYCLE UP OPERATION
1893    000010                          ULOAD   =BIT03                  ;UNLOAD OPERATION
1894    000020                          INOUTS  =BIT04                  ;IN-OUT SEEK OPERATION
1895    000040                          OUTINS  =BIT05                  ;OUT-IN SEEK OPERATION
1896    000100                          FOLWRT  =BIT06                  ;FOLLOWING WRITE OPERATION
1897    000200                          REVSKS  =BIT07                  ;REV SEEK SEQ (ADJ INTERFERENCE)
1898    000400                          FWDSKS  =BIT08                  ;FWD SEEK SEQ (ADJ INTERFERENCE)
1899    001000                          REVSKO  =BIT09                  ;REV SEEK SEQ (OVERWRITE)
1900    002000                          FWDSKO  =BIT10                  ;FWD SEEK SEQ (OVERWRITE)
1901    004000                          BADADD  =BIT11                  ;BAD DISK ADDRESS
1902    010000                          SEEKOP  =BIT12                  ;SEEK OPERATION
1903    020000                          RORWOP  =BIT13                  ;READ OR WRITE OPERATION
1904    040000                          RELDWT  =BIT14                  ;RELOAD WAIT
1905    100000                          HDR40   =BIT15                  ;40 HEADER OPERATION
1906    003760                          MQUALS  =OUTINS!INOUTS!FOLWRT!REVSKS!FWDSKS!REVSKO!FWDSKO
1907                                                                    ;MESSAGE QUALIFIER BITS
1908
1909                                    ;       ERROR FLAGS FROM SUBROUTINES
1910    000001                          TOSLOW  =BIT00                  ;OPERATION TOOK TOO LONG
1911    000002                          NOIRPT  =BIT01                  ;NO INTERRUPT FROM OPERATION
1912    000004                          CONHNG  =BIT02                  ;CONTROLLER HUNG
1913    000010                          NOCLR   =BIT03                  ;BAD CONTROLLER CLEAR
1914
1915    000000                          RLCS    =0                      ;CONTROL AND STATUS REGISTER
1916    000002                          RLBA    =2                      ;BUS ADDRESS REGISTER
1917    000004                          RLDA    =4                      ;DISK ADDRESS REGISTER
1918    000006                          RLMP    =6                      ;MULTI-PURPOSE REGISTER
1919
1920                                    ;       REGISTER BIT DEFINITIONS - CONTROL STATUS REGISTER
1921    000000                          RLCSR   =0                      ;CONTROL AND STATUS REGISTER
1922    100000                          ANYERR  =100000                 ;ANY ERROR BIT
1923    040000                          DRVERR  =40000                  ;DRIVE ERROR BIT
1924    020000                          NXMERR  =20000                  ;NON-EXISTANT MEMORY ERROR
1925    010000                          DLTERR  =10000                  ;DATA LATE ERROR
1926    010000                          HNFERR  =10000                  ;HEADER NOT FOUND ERROR
1927    004000                          DCKERR  =4000                   ;DATA CHECK ERROR
1928    004000                          HCRCERR =4000                   ;HEADER CHECK ERROR
1929    002000                          OPIERR  =2000                   ;OPERATION INCOMPLETE ERROR
1930    001400                          DSMSK   =1400                   ;DRIVE SELECT MASK
1931    000200                          CRDYMSK =200                    ;CONTROLLER READY MASK
1932    000100                          INTEBL  =100                    ;INTERRUPT ENABLE MASK
1933    000060                          BAMSK   =60                     ;BUS ADDRESS UPPER MASK
```

GLOBAL DATA SECTION

```
                        DRDYMSK =1                      ;DRIVE READY MASK
                ;       REGISTER BIT DEFINITIONS  DISK ADDRESS FOR DATA XFER
000077          SAMSK   =77                     ;SECTOR ADDRESS MASK
000100          HSMSK   =100                    ;HEAD SELECT MASK
                ;       REGISTER BIT DEFINITIONS  DISK ADDRESS FOR SEEK
000001          MBSET0  =1                      ;MUST BE SET, BIT 0
000004          DIRBIT  =4                      ;DIRECTION BIT
000020          HDSEL   =20                     ;HEAD SELECT BIT
                ;       REGISTER BIT DEFINITIONS  DISK ADDRESS FOR GET STATUS
000003          GETSTAT =3                      ;GET STATUS SETUP
000010          DRSET   =10                     ;DRIVE RESET MASK
                ;       REGISTER BIT DEFINITIONS - MP FOR DATA XFER
017777          WCMSK   =17777                  ;WORD COUNT MASK
160000          WCRNG   =160000                 ;WORD COUNT RANGE MASK
                ;       REGISTER BIT DEFINITIONS - MP FOR READ HEADER
000077          HDSEC   =77                     ;SECTOR MASK
000100          HDHSEL  =100                    ;HEAD SELECT MASK
                ;       REGISTER BIT DEFINITIONS - MP FOR GET STATUS
000007          STAMSK  =7                      ;STATE MASK
000010          BHSTAT  =10                     ;BRUSH HOME STATUS
000020          HDSTAT  =20                     ;HEADS OUT STATUS
000040          COSTAT  =40                     ;COVER OPEN STATUS
000100          HSSTAT  =100                    ;HEAD SELECT STATUS
000400          DSESTAT =400                    ;DRIVE SELECT ERROR STATUS
001000          VCSTAT  =1000                   ;VOLUME CHECK STATUS
002000          WGESTAT =2000                   ;WRITE GATE ERROR STATUS
004000          SPDSTAT =4000                   ;SPIN ERROR STATUS
010000          STOSTAT =10000                  ;SEEK TIMEOUT ERROR STATUS
020000          WLSTAT  =20000                  ;WRITE LOCK STATUS
040000          HCESTAT =40000                  ;HEAD CURRENT ERROR STATUS
100000          WDESTAT =100000                 ;WRITE DATA ERROR STATUS

                ;       P-CLOCK REGISTERS
172540          CLKCSR  =172540                 ;CLOCK CONTROL AND STATUS REGISTER
172542          CLKCSB  =172542                 ;CLOCK COUNT SET BUFFER
172544          CLKCTR  =172544                 ;CLOCK COUNTER

002240          ENDMOD

                .SBTTL  GLOBAL DATA SECTION

002240          BGNMOD  GLBDAT

                ;       TABLE OF OPERATION MESSAGES

002240  000000  OPMSGS: .WORD   0               ;FILLER
002242  005405          .WORD   MWRCHK          ;MESSAGE FOR WRITE CHECK
002244  005430          .WORD   MGTSTA          ;             GET STATUS
002246  005360          .WORD   MSEEK           ;             SEEK
002250  005375          .WORD   MREADH          ;             READ HEADER
002252  005416          .WORD   MWRITE          ;             WRITE DATA
```

GLOBAL DATA SECTION

```
1991 002254  005364                    .WORD   MREAD           ;           READ DATA
1992 002256  005513                    .WORD   MWRSET          ;           WITH RESET
1993 002260  005442                    .WORD   MDATCP          ;           WITH DATA COMPARE
1994 002262  005461                    .WORD   MHDRCP          ;           WITH HEADER COMPARE
1995 002264  005560                    .WORD   MCYLUP          ;           LOAD HEADS
1996 002266  005547                    .WORD   MULOAD          ;           UNLOAD HEADS
1997 002270  005607                    .WORD   MINOUT          ;               IN-OUT SEQ
1998 002272  005570                    .WORD   MOUTIN          ;               OUT IN SEQ
1999 002274  005630                    .WORD   MFOLWRT         ;               FOLLOWING WRITE
2000 002276  005650                    .WORD   MREVSK          ;               REV SEEK
2001 002300  005701                    .WORD   MFWDSK          ;               FWD SEEK
2002 002302  005766                    .WORD   MRESKO          ;               REV SEEK
2003 002304  005732                    .WORD   MFWSKO          ;               FWD SEEK
2004 002306  006022                    .WORD   MBADAD          ;               BAD DISK ADD FOR WRITE
2005 002310  005477                    .WORD   M40HDR          ;           40 HEADER OPERATION
2006 002312  000000            T.DRIVE: .WORD  0
2007 002314  000000            JJJ:     .WORD  0
2008 002316  000000            MLMTW:   .WORD  0
2009 002320  000000            CLRBYT:  .WORD  0
2010 002322  000000            NXTHL:   .WORD  0
2011 002324  000000            GBND:    .WORD  0
2012 002326  000000            CAMSK:   .WORD  0
2013 002330  000000            DIRMSK:  .WORD  0
2014 002332  000000            HDCYL:   .WORD  0
2015
2016                            ;       TABLE OF RESULT NAME MESSAGE ADDRESSES
2017 002334  007771            RFSTBL:  .WORD   MCERR           ;CONTROLLER ERROR
2018 002336  010102                     .WORD   MDRERR          ;DRIVE ERROR
2019 002340  010320                     .WORD   MNEERR          ;NON-EXISTANT MEMORY ERROR
2020 002342  010272                     .WORD   MFLERR          ;HEADER NOT FOUND-DATA LATE
2021 002344  010255                     .WORD   MHDERR          ;HEADER OR DATA ERROR
2022 002346  010245                     .WORD   MOPERR          ;OPERATION INCOMPLETE
2023 002350  010352                     .WORD   MNDRST          ;NO DRIVE STATUS AVAILABLE
2024 002352  000000                     .WORD   0
2025 002354  010230                     .WORD   MWDERR          ;WRITE DATA ERROR
2026 002356  010212                     .WORD   MHCERR          ;HEAD CURRENT ERROR
2027 002360  000000                     .WORD   0
2028 002362  010176                     .WORD   MSTERR          ;SEEK TIMEOUT ERROR
2029 002364  010143                     .WORD   MSPERR          ;SPINDLE ERROR
2030 002366  010161                     .WORD   MWGERR          ;WRITE GATE ERROR
2031 002370  000000                     .WORD   0
2032 002372  010113                     .WORD   MDSERR          ;DRIVE SELECT ERROR
2033
2034                            ;       PATTERN TABLE
2035 002374  005102            PATTBL:  .WORD   PAT1
2036 002376  005104                     .WORD   PAT2
2037 002400  005144                     .WORD   PAT3
2038 002402  005204                     .WORD   PAT4
2039 002404  005244                     .WORD   PAT5
2040 002406  005252                     .WORD   PAT6
2041 002410  005312                     .WORD   PAT7
2042 002412  005314                     .WORD   PAT8
2043 002414  005354                     .WORD   PAT9
2044 002416  005356                     .WORD   PAT10
2045
2046                            ;       SUBROUTINE CALLING STACK
2047 002420  000000            SUBSTK:  .WORD   0               ;STACK IS 12 WORDS LONG
```

GLOBAL DATA SECTION

```
2048 002422  000000                    .WORD   0
2049 002424  000000                    .WORD   0
2050 002426  000000                    .WORD   0
2051 002430  000000                    .WORD   0
205. 002432  000000                    .WORD   0
2053 002434  000000                    .WORD   0
2054 002436  000000                    .WORD   0
2055 002440  000000                    .WORD   0
2056 002442  000000                    .WORD   0
2057
2058                                ;RL01 TABLE OF CYLINDERS
2059 002444  000002                T25TBL: .WORD   2                ;TABLE OF DIFFERENCES
2060 002446  000006                        .WORD   6
2061 002450  000011                        .WORD   9.
2062 002452  000014                        .WORD   12.
2063 002454  000021                        .WORD   17.
2064 002456  000026                        .WORD   22.
2065 002460  000033                        .WORD   27.
2066 002462  000042                        .WORD   34.
2067 002464  000051                        .WORD   41.
2068 002466  000200                        .WORD   128.
2069 002470  000377                        .WORD   255.
2070
2071                                ;RL02 TABLE OF CYLINDERS
2072 002472  000004                T25TB2: .WORD   4
2073 002474  000014                        .WORD   12.
2074 002476  000022                        .WORD   18.
2075 002500  000030                        .WORD   24.
2076 002502  000042                        .WORD   34.
2077 002504  000054                        .WORD   44.
2078 002506  000066                        .WORD   54.
2079 002510  000104                        .WORD   68.
2080 002512  000122                        .WORD   82.
2081 002514  000400                        .WORD   256.
2082 002516  000777                        .WORD   511.
2083
2084                                ;       TABLE TO BE USED TO BUILD AND STORE THE CYLINDERS
2085
2086 002520                         T33TBL: .BLKW   16.
2087 002560                         TBT:    .BLKW   16.
2088
2089 002620      002                CYLTBL: .BYTE   2                ;TABLE OF DEFAULT CYLINDERS
2090 002621      007                        .BYTE   7.
2091 002622      016                        .BYTE   14.
2092 002623      024                        .BYTE   20.
2093 002624      033                        .BYTE   27.
2094 002625      041                        .BYTE   33.
2095 002626      046                        .BYTE   38.
2096 002627      055                        .BYTE   45.
2097 002630      064                        .BYTE   52.
2098 002631      072                        .BYTE   58.
2099 002632      101                        .BYTE   65.
2100 002633      110                        .BYTE   72.
2101 002634      115                        .BYTE   77.
2102 002635      124                        .BYTE   84.
2103 002636      133                        .BYTE   91.
2104 002637      141                        .BYTE   97.
```

GLOBAL DATA SECTION

```
2105 002640     146                     .BYTE   102.
2106 002641     154                     .BYTE   108.
2107 002642     161                     .BYTE   113.
2108 002643     170                     .BYTE   120.
2109 002644     177                     .BYTE   127.
2110 002645     206                     .BYTE   134.
2111 002646     213                     .BYTE   139.
2112 002647     222                     .BYTE   146.
2113 002650     230                     .BYTE   152.
2114 002651     235                     .BYTE   157.
2115 002652     244                     .BYTE   164.
2116 002653     252                     .BYTE   170.
2117 002654     261                     .BYTE   177.
2118 002655     270                     .BYTE   184.
2119 002656     275                     .BYTE   189.
2120 002657     303                     .BYTE   195.
2121 002660     312                     .BYTE   202.
2122 002661     317                     .BYTE   207.
2123 002662     326                     .BYTE   214.
2124 002663     334                     .BYTE   220.
2125 002664     343                     .BYTE   227.
2126 002665     352                     .BYTE   234.
2127 002666     361                     .BYTE   241.
2128 002667     367                     .BYTE   247.
2129 002670     375                     .BYTE   253.
2130 002671     000                     .BYTE   0
2131 002672  000401                     .WORD   257.
2132 002674  000406                     .WORD   262.
2133 002676  000415                     .WORD   269.
2134 002700  000423                     .WORD   275.
2135 002702  000432                     .WORD   282.
2136 002704  000445                     .WORD   293.
2137 002706  000454                     .WORD   300.
2138 002710  000463                     .WORD   307.
2139 002712  000471                     .WORD   313.
2140 002714  000500                     .WORD   320.
2141 002716  000507                     .WORD   327.
2142 002720  000514                     .WORD   332.
2143 002722  000523                     .WORD   339.
2144 002724  000532                     .WORD   346.
2145 002726  000540                     .WORD   352.
2146 002730  000545                     .WORD   357.
2147 002732  000553                     .WORD   363.
2148 002734  000560                     .WORD   368.
2149 002736  000567                     .WORD   375.
2150 002740  000576                     .WORD   382.
2151 002742  000605                     .WORD   389.
2152 002744  000612                     .WORD   394.
2153 002746  000621                     .WORD   401.
2154 002750  000627                     .WORD   407.
2155 002752  000634                     .WORD   412.
2156 002754  000643                     .WORD   419.
2157 002756  000651                     .WORD   425.
2158 002760  000660                     .WORD   432.
2159 002762  000667                     .WORD   439.
2160 002764  000674                     .WORD   444.
2161 002766  000702                     .WORD   450.
```

GLOBAL DATA SECTION

```
2162 002770  000711                    .WORD   457.
2163 002772  000716                    .WORD   462.
2164 002774  000725                    .WORD   469.
2165 002776  000733                    .WORD   475.
2166 003000  000742                    .WORD   482.
2167 003002  000751                    .WORD   489.
2168 003004  000760                    .WORD   496.
2169 003006  000766                    .WORD   502.
2170 003010  000774                    .WORD   508.
2171 003012  000774                    .WORD   508.
2172 003014  000000                    .WORD   0
2173 003016  000000          SSINDX:   .WORD   0             ;SUBROUTINE STACK INDEX POINTER
2174
2175                         ;         OPERATIONAL FLAGS
2176 003020  000000          OPFLAG:   .WORD   0             ;OPERATION FLAGS
2177 003022  000000          DONE:     .WORD   0             ;OPERATION COMPLETE FLAG
2178 003024  000000          HADONE:   .WORD   0             ;HEAD ALIGNMENT DONE FLAG
2179 003026  000000          ERHEAD:   .WORD   0             ;ADDRESS OF ERROR HEADER
2180 003030  000000          MORECE:   .WORD   0             ;MORE THAN 1 COMPARE ERROR
2181 003032  000000          ERRSWI:   .WORD   0             ;ERROR RETURN SWITCH
2182 003034  000000          BSFLAG:   .WORD   0             ;BAD SECTOR FLAGS
2183 003036  000000          WRTSWI:   .WORD   0             ;WRITE SWITCH
2184 003040  000000          TBLSTR:   .WORD   0             ;TABLE STORAGE
2185
2186 003042  000000          RLBAS:    .WORD   0             ;RL11 BASE ADDRESS
2187 003044  000000          RLVEC:    .WORD   0             ;RL11 VECTOR ADDRESS
2188 003046  000000          RLDRV:    .WORD   0             ;DRIVE NUMBER UNDER TEST
2189
2190 003050  000000          L.CS:     .WORD   0             ;CONTROLLER REGISTER STORAGE
2191 003052  000000          L.BA:     .WORD   0             ;BEFORE OPERATION
2192 003054  000000          L.DA:     .WORD   0
2193 003056  000000          L.MP:     .WORD   0
2194 003060  000000          T.CS:     .WORD   0             ;CONTROLLER REGISTER STORAGE
2195 003062  000000          T.BA:     .WORD   0             ;    AFTER OPERATION
2196 003064  000000          T.DA:     .WORD   0
2197 003066                  T.MP:
2198 003066  000000          HDWRD1:   .WORD   0             ;HEADER WORD STORAGE
2199 003070  000000          HDWRD2:   .WORD   0
2200 003072  000000          HDWRD3:   .WORD   0
2201
2202 003074  000000          T.STAT:   .WORD   0             ;DRIVE STATE STORAGE
2203
2204 003076  000000          RESPARM:  .WORD   0             ;PARAM BLOCK FOR REASON REPORT
2205 003100  000000                    .WORD   0
2206 003102  000000                    .WORD   0
2207 003104  000000                    .WORD   0
2208 003106  000000                    .WORD   0
2209
2210 003110  000000          DRVCNT:   .WORD   0             ;DRIVE COUNT FOR DRIVES UNDER TEST
2211 003112  000000          DIFAUG:   .WORD   0             ;DIFFERENCE AUGMENT FOR SEEK
2212 003114  000000          OLDCYL:   .WORD   0             ;OLD CYLINDER
2213 003116  000000          NEWCYL:   .WORD   0             ;NEW CYLINDER
2214 003120  000000          CURCYL:   .WORD   0             ;CURRENT CYLINDER
2215 003122  000000          DESDIF:   .WORD   0             ;DESIRED DIFFERENCE
2216 003124  000000          DESSGN:   .WORD   0             ;DESIRED SIGN
2217 003126  000000          DESHD:    .WORD   0             ;DESIRED HEAD
2218 003130  000000          DESSEC:   .WORD   0             ;DESIRED SECTOR
```

GLOBAL DATA SECTION

```
2219 003132  000000                TEMP0:  .WORD   0               ;TEMPORARY STORAGE
2220 003134  000000                TEMP1:  .WORD   0               ;TEMPORARY STARAGE
2221 003136  000000                TEMP2:  .WORD   0               ;TEMPORARY STORAGE
2222 003140  000000                TEMP3:  .WORD   0               ;TEMPORARY STORAGE
2223 003142  000000                TEMP4:  .WORD   0               ;TEMPORARY STORAGE
2224 003144  000000                TEMP5:  .WORD   0               ;TEMPORARY STORAGE
2225 003146  000000                TEMP6:  .WORD   0               ;TEMPORARY STORAGE
2226 003150  000000                TEMP7:  .WORD   0               ;TEMPORARY STORAGE
2227 003152  000000                TEMP8:  .WORD   0               ;TEMPORARY STORAGE
2229                               ;       TIMER STORAGE
2230 003154  000000                OFIN:   .WORD   0               ;ONE CYLINDER FORWARD INNER
2231 003156  000000                OFINU:  .WORD   0               ;       UPPER
2232 003160  000000                OFMID:  .WORD   0               ;ONE CYLINDER FORWARD MIDDLE
2233 003162  000000                OFMIDU: .WORD   0               ;       UPPER
2234 003164  000000                OFOUT:  .WORD   0               ;ONE CYLINDER FORWARD OUTER
2235 003166  000000                OFOUTU: .WORD   0               ;       UPPER
2236 003170  000000                ORIN:   .WORD   0               ;ONE CYLINDER REVERSE INNER
2237 003172  000000                ORINU:  .WORD   0               ;       UPPER
2238 003174  000000                ORMID:  .WORD   0               ;ONE CYLINDER REVERSE MIDDLE
2239 003176  000000                ORMIDU: .WORD   0               ;       UPPER
2240 003200  000000                OROUT:  .WORD   0               ;ONE CYLINDER REVERSE OUTER
2241 003202  000000                OROUTU: .WORD   0               ;       UPPER
2242 003204  000000                HFIN:   .WORD   0               ;128 CYLINDER FORWARD INNER
2243 003206  000000                HFINU:  .WORD   0               ;       UPPER
2244 003210  000000                HFOUT:  .WORD   0               ;128 CYLINDER FORWARD OUTER
2245 003212  000000                HFOUTU: .WORD   0               ;       UPPER
2246 003214  000000                HRIN:   .WORD   0               ;128 CYLINDER REVERSE INNER
2247 003216  000000                HRINU:  .WORD   0               ;       UPPER
2248 003220  000000                HROUT:  .WORD   0               ;128 CYLINDER REVERSE OUTER
2249 003222  000000                HROUTU: .WORD   0               ;       UPPER
2250 003224  000000                AFMID:  .WORD   0               ;256 CYLINDER FORWARD
2251 003226  000000                AFMIDU: .WORD   0               ;       UPPER
2252 003230  000000                ARMID:  .WORD   0               ;256 CYLINDER REVERSE
2253 003232  000000                ARMIDU: .WORD   0               ;       UPPER
2254
2255 003234  000226                EXOCYL: .WORD   150.            ;EXPECTED TIME ONE CYLINDER
2256 003236  001046                EXHCYL: .WORD   550.            ;EXPECTED TIME 128 CYLINDER
2257 003240  001750                EXACYL: .WORD   1000.           ;EXPECTED TIME 256 CYLINDER
2258 003242  000372                EXROT:  .WORD   250.            ;EXPECTED ROTATION TIME
2260 003244  000004                ERRVEC: .WORD   4               ;ERROR VECTOR
2261
2262                               ;       MISCELLANEOUS COUNTERS
2263 003246  000000                PASCNT: .WORD   0               ;PASS COUNTER (LOCAL TO A TEST)
2264 003250  000000                COUNT:  .WORD   0               ;A COUNTER (LOCAL TO A TEST)
2265 003252  000000                ERRPOINT: .WORD 0               ;ERROR POINTER
2266 003254                        ERRCNT: .BLKW   64.             ;ERROR COUNTER FOR PROGRAM
2267 003454  000000                PASNUM: .WORD   0               ;PASS NUMBER FOR PROGRAM
2268 003456  000000                PSETNM: .WORD   0               ;COUNTER FOR PARAMETER SET NUMBER IN USE
2269 003460     000                LOCERR: .BYTE   0               ;LOCAL ERROR COUNTER
2270 003461     000                NOERCT: .BYTE   0               ;INHIBIT ERROR COUNTING FLAG
2271 003462  000000                TRPFLG: .WORD   0               ;HARDWARE TRAP OCCURANCE
2272 003464  000000                PWRFLG: .WORD   0               ;POWER FAILURE OCCURANCE
2273 003466  000000                XDELAY: .WORD   0
2274 003470  000000                YDELAY: .WORD   0
2275 003472  000000                MININC: .WORD   0
2276 003474  000000                TEMP:   .WORD   0
2277 003476  000000                TIM.US: .WORD   0
```

GLOBAL DATA SECTION

```
2278 003500  000000                TAG:     .WORD   0
2279 003502  000000                MAJINC:  .WORD   0
2280 003504  000000                CLKFLG:  .WORD   0              ;FLAG INDICATING PRESENCE OF A P-CLOCK
2281 003506  000000                CLKADR:  .WORD   0              ;POINTER TO DIAGNOSTIC MONITOR CLOCK TABLE
2282
2283                               ;        BAD SECTOR TABLES AND POINTERS
2284 003510  000000                BSFVAL:  .WORD   0              ;BAD SECTORS FILES VALID FLAG
2285
2286 003512                        SBSFIL:  .BLKW   76             ;SOFTWARE BAD SECTOR FILE
2287 003706                        FBSFIL:  .BLKW   76             ;FACTORY BAD SECTOR FILE
2288
2289 004102                        IBUFF:   .BLKW   200            ;INPUT BUFFER
2290 004502                        OBUFF:   .BLKW   200            ;OUTPUT BUFFER
2291
2292 005102  000000                PAT1:    .WORD   0              ;PATTERN 1 (ALL ZEROS)
2293 005104  177772                PAT2:    .WORD   177772
2294 005106  177777                         .WORD   177777
2295 005110  177777                         .WORD   177777
2296 005112  052525                         .WORD   052525
2297 005114  052525                         .WORD   052525
2298 005116  052525                         .WORD   052525
2299 005120  177777                         .WORD   177777
2300 005122  177777                         .WORD   177777
2301 005124  052525                         .WORD   052525
2302 005126  052525                         .WORD   052525
2303 005130  177777                         .WORD   177777
2304 005132  052525                         .WORD   052525
2305 005134  177252                         .WORD   177252
2306 005136  177252                         .WORD   177252
2307 005140  172765                         .WORD   172765
2308 005142  172765                         .WORD   172765
2309
2310 005144  000003                PAT3:    .WORD   000003
2311 005146  000000                         .WORD   000000
2312 005150  000000                         .WORD   000000
2313 005152  177777                         .WORD   177777
2314 005154  177777                         .WORD   177777
2315 005156  177777                         .WORD   177777
2316 005160  000000                         .WORD   000000
2317 005162  000000                         .WORD   000000
2318 005164  177777                         .WORD   177777
2319 005166  177777                         .WORD   177777
2320 005170  000000                         .WORD   000000
2321 005172  177777                         .WORD   177777
2322 005174  000000                         .WORD   000000
2323 005176  177777                         .WORD   177777
2324 005200  000000                         .WORD   000000
2325 005202  177777                         .WORD   177777
2326
2327 005204  025252                PAT4:    .WORD   025252
2328 005206  052525                         .WORD   052525
2329 005210  052525                         .WORD   052525
2330 005212  125252                         .WORD   125252
2331 005214  125252                         .WORD   125252
2332 005216  125252                         .WORD   125252
2333 005220  052525                         .WORD   052525
2334 005222  052525                         .WORD   052525
```

GLOBAL DATA SECTION

```
2335 005224  125252                        .WORD   125252
2336 005226  125252                        .WORD   125252
2337 005230  052525                        .WORD   052525
2338 005232  125252                        .WORD   125252
2339 005234  052525                        .WORD   052525
2340 005236  125252                        .WORD   125252
2341 005240  052525                        .WORD   052525
2342 005242  125252                        .WORD   125252
2343
2344 005244  155555                PAT5:   .WORD   155555
2345 005246  133333                        .WORD   133333
2346 005250  066666                        .WORD   066666
2347
2348 005252  121105                PAT6:   .WORD   121105
2349 005254  150442                        .WORD   150442
2350 005256  064221                        .WORD   064221
2351 005260  132110                        .WORD   132110
2352 005262  055044                        .WORD   055044
2353 005264  026442                        .WORD   026442
2354 005266  013211                        .WORD   013211
2355 005270  105504                        .WORD   105504
2356 005272  042642                        .WORD   042642
2357 005274  021321                        .WORD   021321
2358 005276  110550                        .WORD   110550
2359 005300  044264                        .WORD   044264
2360 005302  022132                        .WORD   022132
2361 005304  011055                        .WORD   011055
2362 005306  104426                        .WORD   104426
2363 005310  042213                        .WORD   042213
2364
2365 005312  177777                PAT7:   .WORD   177777
2366
2367 005314  045513                PAT8:   .WORD   045513
2368 005316  122645                        .WORD   122645
2369 005320  151322                        .WORD   151322
2370 005322  064551                        .WORD   064551
2371 005324  132264                        .WORD   132264
2372 005326  055132                        .WORD   055132
2373 005330  026455                        .WORD   026455
2374 005332  113226                        .WORD   113226
2375 005334  045513                        .WORD   045513
2376 005336  122645                        .WORD   122645
2377 005340  151322                        .WORD   151322
2378 005342  064551                        .WORD   064551
2379 005344  132264                        .WORD   132264
2380 005346  055132                        .WORD   055132
2381 005350  026455                        .WORD   026455
2382 005352  113226                        .WORD   113226
2383
2384 005354  125252                PAT9:   .WORD   125252
2385
2386 005356  155555                PAT10:  .WORD   155555
2387
2388 005360                        ENDMOD
2389
2393
2394                               .SBTTL  GLOBAL MESSAGES
```

GLOBAL MESSAGES

```
2395
2396 005360                            BGNMOD  GLBTXT
2397
2398 005360   123   113   040  MSEEK:  .ASCIZ  /SK /
2399 005364   122   104   040  MREAD:  .ASCIZ  /RD DATA /
2400 005375   122   104   040  MREADH: .ASCIZ  /RD HDR /
2401 005405   127   122   124  MWRCHK: .ASCIZ  /WRT CHCK/
2402 005416   127   122   124  MWRITE: .ASCIZ  /WRT DATA /
2403 005430   107   105   124  MGTSTA: .ASCIZ  /GET STAT /
2404 005442   127   111   124  MDATCP: .ASCIZ  /WITH DATA CMP /
2405 005461   127   111   124  MHDRCP: .ASCIZ  /WITH HDR CMP /
2406 005477   106   117   122  M40HDR: .ASCIZ  /FOR 40 HDRS/
2407 005513   127   111   124  MWRSET: .ASCIZ  /WITH RESET /
2408 005527   117   120   105  MOPER:  .ASCIZ  /OPER: /
2409 005536   122   105   123  MRSLT:  .ASCIZ  /RESULT:       /
2410 005547   125   116   114  MULOAD: .ASCIZ  /UNLD DRV/
2411 005560   114   104   040  MCYLUP: .ASCIZ  /LD DRV /
2412 005570   106   117   114  MOUTIN: .ASCIZ  /FOL 0 TO CC SK/
2413 005607   106   117   114  MINOUT: .ASCIZ  /FOL 255 TO CC SK/
2414 005630   106   117   114  MFOLWRT:        .ASCIZ  /FOL WRT (NO SK)/
2415 005650   101   104   112  MREVSK: .ASCIZ  /ADJ CYL WRTTN AFT REV SK/
2416 005701   101   104   112  MFWDSK: .ASCIZ  /ADJ CYL WRTTN AFT FWD SK/
2417 005732   123   113   040  MFWSKO: .ASCIZ  /SK FWD,WRT - SK REV,OVERWRT/
2418 005766   123   113   040  MRESKO: .ASCIZ  /SK REV,WRT - SK FWD,OVERWRT/
2419 006022   117   116   040  MBADAD: .ASCIZ  /ON BAD SEC FILES/
2420 006043   103   101   116  MBADSF: .ASCIZ  /CAN'T GET BAD SEC FILES/
2421 006073   102   101   104  MFMTER: .ASCIZ  /BAD SEC FILE FMT ERR/
2422 006120   124   117   040  MTMBS:  .ASCIZ  /TO MANY BAD SEC /
2423 006141   102   125   123  BASADD: .ASCIZ  /BUS ADD=/
2424 006152   104   122   126  DRVNAM: .ASCIZ  /DRV=/
2425 006157   116   117   040  DRVNAV: .ASCIZ  /NO DRV FOR TST/
2426 006176   104   122   126  NOPWR:  .ASCIZ  /DRV DID NOT REC'R FROM PWR FAIL/
2427 006236   122   114   103  CSNAM:  .ASCIZ  /RLCS/
2428 006243   122   114   102  BANAM:  .ASCIZ  /RLBA/
2429 006250   122   114   104  DANAM:  .ASCIZ  /RLDA/
2430 006255   122   114   115  MPNAM:  .ASCIZ  /RLMP/
2431 006262   117   120   040  LAB1:   .ASCIZ  /OP INIT = /
2432 006275   117   120   040  LAB2:   .ASCIZ  /OP DONE = /
2433 006310   127   117   122  MWORD:  .ASCIZ  /WORD /
2434 006316   111   116   124  MTOSLOW: .ASCIZ /INTRPT TOO LATE/
2435 006336   116   117   040  MDRRES: .ASCIZ  /NO DRV RSPNSE/
2436 006354   116   117   040  MNOINT: .ASCIZ  /NO INTRPT ON CMND COMPLETE/
2437 006407   103   116   124  MCONHNG: .ASCIZ /CNTLR HUNG /
2438 006423   105   122   122  MNOCLR: .ASCIZ  /ERR DID NOT CLR/
2439 006443   126   117   114  VCNRST: .ASCIZ  /VOL CHK NOT RSET/
2440 006464   125   116   130  UNXERR: .ASCIZ  /UNXPCTED ERR/
2441 006501   040   124   105  TSTLAB: .ASCIZ  / TEST/
2459 006507   117   125   124  P2T03E: .ASCIZ  /OUT GRD BAND /
2460 006525   111   116   103  P2T04E: .ASCIZ  /INC SK FWD HD 0/
2461 006545   111   116   103  P2T05E: .ASCIZ  /INC SK REV HD 0/
2462 006565   111   116   103  P2T06E: .ASCIZ  /INC SK FWD HD 1/
2463 006605   111   116   116  P2T07E: .ASCIZ  /INN GRD BAND /
2464 006623   111   116   103  P2T08E: .ASCIZ  /INC SK REV HD 1/
2465 006643   123   113   000  P2T09E: .ASCIZ  /SK/
2466 006646   106   127   104  P2T10E: .ASCIZ  /FWD OSC SK/
2467 006661   122   105   126  P2T11E: .ASCIZ  /REV OSC SK/
2468 006674   123   113   040  P2T12E: .ASCIZ  /SK TIMING/
```

GLOBAL MESSAGES

```
2469 006706   102   123   103  P2T13E: .ASCIZ  /BSC RD DATA/
2470 006722   127   122   124  P2T14E: .ASCIZ  &WRT/RD DATA (P1)&
2471 006743   123   120   111  P2T15E: .ASCIZ  /SPINDLE ROT TIMING/
2472 006766   127   122   124  P2T16E: .ASCIZ  &WRT/RD DATA (P2)&
2473 007007   127   122   124  P2T17E: .ASCIZ  /WRT LCK ERR AND DATA PROT/
2474 007041   101   104   112  P2T18E: .ASCIZ  /ADJ CYL INTERFNCE/
2475 007063   117   126   105  P2T19E: .ASCIZ  /OVERWRT/
2476 007073   123   113   040  SKTMES: .ASCIZ  /SK TIMES /
2477 007105   123   120   111  SRTMES: .ASCIZ  /SPINDLE ROT TIME /
2478 007127   050   111   116  VALDES: .ASCIZ  /(IN 100'S OF U SEC)/
2479 007153   101   120   120  MAPROX: .ASCIZ  /APPROX /
2480 007163   111   116   116  LABIN:  .ASCIZ  /INNER/
2481 007171   115   111   104  LABMID: .ASCIZ  /MIDDLE/
2482 007200   117   125   124  LABOUT: .ASCIZ  /OUTER/
2483 007206   115   101   130  LABEXP: .ASCIZ  /MAX TIME/
2484 007217   061   040   103  LABOCF: .ASCIZ  /1 CYL FWD/
2485 007231   061   040   103  LABOCR: .ASCIZ  /1 CYL REV/
2486 007243   115   111   104  LABMCF: .ASCIZ  /MID CYL FWD/
2487 007257   115   111   104  LABMCR: .ASCIZ  /MID CYL REV/
2488 007273   115   101   130  LABACF: .ASCIZ  /MAX CYL FWD/
2489 007307   115   101   130  LABACR: .ASCIZ  /MAX CYL REV/
2491 007323   110   104   123  HDMOVF: .ASCIZ  /HDS FAILED TO MV IN 10 TRYS/
2509 007357   122   105   123  OPR12:  .ASCIZ  /RESET WRT LCK /
2510 007376   117   116   040  OPR1A:  .ASCIZ  /ON /
2511 007402   117   116   040  OPR1B:  .ASCIZ  /ON DRV /
2512 007412   125   116   104  UNDTST: .ASCIZ  /UNDER TEST/
2513 007425   123   105   124  OPR004: .ASCIZ  /SET WRT LCK /
2514 007442   104   111   106  DIFWD:  .ASCIZ  /DIFF /
2515 007450   123   107   116  SGNWD:  .ASCIZ  /SGN /
2516 007455   110   104   040  HDWD:   .ASCIZ  /HD /
2517 007461   123   105   103  SECWD:  .ASCIZ  /SEC /
2518 007466   103   131   114  CYLWD:  .ASCIZ  /CYL /
2519 007473   106   122   117  FRMWD:  .ASCIZ  /FROM /
2520 007501   040   102   131  BYPSNM: .ASCIZ  / BYPASSED /
2521 007514   122   117   125  SEQMES: .ASCIZ  /ROUTINE TRACE SEQ:/
2522 007537   104   122   126  STAMES: .ASCIZ  /DRV STAT/
2523 007550   102   101   104  BSNSTR: .ASCIZ  /BAD SEC FILES NOT STRD. ALL SEC ASSUMED OK./
2524 007624   124   117   124  TCERR:  .ASCIZ  /TOTAL CMP ERRS: /
2525 007645   104   122   111  NOCTLR: .ASCIZ  /DRIVE DROPPED   NO CONTROLLER/
2526 007703   104   122   111  NOTRDY: .ASCIZ  /DRIVE DROPPED - DID NOT RESPOND WITH "READY"/
2527
2528                            ;       RESULT NAMES
2529 007760   104   122   126  MDRDY:  .ASCIZ  /DRV RDY /
2530 007771   103   117   116  MCERR:  .ASCIZ  /CONT ERR /
2531 010003   110   104   122  MHCRC:   ASCIZ  /HDR CRC/
2532 010013   104   101   124  MDCRC:  .ASCIZ  /DATA CRC/
2533 010024   110   104   122  MHNF:   .ASCIZ  /HDR NOT FND/
2534 010040   104   101   124  MDLT:   .ASCIZ  /DATA LATE/
2535 010052   110   104   122  MHFCRC: .ASCIZ  &HDR NOT FND/HDR CRC/OPI&
2536 010102   104   122   126  MDRERR: .ASCIZ  /DRV ERR /
2545 010113   104   122   126  MDSERR: .ASCIZ  /DRV SEL ERR /
2546 010130   104   122   126  MDRVST: .ASCIZ  /DRV STATE /
2547 010143   123   120   111  MSPERR: .ASCIZ  /SPIN TIMEOUT /
2548 010161   127   122   124  MWGERR: .ASCIZ  /WRT GAT ERR /
2549 010176   123   113   040  MSTERR: .ASCIZ  /SK TIMEOUT /
2550 010212   110   105   101  MHCERR: .ASCIZ  /HEAD CUR ERR /
2551 010230   127   122   124  MWDERR: .ASCIZ  /WRT DAT ERR /
```

GLOBAL MESSAGES

```
2552 010245  117  120  122  MOPERR: .ASCIZ  /OPR INC/
2553 010255  110  104  122  MHDERR: .ASCIZ  &HDR/DAT ERR &
2554 010272  110  104  122  MFLERR: .ASCIZ  &HDR NOT FND/DAT LATE &
2555 010320  116  117  116  MNEERR: .ASCIZ  /NON-EXISTENT MEMORY /
2556 010345  103  131  114  MCYLOC: .ASCIZ  /CYL /
2557 010352  103  101  116  MNDRST: .ASCIZ  /CAN'T GET DRV STAT/
2558 010375  125  116  113  MUNDEF: .ASCIZ  /UNKN DRV STATE NO RDY,NO ERR,HDS OUT/
2559 010442  106  101  111  MRLFAL: .ASCIZ  /FAIL TO RELD HDS AFTER ERR CLR/
2560 010501  127  122  124  MWRTAB: .ASCIZ  /WRT ABRTD/
2561 010513  040  117  126  MEXERS: .ASCIZ  / OVR ERR LIMIT   UNIT DRPPD /
2562 010550  040  105  122  MERRS:  .ASCIZ  / ERR/
2563 010555  207  377  377  BELL:   .ASCIZ  <207><37/><377>
2564
2565                        ;       RESULT SETTINGS
2566 010561  111  123  040  RESE3:  .ASCIZ  /IS /
2567 010565  040  123  102  RESE4:  .ASCIZ  / SB /
2568
2569                        ;       RESULT CONDITIONS
2570 010572  040  111  116  RESE5:  .ASCIZ  / IN /
2571 010577  040  117  106  RESE6:  .ASCIZ  / OF /
2572 010604  123  124  101  STATE2: .ASCIZ  /STATE 2/
2573 010614  123  124  101  STATE3: .ASCIZ  /STATE 3/
2574 010624  123  124  101  STATE5: .ASCIZ  /STATE 5/
2578 010634  061  123  124  C10MS:  .ASCIZ  /1ST 3 MS/
2579 010645  065  060  060  C500MS: .ASCIZ  /500MS/
2580 010653  103  131  103  CCYLUP: .ASCIZ  /CYC UP/
2581 010662  104  101  124  CAFDT:  .ASCIZ  /DATA XFR/
2582 010673  065  040  123  C5SEC:  .ASCIZ  /5 SEC/
2583
2584 010701  045  116  045  FMTOP1: .ASCIZ  /%N%T%N%T%T%O6%S%T%O1%N/
2585 010730  045  116  045  FMTOP2: .ASCIZ  /%N%T%O1%S1%T%O1%N/
2586 010752  045  116  045  FMTOP3: .ASCIZ  /%N%T%O1%S1%T%T%N/
2587 010773  045  124  045  FMT1:   .ASCIZ  /%T%T/
2588 011000  045  116  045  FMT1.1: .ASCIZ  /%N%T%T/
2589 011007  045  124  000  FMT2:   .ASCIZ  /%T/
2590 011012  045  116  000  FMT3:   .ASCIZ  /%N/
2591 011015  045  116  045  FMT4:   .ASCIZ  /%N%T%T%N/
2592 011026  045  116  045  FMT5:   .ASCIZ  /%N%T%O6%S1%T%O1/
2593 011046  045  116  045  FMT6:   .ASCIZ  /%N%S11%T%S4%T%S4%T%S4%T%S4%T%S2%T/
2594 011110  045  116  045  FMT7:   .ASCIZ  /%N%T%O6%S2%O6%S2%O6%S2%O6%S3%O3%S2%O1%N/
2595 011160  045  116  045  FMT8:   .ASCIZ  /%N%T%O6%S2%O6%S2%O6%S2%O6/
2596 011212  045  116  045  FMT9:   .ASCIZ  /%N%T/
2597 011217  045  124  045  FMT11:  .ASCIZ  /%T%O1/
2598 011225  045  124  045  FMT12:  .ASCIZ  /%T%O3/
2599 011233  045  116  045  FMT13:  .ASCIZ  /%N%S11%T%O3%S1%T%O3%S1%T%O1%S1%T%O1/
2600 011277  045  116  045  FMT14:  .ASCIZ  /%N%T%T%D3%S1%T%O6%S1%T%O6/
2601 011331  045  116  045  FMT15:  .ASCIZ  /%N%S11%T%D3%S1%T%O6%S1%T%O6/
2602 011365  045  116  045  FMT16:  .ASCIZ  /%N%S5%O6/
2603 011376  045  123  061  FMT17:  .ASCIZ  /%S10%T%N%S11%O6%N/
2604 011420  045  116  045  FMT18:  .ASCIZ  /%N%S15%T%S5%T%S4%T%S5%T%N/
2605 011452  045  124  045  FMT19:  .ASCIZ  /%T%S4%D6%S4%D6%S4%D6%S4%D6%N/
2606 011507  045  124  045  FMT20:  .ASCIZ  /%T%S2%D6%S14%D6%S4%D6%N/
2607 011537  045  124  045  FMT21:  .ASCIZ  /%T%S12%D6%S14%O6%N/
2608 011562  045  116  045  FMT22:  .ASCIZ  /%N%S11%T%O3%S1%T%O1%S1%T%O2/
2609 011616  045  124  045  FMT23:  .ASCIZ  /%T%T%T%O1%N/
2610 011632  045  116  045  FMT24:  .ASCIZ  /%N%T/
2611 011637  045  116  045  FMT25:  .ASCIZ  /%N%D2%T/
```

GLOBAL MESSAGES

```
2612 011647  045  116  045  FMT26:  .ASCIZ  /≤N≤S1≤T≤D4≤T≤T≤D3≤N/
2613 011673  045  116  045  FMT27:  .ASCIZ  /≤N≤T≤D3≤T≤D3≤N/
2614 011712  045  116  045  FMT28:  .ASCIZ  /≤N≤T≤T≤T/
2615
2616 011723                       ENDMOD
2617
2622
2623                              .SBTTL  ERROR MESSAGES
2624 011724                       BGNMOD  GLBERR
2625                              ;       ERR1    R3 POINTS TO RESULT MESSAGE
2626                              ;               RESULT: (R3)
2627
2628                              ;       ERR2    R3 POINTS TO RESULT NAME
2629                              ;               RESULT: (R3) IS 1 SB 0
2630
2631                              ;       ERR3    R3 POINTS TO RESULT NAME
2632                              ;               RESULT: (R3) IS 0 SB 1
2633
2634                              ;       ERR4    R3 POINTS TO RESULT NAME
2635                              ;               R4 POINTS TO RESULT CONDITIONS
2636                              ;               RESULT: (R3) IS 1 SB 0 (R4)
2637
2638                              ;       ERR5    R3 POINTS TO RESULT NAME
2639                              ;               R4 POINTS TO RESULT CONDITIONS
2640                              ;               RESULT: (R3) IS 0 SB 1 (R4)
2641
2642                              ;       ERR6    RESULT ROUTINE DETERMINES WHICH ERROR(S) ARE SET AND
2643                              ;               REPORTS ALL
2644                              ;               RESULT: "ERROR" IS 1 SB 0
2645
2646                              ;       ERR7    DRIVE STATE ERROR REPORT
2647                              ;               R3 CONTAINS EXPECTED STATE
2648                              ;               T.STAT CONTAINS BAD STATE
2649                              ;               RESULT: DRIVE STATE IS (T.STAT) SB (R3)
2650
2651                              ;       ERR8    HEAD POSITIONING ERROR REPORT
2652                              ;               NEWCYL CONTAINS EXPECTED CYLINDER
2653                              ;               HDWRD1 CONTAINS BAD CYLINDER
2654                              ;               RESULT: CYLINDER IS (HDWRD1) SB (NEWCYL)
2655
2656                              ;       ERR9    UTILITY RESULT REPORT
2657                              ;               R3 POINTS TO RESULT NAME
2658                              ;               R4 POINTS TO VALUE 1
2659                              ;               R5 POINTS TO VALUE 2
2660                              ;               RESULT: (R3-NAME) IS (R4 VALUE 1) SB (R5 VALUE 2)
2661
2662                              ;       ERR10   COMPARE ERROR REPORT
2663                              ;               R3 CONTAINS THE BAD WORD NUMBER
2664                              ;               R4 POINTS TO BAD WORD
2665                              ;               R5 POINTS TO GOOD WORD
2666                              ;               RESULT: WORD (R3) IS (R4) SB (R5)
2667
2668 011724                       BGNMSG  ERR1
2669 011724  105737  003461              TSTB    NOERCT          ;TEST IF ERROR COUNTING INHIBITED
2670 011730  001002                      BNE     1$              ;YES - SKIP
2671 011732  005277  171314              INC     @ERRPOINT       ;ELSE BUMP ERROR COUNT
2672 011736  010146              1$:     MOV     R1, (SP)        ;STORE R1
```

ERROR MESSAGES

```
2673 011740  004737  024542                JSR     PC,RPTOP        ;REPORT OPERATION
2674 011744  012721  000001                MOV     #1,(R1)+        ;SET PARAM NUMBER
2675 011750  010321                        MOV     R3,(R1)+        ;INSERT MESSAGE ADDRESS POINTER
2676 011752  004737  025330                JSR     PC,RPTRES       ;REPORT RESULTS
2677 011756  004737  025536                JSR     PC,RPTREM       ;REPORT REMAINDER
2678 011762  012601                        MOV     (SP)+,R1        ;RESTORE R1
2679 011764  004737  015712                JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
2680 011770                        ENDMSG
     011770                        L10000:
     011770  104423                        TRAP    C$MSG
2681
2682 011772                        BGNMSG  ERR2
2683 011772  005277  171254                INC     @ERRPOINT       ;BUMP ERROR COUNT
2684 011776  010146                        MOV     R1, (SP)        ;STORE R1
2685 012000  004737  024542                JSR     PC,RPTOP        ;REPORT OPERATION
2686 012004  012721  000003                MOV     #3,(R1)+        ;SET PARAM NUMBER
2687 012010  010321                        MOV     R3,(R1)+        ;INSERT NAME ADD POINTER
2688 012012  012721  000001                MOV     #1,(R1)+        ;SET IS VALUE
2689 012016  005021                        CLR     (R1)+           ;SET SB VALUE
2690 012020  004737  025330                JSR     PC,RPTRES       ;REPORT RESULTS
2691 012024  004737  025536                JSR     PC,RPTREM       ;REPORT REMAINDER
2692 012030  012601                        MOV     (SP)+,R1        ;RESTORE R1
2693 012032  004737  015712                JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
2694 012036                        ENDMSG
     012036                        L10001:
     012036  104423                        TRAP    C$MSG
2695
2696 012040                        BGNMSG  ERR3
2697 012040  005277  171206                INC     @ERRPOINT       ;BUMP ERROR COUNT
2698 012044  010146                        MOV     R1,-(SP)        ;STORE R1
2699 012046  004737  024542                JSR     PC,RPTOP        ;REPORT OPERATION
2700 012052  012721  000003                MOV     #3,(R1)+        ;SET PARAM NUMBER
2701 012056  010321                        MOV     R3,(R1)+        ;INSERT NAME ADD POINTER
2702 012060  005021                        CLR     (R1)+           ;SET IS VALUE
2703 012062  012721  000001                MOV     #1,(R1)+        ;SET SB VALUE
2704 012066  004737  025330                JSR     PC,RPTRES       ;REPORT RESULTS
2705 012072  004737  025536                JSR     PC,RPTREM       ;REPORT REMAINDER
2706 012076  012601                        MOV     (SP)+,R1        ;RESTORE R1
2707 012100  004737  015712                JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
2708 012104                        ENDMSG
     012104                        L10002:
     012104  104423                        TRAP    C$MSG
2709
2710 012106                        BGNMSG  ERR4
2711 012106  005277  171140                INC     @ERRPOINT       ;BUMP ERROR COUNT
2712 012112  010146                        MOV     R1 -(SP)        ;STORE R1
2713 012114  004737  024542                JSR     PC,RPTOP        ;REPORT OPERATION
2714 012120  012721  000004                MOV     #4,(R1)+        ;SET PARAM NUMBER
2715 012124  010321                        MOV     R3,(R1)+        ;INSERT NAME ADD POINTER
2716 012126  012721  000001                MOV     #1,(R1)+        ;SET IS VALUE
2717 012132  005021                        CLR     (R1)+           ;SET SB VALUE
2718 012134  010411                        MOV     R4,(R1)         ;INSERT ADD OF CONDITION POINTER
2719 012136  004737  025330                JSR     PC,RPTRES       ;REPORT RESULTS
2720 012142  004737  025536                JSR     PC,RPTREM       ;REPORT REMAINDER
2721 012146  012601                        MOV     (SP)+,R1        ;RESTORE R1
2722 012150  004737  015712                JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
2723 012154                        ENDMSG
```

```
012154
012154  104423                          L10003:
                                                TRAP    C$MSG
2724
2725 012156                             BGNMSG  ERR5
2726 012156  005277  171070                        INC     @ERRPOINT       ;BUMP ERROR COUNT
2727 012162  010146                                MOV     R1,(SP)         ;STORE R1
2728 012164  004737  024542                        JSR     PC,RPTOP        ;REPORT OPERATION
2729 012170  012721  000004                        MOV     #4,(R1)+        ;SET PARAM NUMBER
2730 012174  010321                                MOV     R3,(R1)+        ;INSERT NAME ADD POINTER
2731 012176  005021                                CLR     (R1)+           ;SET IS VALUE
2732 012200  012721  000001                        MOV     #1,(R1)+        ;SET SB VALUE
2733 012204  010411                                MOV     R4,(R1)         ;INSERT ADD OF CONDITION POINTER
2734 012206  004737  025330                        JSR     PC,RPTRES       ;REPORT RESULTS
2735 012212  004737  025536                        JSR     PC,RPTREM       ;REPORT REMAINDER
2736 012216  012601                                MOV     (SP)+,R1        ;RESTORE R1
2737 012220  004737  015712                        JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
2738 012224                             ENDMSG
012224                                  L10004:
012224  104423                                  TRAP    C$MSG
2739
2740 012226                             BGNMSG  ERR6
2741 012226  105737  003461                        TSTB    NOERCT          ;TEST IF ERROR COUNTING INHIBITED
2742 012232  001002                                BNE     17$             ;YES   SKIP
2743 012234  005277  171012                        INC     @ERRPOINT       ;ELSE BUMP ERROR COUNT
2744 012240  010146                     17$:       MOV     R1,(SP)         ;STORE R1
2745 012242  010346                                MOV     R3,(SP)         ;STORE R3
2746 012244  010446                                MOV     R4,(SP)         ;STORE R4
2747 012246  010546                                MOV     R5,(SP)         ;STORE R5
2748 012250  004737  024542                        JSR     PC,RPTOP        ;REPORT OPERATION
2749 012254  012721  000003                        MOV     #3,(R1)+        ;SET PARAM NUMBER
2750 012260  012761  000001  000002                MOV     #1,2(R1)        ;INSERT IS VALUE
2751 012266  005037  003140                        CLR     TEMP3           ;CLEAR FOR STATUS STORAGE
2752 012272  013703  003060                        MOV     T.CS,R3         ;GET T.CS
2753 012276  042703  177761                        BIC     #177761,R3      ;AND CLEAR ALL BUT FUNCTION
2754 012302  022703  000004                        CMP     #4,R3           ;CHECK IF IT WAS GET STATUS
2755 012306  001434                                BEQ     1$              ;YES - STATUS IS IN T.MP, SKIP
2756 012310  012762  000003  000004                MOV     #GETSTAT,RLDA(R2) ;ELSE DO GET STATUS
2757 012316  012703  000004                        MOV     #4,R3
2758 012322  053703  003046                        BIS     RLDRV,R3
2759 012326  010362  000000                        MOV     R3,RLCS(R2)
2760 012332                                        WAITUS  #10.            ;WAIT FOR CONTROLLER READY
2761 012344  032762  000200  000000                BIT     #CRDYMSK,RLCS(R2)       ;TEST IF READY
2762 012352  001003                                BNE     10$             ;YES - SKIP
2763 012354  012703  001000             9$:        MOV     #BIT9,R3        ;ELSE SET NO DRIVE STATUS BIT
2764 012360  000413                                BR      2$              ;IN MESSAGE WORD AND SKIP
2765 012362  016203  000006             10$:       MOV     RLMP(R2),R3     ;STORE STATUS FOR REPORT
2766 012366  010337  003140                        MOV     R3,TEMP3
2767 012372  113703  003141                        MOVB    TEMP3+1,R3      ;GET ERROR BITS IN PROPER POSITION
2768 012376  000402                                BR      13$
2769 012400  113703  003067             1$:        MOVB    T.MP+1,R3       ;GET ERROR BITS FROM MP REG
2770 012404  042703  177442             13$:       BIC     #177442,R3      ;CLEAR UNUSED BITS
2771 012410  013704  003060             2$:        MOV     T.CS,R4         ;GET ERROR BITS FROM CS REG
2772 012414  042704  001777                        BIC     #1777,R4        ;CLEAR UNUSED BITS
2773 012420  050403                                BIS     R4,R3           ;MAKE ONE WORD OF POSSIBLE ERRORS
2774 012422  032703  002000                        BIT     #OPIERR,R3      ;TEST IF OPI SET
2775 012426  001442                                BEQ     11$             ;NO - SKIP
2776 012430  032703  010000                        BIT     #HNFERR,R3      ;TEST IF HDR NOT FOUND ERROR
```

ERROR MESSAGES

```
2777 012434 001026                      BNE     107$            ;YES   SKIP
2778 012436 032703 004000               BIT     #HCRCERR,R3     ;TEST IF HDR CRC ERR
2779 012442 001020                      BNE     105$            ;YES   SKIP
2780 012444 012704 010245               MOV     #MOPERR,R4      ;SET OPI ALONE MESSAGE
2781 012450                     100$:   PRINTB  #FMT28,#MRSLT,R4,#MERRS ;REPORT ERROR
     012450 012746 010550               MOV     #MERRS,-(SP)
     012454 010446                      MOV     R4,-(SP)
     012456 012746 005536               MOV     #MRSLT,-(SP)
     012462 012746 011712               MOV     #FMT28,-(SP)
     012466 012746 000004               MOV     #4,-(SP)
     012472 010600                      MOV     SP,R0
     012474 104414                      TRAP    C$PNTB
     012476 062706 000012               ADD     #12,SP
2782 012502 000430                      BR      120$            ;SKIP
2783 012504 012704 010003       105$:   MOV     #MHCRC,R4       ;HDR CRC MESSAGE
2784 012510 000757                      BR      100$
2785 012512 032703 004000       107$:   BIT     #HCRCERR,R3     ;TEST IF HCRC WITH HDR NOT FND
2786 012516 001003                      BNE     109$            ;YES   SKIP
2787 012520 012704 010024               MOV     #MHNF,R4        ;MESSAGE HEADER NOT FOUND
2788 012524 000751                      BR      100$
2789 012526 012704 010052       109$:   MOV     #MHFCRC,R4      ;HNF AND HCRC MESSAGE
2790 012532 000746                      BR      100$            ;SKIP
2791 012534 032703 004000       115$:   BIT     #DCKERR,R3      ;TEST IF DATA CHECK SET, NOT OPI
2792 012540 001403                      BEQ     118$            ;NO - SKIP
2793 012542 012704 010013               MOV     #MDCRC,R4       ;SET MESSAGE DATA CHECK
2794 012546 000740                      BR      100$            ;SKIP
2795 012550 032703 010000       118$:   BIT     #DLTERR,R3      ;TEST IF DATA LATE ERROR
2796 012554 001403                      BEQ     120$            ;NO - SKIP
2797 012556 012704 010040               MOV     #MDLT,R4        ;SET MESSAGE DATA LATE
2798 012562 000732                      BR      100$            ;SKIP
2799 012564 012705 100000       120$:   MOV     #BIT15,R5       ;SET BIT POINTER FOR TEST
2800 012570 005004                      CLR     R4              ;CLEAR R4 FOR TABLE COUNT
2801 012572 030503              3$:     BIT     R5,R3           ;TEST IF BIT IS SET
2802 012574 001005                      BNE     6$              ;YES - SKIP TO REPORT
2803 012576 005724              4$:     TST     (R4)+           ;ELSE BUMP TABLE POINTER
2804 012600 000241                      CLC                     ;CLEAR CARRY
2805 012602 006005                      ROR     R5              ;SHIFT BIT POINTER TO NEXT BIT
2806 012604 001372                      BNE     3$              ;LOOP IF NOT 0
2807 012606 000405                      BR      7$              ;ELSE REPORT REMAINDER
2808 012610 016411 002334       6$:     MOV     RESTBL(R4),(R1) ;INSERT NAME ADDRESS
2809 012614 004737 025330               JSR     PC,RPTRES       ;REPORT RESULTS
2810 012620 000766                      BR      4$              ;GET NEXT BIT
2811 012622 004737 025536       7$:     JSR     PC,RPTREM       ;REPORT REMAINDER
2812 012626 005737 003140               TST     TEMP3           ;TEST IF ANY NEW STATUS
2813 012632 001414                      BEQ     15$             ;NO    SKIP
2814 012634                             PRINTB  #FMT17,#STAMES,TEMP3
     012634 013746 003140               MOV     TEMP3,-(SP)
     012640 012746 007537               MOV     #STAMES,-(SP)
     012644 012746 011376               MOV     #FMT17,-(SP)
     012650 012746 000003               MOV     #3,-(SP)
     012654 010600                      MOV     SP,R0
     012656 104414                      TRAP    C$PNTB
     012660 062706 000010               ADD     #10,SP
2815 012664 032737 004000 003060 15$:   BIT     #DCKERR,T.CS    ;TEST IF DATA CHECK ERROR
2816 012672 001453                      BEQ     25$             ;NO - SKIP
2817 012674 032737 002000 003060        BIT     #OPIERR,T.CS    ;TEST IF OPI SET
2818 012702 001047                      BNE     25$             ;YES   SKIP
```

ERROR MESSAGES

```
2819 012704 005037 003030                CLR     MORECE          ;CLEAR COMPARE ERROR COUNT
2820 012710 012701 000200                MOV     #128.,R1        ;SET COMPARE LENGTH
2821 012714 012703 000001                MOV     #1,R3           ;SET WORD COUNT
2822 012720 012705 004502                MOV     #OBUFF,R5       ;SET GOOD WORD POINTER
2823 012724 012704 004102                MOV     #IBUFF,R4       ;SET TEST WORD POINTER
2824 012730 021514              18$:     CMP     (R5),(R4)       ;CHECK WORD
2825 012732 001427                       BEQ     19$             ;GOOD   SKIP
2826 012734 023727 003030 000012         CMP     MORECE,#10.     ;TEST IF COMPARE LIMIT REACHED
2827 012742 003021                       BGT     20$             ;YES - SKIP
2828 012744                              PRINTB  #FMT15,#MWORD,R3,#RESE3,(R4),#RESE4,(R5)
     012744 011546                       MOV     (R5), (SP)
     012746 012746 010565                MOV     #RESE4, (SP)
     012752 011446                       MOV     (R4),-(SP)
     012754 012746 010561                MOV     #RESE3,-(SP)
     012760 010346                       MOV     R3,-(SP)
     012762 012746 006310                MOV     #MWORD,-(SP)
     012766 012746 011331                MOV     #FMT15,-(SP)
     012772 012746 000007                MOV     #7,-(SP)
     012776 010600                       MOV     SP,R0
     013000 104414                       TRAP    C$PNTB
     013002 062706 000020                ADD     #20,SP
2829 013006 005237 003030        20$:    INC     MORECE          ;BUMP ERROR COUNTER
2830 013012 022524               19$:    CMP     (R5)+,(R4)+     ;BUMP POINTERS
2831 013014 005203                       INC     R3              ;BUMP COUNTER
2832 013016 005301                       DEC     R1              ;DEC LENGTH COUNT
2833 013020 001343                       BNE     18$             ;LOOP IF NOT DONE
2834 013022 005737 003030        25$:    TST     MORECE          ;TEST IF ANY COMPARE ERRORS
2835 013026 001421                       BEQ     27$             ;NO   SKIP
2836 013030 012701 000200                MOV     #128.,R1        ;SET COMPARE LENGTH
2837 013034                              PRINTB  #FMT27,#TCERR,MORECE,#RESE6,R1
     013034 010146                       MOV     R1,-(SP)
     013036 012746 010577                MOV     #RESE6,-(SP)
     013042 013746 003030                MOV     MORECE,-(SP)
     013046 012746 007624                MOV     #TCERR,-(SP)
     013052 012746 011673                MOV     #FMT27,-(SP)
     013056 012746 000005                MOV     #5,-(SP)
     013062 010600                       MOV     SP,R0
     013064 104414                       TRAP    C$PNTB
     013066 062706 000014                ADD     #14,SP
2838 013072 012605               27$:    MOV     (SP)+,R5        ;RESTORE R5, 4, 3, 1
2839 013074 012604                       MOV     (SP)+,R4
2840 013076 012603                       MOV     (SP)+,R3
2841 013100 012601                       MOV     (SP)+,R1
2842 013102 004737 015712                JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
2843 013106                      ENDMSG
     013106                      L10005:
     013106 104423                       TRAP    C$MSG
2844
2845 013110                      BGNMSG  ERR7
2846 013110 005277 170136                INC     @ERRPOINT       ;BUMP ERROR COUNT
2847 013114 010146                       MOV     R1,-(SP)        ;STORE R1
2848 013116 004737 024542                JSR     PC,RPTOP        ;REPORT OPERATION
2849 013122 012721 000003                MOV     #3,(R1)+        ;SET PARAM NUMBER
2850 013126 012721 010130                MOV     #MDRVST,(R1)+   ;INSERT NAME ADD POINTER
2851 013132 013721 003074                MOV     T.STAT,(R1)+    ;INSERT IS VALUE
2852 013136 010311                       MOV     R3,(R1) ;INSERT SB VALUE
2853 013140 004737 025330                JSR     PC,RPTRES       ;REPORT RESULTS
```

ERROR MESSAGES

```
2854 013144  004737  025536                 JSR     PC,RPTREM       ;REPORT REMAINDER
2855 013150  012601                         MOV     (SP)+,R1        ;RESTORE R1
2856 013152  004737  015712                 JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
2857 013156                         ENDMSG
     013156                         L10006:
     013156  104423                         TRAP    C$MSG
2858
2859 013160                         BGNMSG  ERR8
2860 013160  005277  170066                 INC     @ERRPOINT       ;BUMP ERROR COUNT
2861 013164  010146                         MOV     R1,-(SP)        ;STORE R1
2862 013166  010346                         MOV     R3,(SP)         ;STORE R3
2863 013170  004737  024542                 JSR     PC,RPTOP        ;REPORT OPERATION
2864 013174  012721  000003                 MOV     #3,(R1)+        ;SET PARAM NUMBER
2865 013200  012721  010345                 MOV     #MCYLOC,(R1)+   ;INSERT NAME ADD POINTER
2866 013204  013711  003066                 MOV     HDWRD1,(R1)     ;GET HEADER WORD
2867 013210  012703  000007                 MOV     #7,R3           ;SET SHIFT COUNT
2868 013214  000241                 3$:     CLC
2869 013216  006011                         ROR     (R1)            ;ALIGN CHAR FOR PRINTING
2870 013220  005303                         DEC     R3              ;  AS IS VALUE
2871 013222  001374                         BNE     3$
2872 013224  005721                         TST     (R1)+           ;BUMP PARAM POINTER
2873 013226  013711  003116                 MOV     NEWCYL,(R1)     ;INSERT SB VALUE
2874 013232  004737  025330                 JSR     PC,RPTRES       ;REPORT RESULTS
2875 013236  004737  025536                 JSR     PC,RPTREM       ;REPORT REMAINDER
2876 013242  012603                         MOV     (SP)+,R3        ;RESTORE R3
2877 013244  012601                         MOV     (SP)+,R1        ;RESTORE R1
2878 013246  004737  015712                 JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
2879 013252                         ENDMSG
     013252                         L10007:
     013252  104423                         TRAP    C$MSG
2880
2881 013254                         BGNMSG  ERR9
2882 013254  005277  167772                 INC     @ERRPOINT       ;BUMP ERROR COUNT
2883 013260  010146                         MOV     R1,-(SP)        ;STORE R1
2884 013262  004737  024542                 JSR     PC,RPTOP        ;REPORT OPERATION
2885 013266  012721  000003                 MOV     #3,(R1)+        ;SET PARAM NUMBER
2886 013272  010321                         MOV     R3,(R1)+        ;INSERT NAME ADD POINTER
2887 013274  010421                         MOV     R4,(R1)+        ;SET IS VALUE
2888 013276  010521                         MOV     R5,(R1)+        ;SET SB VALUE
2889 013300  004737  025330                 JSR     PC,RPTRES       ;REPORT RESULTS
2890 013304  004737  025536                 JSR     PC,RPTREM       ;REPORT REMAINDER
2891 013310  012601                         MOV     (SP)+,R1        ;RESTORE R1
2892 013312  004737  015712                 JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
2893 013316                         ENDMSG
     013316                         L10010:
     013316  104423                         TRAP    C$MSG
2894 013320                         BGNMSG  ERR10
2895 013320  010146                         MOV     R1,-(SP)        ;STORE R1
2896 013322  005737  003030                 TST     MORECE          ;TEST IF 2ND BAD LINE
2897 013326  001051                         BNE     3$              ;YES - SKIP
2898 013330  005277  167716                 INC     @ERRPOINT       ;BUMP ERROR COUNT
2899 013334  004737  024542                 JSR     PC,RPTOP        ;REPORT OPERATION
2900 013340                                 PRINTB  #FMT5,#BASADD,RLBAS,#DRVNAM,<8,RLDRV+1> ;REPORT ID
     013340  005046                         CLR     -(SP)
     013342  153716  003047                 BISB    RLDRV+1,(SP)
     013346  012746  006152                 MOV     #DRVNAM,-(SP)
     013352  013746  003042                 MOV     RLBAS,-(SP)
```

ERROR MESSAGES

```
                013356  012746  006141                 MOV     @BASADD,-(SP)
                013362  012746  011026                 MOV     @FMT5, (SP)
                013366  012746  000005                 MOV     #5, (SP)
                013372  010600                         MOV     SP,R0
                013374  104414                         TRAP    C$PNTB
                013376  062706  000014                 ADD     #14,SP
        2901    013402                                 PRINTB  @FMT14,@MRSLT,@MWORD,R3,@RESE3,(R4),@RESE4,(R5)
                013402  011546                         MOV     (R5), (SP)
                013404  012746  010565                 MOV     @RESE4,-(SP)
                013410  011446                         MOV     (R4), (SP)
                013412  012746  010561                 MOV     @RESE3, (SP)
                013416  010346                         MOV     R3, (SP)
                013420  012746  006310                 MOV     @MWORD, (SP)
                013424  012746  005536                 MOV     @MRSLT,-(SP)
                013430  012746  011277                 MOV     @FMT14, (SP)
                013434  012746  000010                 MOV     #10, (SP)
                013440  010600                         MOV     SP,R0
                013442  104414                         TRAP    C$PNTB
                013444  062706  000022                 ADD     #22,SP
        2902    013450  000421                         BR      4$
        2903    013452                          3$:    PRINTB  @FMT15,@MWORD,R3,@RESE3,(R4),@RESE4,(R5) ;REPORT DATA
                013452  011546                         MOV     (R5),-(SP)
                013454  012746  010565                 MOV     @RESE4,-(SP)
                013460  011446                         MOV     (R4),-(SP)
                013462  012746  010561                 MOV     @RESE3,-(SP)
                013466  010346                         MOV     R3,-(SP)
                013470  012746  006310                 MOV     @MWORD,-(SP)
                013474  012746  011331                 MOV     @FMT15,-(SP)
                013500  012746  000007                 MOV     #7, (SP)
                013504  010600                         MOV     SP,R0
                013506  104414                         TRAP    C$PNTB
                013510  062706  000020                 ADD     #20,SP
        2904    013514  005237  003030          4$:    INC     @MORECE         ;INC COMPARE ERROR COUNT
        2905    013520  012601                         MOV     (SP)+,R1        ;RESTORE R1
        2906    013522  004737  015712                 JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
        2907    013526                          ENDMSG
                013526                          L10011:
                013526  104423                          TRAP    C$MSG
        2908    013530                          ENDMOD
        2909
        2910                                    ;LOAD PROTECTION TABLE
        2911    013530                          BGNPROT
        2912    013530  000000                          .WORD    0              ;OFFSET OF CSR IN P-TABLE
        2913    013532  177777                          .WORD   -1              ;NOT A MASS-BUS DRIVE
        2914    013534  000010                          .WORD   10              ;OFFSET OF DRIVE IN P TABLE
        2915    013536                          ENDPROT
        2916
        2917                                            .EVEN
        2918
        2919    013536                          BGNMOD  HPTCODE
        2920    013536                          BGNHW
                013536  000006                          .WORD   L10013-L$HW/2
        2921    013540  174400                          .WORD   174400          ;CSR BASE ADDRESS DEFAULT
        2922    013542  000160                          .WORD   160             ;VECTOR DEFAULT
        2923    013544  000240                          .WORD   240             ;PRIORITY DEFAULT
        2924    013546  000001                          .WORD   1               ;TYPE OF DRIVE
        2925    013550  000000                          .WORD   0               ;DRIVE NUMBER DEFAULT
```

ERROR MESSAGES

```
2926 013552  000001                    .WORD   1               ;RL11 CONTROLLER
2927 013554                      ENDHW
     013554                      L10013:
2928 013554                      ENDMOD
2929
2930 013554                      BGNMOD  SPTCODE
2931 013554                      BGNSW
     013554  000006                      .WORD   L10014-L$SW/2
2932 013556  000000              MISWIW: .WORD   0               ;BIT 0 = USE ALL CYLINDERS
2933                                                             ;BIT 1 = USE ALL SECTORS
2934                                                             ;BIT 2 = EXECUTE DRIVE SELECT TEST
2935                                                             ;BIT 3 = EXECUTE HEAD ALIGNMENT
2936                                                             ;BIT 12 = HEAD SELECT SUPPLIED FLAG
2937                                                             ;BIT 13 = HILIMIT SPECIFIED FLAG
2938                                                             ;BIT 14 = LO LIMIT SPECIFIED FLAG
2939 013560  000000              LOLIMW: .WORD   0
2940 013562  000377              HILIMW: .WORD   255.
2941 013564  000000              HEADW:  .WORD   0
2942 013566  000024              ERLIMW: .WORD   20.             ;ERROR LIMIT
2943 013570  000012              DCLIMW: .WORD   10.             ;COMPARE ERROR LIMIT
2944 013572                      ENDSW
     013572                      L10014:
2945 013572                      ENDMOD
2946
2947 013572                      BGNMOD  DSPCODE
2952 013572                      DISPATCH        9
     013572  000011                      .WORD   9
     013574  026022                      .WORD   T1
     013576  026244                      .WORD   T2
     013600  026454                      .WORD   T3
     013602  026664                      .WORD   T4
     013604  027110                      .WORD   T5
     013606  027316                      .WORD   T6
     013610  027544                      .WORD   T7
     013612  030054                      .WORD   T8
     013614  030352                      .WORD   T9
2954 013616                      ENDMOD
2955
2956                             .SBTTL  INITIALIZATION SECTION
2957
2958 013616                      BGNMOD  INITCODE
2959 013616                      BGNINIT
2960 013616                              SETVEC  #140,#170000,#340       ;ODT STARTING ADDR              ;JSD REV A
     013616  012746  000340              MOV     #340,-(SP)
     013622  012746  170000              MOV     #170000,-(SP)
     013626  012746  000140              MOV     #140,-(SP)
     013632  012746  000003              MOV     #3,-(SP)
     013636  104437                      TRAP    C$SVEC
     013640  062706  000010              ADD     #10,SP
2961                             ;CHECK FOR PRESENCE OF A P-CLOCK
2962 013644  005037  003504              CLR     CLKFLG          ;CLEAR CLOCK FLAG
2963 013650                              CLOCK   P,CLKADR        ;P-CLOCK?
     013650  012700  000120              MOV     #'P,R0
     013654  104462                      TRAP    C$CLCK
     013656  010037  003506              MOV     R0,CLKADR
2964 013662                              BNCOMPLETE      1$      ;BRANCH IF NO P-CLOCK
     013662  103002                      BCC     1$
```

INITIALIZATION SECTION

```
2965 013664  005237  003504                    INC     CLKFLG          ;INDICATE PRESENCE OF A P CLOCK
2966                                    ;1$:    SETPRI  #340            ;SET PRI TO 7 TO INHIBIT ALL INT'S      ;JSD REV A
2967 013670                             1$:     SETPRI  #300            ;SET PRI TO 6 TO INHIBIT MOST INT'S     ;JSD REV A
     013670  012700  000300                    MOV     #300,R0
     013674  104441                            TRAP    C$SPRI
2968 013676                                     BRESET                  ;FOR LSI 11 CPU'S
     013676  104433                            TRAP    C$RESET
2969 013700  042737  10001'  013556            BIC     #MITEST!DRSELT!HDALIGN,MISWIW ;CLEAR ALL MANUAL
2970                                                                    ; INTERVENTION FLAGS
2971 013706  005037  003016                    CLR     SSINDX          ;CLEAR SUBROUTINE STACK INDEX
2972 013712                                     READEF  #EF.PWR         ;POWER FAILURE
     013712  012700  000034                    MOV     #EF.PWR,R0
     013716  104447                            TRAP    C$REFG
2973 013720                                     BNCOMPLETE  4$          ;NO, GO CHECK NEW PASS
     013720  103005                            BCC     4$
2974 013722  013737  002012  003464            MOV     L$UNIT,PWRFLG   ;SET POWER FAIL FLAG
2975 013730  000137  014342                    JMP     PWCON           ;GO SERVICE POWER FAIL
2976 013734                             4$:     READEF  #EF.START       ;CHECK IF START
     013734  012700  000040                    MOV     #EF.START,R0
     013740  104447                            TRAP    C$REFG
2977 013742                                     BNCOMPLETE      RESTART ;NO   SKIP
     013742  103034                            BCC     RESTART
2978
2979                                    ;       ON START INITIALIZE TO START AT FIRST DRIVE, CLEAR INTERNAL
2980                                    ;       PASS COUNT, AND ERROR COUNT.
2981
2982 013744  013737  002012  003110            MOV     L$UNIT,DRVCNT   ;SET UP UNIT COUNT
2983 013752  005037  003454            RSTRT:  CLR     PASNUM          ;CLEAR PASS NUMBER
2984 013756  012700  003254                    MOV     #ERRCNT,R0
2985 013762  012701  000100                    MOV     #64.,R1         ;GET A COUNT
2986 013766  005020                     1$:     CLR     (R0)+           ;CLEAR AN ERROR COUNTER STORAGE AREA
2987 013770  005301                             DEC     R1
2988 013772  001375                             BNE     1$              ;LOOP TILL ALL CLEARED
2989 013774  012737  003252  003252            MOV     #ERRCNT-2,ERRPOINT ;INIT ERROR POINTER
2990 014002  012737  177777  003456            MOV     #-1,PSETNM      ;SET PARAM SELECT TO INITIAL VALUE
2991 014010  012737  177777  003024            MOV     #-1,HADONE      ;PRESET HEAD ALIGN DONE FLAG
2992 014016  032737  040000  013556    LAB:    BIT     #LOCYL,MISWIW   ;TEST IF LO LIMIT SET
2993 014024  001002                             BNE     5$              ;YES - SKIP
2994 014026  005037  013560                    CLR     LOLIMW          ;ELSE CLEAR LO LIMIT
2995 014032  000432                     5$:     BR      SETDON
2996 014034                             RESTART:
2997 014034                                     READEF  #EF.RESTART     ;CHECK IF RESTART
     014034  012700  000037                    MOV     #EF.RESTART,R0
     014040  104447                            TRAP    C$REFG
2998 014042                                     BCOMPLETE       RSTRT   ;NO - SKIP
     014042  103743                            BCS     RSTRT
2999 014044                             CONTINUE:
3000 014044                                     READEF  #EF.CONTINUE    ;TEST IF CONTINUE
     014044  012700  000036                    MOV     #EF.CONTINUE,R0
     014050  104447                            TRAP    C$REFG
3001 014052                                     BCOMPLETE       PWCON
     014052  103533                            BCS     PWCON
3002                                    ;       ON CONTINUE PICK UP UNIT LAST UNDER TEST
3003 014054                                     READEF  #EF.NEW         ;CHECK IF STARTING NEW PASS
     014054  012700  000035                    MOV     #EF.NEW,R0
     014060  104447                            TRAP    C$REFG
3004 014062                                     BCOMPLETE       PASNEW
```

INITIALIZATION SECTION

```
                014062  103403                          BCS     PASNEW
          3005  014064                          NXTPAS:
          3006  014064  005737  003110                  TST     DRVCNT          ;TEST IF ALL UNITS CHECKED
          3007  014070  001013                          BNE     SETDON          ;NO - SKIP
          3008  014072  005237  003454          PASNEW: INC     PASNUM          ;ELSE BUMP PASS COUNT
          3009  014076  012737  003252  003252          MOV     @ERRCNT-2,ERRPOINT ;INIT ERROR POINTER
          3010  014104  013737  002012  003110          MOV     L$UNIT,DRVCNT   ;GET ALL DRIVES
          3011  014112  012737  177777  003456          MOV     @-1,PSETNM      ;SET PARAM SELECT TO INITIAL
          3012  014120  005237  003456          SETDON: INC     PSETNM          ;NEXT SET OF PARAMETERS
          3013  014124  005337  003110                  DEC     DRVCNT          ;DOWN COUNT DRIVE TOTAL
          3014  014130  062737  000002  003252          ADD     @2,ERRPOINT     ;UPDATE THE ERROR POINTER
          3015  014136  013700  003456                  MOV     PSETNM,R0       ;SET UP TO GET PARAMETERS
          3016  014142  012702  003042                  MOV     @RLBAS,R2
          3017  014146                                  GPHARD  R0,R1
                014146  104442                          TRAP    C$GPHRD
                014150  010001                          MOV     R0,R1
          3018  014152                                  BCOMPLETE       7$      ;SKIP IF GOOD PARAM
                014152  103406                          BCS     7$
          3019  014154  005737  003464                  TST     PWRFLG          ;RECENT POWER FAILURE
          3020  014160  001741                          BEQ     NXTPAS  ;NO
          3021  014162  005337  003464                  DEC     PWRFLG          ;ACCOUNT FOR DRIVE
          3022  014166  000736                          BR      NXTPAS
          3023  014170  012122                  7$:     MOV     (R1)+,(R2)+     ;STORE PARAMETERS CSR
          3024  014172  012122                          MOV     (R1)+,(R2)+     ;                   VECTOR
          3025  014174  005721                          TST     (R1)+           ;BUMP PAST PRIORITY
          3026  014176  012137  002312                  MOV     (R1)+,T.DRIVE
          3027  014202  012122                          MOV     (R1)+,(R2)+
          3028  014204  022737  000001  002312          CMP     #1,T.DRIVE
          3029  014212  001426                          BEQ     65$
          3030  014214  012737  000776  002322          MOV     #510.,NXTHL
          3031  014222  012737  000777  002316          MOV     #511.,HLMTW
          3032  014230  012737  001000  002324          MOV     #512.,GBND
          3033  014236  012737  177600  002326          MOV     #177600,CAMSK
          3034  014244  012737  177600  002330          MOV     #177600,DIRMSK
          3035  014252  012737  177600  002332          MOV     #177600,HDCYL
          3036  014260  012737  177000  002320          MOV     #177000,CLRBYT
          3037  014266  000425                          BR      PWCON
          3038
          3039  014270  012737  000377  002316  65$:    MOV     #255.,HLMTW
          3040  014276  012737  000400  002324          MOV     #256.,GBND
          3041  014304  012737  077600  002326          MOV     #77600,CAMSK
          3042  014312  012737  077600  002330          MOV     #77600,DIRMSK
          3043  014320  012737  077600  002332          MOV     #77600,HDCYL
          3044  014326  012737  000376  002322          MOV     #254.,NXTHL
          3045  014334  012737  177400  002320          MOV     #177400,CLRBYT
          3046
          3047  014342  032737  020000  013556  PWCON:  BIT     #HICYL,MISWIW
          3048  014350  001003                          BNE     1$
          3049  014352  013737  002316  013562          MOV     HLMTW,HILIMW
          3050  014360                          1$:     SETVEC  RLVEC,#INTHLR,#340      ;SET UP VECTOR
                014360  012746  000340                  MOV     #340,-(SP)
                014364  012746  015632                  MOV     #INTHLR,-(SP)
                014370  013746  003044                  MOV     RLVEC,-(SP)
                014374  012746  000003                  MOV     #3,-(SP)
                014400  104437                          TRAP    C$SVEC
                014402  062706  000010                  ADD     #10,SP
          3051  014406                                  SETPRI  #0              ;SET PRIORITY
```

```
INITIALIZATION SECTION

     014406  012700  000000                    MOV     #0,R0
     014412  104441                            TRAP    C$SPRI
3052 014414  013702  003042                    MOV     RLBAS,R2        ;SET RL11 BASE ADDRESS POINTER
3063                                 ;CHECK IF POWER FAILURE WAIT IS NEEDED
3064
3065 014420  005737  003464                    TST     PWRFLG          ;NEEDED???
3066 014424  001472                            BEQ     8$              ;NO, SKIP
3067
3068 014426  013705  003046                    MOV     RLDRV,R5        ;DRIVE SELECT
3069 014432  052705  000200                    BIS     #CRDYMSK,R5     ;SET CRDY
3070 014436  010562  000000                    MOV     R5,RLCS(R2)     ;SELECT DRIVE
3071 014442  012701  000170                    MOV     #120.,R1        ;INITIALIZE WAIT COUNT
3072 014446  032762  000001  000000  9$:      BIT     #DRDYMSK,RLCS(R2)       ;DRIVE UP YET?
3073 014454  001056                            BNE     8$              ;YES START TEST
3074
3075 014456                                    WAITMS  #10.            ;WAIT A SECOND
3076 014470  005301                            DEC     R1              ;SIXTY GONE BY
3077 014472  001365                            BNE     9$              ;NO
3078 014474                                    PRINTF  #FMT24,#NOPWR
     014474  012746  006176                    MOV     #NOPWR,-(SP)
     014500  012746  011632                    MOV     #FMT24,-(SP)
     014504  012746  000002                    MOV     #2,-(SP)
     014510  010600                            MOV     SP,R0
     014512  104417                            TRAP    C$PNTF
     014514  062706  000006                    ADD     #6,SP
3079 014520                                    PRINTF  #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
     014520  005046                            CLR     -(SP)
     014522  153716  003047                    BISB    RLDRV+1,(SP)
     014526  012746  006152                    MOV     #DRVNAM,-(SP)
     014532  013746  003042                    MOV     RLBAS,-(SP)
     014536  012746  006141                    MOV     #BASADD,-(SP)
     014542  012746  011026                    MOV     #FMT5,-(SP)
     014546  012746  000005                    MOV     #5,-(SP)
     014552  010600                            MOV     SP,R0
     014554  104417                            TRAP    C$PNTF
     014556  062706  000014                    ADD     #14,SP
3080 014562                                    PRINTF  #FMT3
     014562  012746  011012                    MOV     #FMT3,-(SP)
     014566  012746  000001                    MOV     #1,-(SP)
     014572  010600                            MOV     SP,R0
     014574  104417                            TRAP    C$PNTF
     0145 ?  062706  000004                    ADD     #4,SP
3081 014602                                    DODU    PSETNM          ;DROP DRIVE
     014602  013700  003456                    MOV     PSETNM,R0
     014606  104451                            TRAP    C$DODU
3082 014610                                    DOCLN
     014610  104444                            TRAP    C$DCLN
3083 014612                         8$:
3084
3085 014612                         ENDINIT
     014612                         L10015:
     014612  104411                            TRAP    C$INIT
3086 014614                         ENDMOD
3087
3088                                .SBTTL  AUTO DROP SECTION
3089
3090                                ;THE AUTO DROP SECTION IS INVOKED BY THE DIAGNOSTIC SUPERVISOR WHENEVER THE
```

AUTO DROP SECTION

```
3091                                    ;"ADR" FLAG IS SET BY THE OPERATOR. IT IS EXECUTED AFTER THE INITIALIZATION
3092                                    ;CODE AND CHECKS THE DRIVE TO DETERMINE IF IT IS READY TO RECEIVE A COMMAND.
3093                                    ;IF THE DRIVE IS NOT READY IT IS DROPPED FROM THE TEST CYCLE AND THE NEXT
3094                                    ;DRIVE IS ACCESSED. IF THE DRIVE IS READY THE HARDWARE TESTS ARE PERFORMED
3095                                    ;AFTER WHICH THE NEXT DRIVE IS ACCESSED.
3096
3097 014614                             BGNAUTO
3098 014614  005037  003462                    CLR     TRPFLG          ;CLEAR TRAP FLAG
3099 014620                                    SETVEC  ERRVEC,#TRPHAN,#340     ;SET UP TRAP VECTOR TO DETECT
     014620  012746  000340                    MOV     #340,-(SP)
     014624  012746  015624                    MOV     #TRPHAN, (SP)
     014630  013746  003244                    MOV     ERRVEC,-(SP)
     014634  012746  000003                    MOV     #3,-(SP)
     014640  104437                            TRAP    C$SVEC
     014642  062706  000010                    ADD     #10,SP
3100                                                                    ;/NON-EXISTENT CONTROLLER
3101 014646  013702  003042                    MOV     RLBAS,R2        ;GET RL11 BASE ADDRESS
3102 014652  005762  000000                    TST     RLCS(R2)        ;ACCESS DRIVE CONTROLLER ADDRESS
3103 014656  005737  003462                    TST     TRPFLG          ;DID TRAP OCCUR?
3104 014662  001447                            BEQ     1$              ;BRANCH TO CHECK DRIVE IF TRAP DID NOT OCCUR
3105 014664                                    PRINTF  #FMT24,#NOCTLR  ;ELSE, PRINT MSG. "DRIVE DROPPED   NO CONTROLLER
     014664  012746  007645                    MOV     #NOCTLR,-(SP)
     014670  012746  011632                    MOV     #FMT24,-(SP)
     014674  012746  000002                    MOV     #2,-(SP)
     014700  010600                            MOV     SP,R0
     014702  104417                            TRAP    C$PNTF
     014704  062706  000006                    ADD     #6,SP
3106 014710                                    PRINTF  #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
     014710  005046                            CLR     -(SP)
     014712  153716  003047                    BISB    RLDRV+1,(SP)
     014716  012746  006152                    MOV     #DRVNAM,-(SP)
     014722  013746  003042                    MOV     RLBAS,-(SP)
     014726  012746  006141                    MOV     #BASADD,-(SP)
     014732  012746  011026                    MOV     #FMT5, (SP)
     014736  012746  000005                    MOV     #5,-(SP)
     014742  010600                            MOV     SP,R0
     014744  104417                            TRAP    C$PNTF
     014746  062706  000014                    ADD     #14,SP
3107                                                                    ;PRINT DRIVE INFORMATION
3108 014752                                    PRINTF  #FMT3
     014752  012746  011012                    MOV     #FMT3,-(SP)
     014756  012746  000001                    MOV     #1,-(SP)
     014762  010600                            MOV     SP,R0
     014764  104417                            TRAP    C$PNTF
     014766  062706  000004                    ADD     #4,SP
3109
3110 014772                                    DODU    PSETNM          ;DO DROP UNIT ON DRIVE
     014772  013700  003456                    MOV     PSETNM,R0
     014776  104451                            TRAP    C$DODU
3111 015000  000460                            BR      2$              ;BRANCH TO EXIT
3112 015002  013705  003046            1$:     MOV     RLDRV,R5        ;ELSE, GET DRIVE NUMBER
3113 015006  052705  000200                    BIS     #CRDYMSK,R5     ;SET CONTROLLER READY
3114 015012  010562  000000                    MOV     R5,RLCS(R2)     ;LOAD IN THE DRIVE NUMBER
3115 015016  032762  000001  000000            BIT     #DRDYMSK,RLCS(R2)       ;IS DRIVE READY?
3116 015024  001046                            BNE     2$              ;BRANCH TO PERFORM TESTS IF DRIVE IS READY
3117 015026                                    PRINTF  #FMT24,#NOTRDY  ;PRINT MSG. "DRIVE DROPPED - DID NOT RESPOND
     015026  012746  007703                    MOV     #NOTRDY,-(SP)
```

AUTO DROP SECTION

```
        015032  012746  011632                 MOV     #FMT24, (SP)
        015036  012746  000002                 MOV     #2, (SP)
        015042  010600                         MOV     SP,R0
        015044  104417                         TRAP    C$PNTF
        015046  062706  000006                 ADD     #6,SP
3118                                                            ;/WITH 'READY'"
3119 015052                                    PRINTF  #FMT5,#BASADD,RLBAS,#DRVNAM,<8,RLDRV+1>
        015052  005046                         CLR     -(SP)
        015054  153716  003047                 BISB    RLDRV+1,(SP)
        015060  012746  006152                 MOV     #DRVNAM,-(SP)
        015064  013746  003042                 MOV     RLBAS,-(SP)
        015070  012746  006141                 MOV     #BASADD,-(SP)
        015074  012746  011026                 MOV     #FMT5,-(SP)
        015100  012746  000005                 MOV     #5,-(SP)
        015104  010600                         MOV     SP,R0
        015106  104417                         TRAP    C$PNTF
        015110  062706  000014                 ADD     #14,SP
3120                                                            ;PRINT DRIVE INFORMATION
3121 015114                                    PRINTF  #FMT3
        015114  012746  011012                 MOV     #FMT3,-(SP)
        015120  012746  000001                 MOV     #1,-(SP)
        015124  010600                         MOV     SP,R0
        015126  104417                         TRAP    C$PNTF
        015130  062706  000004                 ADD     #4,SP
3122 015134                                    DODU    PSETNM          ;DO DROP UNIT ON DRIVE
        015134  013700  003456                 MOV     PSETNM,R0
        015140  104451                         TRAP    C$DODU
3123 015142                            2$:     CLRVEC  ERRVEC          ;RELEASE ERROR VECTOR
        015142  013700  003244                 MOV     ERRVEC,R0
        015146  104436                         TRAP    C$CVEC
3124 015150                            ENDAUTO
        015150                         L10016:
        015150  104461                         TRAP    C$AUTO
3125
3126                                   .SBTTL  CLEANUP CODE SECTION
3127
3128 015152                            BGNMOD  CLNCODE
3129 015152                            BGNCLN
3130
3131 015152                                    SETVEC  ERRVEC,#TRPHAN,#340
        015152  012746  000340                 MOV     #340,-(SP)
        015156  012746  015624                 MOV     #TRPHAN,-(SP)
        015162  013746  003244                 MOV     ERRVEC,-(SP)
        015166  012746  000003                 MOV     #3,-(SP)
        015172  104437                         TRAP    C$SVEC
        015174  062706  000010                 ADD     #10,SP
3132
3133                                   ;       SETPRI  #7              ;SET PRIORITY TO 7              ;JSD REV A
3134 015200                                    SETPRI  #PRIO6          ;SET PRIORITY TO 6              ;JSD REV A
        015200  012700  000300                 MOV     #PRIO6,R0
        015204  104441                         TRAP    C$SPRI
3135 015206  032762  000200  000000    2$:     BIT     #CRDYMSK,RLCS(R2)       ;TEST IF CONTROLLER READY
3136 015214  001407                            BEQ     3$              ;NO LOOP UNTIL READY
3137 015216  053762  003046  000000            BIS     RLDRV,RLCS(R2)  ;SET DRIVE NUMBER
3138 015224  032762  000001  000000            BIT     #DRDYMSK,RLCS(R2)       ;TEST IF DRIVE BUSY
3139 015232  001005                            BNE     5$              ;NO - SKIP
3140 015234                            3$:     WAITMS  #3              ;WAIT 300 MS
```

CLEANUP CODE SECTION

```
3141 015246                                5$:     CLRVEC  RLVEC           ;RELEASE VEC
     015246  013700  003044                        MOV     RLVEC,R0
     015252  104436                          TRAP    C$CVEC
3142 015254  005737  003464                        TST     PWRFLG          ;PWR FAIL SFT
3143 015260  001402                        BEQ     7$              ;NO
3144 015262  005337  003464                        DEC     PWRFLG
3145 015266                                7$:     CLRVEC  ERRVEC
     015266  013700  003244                        MOV     ERRVEC,R0
     015272  104436                          TRAP    C$CVEC
                                           BRESET                  ;TAKE CARE OF LSI 11
3146 015274
     015274  104433                        TRAP    C$RESET
3147
3148 015276                                ENDCLN
     015276                                L10017:
     015276  104412                        TRAP    C$CLEAN
3149
3150 015300                                BGNDU
3151 015300  000240                        NOP
3152 015302                                ENDDU
     015302                                L10020:
     015302  104453                        TRAP    C$DU
3153
3154 015304                                ENDMOD
3155
3156                                       .SBTTL  GLOBAL SUBROUTINES
3157
3158 015304                                BGNMOD  GLBSUB
3159
3160 015304  012737  000160  002116  TIME:   MOV     #160,L$DLY              ;GET OUTER DELAY LOOP
3161 015312  005237  003476                  INC     TIM.US                  ;US-WAIT ROUTINE INDICATOR
3162 015316  013737  003466  003472          MOV     XDELAY,MININC           ;SAVE ORIGINAL US-WAIT
3163 015324  005437  003466                  NEG     XDELAY                  ;GET NEGATIVE OF FACTOR
3164 015330                                   READBUS                         ;Q - BUS?
     015330  104407                          TRAP    C$RDBU
3165 015332                                   BCOMPLETE 2$                    ;BRANCH   IF YES
     015332  103420                          BCS     2$
3166                                  ;1$:     DELAY   #1.                     ;WAIT                           ;JSD REV A
3167 015334                           ;3$:     DELAY   1.                      ;WAIT                           ;JSD REV A
     015334  012727  000001                  MOV     #1.,(PC)+
     015340  000000                          .WORD   0
     015342  013727  002116                  MOV     L$DLY,(PC)+
     015346  000000                          .WORD   0
     015350  005367  177772                  DEC     -6(PC)
     015354  001375                          BNE     .-4
     015356  005367  177756                  DEC     -22(PC)
     015362  001367                          BNE     .-20
3168 015364  005237  003466                  INC     XDELAY                  ;WAIT FACTOR EXPIRED?
3169 015370  002761                          BLT     1$                      ;BRANCH - IF NO
3170 015372  000422                          BR      4$                      ;GET TIME
3171 015374  012737  000065  002116  2$:     MOV     #65,L$DLY               ;GET OUTER DELAY LOOP
3172                                  ;3$:     DELAY   #1.                     ;WAIT WITH RESPECT TO FONZ BUS  ;JSD REV A
3173 015402                           3$:      DELAY   1.                      ;WAIT WITH RESPECT TO FONZ BUS  ;JSD REV A
     015402  012727  000001                  MOV     #1.,(PC)+
     015406  000000                          .WORD   0
     015410  013727  002116                  MOV     L$DLY,(PC)+
     015414  000000                          .WORD   0
     015416  005367  177772                  DEC     -6(PC)
```

GLOBAL SUBROUTINES

```
                015422  001375                          BNE     .-4
                015424  005367  177756                  DEC     -22(PC)
                015430  001367                          BNE     .-20
3174            015432  005237  003466                  INC     XDELAY          ;WAIT FACTOR EXPIRED?
3175            015436  002761                          BLT     3$              ;BRANCH - IF NO
3176            015440  063737  003472  003132  4$:     ADD     MININC,TEMPO    ;GET TIME EXPIRED
3177            015446  000207                          RTS     PC              ;RETURN
3178
3179            015450  012737  000160  002116  XTIME:  MOV     #160,L$DLY      ;GET OUTER DELAY LOOP
3180            015456  005037  003476                  CLR     TIM.US          ;MS. WAIT INDICATOR
3181            015462  013737  003470  003502          MOV     YDELAY,MAJINC   ;SAVE ORIGINAL WAIT MS
3182            015470  006337  003470                  ASL     YDELAY          ;MULTIPLY BY FACTOR 4
3183            015474  006337  003470                  ASL     YDELAY          ; --  ----
3184            015500  005437  003470                  NEG     YDELAY          ;GET NEGATIVE OF RESULT
3185            015504                                  READBUS                 ;Q - BUS?
                015504  104407                          TRAP    C$RDBU
                                                        BNCOMPLETE      1$
3186            015506                                  BCC     1$              ;BRANCH   IF NO
                015506  103023
3187            015510  012737  000150  002116          MOV     #150,L$DLY      ;GET OUTER DELAY LOOP
3188                                            ;2$:    DELAY   #20             ;WAIT WITH RESPECT TO FONZ BUS  ;JSD REV A
3189            015516                          2$:     DELAY   20              ;WAIT WITH RESPECT TO FONZ BUS  ;JSD REV A
                015516  012727  000020                  MOV     #20,(PC)+
                015522  000000                          .WORD   0
                015524  013727  002116                  MOV     L$DLY,(PC)+
                015530  000000                          .WORD   0
                015532  005367  177772                  DEC     -6(PC)
                015536  001375                          BNE     .-4
                015540  005367  177756                  DEC     -22(PC)
                015544  001367                          BNE     .-20
3190            015546  005237  003470                  INC     YDELAY          ;WAIT FACTOR EXPIRED
3191            015552  002761                          BLT     2$              ;BRANCH - IF NO
3192            015554  000417                          BR      3$              ;GET TIME
3193                                            ;1$:    DELAY   #10             ;WAIT                           ;JSD REV A
3194            015556                          1$:     DELAY   10              ;WAIT                           ;JSD REV A
                015556  012727  000010                  MOV     #10,(PC)+
                015562  000000                          .WORD   0
                015564  013727  002116                  MOV     L$DLY,(PC)+
                015570  000000                          .WORD   0
                015572  005367  177772                  DEC     -6(PC)
                015576  001375                          BNE     .-4
                015600  005367  177756                  DEC     -22(PC)
                015604  001367                          BNE     .-20
3195            015606  005237  003470                  INC     YDELAY                  ;WAIT FACTOR EXPIRED?
3196            015612  002761                          BLT     1$                      ;BRANCH - IF NO
3197            015614  063737  003502  003474  3$:     ADD     MAJINC,TEMP             ;GET EXPIRED TIME
3198            015622  000207                          RTS     PC                      ;RETURN
3199
3200            015624                          BGNSRV
3201
3202                                            ;TRAP HANDLER. INDICATES OCCURRENCE OF A TRAP.
3203
3204            015624  005237  003462          TRPHAN: INC     TRPFLG
3205
3206            015630                          ENDSRV
                015630                          L10021:
                015630  000002                          RTI
3207
```

GLOBAL SUBROUTINES

```
3208 015632                                BGNSRV
3209
3210                                       ;INTERRUPT HANDLER. ABORTS WAIT TIMER AND STORES RL11 REGISTERS.
3211
3212 015632                                INTMER:
3213
3214 015632  012237  003060                        MOV     (R2)+,T.CS      ;STORE RL REGISTERS
3215 015636  012237  003062                        MOV     (R2)+,T.BA
3216 015642  012237  003064                        MOV     (R2)+,T.DA
3217 015646  011237  003066                        MOV     (R2),T.MP
3218 015652  012737  177777  003022                MOV     #-1,DONE        ;SET DONE FLAG
3219 015660  013702  003042                        MOV     RLBAS,R2        ;RESTORE R2
3220 015664                                        ABORTWAIT
3221
3222 015710                                ENDSRV
     015710                                L10022:
     015710  000002                                RTI
3223
3224                                       ;       ERROR LIMIT CHECKING ROUTINE
3225                                       ;       DROPS DRIVE IF ERROR LIMIT EXCEEDED
3226
3227 015712  027737  165334  013566       CKERLM: CMP     @ERRPOINT,ERLIMW        ;TEST IF ERROR LIMIT EXCEEDED
3228 015720  002453                                BLT     1$              ;NO   SKIP
3229 015722                                        INLOOP                  ;CHECK IF IN ERROR LOOP
     015722  104420                                TRAP    C$INLP
3230 015724                                        BCOMPLETE       1$      ;YES  SKIP
     015724  103451                                BCS     1$
3231 015726                                        PRINTF  @FMT25,ERLIMW,@MEXERS
     015726  012746  010513                        MOV     @MEXERS,-(SP)
     015732  013746  013566                        MOV     ERLIMW,(SP)
     015736  012746  011637                        MOV     @FMT25,(SP)
     015742  012746  000003                        MOV     #3,(SP)
     015746  010600                                MOV     SP,R0
     015750  104417                                TRAP    C$PNTF
     015752  062706  000010                        ADD     #10,SP
3232 015756                                        PRINTF  @FMT5,@BASADD,RLBAS,@DRVNAM,<B,RLDRV+1>
     015756  005046                                CLR     -(SP)
     015760  153716  003047                        BISB    RLDRV+1,(SP)
     015764  012746  006152                        MOV     @DRVNAM,-(SP
     015770  013746  003042                        MOV     RLBAS,-(SP)
     015774  012746  006141                        MOV     @BASADD,-(SP)
     016000  012746  011026                        MOV     @FMT5,-(SP)
     016004  012746  000005                        MOV     #5,-(SP)
     016010  010600                                MOV     SP,R0
     016012  104417                                TRAP    C$PNTF
     016014  062706  000014                        ADD     #14,SP
3233 016020                                        PRINTF  @FMT3
     016020  012746  011012                        MOV     @FMT3,-(SP)
     016024  012746  000001                        MOV     #1,-(SP)
     016030  010600                                MOV     SP,R0
     016032  104417                                TRAP    C$PNTF
     016034  062706  000004                        ADD     #4,SP
3234 016040                                        DODU    PSETNM          ;DROP DRIVE
     016040  013700  003456                        MOV     PSETNM,R0
     016044  104451                                TRAP    C$DODU
3235 016046                                        DOCLN                   ;GO TO CLEAN UP
     016046  104444                                TRAP    C$DCLN
```

GLOBAL SUBROUTINES

```
3236 016050  000207                    1$:     RTS     PC
3237
3238                                    ;       READ AND STORE ALL RL11 REGISTERS
3239 016052  016237  000000  003060    READRL: MOV     RLCSR(R2),T.CS  ;GET CS REG
3240 016060  016237  000002  003062            MOV     RLBA(R2),T.BA   ;GET BUS ADDRESS REG
3241 016066  016237  000004  003064            MOV     RLDA(R2),T.DA   ;GET DISK ADDRESS
3242 016074  016237  000006  003066            MOV     RLMP(R2),T.MP   ;GET MULTI-PURPOSE REG
3243 016102  000207                            RTS     PC              ;RETURN
3244
3245                                    ;       WAIT FOR CONTROLLER TIMEOUT TO FORCE INTERRUPT ROUTINE
3246 016104  011646                    WAITIN: MOV     (SP), (SP)      ;MAKE ROOM FOR ERROR POINTER
3247 016106  005066  000002                    CLR     2(SP)           ;CLEAR FOR POINTER
3248 016112  032762  000200  000000            BIT     #CRDYMSK,RLCSR(R2)      ;TEST IF CONTROLLER READY
3249 016120  001420                            BEQ     4$              ;NO - SKIP TO WAIT
3250 016122  004737  016052                    JSR     PC,READRL       ;READ ALL RL REGS
3251 016126  005737  003022                    TST     DONE            ;TEST IF INTERRUPT OCCURRED
3252 016132  001435                            BEQ     5$              ;NO - GO SET NO INTERRUPT ERR FLAG
3253 016134  012766  006316  000002    1$:     MOV     #MTOSLOW,2(SP)  ;ELSE SET TOO SLOW ERROR POINTER
3254 016142  032737  002000  003060            BIT     #OPIERR,T.CS    ;TEST IF OPI SET
3255 016150  001403                            BEQ     2$              ;NO - SKIP
3256 016152  012766  006336  000002            MOV     #MDRRES,2(SP)   ;SET MESSAGE FOR NO DRIVE RESPONSE
3257 016160  000207                    2$:     RTS     PC              ;RETURN
3258 016162                            4$:     WAITMS  #3              ;WAIT 300 MS FOR TIMEOUT
3259 016174  032762  000200  000000            BIT     #CRDYMSK,RLCS(R2)       ;TEST IF READY NOW SET
3260 016202  001006                            BNE     3$              ;YES - SKIP
3261 016204  004737  016052                    JSR     PC,READRL       ;READ RL REGS
3262 016210  012766  006407  000002            MOV     #MCONHNG,2(SP)  ;SET MESSAGE FOR CONTROLLER HUNG
3263 016216  000760                            BR      2$              ;SKIP
3264 016220  005737  003022            3$:     TST     DONE            ;ELSE CHECK IF INTERRUPT OCCURRED
3265 016224  001343                            BNE     1$              ;YES - SKIP TO SET TOO SLOW
3266 016226  004737  016052            5$:     JSR     PC,READRL       ;READ RL REGS
3267 016232  012766  006354  000002            MOV     #MNOINT,2(SP)   ;ELSE SET NO INTERRUPT FLAG
3268 016240  000747                            BR      2$              ;GO TO RETURN
3269
3270                                    ;       OPERATION AND TEST INITIALIZE ROUTINE
3271 016242  005037  003020            TSTINT: CLR     OPFLAG          ;CLEAR OPERATION FLAGS
3272 016246  105037  003461                    CLRB    NOERCT          ;RESET INHIBIT ERROR COUNTING
3273 016252  005037  003030                    CLR     MORECE          ;RESET MORE COMPARE ERRORS
3274 016256  000207                            RTS     PC
3275
3276                                    ;       GET STATUS AND GET STATUS WITH RESET ROUTINE
3277 016260  013746  003142            GSTATR: MOV     TEMP4,-(SP)     ;STORE TEMP4
3278 016264  012737  000013  003142            MOV     #GETSTAT!DRSET,TEMP4    ;SET FOR RESET
3279 016272  000412                            BR      GSTATG
3280 016274  013746  003142            GSTATC: MOV     TEMP4,-(SP)     ;STORE TEMP4
3281 016300  012737  000003  003142            MOV     #GETSTAT,TEMP4  ;SET FOR NO RESET
3282 016306  000404                            BR      GSTATG
3283 016310  013746  003142            GSTAT:  MOV     TEMP4,-(SP)     ;STORE TEMP4
3284 016314  005037  003142                    CLR     TEMP4           ;SET FOR SAVE L. AND T. REGS
3285 016320  010346                    GSTATG: MOV     R3,-(SP)        ;STORE R3
3286 016322  013703  003016                    MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
3287 016326  005723                            TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
3288 016330  016663  000004  002420            MOV     4(SP),SUBSTK(R3) ;INSERT THIS CALL
3289 016336  162763  000004  002420            SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
3290 016344  010337  003016                    MOV     R3,SSINDX       ;STORE IT BACK
3291 016350  010046                            MOV     R0,-(SP)        ;STORE R0
3292 016352  010146                            MOV     R1,-(SP)        ;STORE R1
```

GLOBAL SUBROUTINES

```
3293 016354  012737  000002  003032          MOV     #2,ERRSWI          ;SET FOR NO ERROR RETURN
3294 016362  032737  000010  003142          BIT     #DRSET,TEMP4       ;TEST IF DRIVE RESET
3295 016370  001460                           BEQ     11$                ;NO - SKIP
3296 016372  032762  040000  000000          BIT     #DRVERR,RLCS(R2)   ;TEST IF DRIVE ERROR SET
3297 016400  001405                           BEQ     49$                ;NO - SKIP
3298 016402                                   WAITMS  #3                 ;WAIT FOR 300 MS FOR DRIVE TO SETTLE
3299 016414  012701  000062          49$:     MOV     #50.,R1            ;INITIALIZE WAIT COUNT
3300 016420  004737  016310          50$:     JSR     PC,GSTAT           ;GET DRIVE STATUS
3301 016424  017110                           3$
3302 016426  032737  000001  003060          BIT     #DRDYMSK,T.CS      ;TEST IF DRIVE READY
3303 016434  001054                           BNE     5$                 ;YES - GO DO CLEAR
3304 016436  032737  000020  003066          BIT     #HOSTAT,T.MP       ;ELSE TEST IF HEADS OUT
3305 016444  001010                           BNE     51$                ;YES - BYPASS RELOAD WAIT FLAG SETTING
3306 016446  032737  144000  003066          BIT     #SPDSTAT!HCESTAT!WDESTAT,T.MP ;TEST IF DRIVE HAS ERROR
3307                                                                              ;THAT CAUSED HEADS TO
3308                                                                              ;UNLOAD
3309 016454  001444                           BEQ     5$                 ;NO - SKIP
3310 016456  052737  040000  003020          BIS     #RELDWT,OPFLAG     ;ELSE SET WAIT FLAG
3311 016464  000440                           BR      5$                 ;SKIP TO CLEAR
3312 016466  032737  040000  003060  51$:     BIT     #DRVERR,T.CS       ;TEST IF DRIVE ERROR NOW
3313 016474  001034                           BNE     5$                 ;YES - SKIP TO CLEAR
3314 016476                                   WAITMS  #1                 ;WAIT FOR DRIVE TO GET ERROR, RDY, OR HEADS OUT
3315 016510  005301                           DEC     R1                 ;DEC WAIT COUNTER
3316 016512  001342                           BNE     50$                ;IF NOT DONE, LOOP
3317 016514  012703  010375                   MOV     #MUNDEF,R3         ;MESSAGE FOR UNDEFINED STATE
3318 016520                                   ERRHRD  10001.,,ERR1
     016520  104456                           TRAP    C$ERHRD
     016522  023421                           .WORD   10001
     016524  000000                           .WORD   0
     016526  011724                           .WORD   ERR1
3319 016530  000565                           BR      14$                ;EXIT
3320 016532  005737  003142          11$:     TST     TEMP4              ;TEST IF SAVE REGISTERS
3321 016536  001013                           BNE     5$                 ;NO SKIP
3322 016540  012701  000004                   MOV     #4,R1              ;SET SAVE COUNT
3323 016544  012703  003060                   MOV     #L.MP+2,R3         ;SET ADDRESS OF FIRST SAVE
3324 016550  014346          8$:      MOV     -(R3),-(SP)        ;PUT REG ON STACK
3325 016552  005301                           DEC     R1                 ;DEC COUNT
3326 016554  001375                           BNE     8$                 ;LOOP UNTIL ALL SAVED
3327 016556  012737  000003  003054          MOV     #GETSTAT,L.DA      ;SET FOR GET STATUS
3328 016564  000403                           BR      6$                 ;SKIP
3329 016566  013737  003142  003054  5$:      MOV     TEMP4,L.DA         ;INSERT PRESET FOR STATUS
3330 016574                           6$:
3331 016574  005037  003022                   CLR     DONE               ;CLEAR INTERRUPT FLAG
3332 016600  013737  003046  003050          MOV     RLDRV,L.CS         ;SET UP TO GET STATUS
3333 016606  042737  002000  003050          BIC     #BIT10,L.CS        ;CLEAR FOR DRIVE 4 - 7 SPEC'D
3334 016614  052737  000104  003050          BIS     #GTSTAT,L.CS
3335 016622  013762  003054  000004          MOV     L.DA,RLDA(R2)      ;LOAD RL REGS
3336 016630  013762  003050  000000          MOV     L.CS,RLCSR(R2)     ;LOAD CS REG
3337 016636                                   WAITUS  #1                 ;WAIT 100 US FOR INTERRUPT
3338 016650  005737  003022                   TST     DONE               ;CHECK IF INTERRUPT OCCURRED
3339 016654  001504                           BEQ     1$                 ;NO - SKIP
3340 016656  013737  003066  003074  4$:      MOV     T.MP,T.STAT        ;STORE MP REGISTER
3341 016664  042737  177770  003074          BIC     #^C<STAMSK>,T.STAT ;CLEAR ALL BUT STATE
3342 016672  032737  000010  003054          BIT     #DRSET,L.DA        ;TEST IF RESET WAS SPECIFIED
3343 016700  001503                           BEQ     3$                 ;NO - SKIP TO EXIT
3344 016702  032737  040000  003020          BIT     #RELDWT,OPFLAG     ;TEST IF RELOAD WAIT FLAG SET
3345 016710  001427                           BEQ     12$                ;NO   SKIP
```

GLOBAL SUBROUTINES

```
3346 016712 012701 001130                MOV     #600.,R1          ;SET WAIT COUNT FOR 60 SECONDS
3347 016716 032762 000001 000000 13$:    BIT     #DRDYMSK,RLCS(R2) ;TEST IF DRIVE NOW READY
3348 016724 001021                       BNE     12$               ;YES - SKIP
3349 016726                              WAITMS  #1                ;CALL WAIT
3350 016740 005301                       DEC     R1                ;DEC COUNT
3351 016742 001365                       BNE     13$               ;LOOP IF NOT 0
3352 016744 004737 016310                JSR     PC,GSTAT          ;GET DRIVE STATUS
3353 016750 017110                       3$                        ;ERROR RETURN
3354 016752 012703 010442                MOV     #MRLFAL,R3        ;SET RESULT MESSAGE POINTER
3355 016756                              ERRHRD  10003.,,ERR1
     016756 104456                       TRAP    C$ERHRD
     016760 023423                       .WORD   10003
     016762 000000                       .WORD   0
     016764 011724                       .WORD   ERR1
3356 016766 000446                       BR      14$               ;GO TO EXIT
3357 016770                      12$:    WAITUS  #10.              ;WAIT FOR 1MS
3358 017002 004737 016310                JSR     PC,GSTAT          ;GET DRIVE STATUS
3359 017006 017110                       3$
3360 017010 032737 100000 003060         BIT     #ANYERR,T.CS      ;TEST IF ANY ERROR
3361 017016 001434                       BEQ     3$                ;NO - SKIP
3362 017020 032737 001000 003066         BIT     #VCSTAT,T.MP      ;CHECK IF VOLUME CHECK RESET
3363 017026 001403                       BEQ     7$                ;YES SKIP
3364 017030 012703 006443                MOV     #VCNRST,R3        ;SET REASON POINTER
3365 017034 000417                       BR      2$                ;EXIT
3366 017036 032737 040000 003060 7$:     BIT     #DRVERR,T.CS      ;CHECK IF DRIVE ERROR
3367 017044 001405                       BEQ     9$                ;NO - SKIP
3368 017046                              ERRHRD  10004.,,ERR6
     017046 104456                       TRAP    C$ERHRD
     017050 023424                       .WORD   10004
     017052 000000                       .WORD   0
     017054 012226                       .WORD   ERR6
3369 017056 000412                       BR      14$               ;EXIT
3370 017060 012703 006464        9$:     MOV     #UNXERR,R3        ;SET REASON POINTER
3371 017064 000403                       BR      2$                ;EXIT
3372 017066 004737 016104        1$:     JSR     PC,WAITIN         ;WAIT FOR INTERRUPT
3373 017072 012603                       MOV     (SP)+,R3          ;STORE REASON POINTER FOR RETURN
3374 017074                      2$:     ERRHRD  10002.,,ERR1
     017074 104456                       TRAP    C$ERHRD
     017076 023422                       .WORD   10002
     017100 000000                       .WORD   0
     017102 011724                       .WORD   ERR1
3375 017104 005037 003032        14$:    CLR     ERRSWI            ;CLEAR FOR ERROR RETURN
3376 017110 005737 003142        3$:     TST     TEMP4             ;TEST IF REGISTERS WERE SAVED
3377 017114 001007                       BNE     22$               ;NO - SKIP
3378 017116 012703 003050                MOV     #L.CS,R3          ;SET POINTER TO RESTORE
3379 017122 012701 000004                MOV     #4,R1             ;SET REGISTER COUNT
3380 017126 012623               20$:    MOV     (SP)+,(R3)+       ;RESTORE REG
3381 017130 005301                       DEC     R1                ;DEC COUNT
3382 017132 001375                       BNE     20$               ;LOOP UNTIL ALL ARE RESTORED
3383 017134 162737 000002 003016 22$:    SUB     #2,SSINDX         ;REMOVE ENTRY FROM SUBROUT STACK
3384 017142 012601                       MOV     (SP)+,R1          ;RESTORE R1
3385 017144 012600                       MOV     (SP)+,R0          ;RESTORE R0
3386 017146 012603                       MOV     (SP)+,R3          ;RESTORE R3
3387 017150 012637 003142                MOV     (SP)+,TEMP4       ;RESTORE TEMP4
3388 017154 005737 003032                TST     ERRSWI            ;TEST IF ERROR RETURN
3389 017160 001403                       BEQ     99$               ;YES - SKIP
3390 017162 063716 003032                ADD     ERRSWI,(SP)       ;ADD IN ERROR RETURN
```

GLOBAL SUBROUTINES

```
3391 017166  000207                          RTS     PC
3392 017170  017616  000000          99$:    MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
3393 017174  000207                          RTS     PC
3394
3395                                 ;       SEEK ROUTINE
3396 017176  012737  177777  003134  XSEEKT: MOV     #-1,TEMP1       ;SET SPECIAL TIMING SEEK FLAG
3397 017204  000402                          BR      XSEEK1
3398 017206  005037  003134          XSEEK:  CLR     TEMP1           ;CLEAR SPECIAL SEEK FOR TIMING FLAG
3399 017212  010346                  XSEEK1: MOV     R3,-(SP)        ;STORE R3
3400 017214  013703  003016                  MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
3401 017220  005723                          TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
3402 017222  016663  000002  002420          MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
3403 017230  162763  000004  002420          SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
3404 017236  010337  003016                  MOV     R3,SSINDX       ;STORE IT BACK
3405 017242  010046                          MOV     R0,-(SP)
3406 017244  010146                          MOV     R1,-(SP)
3407 017246  010546                          MOV     R5,-(SP)        ;STORE REG
3408 017250  012737  000002  003032          MOV     #2,ERRSWI       ;SET FOR NO ERROR RETURN
3409 017256  005037  003112                  CLR     DIFAUG          ;CLEAR DIFFERENCE AUGMENT (FOR SEEKING
3410                                                                 ;   PAST GUARD BAND)
3411 017262  004737  022366                  JSR     PC,GETPOS       ;GET PRESENT POSITION
3412 017266  017720                          65$
3413 017270  013737  003120  003114          MOV     CURCYL,OLDCYL   ;MOVE CURRENT TO OLD CYLINDER
3414 017276  023737  003116  002316          CMP     NEWCYL,HLMTW    ;TEST IF NEW IS GREATER THAN 255
3415 017304  003427                          BLE     3$              ;NO - SKIP
3416 017306  163737  002316  003116          SUB     HLMTW,NEWCYL    ;ELSE SUBTRACT 255.
3417 017314  013737  003116  003112          MOV     NEWCYL,DIFAUG   ;STORE DIFFERENCE AS AUGMENT
3418 017322  013737  002316  003116          MOV     HLMTW,NEWCYL    ;SET NEWCYL AS 255.
3419 017330  022737  000001  002312          CMP     #1,T.DRIVE
3420 017336  001424                          BEQ     6$
3421 017340  162737  000001  003116          SUB     #1,NEWCYL
3422 017346  012737  000001  003124          MOV     #1,DESSGN
3423 017354  012737  000001  003122          MOV     #1,DESDIF
3424 017362  000451                          BR      18$
3425 017364  005737  003116          3$:     TST     NEWCYL          ;TEST IF NEWCYL HAS NEGATIVE VALUE
3426 017370  100007                          BPL     6$              ;NO - SKIP
3427 017372  005437  003116                  NEG     NEWCYL          ;ELSE MAKE IT POSITIVE
3428 017376  013737  003116  003112          MOV     NEWCYL,DIFAUG   ;AND STORE IT AS AUGMENT
3429 017404  005037  003116                  CLR     NEWCYL          ;AND SET NEWCYL TO 0
3430 017410  013705  003120          6$:     MOV     CURCYL,R5       ;COMPUTE DIFFERENCE AND NEW CYLINDER
3431 017414  163705  003116                  SUB     NEWCYL,R5       ;SUB NEWCYL FROM CURCYL
3432 017420  100005                          BPL     13$             ;IF DIFF IS POSITIVE - SKIP(REV SEEK)
3433 017422  012737  000001  003124          MOV     #1,DESSGN       ;ELSE SET SIGN FOR FORWARD
3434 017430  005405                          NEG     R5              ;MAKE DIFFERENCE POSITIVE
3435 017432  000402                          BR      14$             ;SKIP
3436 017434  005037  003124          13$:    CLR     DESSGN          ;SET SIGN FOR REVERSE
3437 017440  010537  003122          14$:    MOV     R5,DESDIF       ;STORE DIFFERENCE
3438 017444  005737  003112                  TST     DIFAUG          ;IS THERE A DIFFERENCE AUGMENT
3439 017450  001416                          BEQ     18$             ;NO - SKIP
3440 017452  023737  003116  002316          CMP     NEWCYL,HLMTW    ;CHECK IF NEW CYL IS 255.
3441 017460  001007                          BNE     17$             ;NO - SKIP
3442 017462  012737  000001  003124          MOV     #1,DESSGN       ;ELSE FORCE SIGN FOR FORWARD
3443                                                                 ;(INNER GUARD BAND)
3444 017470  022737  000001  002312          CMP     #1,T.DRIVE
3445 017476  001003                          BNE     18$
3446 017500  063737  003112  003122  17$:    ADD     DIFAUG,DESDIF
3447 017506                          18$:
```

GLOBAL SUBROUTINES

```
3448 017506  012705  003050                MOV     #L.CS,R5        ;GET L REG ADDRESS
3449 017512  012715  000106                MOV     #SEEK,(R5)      ;SET FOR SEEK
3450 017516  053715  003046                BIS     RLDRV,(R5)      ;INSERT DRIVE NUMBER
3451 017522  042725  002000                BIC     #BIT10,(R5)+    ;CLEAR IF DRIVE 4 - 7 SPEC D
3452 017526  005025                        CLR     (R5)+           ;CLEAR BUS ADDRESS
3453 017530  013715  003122                MOV     DESDIF,(R5)     ;LOAD DIFFERENCE
3454 017534  012700  000007                MOV     #7,R0           ;SET TO SHIFT DIFFERENCE
3455 017540  006315              21$:      ASL     (R5)
3456 017542  005300                        DEC     R0
3457 017544  001375                        BNE     21$             ;LOOP UNTIL ALIGNED
3458 017546  005737  003124                TST     DESSGN          ;TEST SIGN
3459 017552  001402                        BEQ     23$             ;SKIP IF 0
3460 017554  052715  000004                BIS     #DIRBIT,(R5)    ;ELSE INSERT SIGN
3461 017560  005737  003126        23$:    TST     DESHD           ;TEST IF HEAD 0
3462 017564  001402                        BEQ     25$             ;YES - SKIP
3463 017566  052715  000020                BIS     #HDSEL,(R5)     ;ELSE SET HEAD BIT
3464 017572  052725  000001        25$:    BIS     #MBSET0,(R5)+   ;INSERT MARKER BIT
3465 017576  004737  020324                JSR     PC,RDYCHK       ;CHECK IF DRIVE READY
3466 017602  017720                        65$
3467 017604  005037  003022                CLR     DONE            ;CLEAR INTERRUPT FLAG
3468 017610  005737  003134                TST     TEMP1           ;CHECK IF SPECIAL SEEK FLAG SET
3469 017614  001041                        BNE     65$             ;YES - SKIP, DO NOT START SEEK
3470 017616  014562  000004                MOV     -(R5),RLDA(R2)  ;LOAD RL REGISTERS
3471 017622  014562  000002                MOV     -(R5),RLBA(R2)
3472 017626  014562  000000                MOV     -(R5),RLCS(R2)
3473 017632                        30$:    WAITUS  #10.
3474 017644  005737  003022                TST     DONE            ;TEST IF INTERRUPT DONE
3475 017650  001012                        BNE     32$             ;YES - SKIP
3476 017652  004737  016104                JSR     PC,WAITIN       ;GO WAIT FOR INTERRUPT
3477 017656  012603                        MOV     (SP)+,R3        ;GET RESULT MESSAGE POINTER
3478 017660                                ERRHRD  10005,,ERR1
     017660  104456                        TRAP    C$ERHRD
     017662  023425                        .WORD   10005
     017664  000000                        .WORD   0
     017666  011724                        .WORD   ERR1
3479 017670  005037  003032                CLR     ERRSWI          ;CLEAR FOR ERROR RETURN
3480 017674  000411                        BR      65$
3481 017676  005737  003060        32$:    TST     T.CS            ;TEST IF ANY ERROR
3482 017702  100006                        BPL     65$             ;NO - SKIP
3483 017704                                ERRHRD  10006,,ERR6
     017704  104456                        TRAP    C$ERHRD
     017706  023426                        .WORD   10006
     017710  000000                        .WORD   0
     017712  012226                        .WORD   ERR6
3484 017714  005037  003032                CLR     ERRSWI          ;CLEAR FOR ERROR RETURN
3485 017720  162737  000002  003016  65$:  SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
3486 017726  012605                        MOV     (SP)+,R5        ;RESTORE REGISTERS
3487 017730  012601                        MOV     (SP)+,R1
3488 017732  012600                        MOV     (SP)+,R0
3489 017734  012603                        MOV     (SP)+,R3
3490 017736  005737  003032                TST     ERRSWI          ;TEST IF ERROR RETURN
3491 017742  001403                        BEQ     99$             ;YES - SKIP
3492 017744  063716  003032                ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
3493 017750  000207                        RTS     PC
3494 017752  017616  000000        99$:    MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
3495 017756  000207                        RTS     PC
3496
```

GLOBAL SUBROUTINES

```
3553
3555                                    ;       POSITION HEADS ROUTINE. POSITIONS HEADS USING 1 CYLINDER SEEKS
3556                                    ;       TO CYLINDER SPECIFIED IN R5 BY THE CALLING ROUTINE
3557 017760  010346                     POSHDS: MOV     R3,-(SP)        ;SAVE REGS
3558 017762  013703  003016                     MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
3559 017766  005723                             TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
3560 017770  016663  000002  002420             MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
3561 017776  162763  000004  002420             SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
3562 020004  010337  003016                     MOV     R3,SSINDX       ;STORE IT BACK
3563 020010  010346                             MOV     R3,-(SP)
3564 020012  010446                             MOV     R4, (SP)
3565 020014  012737  000002  003032             MOV     #2,ERRSWI       ;SET FOR NO ERROR RETURN
3566 020022  004737  022366                     JSR     PC,GETPOS       ;GET CURRENT POSITION
3567 020026  020266                             PH65$
3568 020030  012704  000012                     MOV     #10.,R4         ;SET RETRY COUNT
3569 020034                             BGNSEG
     020034  104404                             TRAP    C$BSEG
3570 020036                             1$:     INLOOP                  ;CHECK IF IN ERROR LOOP
     020036  104420                             TRAP    C$INLP
3571 020040                                     BNCOMPLETE      5$      ;NO - SKIP
     020040  103012                             BCC     5$
3572 020042  004737  022366                     JSR     PC,GETPOS       ;ELSE GET POSITION
3573 020046  020264                             60$
3574 020050  023737  003120  003116             CMP     CURCYL,NEWCYL   ;CHECK IF AT INTENDED POSITION
3575 020056  001017                             BNE     8$              ;NO - SKIP
3576 020060  004737  020664                     JSR     PC,ONSWAP       ;SWAP OLDCYL AND NEWCYL
3577 020064  000414                             BR      8$              ;SKIP
3578 020066  013737  003120  003114  5$:     MOV     CURCYL,OLDCYL   ;IN NOT LOOPING, STORE CURCYL AS OLDCYL
3579 020074  023705  003120                     CMP     CURCYL,R5       ;CHECK IF HDS AT FINAL POSITION
3580 020100  001471                             BEQ     60$             ;YES - GO TO EXIT
3581 020102  003003                             BGT     7$              ;IF CURCYL > FINAL POSITION   SKIP
3582 020104  005237  003116                     INC     NEWCYL          ;ELSE BUMP NEWCYL (MOVE HDS IN)
3583 020110  000402                             BR      8$              ;SKIP
3584 020112  005337  003116             7$:     DEC     NEWCYL          ;DEC NEWCYL (MOVE HDS OUT)
3585 020116  004737  017206             8$:     JSR     PC,XSEEK        ;DO SEEK
3586 020122  020264                             60$
3587 020124  012701  005670                     MOV     #3000.,R1       ;SET WAIT COUNT 300 MS
3588 020130  004737  022102                     JSR     PC,RDYWAIT      ;WAIT FOR DRIVE READY
3589 020134  020264                             60$
3590 020136  005737  003060                     TST     T.CS            ;TEST IF ANY ERROR
3591 020142  100007                             BPL     10$             ;NO - SKIP
3592 020144                                     ERRHRD  10008.,,ERR6
     020144  104456                             TRAP    C$ERHRD
     020146  023430                             .WORD   10008
     020150  000000                             .WORD   0
     020152  012226                             .WORD   ERR6
3593 020154  005037  003032                     CLR     ERRSWI          ;CLEAR FOR ERROR ERROR RETURN
3594 020160  000441                             BR      60$
3595 020162  004737  022366             10$:    JSR     PC,GETPOS       ;GET POSITION
3596 020166  020264                             60$
3597 020170  023737  003120  003116             CMP     CURCYL,NEWCYL   ;CHECK IF ARRIVED AT DESIRED PLACE
3598 020176  001003                             BNE     15$             ;NO - SKIP
3599 020200  012704  000012             14$:    MOV     #10.,R4         ;ELSE INIT RETRY COUNT
3600 020204  000714                             BR      1$              ;GO DO NEXT SEEK
3601 020206  005737  003124             15$:    TST     DESSGN          ;TEST IF GOING IN
3602 020212  001017                             BNE     17$             ;YES - SKIP
3603 020214  023737  003120  003116             CMP     CURCYL,NEWCYL   ;CHECK IF HEADS DID NOT MOVE IN
```

GLOBAL SUBROUTINES

```
3604 020222  003366                        BGT     14$             ;YES - SKIP
3605 020224  005304                16$:     DEC     R4              ;DEC RETRY COUNT
3606 020226  001333                        BNE     8$              ;DO ANOTHER SEEK IF NOT 0
3607 020230  012703  007323                MOV     @#DMOVF,R3      ;ELSE SET RESULT MESSAGE POINTER
3608 020234                                ERRHRD  10009.,,ERR1
     020234  104456                        TRAP    C$ERHRD
     020236  023431                        .WORD   10009
     020240  000000                        .WORD   0
     020242  011724                        .WORD   ERR1
3609 020244  005037  003032                CLR     ERRSWI          ;CLEAR FOR ERROR ERROR RETURN
3610 020250  000405                        BR      60$
3611 020252  023737  003120  003116  17$:  CMP     CURCYL,NEWCYL   ;HDS SHOULD MOVE OUT, CHK THEY DID
3612 020260  002747                        BLT     14$             ;YES   SKIP
3613 020262  000760                        BR      16$             ;ELSE GO DEC AND RETRY
3614 020264                        20$:
3615 020264                        60$:
3616 020264                        ENDSEG
     020264                        10000$:
     020264  104405                        TRAP    C$ESEG
3617 020266  162737  000002  003016  PH65$: SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
3618 020274  012604                        MOV     (SP)+,R4        ;RESTORE REGISTERS
3619 020276  012600                        MOV     (SP)+,R0
3620 020300  012603                        MOV     (SP)+,R3
3621 020302  005737  003032                TST     ERRSWI          ;TEST IF ERROR RETURN
3622 020306  001403                        BEQ     99$             ;YES - SKIP
3623 020310  063716  003032                ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
3624 020314  000207                        RTS     PC
3625 020316  017616  000000          99$:  MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
3626 020322  000207                        RTS     PC
3627
3629                               ;        DRIVE READY TEST ROUTINE. CHECKS DRIVE IS READY. IF NOT, WAIT
3630                               ;        500MS FOR READY TO SET.
3631 020324  010346                RDYCHK: MOV     R3,-(SP)        ;STORE REGS
3632 020326  013703  003016                MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
3633 020332  005723                        TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
3634 020334  016663  000002  002420        MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
3635 020342  162763  000004  002420        SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
3636 020350  010337  003016                MOV     R3,SSINDX       ;STORE IT BACK
3637 020354  010046                        MOV     R0,-(SP)
3638 020356  010146                        MOV     R1,-(SP)
3639 020360  010446                        MOV     R4,-(SP)
3640 020362  012737  000002  003032        MOV     #2,ERRSWI       ;SET FOR NO ERROR RETURN
3641 020370  012701  011610                MOV     #5000.,R1       ;SET WAIT COUNT
3642 020374  004737  016310          1$:   JSR     PC,GSTAT        ;GET DRIVE STATUS
3643 020400  020534                        4$
3644 020402  032737  000001  003060        BIT     #DRDYMSK,T.CS   ;TEST IF DRIVE READY
3645 020410  001053                        BNE     5$              ;YES - EXIT
3646 020412                                WAITUS  #1
3647 020424  005301                        DEC     R1              ;DEC WAIT COUNT
3648 020426  001362                        BNE     1$              ;LOOP IF NOT 0
3649 020430  012703  007760                MOV     #MDRDY,R3       ;SET RESULT MESSAGE POINTER
3650 020434  012704  010645                MOV     #C500MS,R4      ;SET CONDITION MESSAGE POINTER
3651 020440                                ERRHRD  10010.,,ERR5
     020440  104456                        TRAP    C$ERHRD
     020442  023432                        .WORD   10010
     020444  000000                        .WORD   0
     020446  012156                        .WORD   ERR5
```

GLOBAL SUBROUTINES

```
3652 020450  012701  000062            MOV     #50.,R1          ;SET WAIT COUNT FOR 5 SECONDS
3653 020454  004737  016310      2$:   JSR     PC,GSTAT         ;GET DRIVE STATUS
3654 020460  020534                    4$
3655 020462  032737  000001  003060    BIT     #DRDYMSK,T.CS    ;TEST IF DRIVE READY
3656 020470  001007                    BNE     3$               ;YES - SKIP
3657 020472                            WAITMS  #1               ;WAIT FOR 100MS
3658 020504  005301                    DEC     R1               ;DEC WAIT COUNTER
3659 020506  001362                    BNE     2$               ;LOOP UNTIL TIME DONE
3660 020510  032737  100000  003060  3$:   BIT     #ANYERR,T.CS     ;TEST IF ANYERR SET
3661 020516  001406                    BEQ     4$               ;NO - SKIP
3662 020520                            ERRHRD  10011.,ERR6      ;REPORT ALL ERRORS
     020520  104456                    TRAP    C$ERHRD
     020522  023433                    .WORD   10011
     020524  000000                    .WORD   0
     020526  012226                    .WORD   ERR6
3663 020530  005337  003254            DEC     ERRCNT           ;REDUCE ERROR COUNT FOR DUAL ERRORS
3664 020534  005037  003032      4$:   CLR     ERRSWI           ;CLEAR FOR ERROR RETURN
3665 020540  162737  000002  003016  5$:   SUB     #2,SSINDX        ;REMOVE ENTRY FROM SUBROUT STACK
3666 020546  012604                    MOV     (SP)+,R4         ;RESTORE REGS
3667 020550  012601                    MOV     (SP)+,R1
3668 020552  012600                    MOV     (SP)+,R0
3669 020554  012603                    MOV     (SP)+,R3
3670 020556  005737  003032            TST     ERRSWI           ;TEST IF ERROR RETURN
3671 020562  001403                    BEQ     99$              ;YES - SKIP
3672 020564  063716  003032            ADD     ERRSWI,(SP)      ;ADD IN ERROR RETURN
3673 020570  000207                    RTS     PC
3674 020572  017616  000000      99$:  MOV     @(SP),(SP)       ;SET ERROR RETURN ADDRESS
3675 020576  000207                    RTS     PC
3676
3677                                ;       CHOOSE HEAD ROUTINE. PICKS HEAD 0 UNLESS SPECIFIC HEAD IS
3678                                ;       SELECTED BY SOFTWARE PARAMETER.
3679 020600  005037  003126      CHOSHD: CLR     DESHD            ;CLEAR TO HEAD 0
3680 020604  032737  010000  013556    BIT     #HEADLM,MISWIW   ;TEST IF HEAD SPECIFIED
3681 020612  001403                    BEQ     1$               ;NO - SKIP
3682 020614  013737  013564  003126    MOV     HEADW,DESHD      ;INSERT SPECIFIED HEAD
3683 020622  000207                1$:   RTS     PC
3684
3685                                ;       SWAP HEAD ROUTINE. CHANGES SELECTED HEAD TO HEAD 1
3686                                ;       UNLESS HEAD 0 SPECIFICALLY SELECTED BY SOFTWARE PARAMETER.
3687 020624  032737  010000  013556  SWAPHD: BIT     #HEADLM,MISWIW   ;TEST IF HEAD SPECIFIED
3688 020632  001011                    BNE     2$               ;YES - TAKE ABORT EXIT
3689 020634  005737  003126            TST     DESHD            ;TEST IF HEAD ONE USED
3690 020640  001006                    BNE     2$               ;YES - TAKE ABORT EXIT
3691 020642  012737  000001  003126    MOV     #1,DESHD         ;ELSE SET FOR HEAD ONE
3692 020650  062716  000002            ADD     #2,(SP)          ;BUMP PAST ABORT RETURN
3693 020654  000207                    RTS     PC               ;RETURN
3694 020656  017616  000000      2$:   MOV     @(SP),(SP)       ;GET ABORT DESTINATION
3695 020662  000207                3$:   RTS     PC
3696
3697                                ;       SWAP OLD CYLINDER AND NEW CYLINDER ROUTINE.
3698 020664  010046              ONSWAP: MOV     R0,-(SP)         ;STORE R0
3699 020666  013700  003114            MOV     OLDCYL,R0        ;MOVE OLD TO R0
3700 020672  013737  003116  003114    MOV     NEWCYL,OLDCYL    ;MOVE NEW TO OLD
3701 020700  010037  003116            MOV     R0,NEWCYL        ;PUT OLD IN NEW
3702 020704  012600                    MOV     (SP)+,R0         ;RESTORE R0
3703 020706  000207                    RTS     PC
3704
```

GLOBAL SUBROUTINES

```
3706                                    ;       BAD SECTOR FILES VALID CHECK ROUTINE. CHECKS IF BAD SECTOR
3707                                    ;       FILES HAVE BEEN READ AND STORED. IF NOT, REPORT AND FORCE
3708                                    ;       FILES TO LOOK LIKE ALL SECTORS OK.
3709 020710  005737  003510            CKBSVD: TST     BSFVAL          ;TEST IF BAD SECTORS STORED
3710 020714  001051                            BNE     5$              ;YES - EXIT
3711 020716                                    PRINTF  #FMT9,#BSNSTR   ;REPORT
     020716  012746  007550                    MOV     #BSNSTR,-(SP)
     020722  012746  011212                    MOV     #FMT9,-(SP)
     020726  012746  000002                    MOV     #2,-(SP)
     020732  010600                            MOV     SP,R0
     020734  104417                            TRAP    C$PNTF
     020736  062706  000006                    ADD     #6,SP
3712 020742                                    PRINTF  #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
     020742  005046                            CLR     -(SP)
     020744  153716  003047                    BISB    RLDRV+1,(SP)
     020750  012746  006152                    MOV     #DRVNAM,-(SP)
     020754  013746  003042                    MOV     RLBAS,-(SP)
     020760  012746  006141                    MOV     #BASADD,-(SP)
     020764  012746  011026                    MOV     #FMT5,-(SP)
     020770  012746  000005                    MOV     #5,-(SP)
     020774  010600                            MOV     SP,R0
     020776  104417                            TRAP    C$PNTF
     021000  062706  000014                    ADD     #14,SP
3713 021004                                    PRINTF  #FMT3
     021004  012746  011012                    MOV     #FMT3,-(SP)
     021010  012746  000001                    MOV     #1,-(SP)
     021014  010600                            MOV     SP,R0
     021016  104417                            TRAP    C$PNTF
     021020  062706  000004                    ADD     #4,SP
3714 021024  012737  177777  003512            MOV     #-1,SBSFIL      ;FORCE FILES TO NO ENTRIES
3715 021032  012737  177777  003706            MOV     #-1,FBSFIL
3716 021040  000207                    5$:     RTS     PC
3717
3719                                    ;       READ HEADERS ROUTINE.
3720 021042  012737  000001  003142    XRDHDC: MOV     #1,TEMP4        ;SET FLAG TO BYPASS REG STORAGE
3721 021050  000402                            BR      XRDHDG          ;GO DO IT
3722 021052  005037  003142            XRDHD:  CLR     TEMP4           ;SET FLAG TO SAVE T. AMD L. REGS
3723 021056  010346                    XRDHDG: MOV     R3,-(SP)        ;STORE REGISTERS
3724 021060  013703  003016                    MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
3725 021064  005723                            TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
3726 021066  016663  000002  002420            MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
3727 021074  162763  000004  002420            SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
3728 021102  010337  003016                    MOV     R3,SSINDX       ;STORE IT BACK
3729 021106  010046                            MOV     R0,-(SP)
3730 021110  010146                            MOV     R1,-(SP)
3731 021112  010446                            MOV     R4,-(SP)
3732 021114  012737  000002  003032            MOV     #2,ERRSWI       ;SET FOR NO ERROR RETURN
3733 021122  005737  003142                    TST     TEMP4           ;TEST IF REGISTERS TO BE SAVED
3734 021126  001007                            BNE     2$              ;NO - SKIP
3735 021130  012703  003060                    MOV     #L.MP+2,R3      ;SET POINTER FOR REGS
3736 021134  012701  000004                    MOV     #4,R1           ;SET COUNT
3737 021140  014346                    1$:     MOV     -(R3),-(SP)     ;SAVE REGISTER
3738 021142  005301                            DEC     R1              ;DEC COUNT
3739 021144  001375                            BNE     1$              ;LOOP UNTIL ALL ARE SAVED
3740 021146  004737  020324            2$:     JSR     PC,RDYCHK       ;CHECK DRIVE READY
3741 021152  021422                            65$
3742 021154  005037  003022                    CLR     DONE            ;CLEAR INTERRUPT FLAG
```

GLOBAL SUBROUTINES

```
3743 021160  012701  003050                    MOV     #L.CS,R1          ;GET ADDRESS OF LOAD REGS
3744 021164  013711  003046                    MOV     RLDRV,(R1)        ;LOAD DRIVE NUMBER
3745 021170  042711  002000                    BIC     #BIT10,(R1)       ;CLEAR FOR DRIVE 4   7 SPEC D
3746 021174  052721  000110                    BIS     #RDHEAD,(R1)+     ;INSERT COMMAND
3747 021200  005021                            CLR     (R1)+             ;CLEAR BA
3748 021202  005021                            CLR     (R1)+             ;CLEAR DA
3749 021204  014162  000004                    MOV     -(R1),RLDA(R2)    ;LOAD RL11 REGS
3750 021210  014162  000002                    MOV     -(R1),RLBA(R2)
3751 021214  014162  000000                    MOV     -(R1),RLCSR(R2)
3752 021220                            3$:     WAITUS  #10.              ;WAIT 1MS FOR INTERRUPT
3753 021232  005737  003022                    TST     DONE              ;TEST IN INTERRUPT FLAG SET
3754 021236  001460                            BEQ     14$               ;NO - SKIP
3755 021240  032737  000001  003060  5$:       BIT     #DRDYMSK,T.CS     ;TEST IF DRIVE READY
3756 021246  001035                            BNE     10$               ;YES - SKIP
3757 021250  012703  007760                    MOV     #MDRDY,R3         ;SET NO READY MESSAGE
3758 021254  012704  010662                    MOV     #CAFDT,R4         ;CONDITION OF AFTER DATA XFER
3759 021260                                    ERRHRD  10017...,ERR5
     021260  104456                            TRAP    C$ERHRD
     021262  023441                            .WORD   10017
     021264  000000                            .WORD   0
     021266  012156                            .WORD   ERR5
3760 021270  012701  000062                    MOV     #50.,R1           ;SET WAIT COUNT FOR 5 SECONDS
3761 021274  004737  016310            4$:     JSR     PC,GSTAT          ;GET STATUS
3762 021300  021416                            60$
3763 021302  032737  000001  003060            BIT     #DRDYMSK,T.CS     ;TEST IF DRIVE HAS COME READY
3764 021310  001403                            BEQ     11$               ;NO - SKIP
3765 021312  005037  003032                    CLR     ERRSWI            ;CLEAR ERROR SWITCH
3766 021316  000411                            BR      10$               ;SKIP
3767 021320  005301                    11$:    DEC     R1                ;DEC WAIT COUNT
3768 021322  001364                            BNE     4$                ;LOOP UNTIL TIME DONE
3769 021324  012704  010673                    MOV     #C5SEC,R4         ;SET CONDITION AFTER 5 SECONDS
3770 021330                                    ERRHRD  10014...,ERR5
     021330  104456                            TRAP    C$ERHRD
     021332  023436                            .WORD   10014
     021334  000000                            .WORD   0
     021336  012156                            .WORD   ERR5
3771 021340  000426                            BR      60$               ;EXIT
3772 021342  005737  003060            10$:    TST     T.CS              ;CHECK FOR ANY ERRORS
3773 021346  100005                            BPL     12$               ;NO - SKIP
3774 021350                                    ERRHRD  10016...,ERR6     ;REPORT ALL ERRORS
     021350  104456                            TRAP    C$ERHRD
     021352  023440                            .WORD   10016
     021354  000000                            .WORD   0
     021356  012226                            .WORD   ERR6
3775 021360  000416                            BR      60$
3776 021362  012701  003070            12$:    MOV     #HDWRD2,R1        ;GET POINTER
3777 021366  016221  000006                    MOV     RLMP(R2),(R1)+    ;STORE LAST TWO HEADER WORDS
3778 021372  016221  000006                    MOV     RLMP(R2),(R1)+
3779 021376  000411                            BR      65$               ;EXIT
3780 021400  004737  016104            14$:    JSR     PC,WAITIN         ;WAIT FOR INTERRUPT
3781 021404  012603                            MOV     (SP)+,R3          ;GET RESULTS
3782 021406                                    ERRHRD  10015...,ERR1     ;REPORT
     021406  104456                            TRAP    C$ERHRD
     021410  023437                            .WORD   10015
     021412  000000                            .WORD   0
     021414  011724                            .WORD   ERR1
3783 021416  005037  003032            60$:    CLR     ERRSWI            ;CLEAR FOR ERROR ERROR RETURN
```

GLOBAL SUBROUTINES

```
3784 021422  005737  003142          65$:   TST     TEMP4           ;TEST IF REGISTERS WERE SAVED
3785 021426  001007                         BNE     22$             ;NO   SKIP
3786 021430  012703  003050                 MOV     #L.CS,R3        ;SET POINTER TO RESTORE REGS
3787 021434  012701  000004                 MOV     #4,R1           ;SET COUNT
3788 021440  012623                 20$:    MOV     (SP)+,(R3)+     ;RESTORE REGISTER
3789 021442  005301                         DEC     R1              ;DEC COUNT
3790 021444  001375                         BNE     20$             ;LOOP UNTIL ALL ARE RESTORED
3791 021446  162737  000002  003016 22$:    SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
3792 021454  012604                         MOV     (SP)+,R4        ;RESTORE REGS
3793 021456  012601                         MOV     (SP)+,R1
3794 021460  012600                         MOV     (SP)+,R0
3795 021462  012603                         MOV     (SP)+,R3
3796 021464  005737  003032                 TST     ERRSWI          ;TEST IF ERROR RETURN
3797 021470  001403                         BEQ     99$             ;YES - SKIP
3798 021472  063716  003032                 ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
3799 021476  000207                         RTS     PC
3800 021500  017616  000000          99$:   MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
3801 021504  000207                         RTS     PC
3802
3804                                 ;      VERIFY HEADERS ROUTINE. COMPARES 40 HEADERS FOR CONTENT AND
3805                                 ;      SEQUENCE.
3806 021506  010346                 VERHDR: MOV     R3,-(SP)        ;STORE REGS
3807 021510  013703  003016                 MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
3808 021514  005723                         TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
3809 021516  016663  000002  002420         MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
3810 021524  162763  000004  002420         SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
3811 021532  010337  003016                 MOV     R3,SSINDX       ;STORE IT BACK
3812 021536  010046                         MOV     R0,-(SP)
3813 021540  010146                         MOV     R1,-(SP)
3814 021542  010446                         MOV     R4,-(SP)
3815 021544  010546                         MOV     R5,-(SP)
3816 021546  012737  000002  003032         MOV     #2,ERRSWI       ;SET FOR NO ERROR RETURN
3817 021554  052737  000002  003020         BIS     #HDRCMP,OPFLAG  ;SET HEADER COMPARE FLAG
3818 021562  005037  003030                 CLR     MORECE          ;CLEAR MORE ERRORS FLAG
3819 021566  012704  004102                 MOV     #IBUFF,R4       ;SET POINTER TO HEADERS
3820 021572  012705  003132                 MOV     #TEMP0,R5       ;SET POINTER TO WORK AREA
3821 021576  005003                         CLR     R3              ;CLEAR FOR WORD COUNTER
3822 021600  011415                         MOV     (R4),(R5)       ;MOVE HDR WORD TO WORK AREA
3823 021602  011401                         MOV     (R4),R1         ;PUT WORD IN REG 1
3824 021604  042701  000177                 BIC     #177,R1 ;CLEAR ALL BUT CYLINDER
3825 021610  012700  000007                 MOV     #7,R0           ;SET SHIFT COUNT
3826 021614  006201                 3$:     ASR     R1              ;SHIFT
3827 021616  005300                         DEC     R0              ;DEC
3828 021620  001375                         BNE     3$              ;LOOP
3829 021622  020137  003116                 CMP     R1,NEWCYL       ;CHECK IF CYLINDER PART GOOD
3830 021626  001407                         BEQ     4$              ;YES - SKIP
3831 021630                                 ERRHRD  10018.,ERR10    ;REPORT ERROR
     021630  104456                         TRAP    C$ERHRD
     021632  023442                         .WORD   10018
     021634  000000                         .WORD   0
     021636  013320                         .WORD   ERR10
3832 021640  005037  003032                 CLR     ERRSWI          ;CLEAR FOR ERROR ERROR RETURN
3833 021644  000456                         BR      65$
3834 021646  012701  000050          4$:    MOV     #40.,R1         ;SET HEADER COUNT
3835 021652  042715  000100                 BIC     #HDHSEL,(R5)    ;CLEAR HEAD SELECT AND 0 BIT
3836 021656  005737  003126                 TST     DESHD           ;ARE WE USING HD 0?
3837 021662  001402                         BEQ     5$              ;YES - SKIP
```

GLOBAL SUBROUTINES

```
3838 021664  052715  000100                    BIS     #HDHSEL,(R5)     ;INSERT HEAD BIT
3839 021670  005065  000002          5$:       CLR     2(R5)            ;CLEAR 2ND WORD OF WORK AREA
3840 021674  021524                  6$:       CMP     (R5),(R4)+       ;TEST FIRST WORD OK
3841 021676  001410                            BEQ     8$               ;YES   SKIP
3842 021700  005744                            TST      (R4)            ;ELSE SET POINTER FOR ERROR
3843 021702                                     ERRHRD  10018.,,ERR10    ;REPORT
     021702  104456                            TRAP    C$ERHRD
     021704  023442                            .WORD   10018
     021706  000000                            .WORD   0
     021710  013320                            .WORD   ERR10
3844 021712  005037  003032                    CLR     ERRSWI           ;CLEAR FOR ERROR RETURN
3845 021716  005724                            TST     (R4)+            ;RESET POINTER
3846 021720  005203                  8$:       INC     R3               ;BUMP WORD COUNTER
3847 021722  005724                            TST     (R4)+            ;TEST 2ND WORD IS 0
3848 021724  001410                            BEQ     12$              ;YES - SKIP
3849 021726  022544                            CMP     (R5)+, (R4)      ;ADJUST POINTERS FOR REPORT
3850 021730                                     ERRHRD  10018.,,ERR10    ;REPORT
     021730  104456                            TRAP    C$ERHRD
     021732  023442                            .WORD   10018
     021734  000000                            .WORD   0
     021736  013320                            .WORD   ERR10
3851 021740  005037  003032                    CLR     ERRSWI           ;CLEAR FOR ERROR RETURN
3852 021744  024524                            CMP     -(R5),(R4)+      ;RESET POINTERS
3853 021746  005724                  12$:      TST     (R4)+            ;BUMP PAST ECC WORD
3854 021750  005203                            INC     R3               ;BUMP WORD COUNTER
3855 021752  005215                            INC     (R5)             ;BUMP SECTOR OF EXPECTED HEADER
3856 021754  011500                            MOV     (R5),R0          ;MOVE EXPECTED HDR TO R0
3857 021756  042700  177700                    BIC     #^CHDSEC,R0      ;CLEAR ALL BUT SECTOR
3858 021762  022700  000050                    CMP     #40.,R0          ;TEST IF AT SECTOR 40
3859 021766  001002                            BNE     15$              ;NO - SKIP
3860 021770  042715  000077                    BIC     #HDSEC,(R5)      ;CLEAR SECTOR TO 0
3861 021774  005203                  15$:      INC     R3               ;BUMP HDR WORD COUNTER
3862 021776  005301                            DEC     R1               ;DEC HEADER COUNT
3863 022000  001335                            BNE     6$               ;LOOP IF NOT YET DONE
3864 022002  162737  000002  003016  65$:      SUB     #2,SSINDX        ;REMOVE ENTRY FROM SUBROUT STACK
3865 022010  012605                            MOV     (SP)+,R5         ;RESTORE REGISTERS
3866 022012  012604                            MOV     (SP)+,R4
3867 022014  012601                            MOV     (SP)+,R1
3868 022016  012600                            MOV     (SP)+,R0
3869 022020  012603                            MOV     (SP)+,R3
3870 022022  005737  003032                    TST     ERRSWI           ;TEST IF ERROR RETURN
3871 022026  001403                            BEQ     99$              ;YES - SKIP
3872 022030  063716  003032                    ADD     ERRSWI,(SP)      ;ADD IN ERROR RETURN
3873 022034  000207                            RTS     PC
3874 022036  017616  000000          99$:      MOV     @(SP),(SP)       ;SET ERROR RETURN ADDRESS
3875 022042  000207                            RTS     PC
3876
3878                                ;         POSITION HEAD BIT FROM HEADER OR MULTIPURPOSE REGISTER TO LSB.
3879 022044  013705  003066          POSHW1:   MOV     HDWRD1,R5        ;START FOR POSITION HD BIT IN WD 1
3880 022050  000402                            BR      POSHDO           ;SKIP
3881 022052  013705  003066          POSHSB:   MOV     T.MP,R5 ;START FOR POSITION HD BIT IN MP
3882 022056  010146                  POSHDO:   MOV     R1, (SP)         ;STORE R1
3883 022060  042705  177677                    BIC     #^CHSSTAT,R5     ;CLEAR ALL BUT HEAD SEL BIT
3884 022064  012701  000006                    MOV     #6,R1            ;SET SHIFT COUNT
3885 022070  006205                  1$:       ASR     R5               ;SHIFT FOR RIGHT JUSTIFY
3886 022072  005301                            DEC     R1
3887 022074  001375                            BNE     1$
```

GLOBAL SUBROUTINES

```
3888 022076  012601                          MOV     (SP)+,R1        ;RESTORE R1
3889 022100  000207                          RTS     PC              ;RETURN
3890
3891                                 ;       WAIT FOR READY ROUTINE. DURATION OF WAIT PASSED TO THE ROUTINE
3892                                 ;       FROM THE CALLING ROUTINE IN R1.
3893 022102  010346                 RDYWAIT:        MOV     R3, (SP)        ;STORE R3
3894 022104  013703  003016                 MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
3895 022110  005723                         TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
3896 022112  016663  000002  002420         MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
3897 022120  162763  000004  002420         SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
3898 022126  010337  003016                 MOV     R3,SSINDX       ;STORE IT BACK
3899 022132  010046                         MOV     R0,-(SP)
3900 022134  010146                         MOV     R1, (SP)
3901 022136  010446                         MOV     R4,-(SP)
3902 022140  012737  000002  003032         MOV     #2,ERRSWI       ;SET FOR NO ERROR RETURN
3903 022146  004737  016310          5$:    JSR     PC,GSTAT        ;GET DRIVE STATUS
3904 022152  022322                         10$
3905 022154  032737  000001  003060         BIT     #DRDYMSK,T.CS   ;CHECK IF READY
3906 022162  001061                         BNE     9$              ;YES   SKIP
3907 022164  005301                         DEC     R1              ;DEC WAIT COUNT
3908 022166  001406                         BEQ     7$              ;SKIP IF 0
3909 022170                                  WAITUS  #1
3910 022202  000761                         BR      5$
3911 022204  012703  007760          7$:    MOV     #MDRDY,R3       ;SET NAME MESSAGE PTR
3912 022210                                  ERRHRD  10020,,ERR3     ;REPORT READY ERROR
     022210  104456                         TRAP    C$ERHRD
     022212  023444                         .WORD   10020
     022214  000000                         .WORD   0
     022216  012040                         .WORD   ERR3
3913 022220  012701  000062                 MOV     #50.,R1         ;SET WAIT COUNT FOR 5 SECONDS
3914 022224  004737  016310          6$:    JSR     PC,GSTAT        ;GET DRIVE STATUS
3915 022230  022322                         10$
3916 022232  032737  000001  003060         BIT     #DRDYMSK,T.CS   ;TEST IF DRIVE READY
3917 022240  001016                         BNE     8$              ;YES - SKIP
3918 022242                                  WAITMS  #1              ;WAIT 100 MS
3919 022254  005301                         DEC     R1              ;DEC WAIT COUNT
3920 022256  001362                         BNE     6$              ;LOOP UNTIL TIME DONE
3921 022260  012704  010673                 MOV     #C5SEC,R4       ;SET CONDITION AFTER 5 SECDS
3922 022264                                  ERRHRD  10021,,ERR5
     022264  104456                         TRAP    C$ERHRD
     022266  023445                         .WORD   10021
     022270  000000                         .WORD   0
     022272  012156                         .WORD   ERR5
3923 022274  000410                         BR      11$             ;EXIT
3924 022276  032737  100000  003060  8$:    BIT     #ANYERR,T.CS    ;TEST IF ANY ERROR SET
3925 022304  001406                         BEQ     10$             ;NO - SKIP
3926 022306                                  ERRHRD  10022,,ERR6     ;REPORT ALL ERRORS
     022306  104456                         TRAP    C$ERHRD
     022310  023446                         .WORD   10022
     022312  000000                         .WORD   0
     022314  012226                         .WORD   ERR6
3927 022316  005337  003254          11$:   DEC     ERRCNT          ;DEC FOR DOUBLE ERROR REPORT
3928 022322  005037  003032          10$:   CLR     ERRSWI          ;CLEAR FOR ERROR ERROR RETURN
3929 022326  162737  000002  003016  9$:    SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
3930 022334  012604                         MOV     (SP)+,R4        ;RESTORE REGISTERS
3931 022336  012601                         MOV     (SP)+,R1
3932 022340  012600                         MOV     (SP)+,R0
```

GLOBAL SUBROUTINES

```
3933 022342  012603                          MOV     (SP)+,R3        ;RESTORE R3
3934 022344  005737  003032                  TST     ERRSWI          ;TEST IF ERROR RETURN
3935 022350  001403                          BEQ     99$             ;YES - SKIP
3936 022352  063716  003032                  ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
3937 022356  000207                          RTS     PC
3938 022360  017616  000000          99$:    MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
3939 022364  000207                          RTS     PC
3940
3941                                 ;       GET POSITION ROUTINE. READS A HEADER FROM CURRENT CYLINDER
3942                                 ;       (WHERE IT IS PRESENTLY POSITIONED) AND STORES CYLINDER
3943                                 ;       NUMBER IN CURCYL.
3944 022366  010346                  GETPOS: MOV     R3,-(SP)        ;STORE REGISTERS
3945 022370  013703  003016                  MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
3946 022374  005723                          TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
3947 022376  016663  000002  002420          MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
3948 022404  162763  000004  002420          SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
3949 022412  010337  003016                  MOV     R3,SSINDX       ;STORE IT BACK
3950 022416  010046                          MOV     R0,-(SP)
3951 022420  010546                          MOV     R5,-(SP)
3952 022422  004737  021052                  JSR     PC,XRDHD        ;DO READ HEADER
3953 022426  022456                          65$
3954 022430  013703  003066                  MOV     HDWRD1,R3       ;GET HEADER WORD
3955 022434  012705  000007                  MOV     #7,R5           ;SET SHIFT COUNT
3956 022440  006203                  4$:     ASR     R3              ;SHIFT TO RIGHT JUSTIFY
3957 022442  005305                          DEC     R5
3958 022444  001375                          BNE     4$
3959 022446  042703  177000                  BIC     #177000,R3
3960 022452  010337  003120                  MOV     R3,CURCYL       ;STORE AS CURRENT CYLINDER
3961 022456  162737  000002  003016  65$:    SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
3962 022464  012605                          MOV     (SP)+,R5        ;RESTORE REGISTERS
3963 022466  012600                          MOV     (SP)+,R0
3964 022470  012603                          MOV     (SP)+,R3
3965 022472  005737  003032                  TST     ERRSWI          ;TEST IF ERROR RETURN
3966 022476  001403                          BEQ     99$             ;YES - SKIP
3967 022500  063716  003032                  ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
3968 022504  000207                          RTS     PC
3969 022506  017616  000000          99$:    MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
3970 022512  000207                          RTS     PC
3971
3973                                 ;       VERIFY POSITION ROUTINE. READS A HEADER (USING GETPOS) AND
3974                                 ;       CHECKS HEADS ARE POSITIONED AT NEW CYLINDER (CURCYL = NEWCYL).
3975 022514  010346                  VERPOS: MOV     R3,-(SP)        ;STORE R3
3976 022516  013703  003016                  MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
3977 022522  005723                          TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
3978 022524  016663  000002  002420          MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
3979 022532  162763  000004  002420          SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
3980 022540  010337  003016                  MOV     R3,SSINDX       ;STORE IT BACK
3981
3982 022544  012737  000002  003032          MOV     #2,ERRSWI       ;SET FOR NO ERROR RETURN
3983 022552  004737  022366                  JSR     PC,GETPOS       ;GET POSITION
3984 022556  022604                          65$
3985 022560  023737  003116  003120          CMP     NEWCYL,CURCYL   ;CHECK IF CURRENT CYL IS NEW CYL
3986 022566  001406                          BEQ     1$              ;YES - SKIP
3987 022570                                  ERRHRD  10022.,,ERR8
     022570  104456                          TRAP    C$ERHRD
     022572  023446                          .WORD   10022
     022574  000000                          .WORD   0
```

GLOBAL SUBROUTINES

```
      022576  013160                          .WORD   ERR8
 3988 022600  005037  003032                  CLR     ERRSWI          ;CLEAR FOR ERROR ERROR RETURN
 3989 022604                          1$:
 3990 022604  162737  000002  003016  65$:    SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
 3991 022612  012603                          MOV     (SP)+,R3        ;RESTORE R3
 3992 022614  005737  003032                  TST     ERRSWI          ;TEST IF ERROR RETURN
 3993 022620  001403                          BEQ     99$             ;YES   SKIP
 3994 022622  063716  003032                  ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
 3995 022626  000207                          RTS     PC
 3996 022630  017616  000000          99$:    MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
 3997 022634  000207                          RTS     PC
 3998
 4000                                 ;       READ ALL HEADERS ROUTINE. 40 HEADERS ARE READ AND STORED
 4001                                 ;       IN IBUFF.
 4002 022636  010346                  RDALHD: MOV     R3,-(SP)        ;STORE REGISTERS
 4003 022640  013703  003016                  MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
 4004 022644  005723                          TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
 4005 022646  016663  000002  002420          MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
 4006 022654  162763  000004  002420          SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
 4007 022662  010337  003016                  MOV     R3,SSINDX       ;STORE IT BACK
 4008 022666  010046                          MOV     R0,-(SP)
 4009 022670  010146                          MOV     R1,-(SP)
 4010 022672  010446                          MOV     R4,-(SP)
 4011 022674  012737  000002  003032          MOV     #2,ERRSWI       ;SET FOR NO ERROR RETURN
 4012 022702  012701  000050                  MOV     #40.,R1         ;SET HEADER COUNT
 4013 022706  052737  100000  003020          BIS     #HDR40,OPFLAG   ;SET 40 HDR OP FLAG
 4014 022714  012703  004102                  MOV     #IBUFF,R3       ;SET POINTER TO STORE HDRS
 4015 022720  013704  003042                  MOV     RLBAS,R4        ;GET BASE ADDRESS
 4016 022724  062704  000006                  ADD     #RLMP,R4        ;MAKE IT POINT TO MP REG
 4017 022730  012737  000010  003050          MOV     #10,L.CS        ;LOAD FOR READ HEADER, NO INTERRUPT
 4018 022736  053737  003046  003050          BIS     RLDRV,L.CS      ;INSERT DRIVE NUMBER
 4019 022744  042737  002000  003050          BIC     #BIT10,L.CS     ;CLEAR FOR DRIVE 4 - 7 SPEC'D
 4020 022752  005037  003052                  CLR     L.BA            ;CLEAR BA
 4021 022756  005037  003054                  CLR     L.DA            ;CLEAR DA
 4022 022762  005737  003126                  TST     DESHD           ;TEST IF HEAD 0
 4023 022766  001403                          BEQ     3$              ;YES - SKIP
 4024 022770  052737  000020  003054          BIS     #HDSEL,L.DA     ;ELSE INSERT HEAD 0
 4025 022776  013762  003054  000004  3$:     MOV     L.DA,RLDA(R2)   ;LOAD RLDA REG
 4026 023004  013762  003052  000002          MOV     L.BA,RLBA(R2)   ;LOAD RLBA
 4027 023012  032762  000200  000000          BIT     #CRDYMSK,RLCS(R2)       ;TEST IF CONTROLLER READY
 4028 023020  001003                          BNE     6$              ;YES - SKIP
 4029 023022  004737  020324                  JSR     PC,RDYCHK       ;ELSE CHECK READY
 4030 023026  023144                          65$
 4031 023030  013762  003050  000000  6$:     MOV     L.CS,RLCS(R2)   ;LOAD RLCS REG
 4032 023036  012700  077777                  MOV     #77777,R0       ;SET COUNT FOR WAIT
 4033 023042  032762  000200  000000  7$:     BIT     #CRDYMSK,RLCS(R2) ;CHECK THAT OPERATION COMPLETED
 4034 023050  001016                          BNE     8$              ;YES - SKIP
 4035 023052  005300                          DEC     R0              ;DEC COUNT
 4036 023054  001372                          BNE     7$              ;SKIP IF NOT YET 0
 4037 023056  004737  016052                  JSR     PC,READRL       ;ELSE GET ALL REGISTERS
 4038 023062  004737  016104                  JSR     PC,WAITIN       ;ELSE WAIT FOR TIMEOUT
 4039 023066  012603                          MOV     (SP)+,R3        ;GET RESULT MESSAGE POINTER
 4040 023070                          ERRHRD  10025.,.ERR1
      023070  104456                          TRAP    C$ERHRD
      023072  023451                          .WORD   10025
      023074  000000                          .WORD   0
      023076  011724                          .WORD   ERR1
```

GLOBAL SUBROUTINES

```
4041 023100  005037  003032                CLR     ERRSWI          ;CLEAR FOR ERROR RETURN
4042 023104  000417                        BR      65$
4043 023106  005737  003060          8$:   TST     T.CS            ;TEST FOR ANY ERRORS
4044 023112  100007                        BPL     12$             ;NO   SKIP
4045 023114                                ERRHRD  10026.,,ERR6
     023114  104456                        TRAP    C$ERHRD
     023116  023452                        .WORD   10026
     023120  000000                        .WORD   0
     023122  012226                        .WORD   ERR6
4046 023124  005037  003032                CLR     ERRSWI          ;CLEAR FOR ERROR RETURN
4047 023130  000405                        BR      65$
4048 023132  011423                 12$:   MOV     (R4),(R3)+      ;STORE HEADER WORDS
4049 023134  011423                        MOV     (R4),(R3)+
4050 023136  011423                        MOV     (R4),(R3)+
4051 023140  005301                        DEC     R1              ;DEC HEADER COUNT
4052 023142  001332                        BNE     6$
4053 023144  162737  000002  003016  65$:  SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
4054 023152  012604                        MOV     (SP)+,R4        ;RESTORE REGISTERS
4055 023154  012601                        MOV     (SP)+,R1
4056 023156  012600                        MOV     (SP)+,R0
4057 023160  012603                        MOV     (SP)+,R3
4058 023162  005737  003032                TST     ERRSWI          ;TEST IF ERROR RETURN
4059 023166  001403                        BEQ     99$             ;YES - SKIP
4060 023170  063716  003032                ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
4061 023174  000207                        RTS     PC
4062 023176  017616  000000          99$:  MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
4063 023202  000207                        RTS     PC
4064
4066                                ;       GENERATE DATA ROUTINE. PATTERN TO BE GENERATED IS GIVEN
4067                                ;       IN THE WORD FOLLOWING THE CALL. 128 WORDS ARE GENERATED
4068                                ;       IN OBUFF.
4069 023204  010146                DATGEN: MOV     R1,-(SP)        ;STORE REGISTERS
4070 023206  010346                        MOV     R3,-(SP)
4071 023210  010446                        MOV     R4,-(SP)
4072 023212  012701  004502                MOV     #OBUFF,R1       ;SET POINTER TO OBUFF
4073 023216  012504                        MOV     (R5)+,R4        ;GET DATA PATTERN SELECTOR
4074 023220  006304                        ASL     R4              ;ADJUST IT FOR INDEXING
4075 023222  016403  002374                MOV     PATTBL(R4),R3   ;GET ADDRESS OF PATTERN
4076 023226  011321                        MOV     (R3),(R1)+      ;MOVE FIRST PATTERN WORD
4077 023230  001421                        BEQ     5$              ;SKIP IF PATTERN IS 0
4078 023232  021327  177777                CMP     (R3),#-1        ;CHECK IF PATTERN IS ALL 1'S
4079 023236  001416                        BEQ     5$              ;YES - SKIP
4080 023240  020427  000010                CMP     R4,#8.          ;TEST IF PATTERN 5
4081 023244  001403                        BEQ     3$              ;YES - SKIP
4082 023246  020427  000020                CMP     R4,#16.         ;CHECK IF PATTERN 9 OR 10
4083 023252  002413                        BLT     6$              ;NO - SKIP
4084 023254  005723                 3$:    TST     (R3)+           ;BUMP SOURCE POINTER
4085 023256  012321                        MOV     (R3)+,(R1)+     ;MOVE TWO MORE WORDS FORM SOURCE
4086 023260  012321                        MOV     (R3)+,(R1)+
4087 023262  012704  000015                MOV     #13.,R4         ;SET COUNT
4088 023266  012703  004502                MOV     #OBUFF,R3       ;RESET POINTER
4089 023272  000406                        BR      8$
4090 023274  012703  004502          5$:   MOV     #OBUFF,R3       ;ELSE SET OBUFF AS PATTERN SOURCE
4091 023300  000401                        BR      7$              ;GO TO FILL
4092 023302  005723                 6$:    TST     (R3)+           ;BUMP SOURCE POINTER
4093 023304  012704  000017          7$:   MOV     #15.,R4         ;SET MOVE COUNT
4094 023310  012321                 8$:   MOV     (R3)+,(R1)+     ;MOVE 15 WORDS INTO BUFFER
```

GLOBAL SUBROUTINES

```
4095 023312  005304                          DEC     R4
4096 023314  001375                          BNE     8$
4097 023316  012703  004502                  MOV     #OBUFF,R3       ;SET SOURCE TO TOP OF OBUFF
4098 023322  012704  000160                  MOV     #112.,R4        ;SET COUNT FOR REST OF BUFFER
4099 023326  012321              10$:        MOV     (R3)+,(R1)+     ;REPEAT PATTERN IN BUFFER
4100 023330  005304                          DEC     R4
4101 023332  001375                          BNE     10$
4102 023334  012604                          MOV     (SP)+,R4        ;RESTORE REGISTERS
4103 023336  012603                          MOV     (SP)+,R3
4104 023340  012601                          MOV     (SP)+,R1
4105 023342  000205                          RTS     R5              ;RETURN
4106
4107                             ;           DATA COMPARE ROUTINE. COMPARES THE CONTENTS OF IBUFF AND OBUFF.
4108                             ;           ERROR REPORTING IS LIMITED BY SOFTWARE PARAMETER.
4109 023344  010346              DATCOM:     MOV     R3,-(SP)        ;STORE R3
4110 023346  013703  003016                  MOV     SSINDX,R3       ;GET SUBROUTINE STACK INDEX
4111 023352  005723                          TST     (R3)+           ;BUMP INDEX TO NEXT ENTRY
4112 023354  016663  000002  002420          MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
4113 023362  162763  000004  002420          SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
4114 023370  010337  003016                  MOV     R3,SSINDX       ;STORE IT BACK
4115 023374  010146                          MOV     R1,-(SP)        ;STORE OTHER REGISTERS
4116 023376  010446                          MOV     R4,-(SP)
4117 023400  010546                          MOV     R5,-(SP)
4118 023402  052737  000001  003020          BIS     #DATACMP,OPFLAG ;SET DATA COMPARE FLAG
4119 023410  005037  003030                  CLR     MORECE          ;CLEAR MORE ERROR FLAG
4120 023414  012705  004502                  MOV     #OBUFF,R5       ;SET POINTERS TO DATA FOR COMPARE
4121 023420  012704  004102                  MOV     #IBUFF,R4
4122 023424  012703  000001                  MOV     #1,R3           ;SET WORD COUNTER
4123 023430  012701  000200                  MOV     #128.,R1        ;SET COMPARE COUNT
4124 023434  022425              5$:         CMP     (R4)+,(R5)+     ;COMPARE DATA
4125 023436  001052                          BNE     10$             ;ERROR - SKIP TO REPORT
4126 023440  005203              7$:         INC     R3              ;BUMP WORD COUNT
4127 023442  005301                          DEC     R1              ;DEC COMPARE COUNT
4128 023444  001373                          BNE     5$              ;LOOP IF NOT 0
4129 023446  042737  000001  003020  9$:     BIC     #DATACMP,OPFLAG ;CLEAR DATA COMPARE FLAG
4130 023454  005737  003032                  TST     ERRSWI          ;TEST IF ANY COMPARE ERRORS
4131 023460  001021                          BNE     15$             ;NO - SKIP
4132 023462  012701  000200                  MOV     #128.,R1        ;SET REPORT VALUE
4133 023466                                  PRINTB  #FMT27,#TCERR,MORECE,#RESE6,R1
     023466  010146                          MOV     R1,-(SP)
     023470  012746  010577                  MOV     #RESE6,-(SP)
     023474  013746  003030                  MOV     MORECE,-(SP)
     023500  012746  007624                  MOV     #TCERR,-(SP)
     023504  012746  011673                  MOV     #FMT27,-(SP)
     023510  012746  000005                  MOV     #5,-(SP)
     023514  010600                          MOV     SP,R0
     023516  104414                          TRAP    C$PNTB
     023520  062706  000014                  ADD     #14,SP
4134 023524  162737  000002  003016  15$:    SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
4135 023532  012605                          MOV     (SP)+,R5        ;RESTORE REGS
4136 023534  012604                          MOV     (SP)+,R4
4137 023536  012601                          MOV     (SP)+,R1
4138 023540  012603                          MOV     (SP)+,R3
4139 023542  005737  003032                  TST     ERRSWI          ;TEST IF ERROR RETURN
4140 023546  001403                          BEQ     99$             ;YES - SKIP
4141 023550  063716  003032                  ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
4142 023554  000207                          RTS     PC
```

GLOBAL SUBROUTINES

```
4143 023556  017616  000000          99$:    MOV     @(SP),(SP)        ;SET ERROR RETURN ADDRESS
4144 023562  000207                  RTS     PC
4145 023564  023737  003030  013570  10$:    CMP     MORECE,DCLIMW     ;TEST IF COMPARE ERRORS LIMIT EXCEEDED
4146 023572  002011                  BGE     13$               ;YES - SKIP
4147 023574  024445                  CMP     -(R4), (R5)       ;SET PTRS BACK TO ERROR WORDS
4148 023576                          ERRHRD  10035.,,ERR10     ;REPORT ERROR
     023576  104456                  TRAP    C$ERHRD
     023600  023463                  .WORD   10035
     023602  000000                  .WORD   0
     023604  013320                  .WORD   ERR10
4149 023606  005037  003032          CLR     ERRSWI            ;CLEAR ERROR SWITCH
4150 023612  022425                  CMP     (R4)+,(R5)+       ;BUMP PTRS PAST ERROR WORDS
4151 023614  000711                  BR      7$                ;DO NEXT COMPARE
4152 023616  005237  003030          13$:    INC     MORECE            ;BUMP ERROR COUNTER
4153 023622  000706                  BR      7$                ;DO NEXT COMPARE
4154
4155                                 ;       WRITE AND READ DATA ROUTINE.
4156
4157 023624  012737  177777  003134  XWRITT: MOV     #-1,TEMP1         ;SET SPECIAL WRITE FOR TIMING FLAG
4158 023632  000402                  BR      XWRIT1
4159 023634  005037  003134          XWRITE: CLR     TEMP1             ;CLEAR SPECIAL WRITE FLAG
4160 023640  012737  000112  003150  XWRIT1: MOV     #WTDATA,TEMP7     ;SET FOR WRITE
4161 023646  023737  002316  003120          CMP     HLMTW,CURCYL      ;TEST IF CYLINDER 255 (BAD SEC)
4162 023654  001006                  BNE     1$                ;NO - SKIP
4163 023656  005737  003126          TST     DESHD             ;TEST IF HEAD 1 (BAD SECTOR FILES)
4164 023662  001403                  BEQ     1$                ;NO - SKIP
4165 023664  052737  004000  003020  BIS     #BADADD,OPFLAG    ;SET BAD ADDRESS FLAG
4166 023672  000403                  1$:     BR      XREADG            ;SKIP TO EXECUTE
4167 023674  012737  000114  003150  XREAD:  MOV     #RDDATA,TEMP7     ;SET FOR READ
4168 023702  010346                  XREADG: MOV     R3,-(SP)          ;STORE R3
4169 023704  013703  003016          MOV     SSINDX,R3         ;SET SUBROUTINE INDEX
4170 023710  005723                  TST     (R3)+             ;BUMP TO NEXT STACK ENTRY
4171 023712  016663  000002  002420  MOV     2(SP),SUBSTK(R3)  ;INSERT THIS CALL
4172 023720  162763  000004  002420  SUB     #4,SUBSTK(R3)     ;ADJUST TO POINT TO CALL
4173 023726  010337  003016          MOV     R3,SSINDX         ;STORE IT BACK
4174 023732  010046                  MOV     R0,-(SP)
4175 023734  010146                  MOV     R1,-(SP)          ;STORE OTHER REGISTERS
4176 023736  010446                  MOV     R4,-(SP)
4177 023740  004737  020324          JSR     PC,RDYCHK         ;CHECK IF DRIVE READY
4178 023744  024332                  65$
4179 023746  012703  003050          MOV     #L.CS,R3          ;GET ADDRESS OF LOAD REGS
4180 023752  013713  003150          MOV     TEMP7,(R3)        ;SET COMMAND
4181 023756  053713  003046          BIS     RLDRV,(R3)        ;INSERT DRIVE NUMBER
4182 023762  042713  002000          BIC     #BIT10,(R3)       ;CLEAR FOR DRIVE 4 - 7 SPEC'D
4183 023766  032723  000004          BIT     #BIT2,(R3)+       ;TEST IF WRITE DATA
4184 023772  001403                  BEQ     3$                ;YES - SKIP
4185 023774  012723  004102          MOV     #IBUFF,(R3)+      ;ELSE SET BA FOR READ
4186 024000  000402                  BR      4$
4187 024002  012723  004502          3$:     MOV     #OBUFF,(R3)+      ;SET BA FOR WRITE
4188 024006  013713  003120          4$:     MOV     CURCYL,(R3)       ;GET CURRENT CYLINDER
4189 024012  012704  000007          MOV     #7,R4             ;ALIGN IT IN DA
4190 024016  006313                  5$:     ASL     (R3)
4191 024020  005304                  DEC     R4
4192 024022  001375                  BNE     5$
4193 024024  005737  003126          TST     DESHD             ;TEST IF HEAD 0
4194 024030  001402                  BEQ     7$                ;YES - SKIP
4195 024032  052713  000100          BIS     #HSMSK,(R3)       ;SET FOR HEAD 1
```

GLOBAL SUBROUTINES

```
4196 024036  053723  003130          7$:   BIS     DESSEC,(R3)+    ;INSERT DESIRED SECTOR
4197 024042  012713  177600                MOV     #177600,(R3)    ;INSERT WORD COUNT
4198 024046  005737  003134                TST     TEMP1           ;CHECK IF SPECIAL WRITE FOR TIMING
4199 024052  001402                        BEQ     8$              ;NO   SKIP
4200 024054  012713  177777                MOV     #177777,(R3)    ;ELSE SET FOR 1 WORD TRANSFER
4201 024060  032737  004000  003020  8$:   BIT     #BADADD,OPFLAG  ;TEST IF BAD ADDRESS FLAG SET
4202 024066  001414                        BEQ     2$              ;NO - SKIP
4203 024070  042737  173777  003020        BIC     #^CBADADD,OPFLAG  ;CLEAR ALL BUT THIS FLAG
4204 024076  012703  010501                MOV     #MWRTAB,R3      ;SET RESULT MESSAGE POINTER
4205 024102                                ERRHRD  10032.,,ERR1
     024102  104456                        TRAP    C$ERHRD
     024104  023460                        .WORD   10032
     024106  000000                        .WORD   0
     024110  011724                        .WORD   ERR1
4206 024112  005037  003020                CLR     OPFLAG          ;CLEAR ALL FLAGS
4207 024116  000503                        BR      64$
4208 024120  005037  003022          2$:   CLR     DONE            ;CLEAR INTERRUPT FLAG
4209 024124  005737  003134                TST     TEMP1           ;CHECK IF SPECIAL WRITE FLAG SET
4210 024130  001100                        BNE     65$             ;YES - DO NOT START WRITE
4211 024132  011362  000006                MOV     (R3),RLMP(R2)   ;LOAD RL REGS
4212 024136  014362  000004                MOV     -(R3),RLDA(R2)
4213 024142  014362  000002                MOV     -(R3),RLBA(R2)
4214 024146  014362  000000                MOV     -(R3),RLCS(R2)
4215 024152                         10$:   WAITUS  #3000.          ;WAIT 300MS FOR INTERRUPT
4216 024164  005737  003022                TST     DONE            ;CHECK IF INTERRUPT
4217 024170  001010                        BNE     14$             ;YES - SKIP
4218 024172  004737  016104                JSR     PC,WAITIN       ;WAIT FOR INTERRUPT
4219 024176  012603                        MOV     (SP)+,R3        ;GET RESULT MESSAGE
4220 024200                                ERRHRD  10030.,,ERR1
     024200  104456                        TRAP    C$ERHRD
     024202  023456                        .WORD   10030
     024204  000000                        .WORD   0
     024206  011724                        .WORD   ERR1
4221 024210  000446                        BR      64$
4222 024212  032737  000001  003060  14$:  BIT     #DRDYMSK,T.CS   ;TEST IF DRIVE READY
4223 024220  001033                        BNE     20$             ;YES - SKIP
4224 024222  012703  007760                MOV     #MDRDY,R3       ;SET RESULT MESSAGE
4225 024226  012704  010662                MOV     #CAFDT,R4       ;CONDITION AFTER DATA XFER
4226 024232                                ERRHRD  10032.,,ERR5
     024232  104456                        TRAP    C$ERHRD
     024234  023460                        .WORD   10032
     024236  000000                        .WORD   0
     024240  012156                        .WORD   ERR5
4227 024242  012701  000062                MOV     #50.,R1         ;SET WAIT COUNT FOR 5 SECDS
4228 024246  004737  016310          17$:  JSR     PC,GSTAT        ;GET DRIVE STATUS
4229 024252  024326                        64$
4230 024254  032737  000001  003060        BIT     #DRDYMSK,T.CS   ;TEST IF DRIVE READY NOW
4231 024262  001012                        BNE     20$             ;YES - SKIP
4232 024264  005301                        DEC     R1              ;DEC WAIT COUNT
4233 024266  001367                        BNE     17$             ;LOOP IF NOT TIME DONE
4234 024270  012704  010673                MOV     #C5SEC,R4       ;SET CONDITION 5 SECONDS
4235 024274                                ERRHRD  10033.,,ERR5
     024274  104456                        TRAP    C$ERHRD
     024276  023461                        .WORD   10033
     024300  000000                        .WORD   0
     024302  012156                        .WORD   ERR5
4236 024304  005037  003032                CLR     ERRSWI          ;CLEAR ERROR SWITCH
```

G_OBAL SUBROUTINES

```
4237 024310  005737  003060          20$:   TST     T.CS            ;CHECK IF ANY ERROR
4238 024314  100006                         BPL     65$             ;NO - SKIP
4239 024316                                  ERRHRD  10031.,.ERR6
     024316  104456                         TRAP    C$ERHRD
     024320  023457                          WORD   10031
     024322  000000                         .WORD   0
     024324  012226                         .WORD   ERR6
4240 024326  005037  003032          64$:   CLR     ERRSWI          ;CLEAR ERROR SWITCH
4241 024332  162737  000002  003016  65$:   SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
4242 024340  012604                         MOV     (SP)+,R4        ;RESTORE REGISTERS
4243 024342  012601                         MOV     (SP)+,R1
4244 024344  012600                         MOV     (SP)+,R0
4245 024346  012603                         MOV     (SP)+,R3
4246 024350  005737  003032                 TST     ERRSWI          ;TEST IF ERROR RETURN
4247 024354  001403                         BEQ     99$             ;YES - SKIP
4248 024356  063716  003032                 ADD     ERRSWI,(SP)     ;ELSE ADD IN ERROR RETURN
4249 024362  000207                         RTS     PC
4250 024364  017616  000000          99$:   MOV     @(SP),(SP)      ;ADJUST FOR ERROR RETURN
4251 024370  000207                         RTS     PC
4252
4253                                 ;       BAD SECTOR CHECK ROUTINE. CHECKS IF SECTOR SPECIFIED IN CURCYL,
4254                                 ;       DESHD, AND DESSEC IS LISTED AS BAD IN THE BAD SECTOR FILES.
4255 024372  010046                 BSCHK:  MOV     R0,-(SP)        ;STORE REGISTERS
4256 024374  010146                         MOV     R1,-(SP)
4257 024376  010346                         MOV     R3,-(SP)
4258 024400  005037  003034                 CLR     BSFLAG          ;CLEAR FLAG
4259 024404  012703  003706                 MOV     #FBSFIL,R3      ;GET POINTER TO FACTORY FILE
4260 024410  022713  177777                 CMP     #-1,(R3)        ;CHECK IF ALL ONES
4261 024414  001005                         BNE     4$              ;NO SKIP TO TEST
4262 024416  012703  003512          2$:    MOV     #SBSFIL,R3      ;ELSE SET POINTER TO SOFTWARE FILE
4263 024422  022713  177777                 CMP     #-1,(R3)        ;CHECK IF ALL ONES
4264 024426  001431                         BEQ     20$             ;YES - EXIT
4265 024430  013700  003116          4$:    MOV     NEWCYL,R0       ;BUILD HEADER OF ADDRESS IN QUESTION
4266 024434  012701  000007                 MOV     #7,R1           ;POSITION CYLINDER
4267 024440  006300                 5$:    ASL     R0
4268 024442  005301                         DEC     R1
4269 024444  001375                         BNE     5$
4270 024446  005737  003126                 TST     DESHD           ;CHECK IF HEAD 0
4271 024452  001402                         BEQ     7$              ;YES - SKIP
4272 024454  052700  000100                 BIS     #BIT6,R0        ;INSERT HEAD 1
4273 024460  053700  003130          7$:    BIS     DESSEC,R0       ;INSERT SECTOR
4274 024464  022300                 8$:    CMP     (R3)+,R0        ;CHECK THIS WORD IN FILE
4275 024466  001402                         BEQ     12$             ;YES - EXIT,ERROR
4276 024470  101005                         BHI     15$             ;EXIT- NO ERROR
4277 024472  000774                         BR      8$
4278 024474  012737  000001  003034  12$:   MOV     #1,BSFLAG       ;SET ERROR FLAG
4279 024502  000403                         BR      20$             ;GO TO EXIT
4280 024504  020327  003706          15$:   CMP     R3,#FBSFIL      ;DONE BOTH FILES?
4281 024510  003342                         BGT     2$              ;NO GO DO SOFTWARE FILE
4282 024512  012603                 20$:    MOV     (SP)+,R3        ;ELSE RESTORE REGISTERS
4283 024514  012601                         MOV     (SP)+,R1
4284 024516  012600                         MOV     (SP)+,R0
4285 024520  005737  003034                 TST     BSFLAG          ;CHECK IF ERROR
4286 024524  001003                         BNE     99$             ;YES - SKIP
4287 024526  062716  000002                 ADD     #2,(SP)         ;ELSE BUMP ERROR RETURN
4288 024532  000207                         RTS     PC
4289 024534  017616  000000          99$:   MOV     @(SP),(SP)      ;SET FOR ERROR RETURN
```

GLOBAL SUBROUTINES

```
4290 024540  000207                        RTS     PC
4291
4293                                ;       REPORT OPERATION ROUTINE. PRINTS SUBROUTINE TRACE SEQUENCE AND
4294                                ;       OPERATION BEING PERFORMED PORTION OF ALL
4295                                ;       ERROR MESSAGES.
4296 024542  010446                 RPTOP:  MOV     R4,-(SP)
4297 024544  005737  003016                 TST     SSINDX          ;TEST SUBROUTINE INDEX 0
4298 024550  001433                 BEQ     1$              ;SKIP IF 0
4299 024552  012704  000002         MOV     #2,R4           ;SET INDEXER TO FIRST ENTRY
4300 024556                         PRINTB  #FMT9,#SEQMES   ;PRINT "SUBROUTINE CALL SEQ"
     024556  012746  007514         MOV     #SEQMES,-(SP)
     024562  012746  011212         MOV     #FMT9,-(SP)
     024566  012746  000002         MOV     #2,-(SP)
     024572  010600                 MOV     SP,R0
     024574  104414                 TRAP    C$PNTB
     024576  062706  000006         ADD     #6,SP
4301 024602                 3$:     PRINTB  #FMT16,SUBSTK(R4)       ;PRINT CALLING LOCATION
     024602  016446  002420         MOV     SUBSTK(R4),-(SP)
     024606  012746  011365         MOV     #FMT16,-(SP)
     024612  012746  000002         MOV     #2,-(SP)
     024616  010600                 MOV     SP,R0
     024620  104414                 TRAP    C$PNTB
     024622  062706  000006         ADD     #6,SP
4302 024626  062704  000002         ADD     #2,R4           ;BUMP INDEX
4303 024632  020437  003016         CMP     R4,SSINDX       ;CHECK IF ALL PRINTED
4304 024636  003761                 BLE     3$              ;LOOP IF NOT ALL PRINTED YET
4305 024640                 1$:     PRINTB  #FMT4,ERHEAD,#TSTLAB    ;PRINT ERROR HEADER
     024640  012746  006501         MOV     #TSTLAB, (SP)
     024644  013746  003026         MOV     ERHEAD,-(SP)
     024650  012746  011015         MOV     #FMT4,-(SP)
     024654  012746  000003         MOV     #3,-(SP)
     024660  010600                 MOV     SP,R0
     024662  104414                 TRAP    C$PNTB
     024664  062706  000010         ADD     #10,SP
4306 024670  042737  030000  003020 BIC     #SEEKOP!RORWOP,OPFLAG   ;CLEAR SK & RD OR WRT FLAG
4307 024676  013701  003050         MOV     L.CS,R1         ;GET COMMAND EXECUTED
4308 024702  042701  177741         BIC     #177741,R1      ;STRIP ALL BUT FUNCTION CODE
4309 024706  022701  000006         CMP     #6,R1   ;TEST IF SEEK OPERATION
4310 024712  001003                 BNE     2$              ;NO - SKIP
4311 024714  052737  010000  003020 BIS     #SEEKOP,OPFLAG  ;ELSE SET SEEK FLAG
4312 024722  022701  000012         CMP     #12,R1  2$:     ;TEST IF WRITE
4313 024726  001003                 BNE     20$             ;NO - SKIP
4314 024730  052737  020000  003020 BIS     #RORWOP,OPFLAG  ;SET RD OR WRT FLAG
4315 024736  022701  000014         CMP     #14,R1  20$:    ;TEST IF READ
4316 024742  001003                 BNE     22$             ;NO - SKIP
4317 024744  052737  020000  003020 BIS     #RORWOP,OPFLAG  ;SET RD OR WRT FLAG
4318 024752                 22$:    PRINTB  #FMT1,#MOPER,OPMSGS(R1) ;PRINT OPERATION
     024752  016146  002240         MOV     OPMSGS(R1),-(SP)
     024756  012746  005527         MOV     #MOPER,-(SP)
     024762  012746  010773         MOV     #FMT1,-(SP)
     024766  012746  000003         MOV     #3,-(SP)
     024772  010600                 MOV     SP,R0
     024774  104414                 TRAP    C$PNTB
     024776  062706  000010         ADD     #10,SP
4319 025002  020127  000004         CMP     R1,#4           ;CHECK IF GET STATUS
4320 025006  001007                 BNE     4$              ;NO - SKIP
4321 025010  032737  000010  003054 BIT     #DRSET,L.DA     ;TEST IF RESET INCLUDED
```

GLOBAL SUBROUTINES

```
4322 025016  001403                          BEQ     4$                ;NO - SKIP
4323 025020  012701  000016                  MOV     #16,R1            ;SET TO PRINT WITH RESET
4324 025024  000456                          BR      9$
4325 025026  032737  007777  003020  4$:     BIT     #COMPOP,OPFLAG    ;TEST IF ANY OTHER OPERATION
4326 025034  001424                          BEQ     8$                ;NO - SKIP
4327 025036  013704  003020                  MOV     OPFLAG,R4         ;SET UP TO DETERMINE WHICH ONE
4328 025042  012701  000020                  MOV     #20,R1            ;PRESET THE POINTER
4329 025046  032704  000001          5$:     BIT     #BIT00,R4         ;CHECK THE BIT
4330 025052  001003                          BNE     6$                ;IF SET - SKIP
4331 025054  005721                          TST     (R1)+             ;BUMP POINTER
4332 025056  006204                          ASR     R4
4333 025060  000772                          BR      5$
4334 025062                          6$:     PRINTB  #FMT2,OPMSGS(R1)
     025062  016146  002240                  MOV     OPMSGS(R1),-(SP)
     025066  012746  011007                  MOV     #FMT2,-(SP)
     025072  012746  000002                  MOV     #2,-(SP)
     025076  010600                          MOV     SP,R0
     025100  104414                          TRAP    C$PNTB
     025102  062706  000006                  ADD     #6,SP
4335 025106  032737  100000  003020  8$:     BIT     #HDR40,OPFLAG     ;TEST IF 40 HEADER OPERATION
4336 025114  001415                          BEQ     10$               ;NO - SKIP
4337 025116  012701  000050                  MOV     #50,R1            ;ELSE PRINT IT
4338 025122                          9$:     PRINTB  #FMT2,OPMSGS(R1)
     025122  016146  002240                  MOV     OPMSGS(R1),-(SP)
     025126  012746  011007                  MOV     #FMT2,-(SP)
     025132  012746  000002                  MOV     #2,-(SP)
     025136  010600                          MOV     SP,R0
     025140  104414                          TRAP    C$PNTB
     025142  062706  000006                  ADD     #6,SP
4339 025146  000434                          BR      15$               ;SKIP
4340 025150  032737  010000  003020  10$:    BIT     #SEEKOP,OPFLAG    ;TEST IF SEEK
4341 025156  001430                          BEQ     15$               ;NO - SKIP
4342 025160                                  PRINTB  #FMT13,#FRMWD,OLDCYL,#DIFWD,DESDIF,#SGNWD,DESSGN,#HDWD,DESHD
     025160  013746  003126                  MOV     DESHD,-(SP)
     025164  012746  007455                  MOV     #HDWD,-(SP)
     025170  013746  003124                  MOV     DESSGN,-(SP)
     025174  012746  007450                  MOV     #SGNWD,-(SP)
     025200  013746  003122                  MOV     DESDIF,-(SP)
     025204  012746  007442                  MOV     #DIFWD,-(SP)
     025210  013746  003114                  MOV     OLDCYL,-(SP)
     025214  012746  007473                  MOV     #FRMWD,-(SP)
     025220  012746  011233                  MOV     #FMT13,-(SP)
     025224  012746  000011                  MOV     #11,-(SP)
     025230  010600                          MOV     SP,R0
     025232  104414                          TRAP    C$PNTB
     025234  062706  000024                  ADD     #24,SP
4343 025240  032737  020000  003020  15$:    BIT     #RORWOP,OPFLAG    ;TEST IF READ OR WRITE SET
4344 025246  001424                          BEQ     17$               ;NO - SKIP
4345 025250                                  PRINTB  #FMT22,#CYLWD,CURCYL,#HDWD,DESHD,#SECWD,DESSEC
     025250  013746  003130                  MOV     DESSEC,-(SP)
     025254  012746  007461                  MOV     #SECWD,-(SP)
     025260  013746  003126                  MOV     DESHD,-(SP)
     025264  012746  007455                  MOV     #HDWD,-(SP)
     025270  013746  003120                  MOV     CURCYL,-(SP)
     025274  012746  007466                  MOV     #CYLWD,-(SP)
     025300  012746  011562                  MOV     #FMT22,-(SP)
     025304  012746  000007                  MOV     #7,-(SP)
```

GLOBAL SUBROUTINES

```
                025310  010600                  MOV     SP,R0
                025312  104414                  TRAP    C$PNTB
                025314  062706  000020          ADD     #20,SP
 4346           025320  004737  025772  17$:    JSR     PC,CLRPARM      ;CLEAR PARAM TABLE
 4347           025324  012604                  MOV     (SP)+,R4        ;RESTORE R4
 4348           025326  000207                  RTS     PC
 4349
 4350                                   ;       REPORT REASON ROUTINE
 4351                                   ;       PRINTS REASON PORTION FOR ALL ERROR REPORTS.
 4352           025330  010146          RPTRES: MOV     R1,-(SP)        ;STORE R1
 4353           025332  010346                  MOV     R3,-(SP)        ;STORE R3
 4354           025334  010446                  MOV     R4,-(SP)        ;STORE R4
 4355           025336  012701  003076          MOV     #RESPARM,R1     ;GET START OF PARAM
 4356           025342  012103                  MOV     (R1)+,R3        ;GET NUMBER OF PARAM
 4357           025344                          PRINTB  #FMT1.1,#MRSLT,(R1)     ;PRINT NAME
                025344  011146                  MOV     (R1),-(SP)
                025346  012746  005536          MOV     #MRSLT,-(SP)
                025352  012746  011000          MOV     #FMT1.1,-(SP)
                025356  012746  000003          MOV     #3,-(SP)
                025362  010600                  MOV     SP,R0
                025364  104414                  TRAP    C$PNTB
                025366  062706  000010          ADD     #10,SP
 4358           025372  021127  010352          CMP     (R1),#MNDRST    ;TEST IF MESSAGE IS NO DRV STATUS
 4359           025376  001453                  BEQ     6$              ;YES - SKIP REST OF REPORT
 4360           025400  012704  011217          MOV     #FMT11,R4       ;PRISET FOR FORMAT 11
 4361           025404  022127  010345          CMP     (R1)+,#MCYLOC   ;CHECK IF REPORTING CYLINDER LOC
 4362           025410  001002                  BNE     3$              ;NO - SKIP
 4363           025412  012704  011225          MOV     #FMT12,R4       ;ELSE CHANGE TO FORMAT 12
 4364           025416  005303          3$:     DEC     R3              ;DEC PARAM COUNT
 4365           025420  001442                  BEQ     6$              ;IF 0 - EXIT
 4366           025422                          PRINTB  R4,#RESE3,(R1)+  ;REPORT IS VALUE
                025422  012146                  MOV     (R1)+,-(SP)
                025424  012746  010561          MOV     #RESE3,-(SP)
                025430  010446                  MOV     R4,-(SP)
                025432  012746  000003          MOV     #3,-(SP)
                025436  010600                  MOV     SP,R0
                025440  104414                  TRAP    C$PNTB
                025442  062706  000010          ADD     #10,SP
 4367           025446                          PRINTB  R4,#RESE4,(R1)+  ;REPORT SB VALUE
                025446  012146                  MOV     (R1)+,-(SP)
                025450  012746  010565          MOV     #RESE4,-(SP)
                025454  010446                  MOV     R4,-(SP)
                025456  012746  000003          MOV     #3,-(SP)
                025462  010600                  MOV     SP,R0
                025464  104414                  TRAP    C$PNTB
                025466  062706  000010          ADD     #10,SP
 4368           025472  162703  000002          SUB     #2,R3           ;DEC PARAM COUNT
 4369           025476  001413                  BEQ     6$              ;IF 0 - EXIT
 4370           025500                          PRINTB  #FMT1,#RESE5,(R1)+  ;REPORT CONDITION
                025500  012146                  MOV     (R1)+,-(SP)
                025502  012746  010572          MOV     #RESE5,-(SP)
                025506  012746  010773          MOV     #FMT1,-(SP)
                025512  012746  000003          MOV     #3,-(SP)
                025516  010600                  MOV     SP,R0
                025520  104414                  TRAP    C$PNTB
                025522  062706  000010          ADD     #10,SP
 4371           025526  012604          6$:     MOV     (SP)+,R4        ;RESTORE REGS
```

GLOBAL SUBROUTINES

```
                                        MOV     (SP)+,R3
                                        MOV     (SP)+,R1
                                        RTS     PC                      ;RETURN
                                ;       REPORT PHYSICAL ADDRESS OF DEVICE UNDER TEST
                                ;       AND ALL REGISTER CONTENTS.
                                RPTREM: PRINTB  #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
025536  005046                          CLR      (SP)
025540  153716  003047                  BISB    RLDRV+1,(SP)
025544  012746  006152                  MOV     #DRVNAM, (SP)
025550  013746  003042                  MOV     RLBAS,-(SP)
025554  012746  006141                  MOV     #BASADD,-(SP)
025560  012746  011026                  MOV     #FMT5,-(SP)
025564  012746  000005                  MOV     #5,-(SP)
025570  010600                          MOV     SP,R0
025572  104414                          TRAP    C$PNTB
025574  062706  000014                  ADD     #14,SP
                                ;       REPORT RL11 REGISTERS
025600                                  PRINTB  #FMT6,#CSNAM,#DANAM,#BANAM,#MPNAM,#CYLWD,#HDWD
025600  012746  007455                  MOV     #HDWD,-(SP)
025604  012746  007466                  MOV     #CYLWD,-(SP)
025610  012746  006255                  MOV     #MPNAM,-(SP)
025614  012746  006243                  MOV     #BANAM,-(SP)
025620  012746  006250                  MOV     #DANAM,-(SP)
025624  012746  006236                  MOV     #CSNAM,-(SP)
025630  012746  011046                  MOV     #FMT6,-(SP)
025634  012746  000007                  MOV     #7, (SP)
025640  010600                          MOV     SP,R0
025642  104414                          TRAP    C$PNTB
025644  062706  000020                  ADD     #20,SP
025650                                  PRINTB  #FMT8,#LAB1,L.CS,L.DA,L.BA,L.MP
025650  013746  003056                  MOV     L.MP, (SP)
025654  013746  003052                  MOV     L.BA, (SP)
025660  013746  003054                  MOV     L.DA,-(SP)
025664  013746  003050                  MOV     L.CS,-(SP)
025670  012746  006262                  MOV     #LAB1,-(SP)
025674  012746  011160                  MOV     #FMT8,-(SP)
025700  012746  000006                  MOV     #6,-(SP)
025704  010600                          MOV     SP,R0
025706  104414                          TRAP    C$PNTB
025710  062706  000016                  ADD     #16,SP
025714                                  PRINTB  #FMT7,#LAB2,T.CS,T.DA,T.BA,T.MP,CURCYL,DESHD
025714  013746  003126                  MOV     DESHD,-(SP)
025720  013746  003120                  MOV     CURCYL,-(SP)
025724  013746  003066                  MOV     T.MP,-(SP)
025730  013746  003062                  MOV     T.BA,-(SP)
025734  013746  003064                  MOV     T.DA,-(SP)
025740  013746  003060                  MOV     T.CS,-(SP)
025744  012746  006275                  MOV     #LAB2,     )
025750  012746  011110                  MOV     #FMT7,-(SP)
025754  012746  000010                  MOV     #10,-(SP)
025760  010600                          MOV     SP,R0
025762  104414                          TRAP    C$PNTB
025764  062706  000022                  ADD     #22,SP
025770  000207                          RTS     PC

                                ;       CLEAR PARAMETER BLOCK FOR REPORTING
```

GLOBAL SUBROUTINES

```
4386 025772  010546                    CLRPARM:    MOV     R5, (SP)        ;STORE R5
4387 025774  012701  003076                        MOV     #RESPARM,R1     ;GET ADDRESS OF BLOCK
4388 026000  012705  000005                        MOV     #5,R5           ;SET COUNT
4389 026004  005021                    2$:         CLR     (R1)+           ;CLEAR WORD
4390 026006  005305                                DEC     R5              ;DEC COUNT
4391 026010  001375                                BNE     2$              ;LOOP UNTIL 0
4392 026012  012701  003076                        MOV     #RESPARM,R1     ;RESET POINTER
4393 026016  012605                                MOV     (SP)+,R5        ;RESTORE R5
4394 026020  000207                                RTS     PC
4395
4396 026022                            ENDMOD
4397
4398                                   .TITLE  CNRLJA0 RL01/02 DRIVE TEST 2
4399
4400                                   .SBTTL  *TEST 1         **OUTER GUARD BAND DETECTION
4401
4402 026022                            BGNMOD  HRDWTST
4403 026022                            BGNTST                  ;TEST 1
     026022                                                                    T1::
4404
4405 026022  012737  006507  003026            MOV     #P2T03E,ERHEAD  ;SET ERROR HEADER
4406 026030  004737  016242                    JSR     PC,TSTINT       ;INITIALIZE TEST
4407 026034  004737  016260                    JSR     PC,GSTATR       ;CLEAR DRIVE
4408 026040  026242                            T1965$
4409 026042  004737  020600                    JSR     PC,CHOSHD       ;GO CHOSE HEAD
4410 026046  005005                T197$:      CLR     R5              ;CLEAR FOR POSITION TO 0
4411 026050  004737  017760                    JSR     PC,POSHDS       ;POSITION HEADS
4412 026054  026242                            T1965$
4413 026056                            BGNSUB
     026056                                                                    T1.1:
     026056  104402                            TRAP    C$BSUB
4414 026060  012737  177777  003116            MOV     #-1,NEWCYL      ;SET FOR GUARD BAND SEEK
4415 026066  004737  017206                    JSR     PC,XSEEK        ;DO SEEK
4416 026072  026216                            60$
4417 026074  012701  000002                    MOV     #2,R1           ;INITIALIZE WAIT COUNT
4418 026100  032762  000001  000000  8$:       BIT     #DRDYMSK,RLCS(R2) ;TEST IF DRIVE READY
4419 026106  001414                            BEQ     9$              ;NO-SKIP
4420 026110  004737  016310                    JSR     PC,GSTAT        ;GET DRIVE STATUS
4421 026114  026216                            60$
4422 026116  012703  007760                    MOV     #MDRDY,R3       ;SET NAME MESSAGE PTR
4423 026122  012704  010634                    MOV     #C10MS,R4       ;SET CONDITION MESSAGE PTR
4424 026126                                    ERRHRD  301...ERR4      ;REPORT READY ERROR
     026126  104456                            TRAP    C$ERHRD
     026130  000455                            .WORD   301
     026132  000000                            .WORD   0
     026134  012106                            .WORD   ERR4
4425 026136  000427                            BR      60$             ;EXIT TEST
4426 026140  005301                9$:         DEC     R1              ;DEC WAIT COUNT
4427 026142  001406                            BEQ     12$             ;SKIP IF 0
4428 026144                                    WAITUS  #10.            ;WAIT 1MS
4429 026156  000750                            BR      8$              ;LOOP
4430 026160  012701  000226        12$:        MOV     #150.,R1        ;SET WAIT COUNT FOR 15 MS
4431 026164  004737  022102                    JSR     PC,RDYWAIT      ;WAIT FOR READY & REPORT IF NOT READY
4432 026170  026216                            60$
4433
4434 026172  004737  022366                    JSR     PC,GETPOS       ;GET POSITION
4435 026176  026216                            60$
```

```
*TEST 1          **OUTER GUARD BAND DETECTION

 4436 026200 005737 003120                  TST     CURCYL          ;CHECK IF HEADS STILL AT 0
 4437 026204 001404                         BEQ     15$             ;YES-SKIP
 4438 026206                                ERRHRD  302...ERR8      ;ELSE REPORT CYLINDER ERROR
      026206 104456                         TRAP    C$ERHRD
      026210 000456                         .WORD   302
      026212 000000                         .WORD   0
      026214 013160                         .WORD   ERR8
 4439 026216                          15$:
 4440 026216 012737 000002 003032     60$:  MOV     #2,ERRSWI       ;INIT ERROR SWITCH
 4441 026224                          ENDSUB
      026224                          L10024:
      026224 104403                         TRAP    C$ESUB
 4442 026226                                ESCAPE  TST             ;EXIT TEST IF ERROR
      026226 104410                         TRAP    C$ESCAPE
      026230 000012                         .WORD   L10023 .
 4443 026232 004737 020624                  JSR     PC,SWAPHD       ;GO SWAP TO HEAD 1 OR END TEST
 4444 026236 026242                         17$                     ;ABORT RETURN
 4445 026240 000702                         BR      T197$           ;REDO TEST
 4446 026242                          17$:
 4447 026242                          T1965$:
 4448
 4449 026242                          ENDTST
      026242                          L10023:
      026242 104401                         TRAP    C$ETST
 4450
 4451                                 .SBTTL  *TEST 2         **INCREMENTAL FORWARD SEEK HEAD 0
 4452 026244                          BGNTST                  ;TEST 2
      026244                                                                          T2::
 4453 026244 012737 006525 003026           MOV     #P2T04E,ERHEAD  ;SET ERROR HEADER
 4454 026252 004737 016242                  JSR     PC,TSTINT       ;INITIALIZE TEST
 4455 026256 004737 016260                  JSR     PC,GSTATR       ;CLEAR DRIVE
 4456 026262 026452                         T2065$
 4457 026264 004737 020600                  JSR     PC,CHOSHD       ;GO CHOSE HEAD
 4458 026270 005737 003126                  TST     DESHD           ;TEST IF THIS IS HEAD 0
 4459 026274 001402                         BEQ     2$              ;YES - SKIP
 4460 026276                                EXIT    TST             ;ELSE EXIT TEST
      026276 104432                         TRAP    C$EXIT
      026300 000152                         .WORD   L10025-.
 4461 026302 013705 013560            2$:   MOV     LOLIMW,R5       ;CLEAR TO POSITION HEADS TO LOLIMIT
 4462 026306 004737 017760                  JSR     PC,POSHDS       ;POSITION HEADS
 4463 026312 026452                         T2065$
 4464 026314                          BGNSUB
      026314                                                                          T2.1:
      026314 104402                         TRAP    C$BSUB
 4465 026316 004737 022366            T206$: JSR     PC,GETPOS       ;GET POSITION
 4466 026322 026442                         60$
 4467 026324                                INLOOP                  ;CHECK IF IN ERROR LOOP
      026324 104420                         TRAP    C$INLP
 4468 026326                                BNCOMPLETE      5$      ;NO - SKIP
      026326 103007                         BCC     5$
 4469 026330 023737 003120 003116           CMP     CURCYL,NEWCYL   ;CHECK IF POSITIONED AT DESIRED LOC
 4470 026336 001003                         BNE     5$              ;NO - SKIP
 4471 026340 004737 020664                  JSR     PC,ONSWAP       ;ELSE SWAP NEW AND OLD CYLINDERS
 4472 026344 000405                         BR      7$              ;SKIP
 4473 026346 013737 003120 003116     5$:   MOV     CURCYL,NEWCYL   ;PLACE CURRENT INTO NEW
 4474 026354 005237 003116                  INC     NEWCYL          ;BUMP FOR ONE CYLINDER SEEK
 4475 026360                          7$:
```

```
.TEST 2          **INCREMENTAL FORWARD SEEK HEAD 0

4476 026360  004737  017206                    JSR     PC,XSEEK        ;DO SEEK
4477 026364  026442                            60$
4478 026366  012701  000226                    MOV     #150.,R1        ;SET WAIT TIME 15 MS
4479 026372  004737  022102                    JSR     PC,RDYWAIT      ;WAIT FOR READY
4480 026376  026442                            60$
4481
4482 026400  004737  022514                    JSR     PC,VERPOS       ;GO VERIFY POSITON
4483 026404  026442                            60$
4484
4485 026406  032737  000002  013556            BIT     #ALLSEC,MISWIW  ;TEST IF CHECK ALL SECTORS
4486 026414  001406                            BEQ     11$             ;NO SKIP
4487 026416  004737  022636                    JSR     PC,RDALHD       ;GO READ ALL HEADERS
4488 026422  026442                            60$
4489 026424  004737  021506                    JSR     PC,VERHDR       ;GO VERIFY HEADFR
4490 026430  026442                            60$
4491 026432                          11$:
4492 026432  023737  013562  003116            CMP     HILIMW,NEWCYL   ;CHECK IF HILIMIT REACHED
4493 026440  103726                            BLO     T206$           ;NO-LOOP
4494 026442  012737  000002  003032  60$:      MOV     #2,ERRSWI       ;INIT ERROR SWITCH
4495 026450                          ENDSUB
     026450                          L10026:
     026450  104403                            TRAP    C$ESUB
4496 026452                          T2065$:
4497 026452                          ENDTST
     026452                          L10025:
     026452  104401                            TRAP    C$ETST
4498
4499                                 .SBTTL  *TEST 3         **INCREMENTAL REVERSE SEEK HEAD 0
4500 026454                          BGNTST                  ;TEST 3
     026454                                                                          T3::
4501 026454  012737  006545  003026            MOV     #P2TOSE,ERHEAD  ;SET ERROR HEADER
4502 026462  004737  016242                    JSR     PC,TSTINT       ;INITIALIZE TEST
4503 026466  004737  016260                    JSR     PC,GSTATR       ;CLEAR DRIVE
4504 026472  026662                            T2165$
4505 026474  004737  020600                    JSR     PC,CHOSHD       ;GO CHOSE HEAD
4506 026500  005737  003126                    TST     DESHD           ;TEST IF HEAD 0 SELECTED
4507 026504  001402                            BEQ     2$              ;YES - SKIP
4508 026506                          EXIT    TST                     ;ELSE EXIT TEST
     026506  104432                            TRAP    C$EXIT
     026510  000152                            .WORD   L10027-.
4509 026512  013705  013562          2$:       MOV     HILIMW,R5       ;SET TO POSITION HDS TO HILIMIT
4510 026516  004737  017760                    JSR     PC,POSHDS       ;POSITION HEADS
4511 026522  026662                            T2165$
4512 026524                          BGNSUB
     026524                                                                          T3.1:
     026524  104402                            TRAP    C$BSUB
4513 026526  004737  022366          T216$:    JSR     PC,GETPOS       ;GET POSITION
4514 026532  026652                            60$
4515 026534                                    INLOOP                  ;CHECK IF IN ERROR LOOP
     026534  104420                            TRAP    C$INLP
4516 026536                                    BNCOMPLETE      5$      ;NO - SKIP
     026536  103007                            BCC     5$
4517 026540  023737  003120  003116            CMP     CURCYL,NEWCYL   ;CHECK IF POSITIONED AT DES LOC
4518 026546  001003                            BNE     5$              ;NO - SKIP
4519 026550  004737  020664                    JSR     PC,ONSWAP       ;ELSE SWAP OLD AND NEW CYLINDERS
4520 026554  000405                            BR      7$              ;SKIP
4521 026556  013737  003120  003116  5$:       MOV     CURCYL,NEWCYL   ;PUT CURRENT INTO NEW
```

```
*TEST 3          **INCREMENTAL REVERSE SEEK HEAD 0

4522 026564 005337 003116              DEC     NEWCYL          ;DEC FOR ONE CYLINDER REVERSE SEEK
4523 026570 004737 017206      7$:     JSR     PC,XSEEK        ;SEEK TO IT
4524 026574 026652                     60$
4525 026576 012701 000226              MOV     #150.,R1        ;SET WAIT FOR 15 MS
4526 026602 004737 022102              JSR     PC,RDYWAIT      ;WAIT FOR READY
4527 026606 026652                     60$
4528
4529 026610 004737 022514              JSR     PC,VERPOS       ;VERIFY POSITION
4530 026614 026652                     60$
4531
4532 026616 032737 000002 003020       BIT     #ALLSEC,OPFLAG  ;TEST IF USE ALL SECTORS
4533 026624 001406                     BEQ     11$             ;NO-SKIP
4534 026626 004737 022636              JSR     PC,RDALHD       ;ELSE READ ALL THE HDRS
4535 026632 026652                     60$
4536 026634 004737 021506              JSR     PC,VERHDR       ;VERIFY THE HEADERS
4537 026640 026652                     60$
4538 026642                    11$:
4539 026642 023737 013560 003116       CMP     LOLIMW,NEWCYL   ;CHECK IF REACHED LOLIMIT
4540 026650 103726                     BLO     T216$           ;NO - LOOP
4541 026652 012737 000002 003032 60$:  MOV     #2,ERRSWI       ;INIT ERROR SWITCH
4542 026660                    ENDSUB
     026660                    L10030:
     026660 104403                     TRAP    C$ESUB
4543 026662                    T2165$:
4544 026662                    ENDTST
     026662                    L10027:
     026662 104401                     TRAP    C$ETST
4545
4546                           .SBTTL  *TEST 4          **INCREMENTAL FORWARD SEEK HEAD 1
4547 026664                    BGNTST                   ;TEST 4
     026664                                                                  T4::
4548 026664 012737 006565 003026       MOV     #P2T06E,ERHEAD  ;SET ERROR HEADER
4549 026672 004737 016242              JSR     PC,TSTINT       ;INITIALIZE TEST
4550 026676 004737 016260              JSR     PC,GSTATR       ;CLEAR DRIVE
4551 026702 027106                     T2265$
4552 026704 005037 003126              CLR     DESHD           ;SET HEAD TO 0
4553 026710 013705 013560              MOV     LOLIMW,R5       ;CLEAR FOR POSITION HDS TO LOLIMIT
4554 026714 004737 017760              JSR     PC,POSHDS       ;POSITION HDS
4555 026720 027106                     T2265$
4556 026722 012737 000001 003126       MOV     #1,DESHD        ;SET TO HEAD 1
4557 026730 032737 010000 013556       BIT     #HEADLM,MISWIW  ;TEST IF HEAD SPECIFIED
4558 026736 001405                     BEQ     2$              ;NO - SKIP
4559 026740 005737 013564              TST     HEADW           ;TEST IF IT IS HEAD 0
4560 026744 001002                     BNE     2$              ;NO - SKIP
4561 026746                    EXIT    TST                     ;ELSE EXIT TEST
     026746 104432                     TRAP    C$EXIT
     026750 000136                     .WORD   L10031-.
4562 026752                    2$:
4563 026752                    BGNSUB
     026752                                                                  T4.1:
     026752 104402                     TRAP    C$BSUB
4564 026754 004737 022366      T227$:  JSR     PC,GETPOS       ;GET CURRENT POSITION
4565 026760                            INLOOP                  ;CHECK IF IN ERROR LOOP
     026760 104420                     TRAP    C$INLP
4566 026762                            BNCOMPLETE 5$           ;NO - SKIP
     026762 103007                     BCC     5$
4567 026764 023737 003120 003116       CMP     CURCYL,NEWCYL   ;CHECK IF AT DESIRED LOCATION
```

```
*TEST 4          **INCREMENTAL FORWARD SEEK HEAD 1

 4568 026772 001003                        BNE     5$              ;NO   SKIP
 4569 026774 004737 020664                 JSR     PC,ONSWAP       ;SWAP OLD AND NEW CYLINDER
 4570 027000 000405                        BR      7$              ;SKIP
 4571 027002 013737 003120 003116  5$:     MOV     CURCYL,NEWCYL   ;MOVE CURRENT INTO NEW
 4572 027010 005237 003116                 INC     NEWCYL          ;BUMP NEWCYL FOR ONE CYL FWRD SEEK
 4573 027014                       7$:
 4574 027014 004737 017206                 JSR     PC,XSEEK        ;DO SEEK
 4575 027020 027076                        60$
 4576 027022 012701 000226                 MOV     #150.,R1        ;SET WAIT COUNT 15 MS
 4577 027026 004737 022102                 JSR     PC,RDYWAIT      ;WAIT FOR READY
 4578 027032 027076                        60$
 4579 027034 004737 022514                 JSR     PC,VERPOS       ;VERIFY POSITION IS CORRECT
 4580 027040 027076                        60$
 4581
 4582 027042 032737 000002 013556          BIT     #ALLSEC,MISWIW  ;CHECK IF USE ALL SECTORS
 4583 027050 001406                        BEQ     9$              ;NO-SKIP
 4584 027052 004737 022636                 JSR     PC,RDALHD       ;ELSE READ ALL HEADERS
 4585 027056 027076                        60$
 4586 027060 004737 021506                 JSR     PC,VERHDR       ;VERIFY HEADERS
 4587 027064 027076                        60$
 4588 027066                       9$:
 4589 027066 023737 013562 003116          CMP     MILIMW,NEWCYL   ;CHECK IF DONE
 4590 027074 101327                        BHI     T227$           ;NO   LOOP
 4591 027076 012737 000002 003032  60$:     MOV     #2,ERRSWI       ;INIT ERROR SWITCH
 4592 027104                       ENDSUB
      027104                       L10032:
      027104 104403                        TRAP    C$ESUB
 4593 027106                       T2265$:
 4594 027106                       ENDTST
      027106                       L10031:
      027106 104401                        TRAP    C$ETST
 4595
 4596                              .SBTTL  *TEST 5         **INNER GUARD BAND DETECTION
 4597
 4598 027110                       BGNTST                  ;TEST 5
      027110                                                                        T5::
 4599
 4600 027110 012737 006605 003026          MOV     #P2T07E,ERHEAD  ;SET ERROR HEADER
 4601 027116 004737 016242                 JSR     PC,TSTINT       ;INITIALIZE TEST
 4602 027122 004737 016260                 JSR     PC,GSTATR       ;CLEAR DRIVE
 4603 027126 027314                        T2365$
 4604 027130 004737 020600                 JSR     PC,CHOSHD       ;GO CHOSE HEAD
 4605 027134 013705 002316         T233$:  MOV     HLMTW,R5        ;SET FOR POSITION TO 255.
 4606 027140 004737 017760                 JSR     PC,POSHDS       ;POSITION HEADS
 4607 027144 027314                        T2365$
 4608 027146                       BGNSUB
      027146                                                                        T5.1:
      027146 104402                        TRAP    C$BSUB
 4609 027150 013737 002324 003116          MOV     GBND,NEWCYL     ;SET FOR INNER GUARD BAND SEEK
 4610 027156 004737 017206                 JSR     PC,XSEEK        ;DO IT
 4611 027162 027270                        60$
 4612 027164 012701 000001                 MOV     #1.,R1          ;INITIALIZE WAIT COUNT
 4613 027170 032762 000001 000000  7$:     BIT     #DRDYMSK,RLCS(R2) ;CHECK IF READY
 4614 027176 001414                        BEQ     9$              ;NO-SKIP
 4615 027200 004737 016310                 JSR     PC,GSTAT        ;GET DRIVE STATUS
 4616 027204 027270                        60$
 4617 027206 012703 007760                 MOV     #MDRDY,R3       ;SET NAME MESSAGE PTR
```

```
*TEST 5          **INNER GUARD BAND DETECTION

4618 027212 012704 010634                MOV     #C10MS,R4       ;SET CONDITION MESSAGE PTR
4619 027216                              ERRHRD  701.,,ERR4      ;REPORT READY ERROR
     027216 104456                       TRAP    C$ERHRD
     027220 001275                       .WORD   701
     027222 000000                       .WORD   0
     027224 012106                       .WORD   ERR4
4620 027226 000420                       BR      60$             ;EXIT TEST
4621 027230 005301                9$:    DEC     R1              ;DEC WAIT COUNT
4622 027232 001406                       BEQ     11$             ;SKIP IF 0
4623 027234                              WAITUS  #10.            ;WAIT 1MS
4624 027246 000750                       BR      7$              ;LOOP
4625 027250 012701 000226        11$:    MOV     #150.,R1        ;SET WAIT COUNT 15 MS
4626 027254 004737 022102                JSR     PC,RDYWAIT      ;GO WAIT FOR READY
4627 027260 027270                       60$
4628
4629 027262 004737 022514                JSR     PC,VERPOS       ;GO VERIFY POSITION IS 255
4630 027266 027270                       60$
4631 027270 012737 000002 003032 60$:    MOV     #2,ERRSWI       ;INIT ERROR SWITCH
4632 027276                              ENDSUB
     027276                              L10034:
     027276 104403                               TRAP    C$ESUB
4633 027300                                      ESCAPE  TST             ;EXIT TEST IF ERROR
     027300 104410                               TRAP    C$ESCAPE
     027302 000012                               .WORD   L10033-.
4634 027304 004737 020624                        JSR     PC,SWAPHD       ;GO SWAP TO HEAD 1 OR END TEST
4635 027310 027314                               15$                     ;ABORT RETURN
4636 027312 000710                               BR      T233$           ;REPEAT THE TESTS
4637 027314                              15$:
4638 027314                              T2365$:
4639
4640 027314                              ENDTST
     027314                              L10033:
     027314 104401                               TRAP    C$ETST
4641
4642                                     .SBTTL  *TEST 6         **INCREMENTAL REVERSE SEEK HEAD 1
4643 027316                              BGNTST                  ;TEST 6
     027316                                                                                  T6::
4644 027316 012737 006623 003026                 MOV     #P2TO8E,ERHEAD  ;SET ERROR HEADER
4645 027324 004737 016242                        JSR     PC,TSTINT       ;INITIALIZE TEST
4646 027330 004737 016260                        JSR     PC,GSTATR       ;GET STATUS & CLEAR
4647 027334 027542                               T2465$
4648 027336 005037 003126                        CLR     DESHD           ;SET TO HEAD 0
4649 027342 013705 013562                        MOV     MILIMW,R5       ;SET TO POSITION HDS AT MILIMIT
4650 027346 004737 017760                        JSR     PC,POSHDS       ;POSITION HDS
4651 027352 027542                               T2465$
4652 027354 012737 000001 003126                 MOV     #1,DESHD        ;SET TO SELECT HD 1
4653 027362 032737 010000 013556                 BIT     #HEADLM,MISWIW  ;TEST IF HEAD SPECIFIED
4654 027370 001405                               BEQ     2$              ;NO - SKIP
4655 027372 005737 013564                        TST     HEADW           ;TEST IF HEAD SPECIFIED IS 0
4656 027376 001002                               BNE     2$              ;NO - SKIP
4657 027400                                      EXIT    TST             ;ESLE EXIT TEST
     027400 104432                               TRAP    C$EXIT
     027402 000140                               .WORD   L10035 .
4658 027404                              2$:
4659 027404                              BGNSUB
     027404                                                                                  T6.1:
     027404 104402                               TRAP    C$BSUB
```

```
*TEST 6          **INCREMENTAL REVERSE SEEK HEAD 1

4660 027406  004737  022366          T247$:  JSR     PC,GETPOS       ;GET CURRENT POSITION
4661 027412  027532                          60$
4662 027414                                  INLOOP                  ;CHECK IF IN ERROR LOOP
     027414  104420                          TRAP    C$INLP
4663 027416                                  BNCOMPLETE      5$      ;NO - SKIP
     027416  103007                          BCC     5$
4664 027420  023737  003120  003116          CMP     CURCYL,NEWCYL   ;CHECK IF POSITIONED AT DESIRED LOC
4665 027426  001003                          BNE     5$              ;NO - SKIP
4666 027430  004737  020664                  JSR     PC,ONSWAP       ;ELSE SWAP OLD AND NEW CYLINDER
4667 027434  000405                          BR      7$              ;SKIP
4668 027436  013737  003120  003116  5$:     MOV     CURCYL,NEWCYL   ;MOV CUR TO NEW
4669 027444  005337  003116                  DEC     NEWCYL          ;DEC NEWCYL FOR 1 CYL REV SEEK
4670 027450  004737  017206          7$:     JSR     PC,XSEEK        ;DO SEEK
4671 027454  027532                          60$
4672 027456  012701  000226                  MOV     #150.,R1                ;SET WAIT FOR 15 MS
4673 027462  004737  022102                  JSR     PC,RDYWAIT      ;WAIT FOR READY
4674 027466  027532                          60$
4675 027470  004737  022514                  JSR     PC,VERPOS       ;VERIFY POSITION
4676 027474  027532                          60$
4677 027476  032737  000002  013556          BIT     #ALLSEC,MISWIW  ;TEST IF ALL SECTORS
4678 027504  001406                          BEQ     9$              ;NO-EXIT
4679 027506  004737  022636                  JSR     PC,RDALHD       ;READ ALL HEADERS
4680 027512  027532                          60$
4681 027514  004737  021506                  JSR     PC,VERHDR       ;VERIFY HEADER
4682 027520  027532                          60$
4683 027522                          9$:
4684 027522  023737  013560  003116          CMP     LOLIMW,NEWCYL   ;CHECK IF AT LOLIMIT
4685 027530  103726                          BLO     T247$           ;NO - LOOP
4686 027532  012737  000002  003032  60$:    MOV     #2,ERRSWI       ;INIT ERROR SWITCH
4687 027540                          ENDSUB
     027540                          L10036:
     027540  104403                          TRAP    C$ESUB
4688 027542                          T2465$:
4689 027542                          ENDTST
     027542                          L10035:
     027542  104401                          TRAP    C$ETST
4690
4691                                 .SBTTL  *TEST 7         **SEEK TESTS
4692 027544                          BGNTST          ;TEST 7
     027544                                                          T7::
4693 027544  012737  006643  003026          MOV     #P2T09E,ERHEAD  ;SET ERROR HEADER
4694 027552  004737  016242                  JSR     PC,TSTINT       ;INITIALIZE TEST
4695 027556  004737  016260                  JSR     PC,GSTATR       ;CLEAR DRIVE
4696 027562  030052                          T2565$
4697 027564  004737  020600                  JSR     PC,CHOSHD       ;GO CHOSE HEAD
4698 027570  013705  013560                  MOV     LOLIMW,R5       ;SET TO POSTION HEADS TO LOLIMIT
4699 027574  004737  017760                  JSR     PC,POSHDS       ;POSITION HDS TO LOWLIMIT
4700 027600  030052                          T2565$
4701 027602  004737  022366          T256$:  JSR     PC,GETPOS       ;GET CURRENT POSITION
4702 027606  030052                          T2565$
4703 027610  013737  003120  003116          MOV     CURCYL,NEWCYL   ;PUT CURRENT INTO NEW
4704 027616  012704  002444                  MOV     #T25TBL,R4      ;SET POINTER TO TABLE OF SEEK DIFF FOR RL01
4705 027622  022737  000001  002312  T258$:  CMP     #1,T.DRIVE      ;CHECK TYPE OF DRIVE
4706 027630  001402                          BEQ     T2588$          ;BRANCH IF RL01
4707 027632  012704  002472                  MOV     #T25TB2,R4      ;POINT TO THE RL02 TABLE OF CYLINDERS
4708
4709 027636  012405                  T2588$: MOV     (R4)+,R5        ;PUT FIRST IN R5
```

```
;TEST 7          ;;SEEK TESTS

 4710 027640  013701  013562                    MOV     HILIMW,R1       ;GET HILIMIT
 4711 027644  163701  013560                    SUB     LOLIMW,R1       ;SUBTRACT LOLIMIT
 4712 027650  021401                            CMP     (R4),R1         ;CHECK IF NEW DIFFERENCE IS IN BOUNDS
 4713 027652  101073                            BHI     T2517$$         ;NO - SKIP TEST
 4714 027654  060537  003116            T257$:  ADD     R5,NEWCYL       ;ADD TO PRESENT POSITION
 4715 027660  023737  003116  013560            CMP     NEWCYL,LOLIMW   ;CHECK IF AT OR PAST LOLIMIT
 4716 027666  002004                            BGE     9$              ;NO - SKIP
 4717 027670  013737  013560  003116            MOV     LOLIMW,NEWCYL   ;ELSE SET TO LOLIMIT
 4718 027676  000407                            BR      11$
 4719 027700  023737  003116  013562  9$:       CMP     NEWCYL,HILIMW   ;CHECK IF AT HILIMIT OR GREATER
 4720 027706  003403                            BLE     11$             ;NO - SKIP
 4721 027710  013737  013562  003116            MOV     HILIMW,NEWCYL   ;ELSE SET FOR HILIMIT
 4722 027716                            11$:
 4723 027716                            BGNSUB
      027716                                                                    T7.1:
      027716  104402                            TRAP    C$BSUB
 4724 027720                                    INLOOP                  ;CHECK IF IN ERROR LOOP
      027720  104420                            TRAP    C$INLP
 4725 027722                                    BNCOMPLETE      13$     ;NO - SKIP
      027722  103011                            BCC     13$
 4726 027724  004737  022366                    JSR     PC,GETPOS       ;GET CURRENT POSITION
 4727 027730  027774                            60$
 4728 027732  023737  003120  003116            CMP     CURCYL,NEWCYL   ;CHECK IF HEADS AT DESIRED POSITION
 4729 027740  001002                            BNE     13$             ;NO - SKIP
 4730 027742  004737  020664                    JSR     PC,ONSWAP       ;ELSE SWAP CURRENT AND NEW CYLINDERS
 4731 027746  004737  017206            13$:    JSR     PC,XSEEK        ;DO SEEK
 4732 027752  027774                            60$
 4733 027754  012701  005670                    MOV     #3000.,R1       ;SET WAIT COUNT
 4734 027760  004737  022102                    JSR     PC,RDYWAIT      ;WAIT FOR READY
 4735 027764  027774                            60$
 4736 027766  004737  022514                    JSR     PC,VERPOS       ;VERIFY POSITION
 4737 027772  027774                            60$
 4738 027774  012737  000002  003032  60$:      MOV     #2,ERRSWI       ;INITIALIZE ERROR SWITCH
 4739 030002                            ENDSUB
      030002                            L10040:
      030002  104403                            TRAP    C$ESUB
 4740 030004                                    ESCAPE  TST             ;EXIT TEST IF ERROR
      030004  104410                            TRAP    C$ESCAPE
      030006  000044                            .WORD   L10037-.
 4741 030010  023737  013562  003116            CMP     HILIMW,NEWCYL   ;CHECK IF SEEK WAS TO HILIMIT
 4742 030016  001002                            BNE     15$             ;NO - SKIP
 4743 030020  005405                            NEG     R5              ;ELSE SET R5 TO REPEAT DIFF IN REVERSE
 4744 030022  000714                            BR      T257$
 4745 030024  023737  013560  003116  15$:      CMP     LOLIMW,NEWCYL   ;TEST IF LAST SEEK WAS TO LOLIMIT
 4746 030032  001310                            BNE     T257$           ;NO - GO DO SEEK TEST
 4747 030034  021437  002316                    CMP     (R4),HLMTW      ;CHECK IF ALL TABLE DIFF USED
 4748 030040  001276                            BNE     T2[illegible]   ;NO - SKIP
 4749 030042  004737  020624            T2517$: JSR     PC,[illegible]HD ;GO SWAP TO HEAD 1 OR END TEST
 4750 030046  030052                            T2565$                  ;ABORT RETURN
 4751 030050  000654                            BR      T256$           ;REPEAT TEST HEAD 1
 4752 030052                            T2565$:
 4753 030052                            ENDTST
      030052                            L10037:
      030052  104401                            TRAP    C$ETST
 4754
 4755                                   .SBTTL  ;TEST 8          ;;FORWARD OSCILLATING SEEK
 4756 030054                            BGNTST          ;TEST 8
```

•TEST 8 ••FORWARD OSCILLATING SEEK

```
                                                                                T8::
     030054
4757 030054  012737  006646  003026          MOV     #P2T10E,ERHEAD  ;SET ERROR HEADER
4758 030062  004737  016242                  JSR     PC,TSTINT       ;INITIALIZE TEST
4759 030066  004737  016260                  JSR     PC,GSTATR       ;CLEAR DRIVE
4760 030072  030350                          T2665$
4761 030074  004737  020600                  JSR     PC,CHOSHD       ;GO CHOSE HEAD
4762 030100  012705  000001          T266$:  MOV     #1,R5           ;LOAD R5 FOR FIRST SEEK
4763 030104  032737  020000  013556          BIT     #HICYL,MISWIW   ;TEST IF HI CYLINDER SPECED
4764 030112  001402                          BEQ     2$              ;NO - SKIP
4765 030114  013705  013562                  MOV     HILIMW,R5       ;ELSE SET UPPER LIMIT
4766 030120  005037  003116          2$:     CLR     NEWCYL          ;SET TO SEEK TO CYL 0
4767 030124  032737  040000  013556          BIT     #LOCYL,MISWIW   ;CHECK IF LO CYL SPEC'D
4768 030132  001403                          BEQ     5$              ;NO - SKIP
4769 030134  013737  013560  003116          MOV     LOLIMW,NEWCYL   ;ELSE SET LOWER LIMIT
4770 030142  004737  017206          5$:     JSR     PC,XSEEK        ;DO SEEK
4771 030146  030350                          T2665$
4772 030150  012701  005670                  MOV     #3000.,R1       ;SET WAIT COUNT FOR 120 MS
4773 030154  004737  022102                  JSR     PC,RDYWAIT      ;WAIT FOR READY
4774 030160  030350                          T2665$
4775 030162  004737  022366          T267$:  JSR     PC,GETPOS       ;GET HEAD POSITION
4776 030166  030350                          T2665$
4777 030170  010537  003116                  MOV     R5,NEWCYL       ;LOAD NEW CYLINDER INTO NEWCYL
4778 030174                          BGNSUB
                                                                                T8.1:
     030174
     030174  104402                          TRAP    C$BSUB
4779 030176                                  INLOOP                  ;CHECK IF IN ERROR LOOP
     030176  104420                          TRAP    C$INLP
4780 030200                                  BNCOMPLETE      18$     ;NO - SKIP
     030200  103011                          BCC     18$
4781 030202  004737  022366                  JSR     PC,GETPOS       ;GET POSITION
4782 030206  030304                          60$
4783 030210  023737  003120  003116          CMP     CURCYL,NEWCYL   ;CHECK IF HEADS AT DESIRED LOC
4784 030216  001002                          BNE     18$             ;NO - SKIP
4785 030220  004737  020664                  JSR     PC,ONSWAP               ;SWAP OLD AND NEW
4786 030224  004737  017206          18$:    JSR     PC,XSEEK        ;DO SEEK
4787 030230  030304                          60$
4788 030232  012701  005670                  MOV     #3000.,R1       ;SET WAIT COUNT 120 MS
4789 030236  004737  022102                  JSR     PC,RDYWAIT      ;WAIT FOR READY
4790 030242  030304                          60$
4791 030244  004737  022514                  JSR     PC,VERPOS       ;VERIFY HEAD POSITION
4792 030250  030304                          60$
4793 030252  005737  003124                  TST     DESSGN          ;TEST IF JUST SEEK REV
4794 030256  001412                          BEQ     60$             ;YES - SKIP
4795 030260  005037  003116                  CLR     NEWCYL          ;ELSE SET TO SEEK TO 0
4796 030264  032737  040000  013556          BIT     #LOCYL,MISWIW   ;CHECK IF LO LIMIT SPEC'D
4797 030272  001754                          BEQ     18$             ;NO - SKIP
4798 030274  013737  013560  003116          MOV     LOLIMW,NEWCYL   ;ELSE SET LOW LIMIT FOR SEEK
4799 030302  000750                          BR      18$
4800 030304  012737  000002  003032  60$:    MOV     #2,ERRSWI       ;INIT ERROR SWITCH
4801 030312                          ENDSUB
     030312                          L10042:
     030312  104403                          TRAP    C$ESUB
4802 030314                                  ESCAPE  TST             ;EXIT TEST IF ERROR
     030314  104410                          TRAP    C$ESCAPE
     030316  000032                          .WORD   L10041-.
4803 030320  032737  020000  013556          BIT     #HICYL,MISWIW   ;TEST IF UPPER LIMIT SPEC D
4804 030326  001004                          BNE     20$             ;YES - SKIP
```

```
*TEST 8          **FORWARD OSCILLATING SEEK

 4805 030330  005205                          INC     R5              ;BUMP R5
 4806 030332  020537  002324                  CMP     R5,GBND ;ALL CYLINDERS DONE
 4807 030336  001311                          BNE     T267$           ;NO   GO DO NEXT CYLINDER
 4808 030340  004737  020624          20$:    JSR     PC,SWAPHD       ;GO SWAP TO HEAD 1 OR END TEST
 4809 030344  030350                          T2665$                  ;ABORT RETURN
 4810 030346  000654                          BR      T266$           ;GO DO TESTS
 4811 030350                          T2665$:
 4812 030350                          ENDTST
      030350                          L10041:
      030350  104401                          TRAP    C$ETST
 4813
 4814                                 .SBTTL  *TEST 9         **REVERSE OSCILLATING SEEK
 4815 030352                          BGNTST                  ;TEST 9
      030352                                                                          T9::
 4816 030352  012737  006661  003026          MOV     #P2T11E,ERHEAD  ;SET ERROR HEADER
 4817 030360  004737  016242                  JSR     PC,TSTINT       ;INITIALIZE TEST
 4818 030364  004737  016260                  JSR     PC,GSTATR       ;CLEAR DRIVE
 4819 030370  030646                          T2765$
 4820 030372  004737  020600                  JSR     PC,CHOSHD       ;GO CHOSE HEAD
 4821 030376  013737  002316  003116  T275$:  MOV     HLMTW,NEWCYL    ;SEEK OUT TO 255.
 4822 030404  032737  020000  013556          BIT     #HICYL,MISWIW   ;TEST IF UPPER LIMIT SPEC'D
 4823 030412  001403                          BEQ     2$              ;NO - SKIP
 4824 030414  013737  013562  003116          MOV     HILIMW,NEWCYL   ;ELSE SET UPPER LIMIT
 4825 030422  013705  002322          2$:     MOV     NXTHL,R5        ;SET R5 FOR FIRST SEEKS
 4826 030426  032737  040000  013556          BIT     #LOCYL,MISWIW   ;CHECK IF LO LIMIT SPEC'D
 4827 030434  001402                          BEQ     5$              ;NO - SKIP
 4828 030436  013705  013560                  MOV     LOLIMW,R5       ;SET LOWER LIMIT
 4829 030442  004737  017206          5$:     JSR     PC,XSEEK        ;DO SEEK
 4830 030446  030646                          T2765$
 4831 030450  012701  005670                  MOV     #3000.,R1       ;SET WAIT TO 120 MS
 4832 030454  004737  022102                  JSR     PC,RDYWAIT      ;WAIT FOR DRIVE READY
 4833 030460  030646                          T2765$
 4834 030462  004737  022366          T276$:  JSR     PC,GETPOS       ;GET POSITION
 4835 030466  030646                          T2765$
 4836 030470  010537  003116                  MOV     R5,NEWCYL       ;SET FOR NEXT SEEK
 4837 030474                          BGNSUB
      030474                                                                          T9.1:
      030474  104402                          TRAP    C$BSUB
 4838 030476                                  INLOOP                  ;CHECK IF IN ERROR LOOP
      030476  104420                          TRAP    C$INLP
 4839 030500                                  RNCOMPLETE      18$     ;NO - SKIP
      030500  103011                          BCC     18$
 4840 030502  004737  022366                  JSR     PC,GETPOS       ;ELSE GET POSITION
 4841 030506  030606                          60$
 4842 030510  023737  003120  003116          CMP     CURCYL,NEWCYL   ;CHECK IF AT DESIRED CYL
 4843 030516  001002                          BNE     18$             ;NO - SKIP
 4844 030520  004737  020664                  JSR     PC,ONSWAP       ;ELSE SWAP OLD AND NEW CYL
 4845 030524  004737  017206          18$:    JSR     PC,XSEEK        ;DO SEEK
 4846 030530  030606                          60$
 4847 030532  012701  005670                  MOV     #3000.,R1       ;SET WAIT FOR 120 MS
 4848 030536  004737  022102                  JSR     PC,RDYWAIT      ;WAIT FOR READY
 4849 030542  030606                          60$
 4850 030544  004737  022514                  JSR     PC,VERPOS       ;VERIFY POSITION
 4851 030550  030606                          60$
 4852 030552  005737  003124                  TST     DESSGN          ;CHECK IF JUST SEEK FWD
 4853 030556  001013                          BNE     60$             ;YES - SKIP
 4854 030560  013737  002316  003116          MOV     HLMTW,NEWCYL    ;ELSE SEEK TO TO 255
```

•TEST 9 ••REVERSE OSCILLATING SEEK

```
4855 030566  032737  020000  013556          BIT     #HICYL,MISWIW   ;TEST IF HILIMIT SPEC'D
4856 030574  001753                           BEQ     18$             ;NO - SKIP
4857 030576  013737  013562  003116          MOV     HILIMW,NEWCYL   ;SET TO UPPER LIMIT
4858 030604  000747                           BR      18$
4859 030606  012737  000002  003032  60$:    MOV     #2,ERRSWI       ;INIT ERROR SWITCH
4860 030614                          ENDSUB
     030614                          L10044:
     030614  104403                          TRAP    C$ESUB
4861 030616                                   ESCAPE  TST             ;EXIT TEST IF ERROR
     030616  104410                          TRAP    C$ESCAPE
     030620  000026                          .WORD   L10043 .
4862 030622  032737  040000  013556          BIT     #LOCYL,MISWIW   ;TEST IF LOLIMIT SPEC'D
4863 030630  001002                           BNE     20$             ;YES - SKIP
4864 030632  005305                           DEC     R5              ;DEC CYLINDER COUNT
4865 030634  100312                           BPL     T276$           ;IF STILL POSITIVE, DO SEEKS AGAIN
4866 030636  004737  020624          20$:    JSR     PC,SWAPHD       ;GO SWAP TO HEAD 1 OR END TEST
4867 030642  030646                          T2765$                  ;ABORT RETURN
4868 030644  000654                           BR      T275$           ;LOOP AGAIN
4869 030646                          T2765$:
4870 030646                          ENDTST
     030646                          L10043:
     030646  104401                          TRAP    C$ETST
4871
4872 030650                          ENDMOD
4873
4874                                 .SBTTL  PARAMETER CODING
4875 030650                          BGNMOD  HRDPRM
4876 030650                          BGNHRD
     030650  000030                          .WORD L10045-L$HARD/2
4877 030652                          GPRML   CNTYPE,CNT,1,YES
     030652  005130                          .WORD   T$CODE
     030654  031016                          .WORD   CNTYPE
     030656  000001                          .WORD   1
4878 030660                          GPRMA   CSRMSG,CSR,0,160000,177776,YES
     030660  000031                          .WORD   T$CODE
     030662  030732                          .WORD   CSRMSG
     030664  160000                          .WORD   T$LOLIM
     030666  177776                          .WORD   T$HILIM
4879 030670                          GPRMA   VECMSG,VECT,0,0,776,YES
     030670  001031                          .WORD   T$CODE
     030672  030746                          .WORD   VECMSG
     030674  000000                          .WORD   T$LOLIM
     030676  000776                          .WORD   T$HILIM
4880 030700                          GPRMD   DRMSG,D  B,0,3400,0,7,YES
     030700  004032                          .WORD   T$CODE
     030702  031010                          .WORD   DRMSG
     030704  003400                          .WORD   3400
     030706  000000                          .WORD   T$LOLIM
     030710  000007                          .WORD   T$HILIM
4881 030712                          GPRML   DRTYPE,TYPDR,1,YES
     030712  003130                          .WORD   T$CODE
     030714  030766                          .WORD   DRTYPE
     030716  000001                          .WORD   1
4882 030720                          GPRMD   BRMSG,PRIOR,0,340,0,7,YES
     030720  002032                          .WORD   T$CODE
     030722  030755                          .WORD   BRMSG
     030724  000340                          .WORD   340
```

PARAMETER CODING

```
        030726  000000                          .WORD   T$LOLIM
        030730  000007                          .WORD   T$HILIM
   4883
   4884 030732                          ENDHRD
                                                .EVEN
        030732                          L10045:
   4885
   4886 030732     102     125     123  CSRMSG: .ASCIZ  /BUS ADDRESS/
        030735     040     101     104
        030740     104     122     105
        030743     123     123     000
   4887 030746     126     105     103  VECMSG: .ASCIZ  /VECTOR/
        030751     124     117     122
        030754     000
   4888 030755     102     122     040  BRMSG:  .ASCIZ  /BR LEVEL/
        030760     114     105     126
        030763     105     114     000
   4889 030766     104     122     111  DRTYPE: .ASCIZ  /DRIVE TYPE = RL01/
        030771     126     105     040
        030774     124     131     120
        030777     105     040     075
        031002     040     122     114
        031005     060     061     000
   4890 031010     104     122     111  DRMSG:  .ASCIZ  /DRIVE/
        031013     126     105     000
   4891 031016     122     114     061  CNTYPE: .ASCIZ  /RL11/
        031021     061     000
   4892 031023                          ENDMOD
                                                .EVEN
   4893
   4894
   4895 031024                          BGNMOD  SFTPRM
   4896 031024                          BGNSFT
        031024  000053                          .WORD L10046-L$SOFT/2
   4897
   4899 031026                          GPRML   CYLQ,MISWI,1,YES
        031026  000130                          .WORD   T$CODE
        031030  031154                          .WORD   CYLQ
        031032  000001                          .WORD   1
   4900 031034                          GPRML   SECQ,MISWI,2,YES
        031034  000130                          .WORD   T$CODE
        031036  031170                          .WORD   SECQ
        031040  000002                          .WORD   2
   4906
   4908 031042                          GPRML   LOLIMQ,MISWI,40000,YES
        031042  000130                          .WORD   T$CODE
        031044  031205                          .WORD   LOLIMQ
        031046  040000                          .WORD   40000
   4909 031050                                  XFERF   1$
        031050  006044                          .WORD   T$CODE
   4910 031052                                  GPRMD   LIMVAL,LOLIM,D,255.,0,253.,YES
        031052  001052                          .WORD   T$CODE
        031054  031224                          .WORD   LIMVAL
        031056  000377                          .WORD   255.
        031060  000000                          .WORD   T$LOLIM
        031062  000375                          .WORD   T$HILIM
   4911 031064                          1$:     GPRML   HILIMQ,MISWI,20000,YES
        031064  000130                          .WORD   T$CODE
```

PARAMETER CODING

```
                031066  031232                          .WORD   HILIMQ
                031070  020000                          .WORD   20000
          4912  031072                                  XFERF   2$
                031072  006044                          .WORD   T$CODE
          4913  031074                                  GPRMD   LIMVAL,HILIM,D,255.,0,255.,YES
                031074  002052                          .WORD   T$CODE
                031076  031224                          .WORD   LIMVAL
                031100  000377                          .WORD   255.
                031102  000000                          .WORD   T$LOLIM
                031104  000377                          .WORD   T$HILIM
          4914  031106                          2$:     GPRML   HEADQ,MISWI,10000,YES
                031106  000130                          .WORD   T$CODE
                031110  031253                          .WORD   HEADQ
                031112  010000                          .WORD   10000
          4915  031114                                  XFERF   3$
                031114  006044                          .WORD   T$CODE
          4916  031116                                  GPRMD   HEADV,HEAD,D,17,0,1,YES
                031116  003052                          .WORD   T$CODE
                031120  031275                          .WORD   HEADV
                031122  000017                          .WORD   17
                031124  000000                          .WORD   T$LOLIM
                031126  000001                          .WORD   T$HILIM
          4918  031130                          3$:     GPRMD   ERLIMQ,ERLIM,D,377,0,377,YES
                031130  004052                          .WORD   T$CODE
                031132  031320                          .WORD   ERLIMQ
                031134  000377                          .WORD   377
                031136  000000                          .WORD   T$LOLIM
                031140  000377                          .WORD   T$HILIM
          4920  031142                                  GPRMD   DCLIMQ,DCLIM,D,377,1,377,YES
                031142  005052                          .WORD   T$CODE
                031144  031342                          .WORD   DCLIMQ
                031146  000377                          .WORD   377
                031150  000001                          .WORD   T$LOLIM
                031152  000377                          .WORD   T$HILIM
          4922  031154                          ENDSFT
                                                          .EVEN
                031154                          L10046:
          4923
          4925  031154    125     123     105   CYLQ:   .ASCIZ  /USE ALL CYL/
                031157    040     101     114
                031162    114     040     103
                031165    131     114     000
          4926  031170    125     123     105   SECQ:   .ASCIZ  /USE ALL SECT/
                031173    040     101     114
                031176    114     040     123
                031201    105     103     124
                031204    000
          4933  031205    114     117     127   LOLIMQ: .ASCIZ  /LOW SEEK LIMIT/
                031210    040     123     105
                031213    105     113     040
                031216    114     111     115
                031221    111     124     000
          4934  031224    126     101     114   LIMVAL: .ASCIZ  /VALUE/
                 31227    125     105     000
          4935  031232    125     120     120   HILIMQ: .ASCIZ  /UPPER SEEK LIMIT/
                031235    105     122     040
                031240    123     105     105
```

PARAMETER CODING

```
                031243  113     040     114
                031246  111     115     111
                031251  124     000
 4936           031253  125     123     105     HEADQ:  .ASCIZ  /USE ONLY ONE SURF/
                031256  040     117     116
                031261  114     131     040
                031264  117     116     105
                031267  040     123     125
                031272  122     106     000
 4937           031275  127     110     101     HEADV:  .ASCIZ  /WHAT SURF (0 OR 1)/
                031300  124     040     123
                031303  125     122     106
                031306  040     050     060
                031311  040     117     122
                031314  040     061     051
                031317  000
 4939           031320  111     116     120     ERLIMQ: .ASCIZ  /INPUT ERROR L MIT/
                031323  125     124     040
                031326  105     122     122
                031331  117     122     040
                031334  114     111     115
                031337  111     124     000
 4941           031342  104     101     124     DCLIMQ: .ASCIZ  /DATA CMP ERR LMT/
                031345  101     040     103
                031350  115     120     040
                031353  105     122     122
                031356  040     114     115
                031361  124     000
 4943                                            .EVEN
 4944           031364                          ENDMOD
 4945
 4946           031364                          LASTAD
                                                .EVEN
                031364  000000                  .WORD   0
                031366  000000                  .WORD   0
                031370                  L$LAST::
 4947
 4948                   000001                  .END
```

SYMBOL TABLE

```
ADR    • 000020 G
AFMID    003224
AFMIDU   003226
ALLCYL • 000001
ALLSEC • 000002
ANYERR • 100000
ARMID    003230
ARMIDU   003232
ASSEMB • 000010
BADADD • 004000
BAMSK  • 000060
BANAM    006243
BASADD   006141
BELL     010555
BMSTAT • 000010
BIT0   • 000001 G
BIT00  • 000001 G
BIT01  • 000002 G
BIT02  • 000004 G
BIT03  • 000010 G
BIT04  • 000020 G
BIT05  • 000040 G
BIT06  • 000100 G
BIT07  • 000200 G
BIT08  • 000400 G
BIT09  • 001000 G
BIT1   • 000002 G
BIT10  • 002000 G
BIT11  • 004000 G
BIT12  • 010000 G
BIT13  • 020000 G
BIT14  • 040000 G
BIT15  • 100000 G
BIT2   • 000004 G
BIT3   • 000010 G
BIT4   • 000020 G
BIT5   • 000040 G
BIT6   • 000100 G
BIT7   • 000200 G
BIT8   • 000400 G
BIT9   • 001000 G
BOE    • 000400 G
BRMSG    030755
BSCHK    024372
BSFLAG   003034
BSFVAL   003510
BSNSTR   007550
BYPSNM   007501
CAFDT    010662
CAMSK    002326
CCYLUP   010653
CHOSHD   020600
CKBSVD   020710
CKDATA • 000102
CKERLM   015712
CLKADR   003506
CLKCSB • 172542
CLKCSR • 172540
CLKCTR • 172544
CLKFLG   003504
CLNCOD   015152 G
CLRBYT   002320
CLRPAR   025772
CNT    • 000012
CNTYPE   031016
COMPOP • 007777
CONMNG • 000004
CONTIN   014044
COSTAT • 000040
COUNT    003250
CRDYMS • 000200
CSNAM    006236
CSR    • 000000
CSRMSG   030732
CURCYL   003120
CYLQ     031154
CYLTBL   002620
CYLUP  • 000004
CYLWD    007466
C$AU   • 000052
C$AUTO • 000061
C$BRK  • 000022
C$BSEG • 000004
C$BSUB • 000002
C$CEFG • 000045
C$CLCK • 000062
C$CLEA • 000012
C$CLOS • 000035
C$CLP1 • 000006
C$CVEC • 000036
C$DCLN • 000044
C$DODU • 000051
C$DRPT • 000024
C$DU   • 000053
C$EDIT • 000003
C$ERDF • 000055
C$ERHR • 000056
C$ERRO • 000060
C$ERSF • 000054
C$ERSO • 000057
C$ESCA • 000010
C$ESEG • 000005
C$ESUB • 000003
C$ETST • 000001
C$EXIT • 000032
C$GETB • 000026
C$GETW • 000027
C$GMAN • 000043
C$GPHR • 000042
C$GPLO • 000030
C$GPRI • 000040
C$INIT • 000011
C$INLP • 000020
C$MANI • 000050
C$MEM  • 000031
C$MSG  • 000023
C$OPEN • 000034
C$PNTB • 000014
C$PNTF • 000017
C$PNTS • 000016
C$PNTX • 000015
C$QIO  • 000377
C$RDBU • 000007
C$REFG • 000047
C$RESE • 000033
C$REVI • 000003
C$RFLA • 000021
C$RPT  • 000025
C$SEFG • 000046
C$SPRI • 000041
C$SVEC • 000037
C$TPRI • 000013
C10MS    010634
C5SEC    010673
C500MS   010645
DANAM    006250
DATACM • 000001
DATCOM   023344
DATGEN   023204
DCKERR • 004000
DCLIM  • 000012
DCLIMQ   031342
DCLIMW   013570
DESDIF   003122
DESHD    003126
DESSEC   003130
DESSGN   003124
DIAGMC • 000000
DIFAUG   003112
DIFWD    007442
DIRBIT • 000004
DIRMSK   002330
DLTERR • 010000
DONE     003022
DRDYMS • 000001
DRMSG    031010
DRSB   • 000010
DRSELT • 000004
DRSET  • 000010
DRTYPE   030766
DRVCNT   003110
DRVERR • 040000
DRVNAM   006152
DRVNAV   006157
DSESTA • 000400
DSMSK  • 001400
DSPCOD   013572 G
EF.CON • 000036 G
EF.NEW • 000035 G
EF.PWR • 000034 G
EF.RES • 000037 G
EF.STA • 000040 G
ERHEAD   003026
ERLIM  • 000010
ERLIMQ   031320
ERLIMW   013566
ERRCNT   003254
ERRPOI   003252
ERRSWI   003032
ERRVEC   003244
ERR1     011724 G
ERR10    013320 G
ERR2     011772 G
ERR3     012040 G
ERR4     012106 G
ERR5     012156 G
ERR6     012226 G
ERR7     013110 G
ERR8     013160 G
ERR9     013254 G
EVL    • 000004 G
EXACYL   003240
EXHCYL   003236
EXOCYL   003234
EXROT    003242
E$END  • 002100
E$LOAD • 000035
FBSFIL   003706
FMTOP1   010701
FMTOP2   010730
FMTOP3   010752
FMT1     010773
FMT1.1   011000
FMT11    011217
FMT12    011225
FMT13    011233
FMT14    011277
FMT15    011331
FMT16    011365
FMT17    011376
FMT18    011420
FMT19    011452
FMT2     011007
FMT20    011507
FMT21    011537
FMT22    011562
FMT23    011616
FMT24    011632
FMT25    011637
FMT26    011647
FMT27    011673
FMT28    011712
FMT3     011012
FMT4     011015
FMT5     011026
FMT6     011046
FMT7     011110
FMT8     011160
FMT9     011212
FOLWRT • 000100
FRMWD    007473
FWDSKO • 002000
FWDSKS • 000400
F$AU   • 000015
F$AUTO • 000020
F$BGN  • 000040
F$CLEA • 000007
F$DU   • 000016
F$END  • 000041
F$HARD • 000004
F$HW   • 000013
F$INIT • 000006
F$JMP  • 000050
F$MOF  • 000000
F$MS   • 000011
F$PROT • 000021
F$PWR  • 000017
F$RPT  • 000012
F$SEG  • 000003
F$SOFT • 000005
F$SRV  • 000010
F$SUB  • 000002
F$SW   • 000014
F$TEST • 000001
GBND     002324
GETPOS   022366
GETSTA • 000003
GLBDAT   002240 G
GLBEQA   002240 G
GLBERR   011724 G
GLBSUB   015304 G
GLBTXT   005360 G
GSTAT    016310
GSTATC   016274
GSTATG   016320
GSTATR   016260
GTSTAT • 000104
G$CNTO • 000200
G$DELM • 000372
G$DISP • 000003
G$EXCP • 000400
G$HILI • 000002
G$LOLI • 000001
G$NO   • 000000
G$OFFS • 000400
G$OFSI • 000376
G$PRMA • 000001
G$PRMD • 000002
G$PRML • 000000
G$RADA • 000140
G$RADB • 000000
G$RADO • 000040
G$RADL • 000120
G$RADO • 000020
G$XFER • 000004
```

SYMBOL TABLE

```
G$YES  = 000010
HADONE  003024
HCESTA= 040000
HCRCER= 004000
HDALIG= 000010
HDCYL   002332
HDHSEL= 000100
HDHOVF  007323
HDRCMP= 000002
HDR40  = 100000
HDSEC  = 000077
HDSEL  = 000020
HDWD    007455
HDWRD1  003066
HDWRD2  003070
HDWRD3  003072
HEAD   = 000006
HEADLM= 010000
HEADQ   031253
HEADV   031275
HEADW   013564
HFIN    003204
HFINU   003206
HFOUT   003210
HFOUTU  003212
HICYL  = 020000
HILIM  = 000004
HILIMQ  031232
HILIMW  013562
HLMTW   002316
HNFERR= 010000
HOE    = 100000 G
HOSTAT= 000020
HPTCOD  013536 G
HRDPRM  030650 G
HRDWTS  026022 G
HRIN    003214
HRINU   003216
HROUT   003220
HROUTU  003222
HSMSK  = 000100
HSSTAT= 000100
IBE    = 010000 G
IBUFF   004102
IDU    = 000040 G
IER    = 020000 G
INITCO  013616 G
INOUTS= 000020
INTEBL= 000100
INTHLR  015632
ISR    = 000100 G
IXE    = 004000 G
I$AU   = 000041
I$AUTO= 000041
I$CLN  = 000041
I$DU   = 000041
I$HRD  = 000041
I$INIT= 000041
I$MOD  = 000041
I$MSG  = 000041
I$PROT= 000040
I$PTAB= 000041
I$PWR  = 000041
I$RPT  = 000041
I$SEG  = 000041
I$SETU= 000041
I$SFT  = 000041
I$SRV  = 000041
I$SUB  = 000041
I$TST  = 000041
JJJ     002314
J$JMP  = 000167
LAB     014016
LABACF  007273
LABACR  007307
LABEXP  007206
LABHCF  007243
LABHCR  007257
LABIN   007163
LABMID  007171
LABOCF  007217
LABOCR  007231
LABOUT  007200
LAB1    006262
LAB2    006275
LIMVAL  031224
LOCERR  003460
LOCYL  = 040000
LOE    = 040000 G
LOLIM  = 000002
LOLIMQ  031205
LOLIMW  013560
LOT    = 000010 G
L$ACP   002110 G
L$APT   002036 G
L$AUT   002070 G
L$AUTO  014614 G
L$CCP   002106 G
L$CLEA  015152 G
L$CO    002032 G
L$DEPO  002011 G
L$DESC  002122 G
L$DESP  002076 G
L$DEVP  002060 G
L$DISP  013574 G
L$DLY   002116 G
L$DTP   002040 G
L$DTYP  002034 G
L$DU    015300 G
L$DUT   002072 G
L$DVTY  002226 G
L$EF    002052 G
L$ENVI  002044 G
L$ETP   002102 G
L$EXP1  002046 G
L$EXP4  002064 G
L$EXP5  002066 G
L$HARD  030652 G
L$HIME  002120 G
L$HPCP  002016 G
L$HPTP  002022 G
L$HW    013540 G
L$ICP   002104 G
L$INIT  013616 G
L$LADP  002026 G
L$LAST  031370 G
L$LOAD  002100 G
L$LUN   002074 G
L$MREV  002050 G
L$NAME  002000 G
L$PRIO  002042 G
L$PROT  013530 G
L$PRT   002112 G
L$REPP  002062 G
L$REV   002010 G
L$SOFT  031026 G
L$SPC   002056 G
L$SPCP  002020 G
L$SPTP  002024 G
L$STA   002030 G
L$SW    013556 G
L$TEST  002114 G
L$TIML  002014 G
L$UNIT  002012 G
L.BA    003052
L.CS    003050
L.DA    003054
L.MP    003056
L10000  011770
L10001  012036
L10002  012104
L10003  012154
L10004  012224
L10005  013106
L10006  013156
L10007  013252
L10010  013316
L10011  013526
L10013  013554
L10014  013572
L10015  014612
L10016  015150
L10017  015276
L10020  015302
L10021  015630
L10022  015710
L10023  026242
L10024  026224
L10025  026452
L10026  026450
L10027  026662
L10030  026660
L10031  027106
L10032  027104
L10033  027314
L10034  027276
L10035  027542
L10036  027540
L10037  030052
L10040  030002
L10041  030350
L10042  030312
L10043  030646
L10044  030614
L10045  030732
L10046  031154
MAJINC  003502
MAPROX  007153
MBADAD  006022
MBADSF  006043
MBSETO= 000001
MCERR   007771
MCONHN  006407
MCYLOC  010345
MCYLUP  005560
MDATCP  005442
MDCRC   010013
MDHEDR  002000 G
MDLT    010040
MDRDY   007760
MDRERR  010102
MDRRES  006336
MDRVST  010130
MDSERR  010113
MERRS   010550
MEXERS  010513
MFLERR  010272
MFMTER  006073
MFOLWR  005630
MFWDSK  005701
MFWSKO  005732
MGTSTA  005430
MHCERR  010212
MHCRC   010003
MHDERR  010255
MHDRCP  005461
MHFCRC  010052
MHNF    010024
MININC  003472
MINOUT  005607
MISWI  = 000000
MISWIW  013556
MITEST= 100000
MNDRST  010352
MNEERR  010320
MNOCLR  006423
MNOINT  006354
MOPER   005527
MOPERR  010245
MORECE  003030
MOUTIN  005570
MPNAM   006255
MQUALS= 003760
MREAD   005364
MREADH  005375
MRESKO  005766
MREVSK  005650
MRLFAL  010442
MRSLT   005536
MSEEK   005360
MSPERR  010143
MSTERR  010176
MTMBS   006120
MTOSLO  006316
MULOAD  005547
MUNDEF  010375
MWDERR  010230
MWGERR  010161
MWORD   006310
MWRCHK  005405
MWRITE  005416
MWRSET  005513
MWRTAB  010501
M40HDR  005477
NEWCYL  003116
NOCLR  = 000010
NOCTLR  007645
NOERCT  003461
NOIRPT= 000002
NOOP   = 000100
NOPWR   006176
NOTRDY  007703
NXMERR= 020000
NXTHL   002322
NXTPAS  014064
OBUFF   004502
OFIN    003154
OFINU   003156
OFMID   003160
OFMIDU  003162
OFOUT   003164
OFOUTU  003166
OLDCYL  003114
ONSWAP  020664
OPFLAG  003020
OPIERR= 002000
OPMSGS  002240
OPR004  007425
OPR1A   007376
OPR1B   007402
OPR12   007357
ORIN    003170
ORINU   003172
ORMID   003174
ORMIDU  003176
```

SEQ 0...

SYMBOL TABLE

```
OROUT   003200          P2T12E  006674          STATE2  010604          T$TAGL- 177777          T276$   030462
OROUTU  003202          P2T13E  006706          STATE3  010614          T$TAGN- 010047          T265$   030646
OUTINS- 000040          P2T14E  006722          STATE5  010624          T$TEMP- 000000          T3      026454 G
O$APTS- 000000          P2T15E  006743          STOSTA- 010000          T$TEST- 000011          T3.1    026524
O$AU  - 000000          P2T16E  006766          SUBSTK  002420          T$TSTM- 177777          T33TBL  002520
O$BGNR- 000000          P2T17E  007007          SVCBGL- 000001          T$TSTS- 000001          T4      026664 G
O$BGNS- 000001          P2T18E  007041          SVCGBL- 000000          T$$AUT- 010016          T4.1    026752
O$DU  - 000001          P2T19E  007063          SVCINS- 000000          T$$CLE- 010017          T5      027110 G
O$ERRT- 000000          RDALHD  022636          SVCSUB- 000001          T$$DU - 010020          T5.1    027146
O$GNSW- 000001          RDDATA- 000114          SVCTAG- 000000          T$$HAR- 010045          T6      027316 G
O$POIN- 000001          RDHEAD- 000110          SVCTST- 000001          T$$HW - 010013          T6.1    027404
O$SETU- 000000          RDNOHR- 000116          SWAPHD  020624          T$$INI- 010015          T7      027544 G
PART2 - 000001 G        RDYCHK  020324          S$LSYM- 010000          T$$MSG- 010011          T7.1    027716
PASCNT  003246          RDYWAI  022102          TAG     003500          T$$PRO- 010012          T8      030054 G
PASNEW  014072          READRL  016052          TBLSTR  003040          T$$SEG- 010000          T8.1    030174
PASNUM  003454          RELDWT- 040000          TBT     002560          T$$SOF- 010046          T9      030352 G
PATTBL  002374          RESE3   010561          TCERR   007624          T$$SRV- 010022          T9.1    030474
PAT1    005102          RESE4   010565          TEMP    003474          T$$SUB- 010044          UAM   - 000200 G
PAT10   005356          RESE5   010572          TEMP0   003132          T$$SW - 010014          ULCAD - 000010
PAT2    005104          RESE6   010577          TEMP1   003134          T$$TES- 010043          UNDTST  007412
PAT3    005144          RESPAR  003076          TEMP2   003136          T.BA    003062          UNXERR  006464
PAT4    005204          RESTAR  014034          TEMP3   003140          T.CS    003060          VALDES  007127
PAT5    005244          RESTBL  002334          TEMP4   003142          T.DA    003064          VCNRST  006443
PAT6    005252          REVSKO- 001000          TEMP5   003144          T.DRIV  002312          VCSTAT- 001000
PAT7    005312          REVSKS- 000200          TEMP6   003146          T.MP    003066          VECMSG  030746
PAT8    005314          RLBA  - 000002          TEMP7   003150          T.STAT  003074          VECT  - 000002
PAT9    005354          RLBAS   003042          TEMP8   003152          T1      026022 G        VERHDR  021506
PH65$   020266          RLCS  - 000000          TIME    015304          T1.1    026056          VERPOS  022514
PNT   - 001000 G        RLCSR - 000000          TIM.US  003476          T1965$  026242          WAITIN  016104
POSHDS  017760          RLDA  - 000004          TOSLOW- 000001          T197$   026046          WCMSK - 017777
POSHDO  022056          RLDRV   003046          TRPFLG  003462          T2      026244 G        WCRNG - 160000
POSHSB  022052          RLMP  - 000006          TRPHAN  015624          T2.1    026314          WDESTA- 100000
POSHW1  022044          RLVEC   003044          TSTINT  016242          T206$   026316          WGESTA- 002000
PRI   - 002000 G        RORWOP- 020000          TSTLAB  006501          T2065$  026452          WLSTAT- 020000
PRIOR - 000004          RPTOP   024542          TYPDR - 000006          T216$   026526          WRTSWI  003036
PRIO0 - 000000 G        RPTREM  025536          T$ARGC- 000010          T2165$  026662          WTDATA- 000112
PRIO1 - 000040 G        RPTRES  025330          T$CODE- 005052          T2265$  027106          XDELAY  003466
PRIO2 - 000100 G        RSTRT   013752          T$ERRN- 001275          T227$   026754          XRDHD   021052
PRIO3 - 000140 G        SAMSK - 000077          T$EXCP- 000000          T233$   027134          XRDHDC  021042
PRIO4 - 000200 G        SBSFIL  003512          T$FLAG- 000040          T2365$  027314          XRDHDG  021056
PRIO5 - 000240 G        SECQ    031170          T$GMAN- 000000          T2465$  027542          XREAD   023674
PRIO6 - 000300 G        SECWD   007461          T$HILI- 000377          T247$   027406          XREADG  023702
PRIO7 - 000340 G        SEEK  - 000106          T$LAST- 000001          T25TBL  002444          XSEEK   017206
PSETNM  003456          SEEKOP- 010000          T$LOLI- 000001          T25TB2  002472          XSEEKT  017176
PWCON   014342          SEQMES  007514          T$LSYM- 010000          T2517$  030042          XSEEK1  017212
PWRFLG  003464          SETDON  014120          T$LTNO- 000011          T256$   027602          XTIME   015450
P2T03E  006507          SFTPRM  031024 G        T$NEST- 177777          T2565$  030052          XWRITE  023634
P2T04E  006525          SGNWD   007450          T$NS0 - 000000          T257$   027654          XWRITT  023624
P2T05E  006545          SKTMES  007073          T$NS1 - 000005          T258$   027622          XWRIT1  023640
P2T06E  006565          SPDSTA- 004000          T$NS2 - 000002          T2588$  027636          X$ALWA- 000000
P2T07E  006605          SPTCOD  013554 G        T$PTNU- 000000          T266$   030100          X$FALS- 000040
P2T08E  006623          SRTMES  007105          T$SAVL- 177777          T2665$  030350          X$OFFS- 000400
P2T09E  006643          SSINDX  003016          T$SEGL- 177777          T267$   030162          X$TRUE- 000020
P2T10E  006646          STAMES  007537          T$SEKO- 010000          T275$   030376          YDELAY  003470
P2T11E  006661          STAMSK- 000007          T$SUBN- 000001

. ABS.  031370      000
```

SYMBOL TABLE

000000 001
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 29072 WORDS (114 PAGES)
DYNAMIC MEMORY: 20060 WORDS (77 PAGES)
ELAPSED TIME: 00:27:14
CNRLJA.BIN,CNRLJA.LST/-SP=SVC34.MLB/ML,CNRLJA.MAC

.