RL11,RLV11

RL01,02 DR TST 2

CZRLJA0

AH-F122A-MC

FICHE 1 OF 1

MAY 1979

digital

MADE IN USA

IDENTIFICATION

PRODUCT CODE: AC-F123A-MC

PRODUCT NAME: CZRLJA0 RL01/02 DRIVE TEST PART 2

DATE CREATED: 5-JAN-79

MAINTAINER: DIAGNOSTIC ENGINEERING

AUTHOR: D. DEKNIS, C. CAMPBELL

TABLE OF CONTENTS

1.0 GENERAL INFORMATION

1.1 PROGRAM ABSTRACT

1.1.1 STRUCTURE OF PROGRAM

THIS DIAGNOSTIC OCCUPIES 14.5K WORDS OF MEMORY AND IS COMPATIBLE WITH BOTH XXDP AND ACT. IT CAN BE RUN STANDALONE UNDER XXDP, AND CAN BE CHAINED UNDER XXDP, ACT AND APT IN ACT MODE (SEE "CREATE CORE IMAGE" COMMAND BELOW FOR DETAILS OF CHAINING PROCEDURE). IT IS A SINGLE PROGRAM FROM THE STANDPOINT OF THE DIAGNOSTIC USER, BUT WE HAVE INCORPORATED INTO IT A CONTROL MODULE WHICH WILL LATER BE RELEASED INDEPENDENTLY AS A DIAGNOSTIC SUPERVISOR.

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 (DS B>). AT COMMAND MODE THE OPERATOR MAY ENTER ANY OF SEVERAL COMMANDS AS DESCRIBED BELOW.

THE SUPERVISOR CODING FOLLOWS IMMEDIATELY THE DIAGNOSTIC TEST CODING, BUT THE SUPERVISOR LISTING HAS BEEN SUPPRESSED FOR GENERAL DISTRIBUTION. A LIMITED DISTRIBUTION HAS BEEN MADE TO FIELD SERVICE OF THE SUPERVISOR ASSEMBLY LISTING, AND IT MAY BE CONSULTED IN EVENT OF A SOFTWARE PROBLEM.

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 FIRST TESTS THE RL01/02 INTERFACE AND BASIC DRIVE LOGIC. IT THEN BEGINS TESTING THE SEEK OPERATIONS USING SINGLE DIFFERENCES, PROCEEDING INTO SEEKS OF GREATER DIFFERENCES. SEEK TIMING IS DONE AFTER THE SEEK LOGIC HAS BEEN TESTED.

DATA TRANSFERS ARE DONE AFTER ALL THE SEEK TESTS. THE FIRST DATA TRANSFER IS READING OF THE BAD SECTOR FILES WHICH ARE STORED AND USED LATER TO PREVENT TESTING ON BAD SECTORS. FOLLOWING DATA READ AND WRITE TESTING, THE PROGRAM TESTS FOR OVERWRITE PROBLEMS AND ADJACENT CYLINDER INTERFERENCE.

SEEK TIMING, ROTATIONAL TIMING, AND WRITE LOCK DATA PROTECTION ARE DONE IF MANUAL INTERVENTION IS REQUESTED.

1.2 SYSTEM REQUIREMENTS

1.2.1 HARDWARE REQUIREMENTS

- PDP-11/LSI-11 PROCESSOR WITH 16K OR MORE OF MEMORY
- 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'
- KW11P, KW11L (OPTIONAL)
- LINE PRINTER (OPTIONAL)

1.2.2 SOFTWARE REQUIREMENTS

CZRLJA0 RL01/02 DRIVE TEST PART 2 (FORMERLY CZRLDB0)

1.3 RELATED DOCUMENTS AND STANDARDS

RL01/02 USERS MANUAL (EK-RL01/02-PG-PRE)
XXDP USERS MANUAL

1.4 DIAGNOSTIC HIERARCY PREREQUISITES

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

CVRLAA0	RLV11 RL01 DISKLESS TEST (RLV11 ONLY)
CZRLGA0	RL11/RLV11 RL01/02 CONTROLLER TEST (PART 1)
CZRLHA0	RL11/RLV11 RL01/02 CONTROLLER TEST (PART 2)
CZRLIA0	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 SIX STEPS OF EXECUTION

THIS DIAGNOSTIC SHOULD BE LOADED AND STARTED USING NORMAL XXDP PROCEDURES. THE START COMMAND SHOULD NOT SPECIFY AN ADDRESS, BECAUSE THE DIAGNOSTIC HAS THE PROPER TRANSFER ADDRESS CODED INTO IT.

WHEN THIS DIAGNOSTIC IS STARTED, THE FOLLOWING STEPS WILL OCCUR:

****
* STEP 1 *
****

A SHORT SERIES OF "HARDCORE QUESTIONS" WILL BE ASKED:

QUESTION	MEANING
L-CLK (L) N ?	IS THERE AN L-CLOCK?
P-CLK (L) N ?	" " " P-CLOCK?
50HZ (L) N ?	IS THE POWER 50 CYCLES (EUROPE)?
LSI (L) N ?	IS MACHINE AN LSI?
LPT (L) N ?	IS THERE A LINE PRINTER?
MEM (K) (D) 16 ?	HOW MANY K OF MEMORY ARE THERE?

THE DEFAULTS (SHOWN AFTER EACH QUESTION) CAN BE SELECTED BY HITTING CARRIAGE RETURN. IT IS POSSIBLE THAT NOT ALL OF THE QUESTIONS WILL BE ASKED: FOR EXAMPLE, IF YOU SAY "YES" TO THE L-CLOCK QUESTION, THE P-CLOCK QUESTION WILL NOT BE ASKED.

IF NEITHER P OR L CLOCK ARE ANSWERED YES THE OPERATOR WILL BE ASKED TO TYPE TWO CHARACTERS 4 SECONDS APART.

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

WHEN YOU HAVE ANSWERED ALL THE HARDCORE QUESTIONS, THE DIAGNOSTIC WILL ISSUE THE PROMPT "DS-B>". FROM THIS POINT UNTIL THE TIME WHEN YOU RESTART XXDP, YOU WILL BE TALKING TO THE DIAGNOSTIC, NOT XXDP. WE WILL REFER TO THE PRESENCE OF THIS PROMPT AS BEING IN DIAGNOSTIC COMMAND MODE, AS OPPOSED TO XXDP COMMAND MODE.

AT THIS POINT YOU WILL ENTER A "START" COMMAND. THIS IS NOT THE SAME AS THE XXDP "START" COMMAND, WHICH YOU ALREADY ISSUED IN RESPONSE TO THE XXDP 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 "DS-8>" 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:

```
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 3 *
**********
```

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 4 *
****

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 5 *
****

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 6 *
****

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 DS-B>).

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 OCCURED.

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 2, 3, 4, 5, AND 6 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 OCCURED. 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

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 CZRLJA                               O

CZRLJ                                   D

L-CLK (L) N ? Y                         D,O
50HZ (L) N ?                            D
LSI (L) N ?                             D
LPT (L) N ?                             D
MEM (K) (D) 16 ?                        D

DS-B>STA/PASS:1/FLAGS:HOE               D,O

# UNITS (D) ? 2                        D,O

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

UNIT 2                                  D
RL11 (L) Y ?                            D,O
BUS ADDRESS (O) 174400 ?                D,O
VECTOR (O) 160 ?                        D,O
DRIVE (O) 0 ? 1                         D,O
DRIVE TYPE = RL01 (L) ? N               D,O (N=RL02)
BR LEVEL (O) 5 ?                        D,O

CHANGE SW (L) ? N                       D,O


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

DS-B>PRO/FLAGS:IER:LOE:HOE=0            D,O


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

DS-B>CON/FLAGS:HOE:IER:LOE=0                D,0

CHANGE SW (L) ? N                           D,0

CZRLJ EOP 1                                 D
^C

DS-B>RESTART/PASS:1                         D,0

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

2.2 HOW TO CREATE A CHAINABLE FILE

THE DIAGNOSTIC AS RECEIVED FROM RELEASE ENGINEERING CANNOT BE RUN IN CHAIN MODE. THAT IS WHY IT BEARS THE EXTENSION "BIN" INSTEAD OF "BIC". THERE IS A WAY, HOWEVER, TO CREATE A CHAINABLE PROGRAM FROM WHAT YOU'VE GOT.

IT CONSISTS OF RUNNING THE PROGRAM WITH THE SPECIAL COMMAND "CCI" ISSUED WHERE YOU WOULD NORMALLY ISSUE A START COMMAND (TO THE PROMPT DS-B>. THIS COMMAND CAUSES THE DIAGNOSTIC TO GO THRU ALL THE QUESTIONS AND ANSWERS AND THEN TO HALT, JUST WHERE IT WOULD ORDINARILY BEGIN EXECUTION OF THE HARDWARE TEST CODE. AT THIS POINT YOU CAN DUMP THE PROGRAM AS IT SITS IN CORE TO THE LOAD MEDIUM, WITH THE NEW EXTENSION "BIC".

HERE IS A SAMPLE DIALOGUE TO ACCOMPLISH THIS:

```
.R UPD2
RESTART:  XXXXXX
*CLR
*LOAD DIAG.BIN
XFER:200    CORE:0,60602
*START 200
L-CLK (L) N ?
-------------
-------------
-------------
```

```
DS-B>CCI
  UNITS (D) ? 4
-------------
-------------
-------------
CHANGE SW (L) ? N
PTAB END:  60632

****************************************
*AT THIS POINT THE MACHINE HALTS AND*
*YOU MUST RESTART AT ADDRESS  XXXXXX*
****************************************

*HICORE 60632
CORE:  0,60632
*DUMP DK0: DIAG.BIC
```

THE RESULT OF DOING THIS IS THAT YOU CAN NOW BUILD AN XXDP CHAIN FILE CONTAINING THE XXDP COMMAND

```
    .R DIAG.BIC
```

AND THE DIAGNOSTIC WILL EXECUTE WITHOUT MANUAL INTERVENTION, USING THE ANSWERS THAT YOU GAVE IT WHEN YOU DID THE CCI COMMAND.

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
2.	DIAGNOSTIC HAS FINISHED ALL ITS REQUESTED PASSES	START RESTART PRINT DISPLAY FLAGS ZFLAGS
3.	OPERATOR INTERRUPTED THE DIAGNOSTIC WITH CTRL/C	START RESTART CONTINUE PRINT DISPLAY FLAGS ZFLAGS
4.	AN ERROR WAS ENCOUNTERED WITH THE HOE FLAG SET SET	START RESTART CONTINUE PROCEED PRINT DISPLAY FLAGS ZFLAGS

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 EXE-

CUTING C) ERROR WAS ENCOUNTERED WITH HOE FLAG SET D) OPERATOR ENTERED CONTROL/C. AFTER THE OPERATOR RESPONDS TO "# UNITS?", THE HARDWARE DIALOGUE IS INITIATED. WHEN IT IS COMPLETED, THE QUESTIONS "CHANGE SW?" IS ISSUED, AND THE ANSWERS, IF GIVEN, BECOME THE NEW DEFAULTS. THEREFORE IT IS NECESSARY TO RELOAD THE PROGRAM IN ORDER TO RETURN TO THE LOAD DEFAULTS.

THE SWITCH ARGUMENTS ARE AS FOLLOWS:

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

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

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

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

IER INHIBIT ERROR REPORTING

IBE INHIBIT BASIC ERROR REPORTS

IXE INHIBIT EXTENDED ERROR REPORTS

PRI DIRECT ALL MESSAGES TO A LINE PRINTER

PNT PRINT NUMBER OF TEST BEING EXECUTED

BOE BELL ON ERROR

UAM RUN IN UNATTENDED MODE, BYPASSING MANUAL INTERVENTION TESTS

ISR INHIBIT STATISTICAL REPORTS

IDU INHIBIT DROPPING OF UNITS BY DIAGNOSTIC

THE FLAGS NAMED OR EQUATED TO 1 ARE SET, THOSE EQUATED TO 0 ARE CLEARED. A FLAG NOT SPECIFIED IS CLEARED. IF THE FLAGS SWITCH IS NOT GIVEN ALL FLAGS ARE CLEARED.

"EOP-INCR" IS A DECIMAL NUMBER INDICATING HOW OFTEN (IN TERMS OF PASSES) IT IS DESIRED THAT THE END OF PASS MESSAGE BE PRINTED. THE DEFAULT IS AT THE END OF EVERY PASS.

****
RES(TART)/TEST:TEST-LIST/PASS:PASS-CNT/FLAGS:FLAG-LIST/EOP:EOP-INCR/UNITS
:UNIT-LIST
****

THE DIAGNOSTIC IN CORE IS EXECUTED IN ACCORDANCE WITH THE SWITCHES SPECIFIED. HOWEVER, NEW "P-TABLES" ARE NOT BUILT. INSTEAD, THE ONES IN CORE ARE USED.

THE QUESTION "CHANGE SW?" IS ASKED AND THE ANSWERS GIVEN BECOME THE NEW DEFAULTS. THE COMMAND MAY BE ISSUED WHEN COMAND MODE HAS BEEN ENTERED VIA A) DIAGNOSTIC IS FINISHED B) HALT ON ERROR C) CONTROL/C.

THE SWITCH ARGUMENTS ARE AS IN THE START COMMAND EXCEPT:

1. "UNIT-LIST" IS A SEQUENCE OF LOGICAL UNIT NUMBERS RANGING FROM 1 THRU N (N = NUMBER OF UNITS BEING TESTED) SPECIFYING WHICH UNITS ARE TO BE TESTED. THE LOGICAL UNIT NUMBER DESIGNATES THE POSITION OF THE P-TABLE IN CORE, ACCORDING TO THE ORDER IN WHICH THEY WERE BUILT. THE UNITS SPECIFIED MUST NOT HAVE BEEN DROPPED BY THE OPERATOR DROP COMMAND. THE UNIT-LIST DEFAULTS TO "ALL THAT HAVE NOT BEEN DROPPED BY OPERATOR COMMAND". THE EFFECT OF THE UNIT-LIST LASTS UNTIL THE NEXT START (WHERE IT IS AUTOMATICALLY RESET TO "ALL") OR THE NEXT RESTART.

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. DEFALT 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

****
CCI/TEST:TEST-LIST/PASS:PASS-CNT/FLAGS:FLAG-LIST/EOP:EOP-INCR
****

THE DIAGNOSTIC EXECUTES THRU ALL OPERATOR DIALOGUE AND HALTS AT THE HARDWARE TEST CODE. NOW THE OPERATOR CAN DUMP THE CORE IMAGE TO THE MEDIUM WITH A BIC EXTENSION.

THE BIC FILE MUST BE HANDLED DIFFERENTLY DEPENDING ON WHETHER IT IS RUN MANUALLY OR IN CHAIN MODE. IF RUN MANUALLY IT CAN BE INVOKED EITHER WITH A "START" (IN WHICH CASE IT WILL BEHAVE LIKE THE BIN FILE: THE PRE-GENERATED ANSWERS TO OPERATOR QUESTIONS WILL BE IGNORED) OR WITH A "RESTART" (IN WHICH CASE THE PRE-GENERATED OPERATOR ANSWERS WILL BE USED).

IF RUN IN CHAIN MODE, AUTOMATIC EXECUTION WILL COMMENCE IMMEDIATELY FROM THE XXDP COMMAND ".R DIAG". THE COMMAND PROMPT "DS-B>" WILL NOT BE ISSUED.

ANY SWITCHES SPECIFIED ON THE CCI COMMAND WILL CARRY OVER WHEN THE BIC FILE IS RUN IN CHAIN MODE (EXCEPT THAT UAM IS ALWAYS SET THERE) BUT WILL NOT CARRY OVER WHEN IT IS RUN MANUALLY.

TO DO A CCI ON A FULL SIZED DIAGNOSTIC (14.5K WORDS), A MACHINE SIZE LARGER THAN 16K IS REQUIRED. THE EXACT SIZE NEEDED DEPENDS ON WHICH UTILITY IS USED TO EXECUTE THE DIAGNOSTIC AT CCI TIME.

****
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.

****
ADD/UNITS:UNIT-LIST
****

THE UNITS SPECIFIED ARE ADDED BACK (THEY MUST HAVE BEEN PREVIOUSLY DROPPED BY THE DROP COMMAND) TO THE TEST SEQUENCE. AN ADD CANNOT BE FOLLOWED BY A PROCEED.

****
PRI(NT)
****

ALL STATISTICS TABLES ACCUMULATED BY THE DIAGNOSTIC ARE PRINTED. THE ISR (INHIBIT STATISTICAL REPORTING) FLAG IS CLEARED.

****
DIS(PLAY)/UNITS:<UNIT-LIST>
****

THE HARDWARE P-TABLES FOR ALL UNITS UNDER TEST ARE PRINTED OUT IN THE FORMAT IN WHICH THEY WERE ENTERED. ANY UNITS THAT WERE DROPPED BY THE OPERATOR "DROP" COMMAND ARE SO DESIGNATED.

****
FLA(GS)
****

THE CURRENT SETTINGS OF ALL FLAGS ARE PRINTED.

****
ZFL(AGS)
****

ALL FLAGS ARE CLEARED.

2.4 EXTENDED P-TABLE DIALOGUE

THE FULL CAPABILITY OF THE HARDWARE DIALOGUE IS REVEALED BY THE FOLLOWING DISCUSSION OF WHAT HAPPENS INTERNALLY.

AS SOON AS THE QUESTION "# UNITS?" IS ANSWERED (WITH THE NUMBER N), SPACE IN CORE IS ALLOCATED FOR "N" P-TABLES. ALL OF THE P-TABLES ARE OF THE SAME FORMAT, AND THERE IS A ONE-TO-ONE CORRESPONDENCE BETWEEN THE HARDWARE PARAMETER QUESTIONS AND THE SLOTS IN THE P-TABLE FORMAT.

IN GIVING A STRING OF VALUES, COMMAS WITHOUT INTERVENING VALUES MAY 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 TRANLATES 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 RL0X 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 1
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 5
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 RESPONCE.

RL11 (L) Y?

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

BUS ADDRESS (O) 174400?

ANSWER WITH THE BUS ADDRESS OF THE CONTROLLER.

VECTOR (O) 160?

ANSWER WITH THE INTERRUPT VECTOR OF THE CONTROLLER.

DRIVE (O) 0?

ANSWER WITH THE DRIVE(S) CONNECTED TO THE CONTROLLER

DRIVE TYPE = RL01 (L) ?

ANSWER NO (N) IF DRIVE IS AN RL02

BR LEVEL (O) 5?

ANSWER WITH THE INTERRUPT PRIORITY OF THE CONTROLLER.

2.6 SOFTWARE PARAMETERS

THE FOLLOWING QUESTIONS ARE ASKED IF REQUESTED ON A START, RESTART, OR CONTINUE. THEY ALLOW FLEXABILITY 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.

EXECUTE MANUAL INTERVENTION TESTS (N)?

IF "YES", SEEK TIMING, ROTATIONAL TIMING, AND WRITE LOCK ERROR AND DATA PROTECTION TESTS ARE EXECUTED. THE ONLY TEST THAT ACTUALLY REQUIRES MANUAL INTERVENTION IS THE WRITE LOCK TEST AND THAT TEST WILL BYPASS AUTOMATICALLY AFTER WAITING 30 SECONDS FOR WRITE LOCK TO BE SET.

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.

DROP DRIVE IF NO RESPONSE (N)?

IF THIS PARAMETER IS SPECIFIED AS YES, THE PROGRAM WILL CHECK IF THE DRIVE IS READY OR IF IT WILL RESPOND TO A GET STATUS BEFORE TESTING STARTS ON THAT DRIVE. IF IT IS NOT READY AND WILL NOT RESPOND TO A GET STATUS THE DRIVE IS DROPPED AND A MESSAGE IS PRINTED.

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 DE-

TECTED 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:

RESULT:(VAR 1) IS (VAR 2) SB (VAR 3) (OPTIONAL QUALIFIER)
WHERE VARIABLE 1 CAN BE ONE OF THE FOLLOWING:

CONT ERR	(CONTROLLER ERROR)
DRV ERR	(DRIVE ERROR)
NON-EXSTNT MEM	(NON-EXISTANT MEMORY)
HDR CRC	(HEADER CRC ERROR)
DATA CRC	
HDR NOT FND	(HEADER NOT FOUND)
DATA LATE	
HDR NOT FND/HDR CRC/OPI	(ALL 3 BITS SET)
DRV RDY	(DRIVE READY)
SELECTED HEAD	
VOL CHK	(VOLUME CHECK)
COVER OPEN	
BRUSH HME	(BRUCH HOME)
WRT LCK	(WRITE LOCK)
HDS OUT	(HEADER OUT)
DRV SEL ERR	(DRIVE SELECT ERROR)
DRV STATE	(DRIVE STATE)
SPIN TIMEOUT	(SPINDLE TIMEOUT SPD ERROR)
WRT GAT ERR	(WRITE GATE ERROR)
SEEK TIMEOUT	(SKTO ERROR)
CUR HEAD ERR	(CURRENT IN HEAD ERROR)
WRT DAT ERR	(WRITE DATA ERROR)
OP INCOMPLETE	(OPI ERROR)
HDR/DAT ERR	(HDR CRC OR DATA CRC ERROR BIT 11 OF CS REGISTER)
HDR NOT FND/DAT LATE	(HDR NOT FOUND OR DATA LATE ERROR BIT 12 OF CS REGISTER)
CYL	(CYLINDER WHEN REPORTING A SEEK ERROR)

VARIABLE 2 WILL BE A VALUE THAT DEFINES WHAT THE RESULT ACTUALLY IS. THIS CAN BE A 1 OR 0 TO INDICATE A SET OF RESULT CONDITIONS, A NUMBER 0 TO 7 TO INDICATE THE DRIVE STATE, OR A NUMBER 0 TO 377 (OCTAL) TO IDENTIFY A CYLINDER NUMBER.

VARIABLE 3 DEFINES THAT THE VALUE GIVEN IS VARIABLE 2 SHOULD BE. THE OPTIONAL QUALIFIER IS PROVIDED WHEN IT IS USEFUL TO KNOW WHEN THE ERROR WAS DETECTED IN THE OPERATION BEING PERFORMED. THIS QUALIFIER IS USED

TO REPORT RESULTS SUCH AS:

```
BRUSH HME IS 1 SB 0 IN STATE 2
HEADS OUT IS 0 SB 1 IN STATE 3
DRV RDY   IS 0 SB 1 IN DATA XFER
SELECTED HEAD IS 1 SB 0 IN CYCLE UP
DRV RDY   IS 0 SB 1 IN STATE 5
DRV RDY   IS 1 SB 0 IN SEEK W/O MOTION
DRV RDY   IS 0 SB 1 IN 10MS
DRV RDY   IS 0 SB 1 IN 500MS
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 TO 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.

"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
(7)                RLCS    RLDA    RLBA    RLMP    CYL     HD
(8)   OP INIT
(9)   OP DONE
(10)       DRIVE STATUS
(11)             WORD NUM IS (XXXXXX) SB (YYYYYY)
(12)   TOTAL COMPARE ERRS: (ZZZ) OF (128)
```

THE ONLY EXCEPTION TO THE ABOVE FORMAT IS PURE DATA COMPARE ERRORS (NOT DETECTED BY READ ERROR). THEN THE FORMAT DOES NOT INCLUDE LINES 5 THROUGH 10.

LINE 1 IS THE ERROR HEADER AND IS PROVIDED BY THE SUPERVISOR. THE PROGRAM IS IDENTIFIED BY NAME WITH THE NUMBER OF TEST AND SUBTEST PRESENTLY BEING EXECUTED.

THE SUBTEST NUMBER IS UNIQUE IN THIS PROGRAM IN THAT IT DOES NOT REFER TO A PHYSICAL SUBTEST WITHIN A GIVEN TEST. RATHER IT REFLECTS THE NUMBER OF TIMES A SUBTEST HAS BEEN EXECUTED WITHIN A TEST. CONSEQUENTLY, ON A TEST THAT TESTS AN INCREMENTAL TYPE OF OPERATION (SUCH A INCREMENTAL SEEKS, READ ALL HEADERS FROM BOTH SURFACES, ETC.) THE SUBTEST WILL BE DESCRIPTIVE OF WHERE IN THE TEST THE ERROR OCCURRED.

THE ERROR P.C. IS THE PHYSICAL MEMORY LOCATION WHERE THE ERROR REPORT WAS INITIATED. SINCE MANY FUNCTIONS ARE SUBROUTINED, AND ERRORS ARE REPORTED FROM SUBROUTINES, THE ERROR P.C. IS NOT SUFFICIENT TO IDENTIFY THE LOCATION OF THE ERROR CALL AND THE ROUTINE TRACE SEQUENCE IS PROVIDED.

LINE 2 IS THE ROUTINE TRACE SEQUENCE. IF THE ERROR CALL IS INITIATED FROM WITHIN THE TEST (AS OPPOSED TO WITHIN A ROUTINE), THIS PORTION OF THE REPORT IS OMITTED. IF THE CALL IS INITIATED FROM A ROUTINE (WHICH MAY BE CALLED BY ANOTHER ROUTINE, WHICH MAY BE CALLED BY ANOTHER ROUTINE, ETC. SEVERAL LEVELS DEEP) THE ROUTINE TRACE SEQUENCE PROVIDES A TRAIL TO THE ACTUAL LOCATION WITHIN THE TEST THAT CALLED THE FIRST ROUTINE. THE FIRST ENTRY LISTED IS THE LOCATION WHERE THE FIRST ROUTINE WAS CALLED.

LINE 3 IS THE TEST DESCRIPTION AND IS ROUGHLY IDENTICAL TO THE NAME OF THE TEST BEING PERFORMED.

LINE 4 IDENTIFIES THE ACTUAL HARDWARE FUNCTION THAT IS BEING PERFORMED. ADDITIONAL INFORMATION ON THIS LINE IS DESCRIPTIVE OF SPECIFIC USE OF THE FUNCTION. FOR EXAMPLE, THE OPERATION LINE WILL READ "READ HEADERS FOR 40 HEADERS" WHEN ALL HEADERS ARE BEING READ FROM A TRACK.

LINE 5 IDENTIFIES THE ERROR THAT HAS BEEN DETECTED. THE CONTENT OF LINE 5 IDENTIFIES WHAT WAS BENG TESTED (SUCH AS DRIVE READY, CONTROLLER ERROR, DRIVE STATE, ETC.), WHAT IT IS AND WHAT IT SHOULD BE. LINE 5 MAY BE REPEATED IF MORE THAN ONE TESTED ITEM IS FOUND 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)

FOR READ/WRITE FUNCTIONS

BIT 15-7 - CYLINDER ADDRESS FOR TRANSFER
BIT 6 - SURFACE FOR TRANSFER
BIT 5-0 - SECTOR FOR TRANSFER (1-40.)

FOR SEEK FUNCTION

BIT 15-7 - DIFFERENCE TO NEW CYLINDER
BIT 6-5 - MUST BE ZERO (0)
BIT 4 - SURFACE (0=UPPER, 1=LOWER)
BIT 3 - MUST BE ZERO (0)
BIT 2 - SEEK DIRECTION(1=IN / 0=OUT)
BIT 1 - MUST BE ZERO (0)
BIT 0 - MUST BE ONE (1)

FOR GET STATUS FUNCTION

BIT 15-4 - IGNORED SHOULD BE ZERO (0)
BIT 3 - DRIVE RESET
BIT 2 - MUST BE ZERO (0)
BIT 1 - MUST BE ONE (1)
BIT 0 - MUST BE ONE (1)

RLMP - MULTIPURPOSE REGISTER

FOR READ/WRITE FUNCTION

BIT 15 - 0 - WORD COUNT (TWO'S COMPLIMENT)

FOR READ HEADER FUNCTION

BIT 15-0 - DISK HEADER OF SECTOR (FIRST READ)
- ZERO WORD (SECOND READ)
- HEADER CRC (THIRD READ)

FOR GET STATUS FUNCTION

HAS DRIVE STAIUS

BIT 15 - WRITE DATA ERROR
BIT 14 - CURRENT HEAD ERROR (CHE)
BIT 13 - WRITE LOCK STATUS (WL)
BIT 12 - SEEK TIME OUT (SKTO)
BIT 11 - SPIN ERROR (SPE)
BIT 10 - WRITE GATE ERROR (WGE)
BIT 9 - VOLUME CHECK (VC)
BIT 8 - DRIVE SELECT ERROR (DSE)
BIT 7 - DRIVE TYPE IS RL02 IF SET
BIT 6 - SURFACE (0=UPPPER, 1=LOWER)
BIT 5 - COVER OPEN
BIT 4 - HEADS HOME
BIT 3 - BRUSHES HOME
BIT 2-0 -STATE BITS
0 - LOAD STATE
1 - SPIN UP
2 - BRUSH CYCLE
3 - LOAD HEADS
4 - SEEK - TRACK COUNTING
5 - SEEK - LINEAR MODE
6 - UNLOAD HEADS
7 - SPIN DOWN

6.0 TEST SUMMARIES

TEST 1 OUTER GUARD BAND DETECTION TEST

DO READ HEADER, WAIT FOR INTERRUPT. CHECK IF AT CYLINDER 0. IF NOT, SEEK REVERSE 1 CYLINDER AT A TIME UNTIL CYLINDER 0 IS REACHED. IF ANY REVERSE SEEK FAILS TO MOVE THE HEADS IN 10 TRIES:

DETECTION OF GUARD BAND PREMATURE.

WHEN AT CYLINDER 0, DO SEEK DIFFERENCE OF 1, SIGN 0, HEAD 0. WAIT FOR INTERRUPT, WAIT FOR READY. READY SHOULD SET IN 20MS>T>15MS. IF NOT:

FAILED TO DETECT GUARD BAND

DO READ HEADER. WAIT FOR INTERRUPT. CHECK FOR CYLINDER 0. IF NOT:

FAILED TO SEEK BACK TO ZERO

DO SEEK WITH DIFFERENCE OF 1, SIGN 0, HEAD 1. DO SAME TESTS

AS ABOVE WITH REGARD TO READY VS TIME AND CYLINDER FOUND IN HEADER.

NOTE: CHOOSING A SINGLE SURFACE WILL LIMIT THE TESTING TO THAT SURFACE.

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).

NOTE: 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 4 INCREMENTAL FORWARD SEEK HEAD 1 TEST
******

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

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

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 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 0 IS CHOSEN.

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

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

WHEN AT MAX CYLINDER, DO SEEK WITH DIFFERENCE OF 1, SIGN 1, HEAD 0. WAIT FOR INTERRUPT, WAIT FOR DRIVE READY. READY SHOULD SET IN 20MS>T>15MS. IF NOT:

FAILED TO DETECT GUARD BAND

DO READ HEADER. WAIT FOR INTERRUPT. CHECK FOR MAX. CYLINDER IF NOT:

FAILED TO SEEK BACK TO MAX CYLINDER

DO SEEK WITH DIFFERENCE OF 1, SIGN 1, HEAD 1. DO SAME TESTS AS ABOVE.

NOTE: CHOOSING A SINGLE SURFACE WILL LIMIT THE TESTING TO THAT SURFACE.

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

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

DO SEEK WITH DIFFERENCE OF 1, SIGN 0, HEAD 1. 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 1: IF PROGRAM MODE 2 IS USED AND 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 0 IS CHOSEN.

TEST 7 SEEK TESTS
******

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

DO READ HEADER, RECORD POSITION. DO SEEK WITH DIFFERENCE OF 2 (MAX DISTANCE AT 3 IPS), SIGN 1, HEAD 0. DO READ HEADER, CHECK NEW CYLINDER IS OLD CYLINDER + DISTANCE. IF NOT:

TRACK CROSSING DETECTION FAILURE
DIFFERENCE COUNTER FAILURE
COUNT PULSE GENERATION FAILURE
VELOCITY ROM FAILURE

REPEAT ABOVE UNTIL OLD CYLINDER + DISTANCE > MAX. POSITION AT MAX.

DO READ HEADER, RECORD POSITION. DO SEEK WITH DIFFERENCE OF 2 (MAX DISTANCE AT 3 IPS), SIGN 0, HEAD 0. DO READ HEADER, CHECK NEW CYLINDER IS OLD CYLINDER - DISTANCE. IF NOT:

TRACK CROSSING DETECTION FAILURE

REPEAT UNTIL OLD CYLINDER - DISTANCE < 0. REPEAT ALL OF THE ABOVE USING HEAD 1.

REPEAT ALL OF THE ABOVE TESTS USING THE FOLLOWING DISTANCES: 2, 6, 9, 12, 17, 22, 27, 34, 41, 128, 256 FOR RL01 OR 4, 12, 18, 24, 34, 44, 54, 68, 82, 256, 512 FOR RL02. THESE DISTANCES ARE SPECIFIED BECAUSE THEY REPRESENT THE MAXIMUM DISTANCE FOR EACH VELOCITY LEVEL USED IN THE DRIVE.

NOTE: TESTING WILL BE DONE BETWEEN UPPER AND LOWER CYLINDER LIMITS. CHOOSING A SINGLE SURFACE WILL LIMIT TESTING TO THAT SURFACE.

TEST 8 FORWARD OSCILLATING SEEK TEST
******

POSITION HEADS AT CYLINDER 0.

DO OSCILLATING SEEK USING HEAD 0 (SEEK FROM 0 TO 1 TO 0, 0 TO 2 TO 0, 0 TO 3 TO 0, ...0 TO MAX CYL TO 0). AFTER EACH SEEK READ HEADER AND VERIFY POSITION.

REPEAT TEST USING HEAD 1.

NOTE: IF EITHER CYLINDER LIMIT IS SPECIFIED, THE TEST WILL SEEK BETWEEN UPPER AND LOWER LIMITS FOR EACH SURFACE. CHOOSING A SINGLE SURFACE WILL LIMIT TESTING TO THAT SURFACE. NOTE THAT LOOPING ON TEST THEN PROVIDES A FIXED DISTANCE SEEK LOOP.

TEST 9 REVERSE OSCILLATING SEEK TEST
******

POSITION HEADS AT MAX CYLINDER. DO OSCILLATING SEEK USING HEAD 0. (IF RL01 SEEK FROM 255 TO 254 TO 255, 255 TO 253 TO 255,....255 TO 0 TO 255.) AFTER EACH SEEK READ HEADER AND VERIFY POSITION.

REPEAT TEST USING HEAD 1.

NOTE: IF EITHER CYLINDER LIMIT IS SPECIFIED, THE TEST WILL SEEK BETWEEN UPPER AND LOWER LIMITS FOR EACH SURFACE.

CHOOSING A SINGLE SURFACE WILL LIMIT TESTING TO THAT SURFACE. NOTE THAT LOOPING ON TEST THEN PROVIDES A FIXED DISTANCE SEEK LOOP.

TEST 10 SEEK TIMING
****

POSITION HEADS AT CYLINDER 0.

DO 64 SEEKS FROM 0 TO 1 AND 1 TO 0, MEASURING THE SEEK TIME FOR EACH SEEK. AVERAGE THE SEEK TIMES (FORWARD AND REVERSE INDEPENDENTLY) AND REPORT.

REPEAT ABOVE SEEKING BETWEEN CYLINDER 127 TO 128 AND 254 TO 255 FOR RL01 AND 255 TO 256 AND 256 TO 511 FOR RL02.

REPEAT ABOVE SEEKING BETWEEN CYLINDER 0 TO 127 AND 128 TO 256 FOR RL01 AND CYLINDER 0 TO 256 AND 256 TO 511 FOR RL02.

REPEAT ABOVE SEEKING BETWEEN CYLINDER 0 AND 255 FOR RL01 AND 0 TO 511 FOR RL02.

THE SEEK TIMES WILL BE REPORTED AS SHOWN BELOW. THE TIME MEASURED IS FROM START OF SEEK COMMAND UNTIL INTERRUPT IS RECEIVED.

	INNER	MIDDLE	OUTER	EXPECTED
1 CYL FWD	X	X	X	X
1 CYL REV	X	X	X	X
MID CYL FWD	X		X	X
MID CYL REV	X		X	X
MAX CYL FWD		X		X
MAX CYL REV		X		X

THE X INDICATES WHERE TIME WILL BE REPORTED.

NOTE: THE ABOVE REPORT WILL BE PRINTED IN THE FIRST PASS FOR EACH DRIVE UNDER TEST IF MANUAL INTERVENTION TESTS WERE RUN. THE EXPECTED TIMES ARE FOR USER COMPARISON ONLY. THE PROGRAM WILL NOT REPORT DEVIATION AS AN ERROR.

TEST 11 BASIC READ DATA TEST
****

POSITION HEADS AT MAX CYLINDER.

DO READ DATA, HEAD 1. CHECK FOR ANY ERRORS AND REPORT. IF ERROR, READ SECTOR 1 THROUGH 19 UNTIL NO ERROR ON READ. REPORT ALL ERRORS BUT DO NOT INCREMENT ERROR COUNT. IF NONE CAN BE READ, SUCCESSFULLY, REPORT THAT FACTORY BAD SECTOR FILE

CANNOT BE READ, INCREMENT ERROR COUNT AND PROCEED WITH READ OF SECTOR 20.

ON SECTOR WITH NO CRC ERROR, VERIFY DATA FORMAT (WORD 0 AND 1 ARE NOT 0, WORD 2 AND 3 ARE 0, LOCATE FIRST WORD OF ALL ONE'S AND THAT WORD TO WORD 127 ARE ALL ONE'S.) STORE BAD SECTOR DATA.

READ DATA, HEAD ONE, SECTOR 20. CHECK FOR ANY ERRORS AND

REPORT. IF ERROR, READ SECTOR 21 THROUGH 39 UNTIL NO ERROR ON READ. REPORT ALL ERRORS BUT DO NOT INCREMENT ERROR COUNT. IF NONE CAN BE READ SUCCESSFULLY, REPORT THAT SOFTWARE BAD SECTOR FILES CANNOT BE READ, INCREMENT ERROR COUNT AND EXIT TEST.

ON SECTOR WITH NO CRC ERROR, VERIFY DATA AS ABOVE. STORE BAD SECTOR DATA.

NOTE: IF SURFACE 0 IS SELECTED THIS TEST WILL BE BYPASSED.

TEST 12 WRITE/READ DATA TEST (PART 1)
****

POSITION HEADS AT CYLINDER 0

WRITE PATTERN 1 ON HEAD 0, SECTOR 0. CHECK FOR ANY ERROR.

READ HEAD 0, SECTOR 0. CHECK FOR CRC ERROR. COMPARE DATA.

REPEAT FOR OTHER DATA PATTERNS (2 THROUGH 8).

CHECK IF CYLINDER 0, TRACK 1, SECTOR 0 IS LISTED IN BAD SECTOR DATA. IF NOT, REPEAT ABOVE TEST AT CYLINDER 0, TRACK 1, SECTOR 0. IF IT IS LISTED AS BAD, LOCATE FIRST SECTOR 0, TRACK 1 THAT IS GOOD AND DO ABOVE TESTS.

NOTE: CYLINDER LIMITS ARE IGNORED, TESTING IS DONE AT CYLINDER 0. HOWEVER, CHOOSING A SINGLE SURFACE WILL LIMIT TESTING TO THAT SURFACE.

TEST 13 SPINDLE TIMING TEST
****

POSITION HEADS TO CYLINDER 0.

DO WRITE DATA TO CYLINDER 0, HEAD 0, SECTOR 0. WAIT FOR INTERRUPT.

DO WRITE DATA TO CYLINDER 0, HEAD 0, SECTOR 0. START TIMING. WHEN INTERRUPT OCCURS, STOP TIMING. RESULT IS SPINDLE ROTATION TIME.

REPEAT TEST 64 TIMES. REPORT THE AVERAGE AS SPINDLE ROTATION

TIME. THE TIME REPORTED IS IN 100'S OR MICROSECONDS.

NOTE: THIS TEST WILL BE RUN ONLY IN THE FIRST PASS AND ONLY IF MANUAL INTERVENTION TESTS WERE RUN.

TEST 14 WRITE/READ TEST (PART 2)
****

CC IS CURRENT CYLINDER SELECTED FROM SET.
LET SELECTED CYLINDER SET BE AS DEFINED IN PARAGRAPH 4.3.

SEEK FORWARD TO CC. WRITE PATTERNS 1 THROUGH 8 REPEATED 5 TIMES ON HEAD 0. READ/COMPARE ALL DATA.

SEEK REVERSE TO "LOLIMIT". SEEK FORWARD TO CC. READ/COMPARE ALL DATA. SEEK FORWARD TO "HILIMIT". SEEK REVERSE TO CC. READ/COMPARE ALL DATA. REWRITE DATA PATTERNS 1 THROUGH 8 REPEATED 5 TIMES ON HEAD 0. READ COMPARE ALL DATA.

SEEK FORWARD TO "HILIMIT". SEEK REVERSE TO CC. READ/COMPARE ALL DATA. SEEK REVERSE TO "LOLIMIT". SEEK FORWARD TO CC. READ/COMPARE ALL DATA.

REPEAT ABOVE TEST FOR HEAD 1.

REPEAT ABOVE TESTS FOR ALL CYLINDERS IN SELECTED CYLINDER SET.

NOTE 1: IF ANY OF THE SECTORS IN THE SELECTED CYLINDER SET ARE LISTED AS BAD, THAT SECTOR WILL BE BYPASSED.

NOTE 2: IF THE "USE ALL CYLINDERS" PARAMETER IS SPECIFIED AS "Y", THE TEST WILL INCLUDE ALL CYLINDERS IN THE SELECTED PARAMETER SET.

NOTE 3: IN THE FIRST PASS OF THE PROGRAM THIS TEST IS EXECUTED ON ONLY 6 OF THE CYLINDERS LISTED IN THE CYLINDER SET. THOSE USED WILL BE EVERY 8TH ENTRY IN THE TABLE. ON THE SECOND AND SUBSEQUENT PASSES ALL ENTRIES IN THE SELECTED CYLINDER SET ARE USED.

NOTE 4: TESTING WILL BE DONE BETWEEN UPPER AND LOWER LIMITS. CYLINDERS IN THE CYLINDER SET BEYOND THESE LIMITS WILL NOT BE TESTED. CHOOSING A SINGLE SURFACE WILL LIMIT TESTING TO THAT SURFACE.

TEST 15 WRITE LOCK ERROR AND DATA PROTECTION TEST
****

DO WRITE DATA PATTERN 0 AT SECTOR 0. READ DATA AND VERIFY.

ASK OPERATOR TO WRITE LOCK DRIVE. DO GET STATUS LOOP UNTIL WRITE LOCK IS SET. IF NOT SET IN 30 SECONDS, ABORT THE TEST.

WHEN WRITE LOCK IS SET, DO WRITE DATA PATTERN 1 AT SECTOR 0. REPORT FAILURE IF DRIVE ERROR DOES NOT SET OR IF ANY OTHER ERROR SETS. CLEAR ERROR AND READ DATA AT SECTOR 0. CHECK THAT DATA HAS NOT BEEN DISTURBED.

REQUEST OPERATOR TO RESET WRITE LOCK. DO GET STATUS LOOP UNTIL WRITE LOCK IS RESET. IF NOT RESET IN 30 SECONDS, REPEAT THE REQUEST.

NOTE: THIS TEST IS EXECUTED ONLY IF THE PROGRAM OPERATION MODE 2 IS SELECTED, MANUAL INTERVENTION TESTING IS REQUESTED, AND IS RUN IN FIRST PASS ONLY.

TEST 16 ADJACENT CYLINDER INTERFERENCE TEST
****

CC IS CURRENT CYLINDER SELECTED FROM SET
LET SELECTED CYLINDER SET BE AS DEFINED IN PARAGRAPH 4.3.
DATA PATTERN IS 155555.

SEEK FORWARD TO CYLINDER CC. WRITE PATTERN ON TRACK 0, ALL SECTORS. READ/COMPARE DATA.

SEEK FORWARD TO "HILIMIT". SEEK REVERSE TO CC-1. WRITE PATTERN. SEEK FORWARD TO "HILIMIT". SEEK REVERSE TO CC. WRITE PATTERN. (THIS HAS BRACKETED ORIGINAL WRITE WITH WRITES IN ADJACENT CYLINDERS. NOTE ADJACENT CYLINDERS WERE WRITTEN AFTER HEADS CAME ON CYLINDER IN REVERSE DIRECTION WHICH IS OPPOSITE OF CENTER CYLINDER.)

SEEK REVERSE TO "LOLIMIT". SEEK FORWARD TO CC. READ/COMPARE DATA FROM ALL SECTORS. ANY ERRORS (READ OR COMPARE) ARE ATTRIBUTED TO ADJACENT CYLINDER INTERFERENCE.

SEEK FORWARD TO "HILIMIT". SEEK REVERSE TO CC. WRITE DATA PATTERN. SEEK REVERSE TO "LOLIMIT". SEEK FORWARD TO CC-1, WRITE PATTERN. SEEK REVERSE TO "LOLIMIT", SEEK FORWARD TO CC+1, WRITE PATTERN. SEEK FORWARD TO "HILIMIT", SEEK REVERSE TO CC. READ/COMPARE DATA IN ALL SECTORS. ANY ERRORS (READ OR COMPARE) ARE ATTRIBUTED TO ADJACENT CYLINDER INTERFERENCE.

REPEAT ABOVE TESTS ON HEAD 1.

NOTE 1: IF ANY SECTOR ON A SELECTED CYLINDER IS LISTED BAD, THAT SECTOR WILL BE BYPASSED.

NOTE 2: IF THE "USE ALL CYLINDERS" PARAMETER IS SPECIFIED AS "Y", THE TEST WILL INCLUDE ALL CYLINDERS (EXCEPT 0 AND MAX CYL) IN THE SELECTED PARAMETER SET.

NOTE 3: IN THE FIRST PASS OF THE PROGRAM THIS TEST IS EXECUTED ON ONLY 3 OF THE CYLINDERS LISTED IN THE CYLINDER SET. THOSE USED WILL BE THE FIRST,

TWENTYFIRST, AND FORTYFIRST ENTRIES IN THE TABLE. ON SECOND AND SUBSEQUENT PASSES EVERY FOURTH CYLINDER SET ENTRY WILL BE TESTED.

NOTE 4: TESTING WILL BE DONE BETWEEN UPPER AND LOWER LIMITS. CYLINDERS IN THE CYLINDER SET BEYOND THESE LIMITS WILL NOT BE TESTED. CHOOSING A SINGLE SURFACE WILL LIMIT TESTING TO THAT SURFACE.

TEST 17 OVERWRITE TEST
****

CC IS CURRENT CYLINDER SELECTED FROM SET
SELECTED CYLINDER SET DEFINED IN PARAGRAPH 4.3.
PATTERN A = 125252
PATTERN B = 000000

SEEK FORWARD TO CC. WRITE DATA OF PATTERN A IN ALL SECTORS, HEAD 0. READ/COMPARE DATA.

SEEK FORWARD TO "HILIMIT", SEEK REVERSE TO CC. WRITE PATTERN B. SEEK REVERSE TO "LOLIMIT", SEEK FORWARD TO CC, READ/COMPARE DATA.

SEEK FORWARD TO "HILIMIT", SEEK REVERSE TO CC. WRITE DATA PATTERN A. READ/COMPARE DATA. SEEK REVERSE TO "LOLIMIT", SEEK FORWARD TO CC. WRITE PATTERN B. SEEK FORWARD TO "HILIMIT" SEEK REVERSE TO CC. READ/COMPARE DATA.

ANY FAILURES (READ OR COMPARE) ARE ATTRIBUTED TO OVERWRITE PROBLEM.

REPEAT ABOVE TESTS ON HEAD 1.

NOTE 1: IF ANY SECTOR ON A SELECTED CYLINDER IS LISTED AS BAD, THAT SECTOR WILL BE BYPASSED.

NOTE 2: IF THE "USE ALL CYLINDERS" PARAMETER IS SPECIFIED AS "Y", THE TEST WILL INCLUDE ALL CYLINDERS IN THE SELECTED PARAMETER SET.

NOTE 3. IN THE FIRST PASS OF THE PROGRAM THIS TEST IS EXECUTED ON ONLY 3 OF THE CYLINDERS LISTED IN THE CYLINDER SET. THOSE USED WILL BE THE FIRST, TWENTYFIRST, AND FORTYFIRST ENTRIES IN THE TABLE. ON SECOND AND SUBSEQUENT PASSES EVERY FOURTH CYLINDER SET ENTRY WILL BE TESTED.

NOTE 4: TESTING WILL BE DONE BETWEEN UPPER AND LOWER LIMITS. CYLINDERS IN THE CYLINDER SET BEYOND THESE LIMITS WILL NOT BE TESTED. CHOOSING A SINGLE SURFACE WILL LIMIT TESTING TO THAT SURFACE.

TABLE OF CONTENTS

ASSEMBLY ROUTINES MACRO V03.01 9-FEB-79 19:23:55 PAGE 1

SEQ 0043

```
1          000001.                 PART2==1
```

```
                                        .NLIST  CND,MD,ME
002000                                  .ENABLE ABS
                                        .ENABLE AMA
        002000                          .=2000
                                        .MCALL  SVC

002000                                  SVC
        000001                          SVCTST=1
        000001                          SVCSUB=1
        000001                          SVCBGL=1
        000000                          SVCINS=0
        000000                          SVCTAG=0
002000                                  POINTER BGNSW,BGNSFT,BGNDU

002000                          BGNMOD  MDHEDR
002000                          HEADER  CZRLJ,A,0,30000,30000,300,RL01
002000      103                         .ASCII /C/
002001      132                         .ASCII /Z/
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/
002012   000000                         .WORD   0
002014   000300                         .WORD   300
002016   037532                         .WORD   L$HARD
002020   037676                         .WORD   L$SOFT
002022   013330                         .WORD   L$HW
002024   013346                         .WORD   L$SW
002026   040336                         .WORD   L$LAST
002030   000000                         .WORD   0
002032   000000                         .WORD   0
002034   000000                         .WORD   0
002036   000000                         .WORD   0
002040   013364                         .WORD   L$DISPATCH
002042   000000                         .WORD   0
002044   000000                         .WORD   0
002046   000000                         .WORD   0
002050      002                         .BYTE   C$REVISION
002051      002                         .BYTE   C$EDIT
002052   030000                         .WORD   30000
002054   030000                         .WORD   30000
002056   000000                         .WORD   0
002060   000000                         .WORD   0
002062   000000                         .WORD   0
002064   002114                         .WORD   L$DVTYP
002066   000000                         .WORD   0
002070   002112                         .WORD   L$DR
002072   002112                         .WORD   L$DRST
002074   000000                         .WORD   0
002076   014700                         .WORD   L$DU
002100   000014                         .WORD   14
002102   000000                         .WORD   0
002104   013426                         .WORD   L$INIT
```

```
     002106  014560                          .WORD   L$CLEAN
  22 002110                          ENDMOD
  23 002110                                  DEVREG
     002110  000000                          .WORD   0
                                              .BLKW
  24 002114                                  DEVTYP  <RL01,RL02>
     002114      122     114     060          .ASCIZ  /RL01,RL02/
     002117      061     054     122
     002122      114     060     062
     002125      000
                                              .EVEN
  25
  26                                          ;COPYRIGHT (C) 1978, 1977
  27                                          ;THIS SOFTWARE IS FURNISHED UNDER LICENSE FOR USE ONLY
  28                                          ;ON A SINGLE COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH
  29                                          ;THE INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS
  30                                          ;SOFTWARE, OR ANY COPIES THEREOF, MAY NOT BE PROVIDED
  31                                          ;OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT
  32                                          ;FOR USE ON SUCH SYSTEM, AND TO ONE WHO AGREES TO THESE
  33                                          ;LICENSE TERMS. TITLE TO OWNERSHIP OF THE SOFTWARE SHALL
  34                                          ;AT ALL TIMES REMAIN IN DEC.
  35                                          ;
  36                                          ;THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE
  37                                          ;WITHOUT NOTICE AND SHALL NOT BE CONSTRUED AS A COMMITMENT
  38                                          ;BY DIGITAL EQUIPMENT CORPORATION.
  39                                          ;
  40                                          ;DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY
  41                                          ;OF ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
  42
  43 002126                          BGNMOD  GLBEQAT
  44
  45 002126                                  EQUALS
  46                                  ;       OFFSETS FOR HARDWARE P-TABLE
  47          000000                  CSR     =0                      ;BUS ADDRESS
  48          000002                  VECT    =2                      ;VECTOR ADDRESS
  49          000004                  PRIOR   =4                      ;PRIORITY
  50          000006                  TYPDR=6
  51          000010                  DRSB    =10                     ;DRIVE SELECT BIT
  52          000012                  CNT     =12                     ;CONTROLLER TYPE
  53
  54                                  ;       OFFSET FOR SOFTWARE P-TABLE
  55          000000                  MISWI   =0                      ;SOFTWARE PARAMETERS SWITCHES
  56          000002                  LOLIM   =2                      ;CYLINDER LOWER LIMIT
  57          000004                  HILIM   =4                      ;CYLINDER HIGH LIMIT
  58          000006                  HEAD    =6                      ;SELECTED HEAD FOR RUNNING TESTS
  59          000010                  ERLIM   =10                     ;ERROR LIMIT
  60          000012                  DCLIM   =12                     ;DATA COMPARE ERROR LIMIT
  61
  62                                  ;       BIT ASSIGNMENT FOR SOFTWARE P-TABLE SWITCHES
  63          000001                  ALLCYL  =BIT00                  ;USE ALL CYLINDERS
  64          000002                  ALLSEC  =BIT01                  ;USE ALL SECTORS
  65          000004                  DRSELT  =BIT02                  ;EXECUTE DRIVE SELECT TEST
  66          000010                  HDALIGN =BIT03                  ;EXECUTE HEAD ALIGNMENT TEST
  67          000020                  AUTOSZ  =BIT04                  ;AUTO SIZE FOR DRIVE-DROP IF NO RESPONSE
  68          010000                  HEADLM  =BIT12                  ;HEAD LIMIT SPECIFIED FLAG
  69          020000                  HICYL   =BIT13                  ;HI LIMIT SPECIFIED FLAG
  70          040000                  LOCYL   =BIT14                  ;LO LIMIT SPECIFIED
```

```
                100000          MITEST  =BIT15                  ;EXECUTE MANUAL INTERVENTION TESTS
                                ;       SUBSYSTEM FUNCTIONS
                000102          CKDATA  =102                    ;WRITE CHECK
                000104          GTSTAT  =104                    ;GET STATUS
                000106          SEEK    =106                    ;SEEK
                000110          RDHEAD  =110                    ;READ HEADER
                000112          WTDATA  =112                    ;WRITE DATA
                000114          RDDATA  =114                    ;READ DATA
                000116          RDNOHR  =116                    ;READ DATA, IGNORE HEADERS
                000100          NOOP    =100                    ;NO OPERATION
                                ;       OPERATION FLAGS
                007777          COMPOP  =7777                   ;COMPOSITE OPERATION FLAGS
                000002          HDRCMP  =BIT01                  ;HEADER COMPARE OPERATION
                000001          DATACMP =BIT00                  ;DATA COMPARE OPERATION
                000004          CYLUP   =BIT02                  ;CYCLE UP OPERATION
                000010          ULOAD   =BIT03                  ;UNLOAD OPERATION
                000020          INOUTS  =BIT04                  ;IN-OUT SEEK OPERATION
                000040          OUTINS  =BIT05                  ;OUT-IN SEEK OPERATION
                000100          FOLWRT  =BIT06                  ;FOLLOWING WRITE OPERATION
                000200          REVSKS  =BIT07                  ;REV SEEK SEQ (ADJ INTERFERENCE)
                000400          FWDSKS  =BIT08                  ;FWD SEEK SEQ (ADJ INTERFERENCE)
                001000          REVSKO  =BIT09                  ;REV SEEK SEQ (OVERWRITE)
                002000          FWDSKO  =BIT10                  ;FWD SEEK SEQ (OVERWRITE)
                004000          BADADD  =BIT11                  ;BAD DISK ADDRESS
                010000          SEEKOP  =BIT12                  ;SEEK OPERATION
                020000          RORWOP  =BIT13                  ;READ OR WRITE OPERATION
                040000          RELDWT  =BIT14                  ;RELOAD WAIT
                100000          HDR40   =BIT15                  ;40 HEADER OPERATION
                003760          MQUALS  =OUTINS!INOUTS!FOLWRT!REVSKS!FWDSKS!REVSKO!FWDSKO
                                                                ;MESSAGE QUALIFIER BITS

                                ;       ERROR FLAGS FROM SUBROUTINES
                000001          TOSLOW  =BIT00                  ;OPERATION TOOK TO LONG
                000002          NOIRPT  =BIT01                  ;NO INTERRUPT FROM OPERATION
                000004          CONHNG  =BIT02                  ;CONTROLLER HUNG
                000010          NOCLR   =BIT03                  ;BAD CONTROLLER CLEAR

                000000          RLCS    =0              ;CONTROL AND STATUS REGISTER
                000002          RLBA    =2              ;BUS ADDRESS REGISTER
                000004          RLDA    =4              ;DISK ADDRESS REGISTER
                000006          RLMP    =6              ;MULTI-PURPOSE REGISTER

                                ;       REGISTER BIT DEFINITIONS - CONTROL STATUS REGISTER
                000000          RLCSR   =0                      ;CONTROL AND STATUS REGISTER
                100000          ANYERR  =100000                 ;ANY ERROR BIT
                040000          DRVERR  =40000                  ;DRIVE ERROR BIT
                020000          NXMERR  =20000                  ;NON-EXISTANT MEMORY ERROR
                010000          DLTERR  =10000                  ;DATA LATE ERROR
                010000          HNFERR  =10000                  ;HEADER NOT FOUND ERROR
                004000          DCKERR  =4000                   ;DATA CHECK ERROR
                004000          HCRCERR =4000                   ;HEADER CHECK ERROR
                002000          OPIERR  =2000                   ;OPERATION INCOMPLETE ERROR
                001400          DSMSK   =1400                   ;DRIVE SELECT MASK
                000200          CRDYMSK =200                    ;CONTROLLER READY MASK
                000100          INTEBL  =100                    ;INTERRUPT ENABLE MASK
```

```
                                BAMSK   =60                     ;BUS ADDRESS UPPER MASK
                                DRDYMSK =1                      ;DRIVE READY MASK

                                ;       REGISTER BIT DEFINITIONS - DISK ADDRESS FOR DATA XFER
                                SAMSK   =77                     ;SECTOR ADDRESS MASK
                                HSMSK   =100                    ;HEAD SELECT MASK

                                ;       REGISTER BIT DEFINITIONS - DISK ADDRESS FOR SEEK
                                MBSET0  =1                      ;MUST BE SET, BIT 0
                                DIRBIT  =4                      ;DIRECTION BIT
                                HDSEL   =20                     ;HEAD SELECT BIT

                                ;       REGISTER BIT DEFINITIONS - DISK ADDRESS FOR GET STATUS
                                GETSTAT =3                      ;GET STATUS SETUP
                                DRSET   =10                     ;DRIVE RESET MASK

                                ;       REGISTER BIT DEFINITIONS - MP FOR DATA XFER
                                WCMSK   =17777                  ;WORD COUNT MASK
                                WCRNG   =160000                 ;WORD COUNT RANGE MASK

                                ;       REGISTER BIT DEFINITIONS - MP FOR READ HEADER
                                HDSEC   =77                     ;SECTOR MASK
                                HDHSEL  =100                    ;HEAD SELECT MASK

                                ;       REGISTER BIT DEFINITIONS - MP FOR GET STATUS
                                STAMSK  =7                      ;STATE MASK
                                BHSTAT  =10                     ;BRUSH HOME STATUS
                                HOSTAT  =20                     ;HEADS OUT STATUS
                                COSTAT  =40                     ;COVER OPEN STATUS
                                HSSTAT  =100                    ;HEAD SELECT STATUS
                                DSESTAT =400                    ;DRIVE SELECT ERROR STATUS
                                VCSTAT  =1000                   ;VOLUME CHECK STATUS
                                WGESTAT =2000                   ;WRITE GATE ERROR STATUS
                                SPDSTAT =4000                   ;SPIN ERROR STATUS
                                STOSTAT =10000                  ;SEEK TIMEOUT ERROR STATUS
                                WLSTAT  =20000                  ;WRITE LOCK STATUS
                                HCESTAT =40000                  ;HEAD CURRENT ERROR STATUS
                                WDESTAT =100000                 ;WRITE DATA ERROR STATUS

002126                          ENDMOD
002126                          BGNMOD  GLBDAT

                                ;       TABLE OF OPERATION MESSAGES
002126  000000                  OPMSGS: .WORD   0               ;FILLER
002130  005253                          .WORD   MWRCHK          ;MESSAGE FOR WRITE CHECK
002132  005277                          .WORD   MGTSTA          ;           GET STATUS
002134  005224                          .WORD   MSEEK           ;           SEEK
002136  005243                          .WORD   MREADH          ;           READ HEADER
002140  005265                          .WORD   MWRITE          ;           WRITE DATA
002142  005232                          .WORD   MREAD           ;           READ DATA
002144  005362                          .WORD   MWRSET          ;           WITH RESET
002146  005311                          .WORD   MDATCP          ;           WITH DATA COMPARE
002150  005330                          .WORD   MHDRCP          ;           WITH HEADER COMPARE
002152  005427                          .WORD   MCYLUP          ;           LOAD HEADS
002154  005416                          .WORD   MULOAD          ;           UNLOAD HEADS
002156  005460                          .WORD   MINOUT          ;              IN-OUT SEQ
002160  005437                          .WORD   MOUTIN          ;              OUT-IN SEQ
```

```
                                        .WORD   MFOLWRT         ;               FOLLOWING WRITE
                                        .WORD   MREVSK          ;               REV SEEK
                                        .WORD   MFWDSK          ;               FWD SEEK
                                        .WORD   MRESKO          ;               REV SEEK
                                        .WORD   MFWSKO          ;               FWD SEEK
                                        .WORD   MBADAD          ;               BAD DISK ADD FOR WRITE
                                        .WORD   M40HDR          ;             40 HEADER OPERATION
                                T.DRIVE:        .WORD   0
                                JJJ:    .WORD   0
                                HLMTW:  .WORD   0
                                CLRBYT: .WORD   0
                                NXTHL:  .WORD   0
                                GBND:   .WORD   0
                                CAMSK:  .WORD   0
                                DIRMSK: .WORD   0
                                HDCYL:  .WORD   0

                                ;       TABLE OF RESULT NAME MESSAGE ADDRESSES
                                RESTBL: .WORD   MCERR            ;CONTROLLER ERROR
                                        .WORD   MDRERR           ;DRIVE ERROR
                                        .WORD   MNEERR           ;NON-EXISTANT MEMORY ERROR
                                        .WORD   MFLERR           ;HEADER NOT FOUND-DATA LATE
                                        .WORD   MHDERR           ;HEADER OR DATA ERROR
                                        .WORD   MOPERR           ;OPERATION INCOMPLETE
                                        .WORD   MNDRST           ;NO DRIVE STATUS AVAILABLE
                                        .WORD   0
                                        .WORD   MWDERR           ;WRITE DATA ERROR
                                        .WORD   MHCERR           ;HEAD CURRENT ERROR
                                        .WORD   0
                                        .WORD   MSTERR           ;SEEK TIMEOUT ERROR
                                        .WORD   MSPERR           ;SPINDLE ERROR
                                        .WORD   MWGERR           ;WRITE GATE ERROR
                                        .WORD   0
                                        .WORD   MDSERR           ;DRIVE SELECT ERROR

                                ;       PATTERN TABLE
                                PATTBL: .WORD   PAT1
                                        .WORD   PAT2
                                        .WORD   PAT3
                                        .WORD   PAT4
                                        .WORD   PAT5
                                        .WORD   PAT6
                                        .WORD   PAT7
                                        .WORD   PAT8
                                        .WORD   PAT9
                                        .WORD   PAT10


                                ;       SUBROUTINE CALLING STACK
                                SUBSTK: .WORD   0                ;STACK IS 12 WORDS LONG
                                        .WORD   0
                                        .WORD   0
                                        .WORD   0
                                        .WORD   0
                                        .WORD   0
                                        .WORD   0
                                        .WORD   0
```

```
                                        .WORD   0
                                        .WORD   0

                                ;RL01 TABLE OF CYLINDERS
T25TBL: .WORD   2                       ;TABLE OF DIFFERENCES TO BE USED
        .WORD   6                       ;IN TEST 25
        .WORD   9.
        .WORD   12.
        .WORD   17.
        .WORD   22.
        .WORD   27.
        .WORD   34.
        .WORD   41.
        .WORD   128.
        .WORD   255.

                                ;RL02 TABLE OF CYLINDERS
T25TB2: .WORD   4
        .WORD   12.
        .WORD   18.
        .WORD   24.
        .WORD   34.
        .WORD   44.
        .WORD   54.
        .WORD   68.
        .WORD   82.
        .WORD   256.
        .WORD   511.

;       TABLE TO BE USED IN TEST 33 AND 34 TO BUILD AND STORE THE
;       CYLINDERS TO BE USED IN THE TEST.

T33TBL: .BLKW   16.
TBT:    .BLKW   16.


CYLTBL: .BYTE   2                       ;TABLE OF DEFAULT CYLINDERS
        .BYTE   7.
        .BYTE   14.
        .BYTE   20.
        .BYTE   27.
        .BYTE   33.
        .BYTE   38.
        .BYTE   45.
        .BYTE   52.
        .BYTE   58.
        .BYTE   65.
        .BYTE   72.
        .BYTE   77.
        .BYTE   84.
        .BYTE   91.
        .BYTE   97.
        .BYTE   102.
        .BYTE   108.
        .BYTE   113.
        .BYTE   120.
        .BYTE   127.
```

```
 299 002533      206                    .BYTE   134.
 300 002534      213                    .BYTE   139.
 301 002535      222                    .BYTE   146.
 302 002536      230                    .BYTE   152.
 303 002537      235                    .BYTE   157.
 304 002540      244                    .BYTE   164.
 305 002541      252                    .BYTE   170.
 306 002542      261                    .BYTE   177.
 307 002543      270                    .BYTE   184.
 308 002544      275                    .BYTE   189.
 309 002545      303                    .BYTE   195.
 310 002546      312                    .BYTE   202.
 311 002547      317                    .BYTE   207.
 312 002550      326                    .BYTE   214.
 313 002551      334                    .BYTE   220.
 314 002552      343                    .BYTE   227.
 315 002553      352                    .BYTE   234.
 316 002554      361                    .BYTE   241.
 317 002555      367                    .BYTE   247.
 318 002556      375                    .BYTE   253.
 319 002557      000                    .BYTE   0
 320 002560   000401                    .WORD   257.
 321 002562   000406                    .WORD   262.
 322 002564   000415                    .WORD   269.
 323 002566   000423                    .WORD   275.
 324 002570   000432                    .WORD   282.
 325 002572   000445                    .WORD   293.
 326 002574   000454                    .WORD   300.
 327 002576   000463                    .WORD   307.
 328 002600   000471                    .WORD   313.
 329 002602   000500                    .WORD   320.
 330 002604   000507                    .WORD   327.
 331 002606   000514                    .WORD   332.
 332 002610   000523                    .WORD   339.
 333 002612   000532                    .WORD   346.
 334 002614   000540                    .WORD   352.
 335 002616   000545                    .WORD   357.
 336 002620   000553                    .WORD   363.
 337 002622   000560                    .WORD   368.
 338 002624   000567                    .WORD   375.
 339 002626   000576                    .WORD   382.
 340 002630   000605                    .WORD   389.
 341 002632   000612                    .WORD   394.
 342 002634   000621                    .WORD   401.
 343 002636   000627                    .WORD   407.
 344 002640   000634                    .WORD   412.
 345 002642   000643                    .WORD   419.
 346 002644   000651                    .WORD   425.
 347 002646   000660                    .WORD   432.
 348 002650   000667                    .WORD   439.
 349 002652   000674                    .WORD   444.
 350 002654   000702                    .WORD   450.
 351 002656   000711                    .WORD   457.
 352 002660   000716                    .WORD   462.
 353 002662   000725                    .WORD   469.
 354 002664   000733                    .WORD   475.
 355 002666   000742                    .WORD   482.
```

```
356 002670  000751                        .WORD   489.
357 002672  000760                        .WORD   496.
358 002674  000766                        .WORD   502.
359 002676  000774                        .WORD   508.
360 002700  000774                        .WORD   508.
361 002702  000000                        .WORD   0
362 002704  000000              SSINDX:   .WORD   0             ;SUBROUTINE STACK INDEX POINTER
363
364                             ;         OPERATIONAL FLAGS
365 002706  000000              OPFLAG:   .WORD   0             ;OPERATION FLAGS
366 002710  000000              DONE:     .WORD   0             ;OPERATION COMPLETE FLAG
367 002712  000000              HADONE:   .WORD   0             ;HEAD ALIGNMENT DONE FLAG
368 002714  000000              ERHEAD:   .WORD   0             ;ADDRESS OF ERROR HEADER
369 002716  000000              MORECE:   .WORD   0             ;MORE THAN 1 COMPARE ERROR
370 002720  000000              ERRSWI:   .WORD   0             ;ERROR RETURN SWITCH
371 002722  000000              BSFLAG:   .WORD   0             ;BAD SECTOR FLAGS
372 002724  000000              WRTSWI:   .WORD   0             ;WRITE SWITCH
373 002726  000000              TBLSTR:   .WORD   0             ;TABLE STORAGE
374
375 002730  000000              RLBAS:    .WORD   0             ;RL11 BASE ADDRESS
376 002732  000000              RLVEC:    .WORD   0             ;RL11 VECTOR ADDRESS
377 002734  000000              RLDRV:    .WORD   0             ;DRIVE NUMBER UNDER TEST
378
379 002736  000000              L.CS:     .WORD   0             ;CONTROLLER REGISTER STORAGE
380 002740  000000              L.BA:     .WORD   0             ;BEFORE OPERATION
381 002742  000000              L.DA:     .WORD   0
382 002744  000000              L.MP:     .WORD   0
383 002746  000000              T.CS:     .WORD   0             ;CONTROLLER REGISTER STORAGE
384 002750  000000              T.BA:     .WORD   0             ;   AFTER OPERATION
385 002752  000000              T.DA:     .WORD   0
386 002754                      T.MP:
387 002754  000000              HDWRD1:   .WORD   0             ;HEADER WORD STORAGE
388 002756  000000              HDWRD2:   .WORD   0
389 002760  000000              HDWRD3:   .WORD   0
390
391 002762  000000              T.STAT:   .WORD   0             ;DRIVE STATE STORAGE
392
393 002764  000000              RESPARM:          .WORD   0     ;PARAM BLOCK FOR REASON REPORT
394 002766  000000                        .WORD   0
395 002770  000000                        .WORD   0
396 002772  000000                        .WORD   0
397 002774  000000                        .WORD   0
398
399 002776  000000              DRVCNT:   .WORD   0             ;DRIVE COUNT FOR DRIVES UNDER TEST
400 003000  000000              DIFAUG:   .WORD   0             ;DIFFERENCE AUGMENT FOR SEEK
401 003002  000000              OLDCYL:   .WORD   0             ;OLD CYLINDER
402 003004  000000              NEWCYL:   .WORD   0             ;NEW CYLINDER
403 003006  000000              CURCYL:   .WORD   0             ;CURRENT CYLINDER
404 003010  000000              DESDIF:   .WORD   0             ;DESIRED DIFFERENCE
405 003012  000000              DESSGN:   .WORD   0             ;DESIRED SIGN
406 003014  000000              DESHD:    .WORD   0             ;DESIRED HEAD
407 003016  000000              DESSEC:   .WORD   0             ;DESIRED SECTOR
408 003020  000000              TEMP0:    .WORD   0             ;TEMPORARY STORAGE
409 003022  000000              TEMP1:    .WORD   0             ;TEMPORARY STARAGE
410 003024  000000              TEMP2:    .WORD   0             ;TEMPORARY STORAGE
411 003026  000000              TEMP3:    .WORD   0             ;TEMPORARY STORAGE
412 003030  000000              TEMP4:    .WORD   0             ;TEMPORARY STORAGE
```

```
413 003032  000000                TEMP5:  .WORD   0               ;TEMPORARY STORAGE
414 003034  000000                TEMP6:  .WORD   0               ;TEMPORARY STORAGE
415 003036  000000                TEMP7:  .WORD   0               ;TEMPORARY STORAGE
416 003040  000000                TEMP8:  .WORD   0               ;TEMPORARY STORAGE
418                               ;       TIMER STORAGE
419 003042  000000                OFIN:   .WORD   0               ;ONE CYLINDER FORWARD INNER
420 003044  000000                OFINU:  .WORD   0               ;       UPPER
421 003046  000000                OFMID:  .WORD   0               ;ONE CYLINDER FORWARD MIDDLE
422 003050  000000                OFMIDU: .WORD   0               ;       UPPER
423 003052  000000                OFOUT:  .WORD   0               ;ONE CYLINDER FORWARD OUTER
424 003054  000000                OFOUTU: .WORD   0               ;       UPPER
425 003056  000000                ORIN:   .WORD   0               ;ONE CYLINDER REVERSE INNER
426 003060  000000                ORINU:  .WORD   0               ;       UPPER
427 003062  000000                ORMID:  .WORD   0               ;ONE CYLINDER REVERSE MIDDLE
428 003064  000000                ORMIDU: .WORD   0               ;       UPPER
429 003066  000000                OROUT:  .WORD   0               ;ONE CYLINDER REVERSE OUTER
430 003070  000000                OROUTU: .WORD   0               ;       UPPER
431 003072  000000                HFIN:   .WORD   0               ;128 CYLINDER FORWARD INNER
432 003074  000000                HFINU:  .WORD   0               ;       UPPER
433 003076  000000                HFOUT:  .WORD   0               ;128 CYLINDER FORWARD OUTER
434 003100  000000                HFOUTU: .WORD   0               ;       UPPER
435 003102  000000                HRIN:   .WORD   0               ;128 CYLINDER REVERSE INNER
436 003104  000000                HRINU:  .WORD   0               ;       UPPER
437 003106  000000                HROUT:  .WORD   0               ;128 CYLINDER REVERSE OUTER
438 003110  000000                HROUTU: .WORD   0               ;       UPPER
439 003112  000000                AFMID:  .WORD   0               ;256 CYLINDER FORWARD
440 003114  000000                AFMIDU: .WORD   0               ;       UPPER
441 003116  000000                ARMID:  .WORD   0               ;256 CYLINDER REVERSE
442 003120  000000                ARMIDU: .WORD   0               ;       UPPER
443
444 003122  000226                EXOCYL: .WORD   150.            ;EXPECTED TIME ONE CYLINDER
445 003124  001046                EXHCYL: .WORD   550.            ;EXPECTED TIME 128 CYLINDER
446 003126  001750                EXACYL: .WORD   1000.           ;EXPECTED TIME 256 CYLINDER
447 003130  000372                EXROT:  .WORD   250.            ;EXPECTED ROTATION TIME
449 003132  000004                ERRVEC: .WORD   4               ;ERROR VECTOR USED WHEN AUTO SIZING
450
451                               ;       MISCELLANEOUS COUNTERS
452 003134  000000                PASCNT: .WORD   0               ;PASS COUNTER (LOCAL TO A TEST)
453 003136  000000                COUNT:  .WORD   0               ;A COUNTER (LOCAL TO A TEST)
454 003140  000000                ERRPOINT: .WORD 0               ;ERROR POINTER
455 003142                        ERRCNT: .BLKW   64.             ;ERROR COUNTER FOR PROGRAM
456 003342  000000                PASNUM: .WORD   0               ;PASS NUMBER FOR PROGRAM
457 003344  000000                PSETNM: .WORD   0               ;COUNTER FOR PARAMETER SET NUMBER IN USE
458 003346      000               LOCERR: .BYTE   0               ;LOCAL ERROR COUNTER
459 003347      000               NOERCT: .BYTE   0               ;INHIBIT ERROR COUNTING FLAG
460 003350  000000                TRPFLG: .WORD   0               ;HARDWARE TRAP OCCURANCE
461 003352  000000                PWRFLG: .WORD   0               ;POWER FAILURE OCCURANCE
462
463                               ;       BAD SECTOR TABLES AND POINTERS
464 003354  000000                BSFVAL: .WORD   0               ;BAD SECTORS FILES VALID FLAG
465
466 003356                        SBSFIL: .BLKW   76              ;SOFTWARE BAD SECTOR FILE
467 003552                        FBSFIL: .BLKW   76              ;FACTORY BAD SECTOR FILE
468
469 003746                        IBUFF:  .BLKW   200             ;INPUT BUFFER
470 004346                        OBUFF:  .BLKW   200             ;OUTPUT BUFFER
471
```

```
472 004746  000000                PAT1:   .WORD   0               ;PATTERN 1 (ALL ZEROS)
473 004750  177772                PAT2:   .WORD   177772
474 004752  177777                        .WORD   177777
475 004754  177777                        .WORD   177777
476 004756  052525                        .WORD   052525
477 004760  052525                        .WORD   052525
478 004762  052525                        .WORD   052525
479 004764  177777                        .WORD   177777
480 004766  177777                        .WORD   177777
481 004770  052525                        .WORD   052525
482 004772  052525                        .WORD   052525
483 004774  177777                        .WORD   177777
484 004776  052525                        .WORD   052525
485 005000  177252                        .WORD   177252
486 005002  177252                        .WORD   177252
487 005004  172765                        .WORD   172765
488 005006  172765                        .WORD   172765
489
490 005010  000003                PAT3:   .WORD   000003
491 005012  000000                        .WORD   000000
492 005014  000000                        .WORD   000000
493 005016  177777                        .WORD   177777
494 005020  177777                        .WORD   177777
495 005022  177777                        .WORD   177777
496 005024  000000                        .WORD   000000
497 005026  000000                        .WORD   000000
498 005030  177777                        .WORD   177777
499 005032  177777                        .WORD   177777
500 005034  000000                        .WORD   000000
501 005036  177777                        .WORD   177777
502 005040  000000                        .WORD   000000
503 005042  177777                        .WORD   177777
504 005044  000000                        .WORD   000000
505 005046  177777                        .WORD   177777
506
507 005050  025252                PAT4:   .WORD   025252
508 005052  052525                        .WORD   052525
509 005054  052525                        .WORD   052525
510 005056  125252                        .WORD   125252
511 005060  125252                        .WORD   125252
512 005062  125252                        .WORD   125252
513 005064  052525                        .WORD   052525
514 005066  052525                        .WORD   052525
515 005070  125252                        .WORD   125252
516 005072  125252                        .WORD   125252
517 005074  052525                        .WORD   052525
518 005076  125252                        .WORD   125252
519 005100  052525                        .WORD   052525
520 005102  125252                        .WORD   125252
521 005104  052525                        .WORD   052525
522 005106  125252                        .WORD   125252
523
524 005110  155555                PAT5:   .WORD   155555
525 005112  133333                        .WORD   133333
526 005114  066666                        .WORD   066666
527
528 005116  121105                PAT6:   .WORD   121105
```

```
005120  150442                           .WORD   150442
005122  064221                           .WORD   064221
005124  132110                           .WORD   132110
005126  055044                           .WORD   055044
005130  026442                           .WORD   026442
005132  013211                           .WORD   013211
005134  105504                           .WORD   105504
005136  042642                           .WORD   042642
005140  021321                           .WORD   021321
005142  110550                           .WORD   110550
005144  044264                           .WORD   044264
005146  022132                           .WORD   022132
005150  011055                           .WORD   011055
005152  104426                           .WORD   104426
005154  042213                           .WORD   042213

005156  177777                  PAT7:    .WORD   177777

005160  045513                  PAT8:    .WORD   045513
005162  122645                           .WORD   122645
005164  151322                           .WORD   151322
005166  064551                           .WORD   064551
005170  132264                           .WORD   132264
005172  055132                           .WORD   055132
005174  026455                           .WORD   026455
005176  113226                           .WORD   113226
005200  045513                           .WORD   045513
005202  122645                           .WORD   122645
005204  151322                           .WORD   151322
005206  064551                           .WORD   064551
005210  132264                           .WORD   132264
005212  055132                           .WORD   055132
005214  026455                           .WORD   026455
005216  113226                           .WORD   113226

005220  125252                  PAT9:    .WORD   125252

005222  155555                  PAT10:   .WORD   155555

005224                          ENDMOD

005224                          BGNMOD   GLBTXT
005224     123     105     105  MSEEK:   .ASCIZ  /SEEK /
005232     122     104     040  MREAD:   .ASCIZ  /RD DATA /
005243     122     104     040  MREADH:  .ASCIZ  /RD HDR /
005253     127     122     124  MWRCHK:  .ASCIZ  /WRT CHECK/
005265     127     122     124  MWRITE:  .ASCIZ  /WRT DATA /
005277     107     105     124  MGTSTA:  .ASCIZ  /GET STAT /
005311     127     111     124  MDATCP:  .ASCIZ  /WITH DATA CMP /
005330     127     111     124  MHDRCP:  .ASCIZ  /WITH HDR CMP /
005346     106     117     122  M40HDR:  .ASCIZ  /FOR 40 HDRS/
005362     127     111     124  MWRSET:  .ASCIZ  /WITH RESET /
005376     117     120     105  MOPER:   .ASCIZ  /OPER: /
005405     122     105     123  MRSLT:   .ASCIZ  /RESULT:         /
005416     125     116     114  MULOAD:  .ASCIZ  /UNLD DRV/
005427     114     104     040  MCYLUP:  .ASCIZ  /LD DRV /
005437     106     117     114  MOUTIN:  .ASCIZ  /FOL 0 TO CC SEEK/
```

```
 589 005460  106  117  114  MINOUT: .ASCIZ  /FOL 255 TO CC SEEK/
 590 005503  106  117  114  MFOLWRT:        .ASCIZ  /FOL WRT (NO SEEK)/
 591 005525  101  104  112  MREVSK: .ASCIZ  /ADJ CYL WRTTN AFTER REV SK/
 592 005560  101  104  112  MFWDSK: .ASCIZ  /ADJ CYL WRTTN AFTER FWD SK/
 593 005613  123  113  040  MFWSKO: .ASCIZ  /SK FWD,WRT - SK REV,OVERWRT/
 594 005647  123  113  040  MRESKO: .ASCIZ  /SK REV,WRT - SK FWD,OVERWRT/
 595 005703  117  116  040  MBADAD: .ASCIZ  /ON BAD SEC FILES/
 596 005724  103  101  116  MBADSF: .ASCIZ  /CAN'T GET BAD SEC FILES/
 597 005754  102  101  104  MFMTER: .ASCIZ  /BAD SEC FILE FMT ERR/
 598 006001  124  117  040  MTMBS:  .ASCIZ  /TO MANY BAD SEC /
 599 006022  102  125  123  BASADD: .ASCIZ  /BUS ADD=/
 600 006033  104  122  126  DRVNAM: .ASCIZ  /DRV=/
 601 006040  116  117  040  DRVNAV: .ASCIZ  /NO DRV FOR TST/
 602 006057  104  122  126  NOPWR:  .ASCIZ  /DRV DID NOT REC'R FROM PWR FAIL/
 603 006117  122  114  103  CSNAM:  .ASCIZ  /RLCS/
 604 006124  122  114  102  BANAM:  .ASCIZ  /RLBA/
 605 006131  122  114  104  DANAM:  .ASCIZ  /RLDA/
 606 006136  122  114  115  MPNAM:  .ASCIZ  /RLMP/
 607 006143  117  120  040  LAB1:   .ASCIZ  /OP INIT = /
 608 006156  117  120  040  LAB2:   .ASCIZ  /OP DONE = /
 609 006171  127  117  122  MWORD:  .ASCIZ  /WORD /
 610 006177  111  116  124  MTOSLOW:        .ASCIZ  /INTRPT TO LATE/
 611 006216  116  117  040  MDRRES: .ASCIZ  /NO DRV RESPONSE/
 612 006236  116  117  040  MNOINT: .ASCIZ  /NO INTRPT ON CMND COMPLETE/
 613 006271  103  116  124  MCONHNG:        .ASCIZ  /CNTLR HUNG /
 614 006305  105  122  122  MNOCLR: .ASCIZ  /ERR DID NOT CLR/
 615 006325  126  117  114  VCNRST: .ASCIZ  /VOL CHK NOT RSET/
 616 006346  125  116  130  UNXERR: .ASCIZ  /UNXPCTED ERR/
 617 006363  040  124  105  TSTLAB: .ASCIZ  / TEST/
 635 006371  117  125  124  P2T03E: .ASCIZ  /OUT GRD BAND /
 636 006407  111  116  103  P2T04E: .ASCIZ  /INC SEEK FWD HD 0/
 637 006431  111  116  103  P2T05E: .ASCIZ  /INC SEEK REV HD 0/
 638 006453  111  116  103  P2T06E: .ASCIZ  /INC SEEK FWD HD 1/
 639 006475  111  116  116  P2T07E: .ASCIZ  /INN GRD BAND /
 640 006513  111  116  103  P2T08E: .ASCIZ  /INC SEEK REV HD 1/
 641 006535  123  105  105  P2T09E: .ASCIZ  /SEEK/
 642 006542  106  127  104  P2T10E: .ASCIZ  /FWD OSC SEEK/
 643 006557  122  105  126  P2T11E: .ASCIZ  /REV OSC SEEK/
 644 006574  123  105  105  P2T12E: .ASCIZ  /SEEK TIMING/
 645 006610  102  101  123  P2T13E: .ASCIZ  /BASIC RD DATA/
 646 006626  127  122  124  P2T14E: .ASCIZ  &WRT/RD DATA (P1)&
 647 006647  123  120  111  P2T15E: .ASCIZ  /SPINDLE ROT TIMING/
 648 006672  127  122  124  P2T16E: .ASCIZ  &WRT/RD DATA (P2)&
 649 006713  127  122  124  P2T17E: .ASCIZ  /WRT LCK ERR AND DATA PROT/
 650 006745  101  104  112  P2T18E: .ASCIZ  /ADJ CYL INTERFERENCE/
 651 006772  117  126  105  P2T19E: .ASCIZ  /OVERWRT/
 652 007002  123  105  105  SKTMES: .ASCIZ  /SEEK TIMES /
 653 007016  123  120  111  SRTMES: .ASCIZ  /SPINDLE ROT TIME /
 654 007040  050  111  116  VALDES: .ASCIZ  /(IN 100'S OF U-SEC)/
 655 007064  101  120  120  MAPROX: .ASCIZ  /APPROX /
 656 007074  111  116  116  LABIN:  .ASCIZ  /INNER/
 657 007102  115  111  104  LABMID: .ASCIZ  /MIDDLE/
 658 007111  117  125  124  LABOUT: .ASCIZ  /OUTER/
 659 007117  105  130  120  LABEXP: .ASCIZ  /EXPECTED/
```

```
 1 007130  061  040  103  LABOCF: .ASCIZ  /1 CYL FWD/
 2 007142  061  040  103  LABOCR: .ASCIZ  /1 CYL REV/
 3 007154  115  111  104  LABHCF: .ASCIZ  /MID CYL FWD/
 4 007170  115  111  104  LABHCR: .ASCIZ  /MID CYL REV/
 5 007204  115  101  130  LABACF: .ASCIZ  /MAX CYL FWD/
 6 007220  115  101  130  LABACR: .ASCIZ  /MAX CYL REV/
 8 007234  110  104  123  HDMOVF: .ASCIZ  /HDS FAILED TO MOVE IN 10 TRIES/
26 007273  122  105  123  OPR12:  .ASCIZ  /RESET WRT LCK /
27 007312  117  116  040  OPR1A:  .ASCIZ  /ON /
28 007316  117  116  040  OPR1B:  .ASCIZ  /ON DRV /
29 007326  125  116  104  UNDTST: .ASCIZ  /UNDER TEST/
30 007341  123  105  124  OPR004: .ASCIZ  /SET WRT LCK /
31 007356  104  111  106  DIFWD:  .ASCIZ  /DIFF /
32 007364  123  107  116  SGNWD:  .ASCIZ  /SGN /
33 007371  110  104  040  HDWD:   .ASCIZ  /HD /
34 007375  123  105  103  SECWD:  .ASCIZ  /SEC /
35 007402  103  131  114  CYLWD:  .ASCIZ  /CYL /
36 007407  106  122  117  FRMWD:  .ASCIZ  /FROM /
37 007415  040  102  131  BYPSNM: .ASCIZ  / BYPASSED /
38 007430  122  117  125  SEQMES: .ASCIZ /ROUTINE TRACE SEQ:/
39 007453  104  122  126  STAMES: .ASCIZ  /DRV STAT/
40 007464  102  101  104  BSNSTR: .ASCIZ  /BAD SEC FILES NOT STRD. ALL SEC ASSUMED OK./
41 007540  124  117  124  TCERR:  .ASCIZ  /TOTAL CMP ERRS: /
42
43                        ;       RESULT NAMES
44 007561  104  122  126  MDRDY:  .ASCIZ  /DRV RDY /
45 007572  103  117  116  MCERR:  .ASCIZ  /CONT ERR /
46 007604  110  104  122  MHCRC:  .ASCIZ  /HDR CRC/
47 007614  104  101  124  MDCRC:  .ASCIZ  /DATA CRC/
48 007625  110  104  122  MHNF:   .ASCIZ  /HDR NOT FND/
49 007641  104  101  124  MDLT:   .ASCIZ  /DATA LATE/
50 007653  110  104  122  MHFCRC: .ASCIZ  &HDR NOT FND/HDR CRC/OPI&
51 007703  104  122  126  MDRERR: .ASCIZ  /DRV ERR /
60 007714  104  122  126  MDSERR: .ASCIZ  /DRV SEL ERR /
61 007731  104  122  126  MDRVST: .ASCIZ  /DRV STATE /
62 007744  123  120  111  MSPERR: .ASCIZ  /SPIN TIMEOUT /
63 007762  127  122  124  MWGERR: .ASCIZ  /WRT GAT ERR /
64 007777  123  105  105  MSTERR: .ASCIZ  /SEEK TIMEOUT /
65 010015  110  105  101  MHCERR: .ASCIZ  /HEAD CUR ERR /
66 010033  127  122  124  MWDERR: .ASCIZ  /WRT DAT ERR /
67 010050  117  120  122  MOPERR: .ASCIZ  /OPR-INC/
68 010060  110  104  122  MHDERR: .ASCIZ  &HDR/DAT ERR &
69 010075  110  104  122  MFLERR: .ASCIZ  &HDR NOT FND/DAT LATE &
70 010123  116  055  130  MNEERR: .ASCIZ  /N-X-MEM /
71 010134  103  131  114  MCYLOC: .ASCIZ  /CYL /
72 010141  103  101  116  MNDRST: .ASCIZ  /CAN'T GET DRV STAT/
73 010164  125  116  113  MUNDEF: .ASCIZ  /UNKN DRV STATE-NO RDY,NO ERR,HDS OUT/
74 010231  106  101  111  MRLFAL: .ASCIZ  /FAIL TO RELD HDS AFTER ERR CLEAR/
75 010272  127  122  124  MWRTAB: .ASCIZ  /WRT ABORTED/
76 010306  040  117  126  MEXERS: .ASCIZ  / OVER ERR LIMIT - UNIT DROPPED /
77 010346  040  105  122  MERRS:  .ASCIZ  / ERROR/
78 010355  207  377  377  BELL:   .ASCIZ  <207><377><377>
79
80                        ;       RESULT SETTINGS
81 010361  111  123  040  RESE3:  .ASCIZ  /IS /
82 010365  040  123  102  RESE4:  .ASCIZ  / SB /
83
```

```
 84                                          ;       RESULT CONDITIONS
 85 010372  040  111  116  RESE5:  .ASCIZ  / IN /
 86 010377  040  117  106  RESE6:  .ASCIZ  / OF /
 87 010404  123  124  101  STATE2: .ASCIZ  /STATE 2/
 88 010414  123  124  101  STATE3: .ASCIZ  /STATE 3/
 89 010424  123  124  101  STATE5: .ASCIZ  /STATE 5/
 93 010434  061  123  124  C10MS:  .ASCIZ  /1ST 3 MS/
 94 010445  065  060  060  C500MS: .ASCIZ  /500MS/
 95 010453  103  131  103  CCYLUP: .ASCIZ  /CYCLE UP/
 96 010464  104  101  124  CAFDT:  .ASCIZ  /DATA XFR/
 97 010475  065  040  123  C5SEC:  .ASCIZ  /5 SEC/
 98
 99 010503  045  116  045  FMTOP1: .ASCIZ  /%N%T%N%T%T%06%S%T%01%N/
100 010532  045  116  045  FMTOP2: .ASCIZ  /%N%T%01%S1%T%01%N/
101 010554  045  116  045  FMTOP3: .ASCIZ  /%N%T%01%S1%T%T%N/
102 010575  045  124  045  FMT1:   .ASCIZ  /%T%T/
103 010602  045  116  045  FMT1.1: .ASCIZ  /%N%T%T/
104 010611  045  124  000  FMT2:   .ASCIZ  /%T/
105 010614  045  116  000  FMT3:   .ASCIZ  /%N/
106 010617  045  116  045  FMT4:   .ASCIZ  /%N%T%T%N/
107 010630  045  116  045  FMT5:   .ASCIZ  /%N%T%06%S1%T%01/
108 010650  045  116  045  FMT6:   .ASCIZ  /%N%S11%T%S4%T%S4%T%S4%T%S4%T%S2%T/
109 010712  045  116  045  FMT7:   .ASCIZ  /%N%T%06%S2%06%S2%06%S2%06%S3%03%S2%01%N/
110 010762  045  116  045  FMT8:   .ASCIZ  /%N%T%06%S2%06%S2%06%S2%06/
111 011014  045  116  045  FMT9:   .ASCIZ  /%N%T/
112 011021  045  124  045  FMT11:  .ASCIZ  /%T%01/
113 011027  045  124  045  FMT12:  .ASCIZ  /%T%03/
114 011035  045  116  045  FMT13:  .ASCIZ  /%N%S11%T%03%S1%T%03%S1%T%01%S1%T%01/
115 011101  045  116  045  FMT14:  .ASCIZ  /%N%T%T%D3%S1%T%06%S1%T%06/
116 011133  045  116  045  FMT15:  .ASCIZ  /%N%S11%T%D3%S1%T%06%S1%T%06/
117 011167  045  116  045  FMT16:  .ASCIZ  /%N%S5%06/
118 011200  045  123  061  FMT17:  .ASCIZ  /%S10%T%N%S11%06%N/
119 011222  045  116  045  FMT18:  .ASCIZ  /%N%S15%T%S5%T%S4%T%S5%T%N/
120 011254  045  124  045  FMT19:  .ASCIZ  /%T%S4%D6%S4%D6%S4%D6%S4%D6%N/
121 011311  045  124  045  FMT20:  .ASCIZ  /%T%S2%D6%S14%D6%S4%D6%N/
122 011341  045  124  045  FMT21:  .ASCIZ  /%T%S12%D6%S14%D6%N/
123 011364  045  116  045  FMT22:  .ASCIZ  /%N%S11%T%03%S1%T%01%S1%T%02/
124 011420  045  124  045  FMT23:  .ASCIZ  /%T%T%T%01%N/
125 011434  045  116  045  FMT24:  .ASCIZ  /%N%T/
126 011441  045  116  045  FMT25:  .ASCIZ  /%N%D2%T/
127 011451  045  116  045  FMT26:  .ASCIZ  /%N%S1%T%D4%T%T%D3%N/
128 011475  045  116  045  FMT27:  .ASCIZ  /%N%T%D3%T%D3%N/
129 011514  045  116  045  FMT28:  .ASCIZ  /%N%T%T%T/
130 011525                   ENDMOD
135
136 011526                   BGNMOD  GLBERR
137                          ;       ERR1    R3 POINTS TO RESULT MESSAGE
138                          ;               RESULT: (R3)
139
140                          ;       ERR2    R3 POINTS TO RESULT NAME
141                          ;               RESULT: (R3) IS 1 SB 0
142
143                          ;       ERR3    R3 POINTS TO RESULT NAME
144                          ;               RESULT: (R3) IS 0 SB 1
145
146                          ;       ERR4    R3 POINTS TO RESULT NAME
147                          ;               R4 POINTS TO RESULT CONDITIONS
```

```
                                ;               RESULT: (R3) IS 1 SB 0 (R4)
                                ;
                                ;       ERR5    R3 POINTS TO RESULT NAME
                                ;               R4 POINTS TO RESULT CONDITIONS
                                ;               RESULT: (R3) IS 0 SB 1 (R4)
                                ;
                                ;       ERR6    RESULT ROUTINE DETERMINES WHICH ERROR(S) ARE SET AND
                                ;               REPORTS ALL
                                ;               RESULT: "ERROR" IS 1 SB 0
                                ;
                                ;       ERR7    DRIVE STATE ERROR REPORT
                                ;               R3 CONTAINS EXPECTED STATE
                                ;               T.STAT CONTAINS BAD STATE
                                ;               RESULT: DRIVE STATE IS (T.STAT) SB (R3)
                                ;
                                ;       ERR8    HEAD POSITIONING ERROR REPORT
                                ;               NEWCYL CONTAINS EXPECTED CYLINDER
                                ;               HDWRD1 CONTAINS BAD CYLINDER
                                ;               RESULT: CYLINDER IS (HDWRD1) SB (NEWCYL)
                                ;
                                ;       ERR9    UTILITY RESULT REPORT
                                ;               R3 POINTS TO RESULT NAME
                                ;               R4 POINTS TO VALUE 1
                                ;               R5 POINTS TO VALUE 2
                                ;               RESULT: (R3-NAME) IS (R4-VALUE 1) SB (R5-VALUE 2)
                                ;
                                ;       ERR10   COMPARE ERROR REPORT
                                ;               R3 CONTAINS THE BAD WORD NUMBER
                                ;               R4 POINTS TO BAD WORD
                                ;               R5 POINTS TO GOOD WORD
                                ;               RESULT: WORD (R3) IS (R4) SB (R5)
                                ;
                                ;
011526                          BGNMSG  ERR1
011526  105737  003347                  TSTB    NOERCT          ;TEST IF ERROR COUNTING INHIBITED
011532  001002                          BNE     1$              ;YES - SKIP
011534  005277  171400                  INC     @ERRPOINT       ;ELSE BUMP ERROR COUNT
011540  010146                  1$:     MOV     R1,-(SP)        ;STORE R1
011542  004737  023422                  JSR     PC,RPTOP        ;REPORT OPERATION
011546  012721  000001                  MOV     #1,(R1)+        ;SET PARAM NUMBER
011552  010321                          MOV     R3,(R1)+        ;INSERT MESSAGE ADDRESS POINTER
011554  004737  024210                  JSR     PC,RPTRES       ;REPORT RESULTS
011560  004737  024416                  JSR     PC,RPTREM       ;REPORT REMAINDER
011564  012601                          MOV     (SP)+,R1        ;RESTORE R1
011566  004737  014750                  JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
011572                          ENDMSG
011572                          L10000:
011572  104023                          EMT     C$MSG
                                ;
011574                          BGNMSG  ERR2
011574  005277  171340                  INC     @ERRPOINT       ;BUMP ERROR COUNT
011600  010146                          MOV     R1,-(SP)        ;STORE R1
011602  004737  023422                  JSR     PC,RPTOP        ;REPORT OPERATION
011606  012721  000003                  MOV     #3,(R1)+        ;SET PARAM NUMBER
011612  010321                          MOV     R3,(R1)+        ;INSERT NAME ADD POINTER
011614  012721  000001                  MOV     #1,(R1)+        ;SET IS VALUE
011620  005021                          CLR     (R1)+           ;SET SB VALUE
```

```
                                                JSR     PC,RPTRES       ;REPORT RESULTS
                                                JSR     PC,RPTREM       ;REPORT REMAINDER
                                                MOV     (SP)+,R1        ;RESTORE R1
                                                JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
                                        ENDMSG
                                        L10001:
                                                EMT     C$MSG

                                        BGNMSG  ERR3
                                                INC     @ERRPOINT       ;BUMP ERROR COUNT
                                                MOV     R1,-(SP)        ;STORE R1
                                                JSR     PC,RPTOP        ;REPORT OPERATION
                                                MOV     #3,(R1)+        ;SET PARAM NUMBER
                                                MOV     R3,(R1)+        ;INSERT NAME ADD POINTER
                                                CLR     (R1)+           ;SET IS VALUE
                                                MOV     #1,(R1)+        ;SET SB VALUE
                                                JSR     PC,RPTRES       ;REPORT RESULTS
                                                JSR     PC,RPTREM       ;REPORT REMAINDER
                                                MOV     (SP)+,R1        ;RESTORE R1
                                                JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
                                        ENDMSG
                                        L10002:
                                                EMT     C$MSG

                                        BGNMSG  ERR4
                                                INC     @ERRPOINT       ;BUMP ERROR COUNT
                                                MOV     R1,-(SP)        ;STORE R1
                                                JSR     PC,RPTOP        ;REPORT OPERATION
                                                MOV     #4,(R1)+        ;SET PARAM NUMBER
                                                MOV     R3,(R1)+        ;INSERT NAME ADD POINTER
                                                MOV     #1,(R1)+        ;SET IS VALUE
                                                CLR     (R1)+           ;SET SB VALUE
                                                MOV     R4,(R1)         ;INSERT ADD OF CONDITION POINTER
                                                JSR     PC,RPTRES       ;REPORT RESULTS
                                                JSR     PC,RPTREM       ;REPORT REMAINDER
                                                MOV     (SP)+,R1        ;RESTORE R1
                                                JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
                                        ENDMSG
                                        L10003:
                                                EMT     C$MSG

                                        BGNMSG  ERR5
                                                INC     @ERRPOINT       ;BUMP ERROR COUNT
                                                MOV     R1,-(SP)        ;STORE R1
                                                JSR     PC,RPTOP        ;REPORT OPERATION
                                                MOV     #4,(R1)+        ;SET PARAM NUMBER
                                                MOV     R3,(R1)+        ;INSERT NAME ADD POINTER
                                                CLR     (R1)+           ;SET IS VALUE
                                                MOV     #1,(R1)+        ;SET SB VALUE
                                                MOV     R4,(R1)         ;INSERT ADD OF CONDITION POINTER
                                                JSR     PC,RPTRES       ;REPORT RESULTS
                                                JSR     PC,RPTREM       ;REPORT REMAINDER
                                                MOV     (SP)+,R1        ;RESTORE R1
                                                JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
                                        ENDMSG
                                        L10004:
                                                EMT     C$MSG
```

```
252
253 012030                              BGNMSG  ERR6
254 012030  105737  003347                      TSTB    NOERCT              ;TEST IF ERROR COUNTING INHIBITED
255 012034  001002                              BNE     17$                 ;YES - SKIP
256 012036  005277  171076                      INC     @ERRPOINT           ;ELSE BUMP ERROR COUNT
257 012042  010146                      17$:    MOV     R1,-(SP)            ;STORE R1
258 012044  010346                              MOV     R3,-(SP)            ;STORE R3
259 012046  010446                              MOV     R4,-(SP)            ;STORE R4
260 012050  010546                              MOV     R5,-(SP)            ;STORE R5
261 012052  004737  023422                      JSR     PC,RPTOP            ;REPORT OPERATION
262 012056  012721  000003                      MOV     #3,(R1)+            ;SET PARAM NUMBER
263 012062  012761  000001  000002              MOV     #1,2(R1)            ;INSERT IS VALUE
264 012070  005037  003026                      CLR     TEMP3               ;CLEAR FOR STATUS STORAGE
265 012074  013703  002746                      MOV     T.CS,R3             ;GET T.CS
266 012100  042703  177761                      BIC     #177761,R3          ;AND CLEAR ALL BUT FUNCTION
267 012104  022703  000004                      CMP     #4,R3               ;CHECK IF IT WAS GET STATUS
268 012110  001432                              BEQ     1$                  ;YES - STATUS IS IN T.MP, SKIP
269 012112  012762  000003  000004              MOV     #GETSTAT,RLDA(R2) ;ELSE DO GET STATUS
270 012120  012703  000004                      MOV     #4,R3
271 012124  053703  002734                      BIS     RLDRV,R3
272 012130  010362  000000                      MOV     R3,RLCS(R2)
273 012134                                      WAITUS  #10.                ;WAIT FOR CONTROLLER READY
    012134  012700  000012                      MOV     #10.,R0
    012140  104027                              EMT     C$WTU
274 012142  032762  000200  000000              BIT     #CRDYMSK,RLCS(R2)       ;TEST IF READY
275 012150  001003                              BNE     10$                 ;YES - SKIP
276 012152  012703  001000              9$:     MOV     #BIT9,R3            ;ELSE SET NO DRIVE STATUS BIT
277 012156  000413                              BR      2$                  ;IN MESSAGE WORD AND SKIP
278 012160  016203  000006              10$:    MOV     RLMP(R2),R3         ;STORE STATUS FOR REPORT
279 012164  010337  003026                      MOV     R3,TEMP3
280 012170  113703  003027                      MOVB    TEMP3+1,R3          ;GET ERROR BITS IN PROPER POSITION
281 012174  000402                              BR      13$
282 012176  113703  002755              1$:     MOVB    T.MP+1,R3           ;GET ERROR BITS FROM MP REG
283 012202  042703  177442              13$:    BIC     #177442,R3          ;CLEAR UNUSED BITS
284 012206  013704  002746              2$:     MOV     T.CS,R4             ;GET ERROR BITS FROM CS REG
285 012212  042704  001777                      BIC     #1777,R4            ;CLEAR UNUSED BITS
286 012216  050403                              BIS     R4,R3               ;MAKE ONE WORD OF POSSIBLE ERRORS
287 012220  032703  002000                      BIT     #OPIERR,R3          ;TEST IF OPI SET
288 012224  001442                              BEQ     115$                ;NO - SKIP
289 012226  032703  010000                      BIT     #HNFERR,R3          ;TEST IF HDR NOT FOUND ERROR
290 012232  001026                              BNE     107$                ;YES - SKIP
291 012234  032703  004000                      BIT     #HCRCERR,R3         ;TEST IF HDR CRC ERR
292 012240  001020                              BNE     105$                ;YES - SKIP
293 012242  012704  010050                      MOV     #MOPERR,R4          ;SET OPI ALONE MESSAGE
294 012246                              100$:   PRINTB  #FMT28,#MRSLT,R4,#MERRS ;REPORT ERROR
    012246  012746  010346                      MOV     #MERRS,-(SP)
    012252  010446                              MOV     R4,-(SP)
    012254  012746  005405                      MOV     #MRSLT,-(SP)
    012260  012746  011514                      MOV     #FMT28,-(SP)
    012264  012746  000004                      MOV     #4,-(SP)
    012270  010600                              MOV     SP,R0
    012272  104014                              EMT     C$PNTB
    012274  062706  000012                      ADD     #12,SP
295 012300  000430                              BR      120$                ;SKIP
296 012302  012704  007604              105$:   MOV     #MHCRC,R4           ;HDR CRC MESSAGE
297 012306  000757                              BR      100$
298 012310  032703  004000              107$:   BIT     #HCRCERR,R3         ;TEST IF HCRC WITH HDR NOT FND
```

```
                                        BNE     109$            ;YES - SKIP
                                        MOV     #MHNF,R4        ;MESSAGE HEADER NOT FOUND
                                        BR      100$
                        109$:           MOV     #MHFCRC,R4      ;HNF AND HCRC MESSAGE
                                        BR      100$            ;SKIP
                        115$:           BIT     #DCKERR,R3      ;TEST IF DATA CHECK SET, NOT OPI
                                        BEQ     118$            ;NO - SKIP
                                        MOV     #MDCRC,R4       ;SET MESSAGE DATA CHECK
                                        BR      100$            ;SKIP
                        118$:           BIT     #DLTERR,R3      ;TEST IF DATA LATE ERROR
                                        BEQ     120$            ;NO - SKIP
                                        MOV     #MDLT,R4        ;SET MESSAGE DATA LATE
                                        BR      100$            ;SKIP
                        120$:           MOV     #BIT15,R5       ;SET BIT POINTER FOR TEST
                                        CLR     R4              ;CLEAR R4 FOR TABLE COUNT
                        3$:             BIT     R5,R3           ;TEST IF BIT IS SET
                                        BNE     6$              ;YES - SKIP TO REPORT
                        4$:             TST     (R4)+           ;ELSE BUMP TABLE POINTER
                                        CLC                     ;CLEAR CARRY
                                        ROR     R5              ;SHIFT BIT POINTER TO NEXT BIT
                                        BNE     3$              ;LOOP IF NOT 0
                                        BR      7$              ;ELSE REPORT REMAINDER
                        6$:             MOV     RESTBL(R4),(R1) ;INSERT NAME ADDRESS
                                        JSR     PC,RPTRES       ;REPORT RESULTS
                                        BR      4$              ;GET NEXT BIT
                        7$:             JSR     PC,RPTREM       ;REPORT REMAINDER
                                        TST     TEMP3           ;TEST IF ANY NEW STATUS
                                        BEQ     15$             ;NO - SKIP
                                        PRINTB  #FMT17,#STAMES,TEMP3
                                        MOV     TEMP3,-(SP)
                                        MOV     #STAMES,-(SP)
                                        MOV     #FMT17,-(SP)
                                        MOV     #3,-(SP)
                                        MOV     SP,R0
                                        EMT     C$PNTB
                                        ADD     #10,SP
                        15$:            BIT     #DCKERR,T.CS    ;TEST IF DATA CHECK ERROR
                                        BEQ     25$             ;NO - SKIP
                                        BIT     #OPIERR,T.CS    ;TEST IF OPI SET
                                        BNE     25$             ;YES - SKIP
                                        CLR     MORECE          ;CLEAR COMPARE ERROR COUNT
                                        MOV     #128.,R1        ;SET COMPARE LENGTH
                                        MOV     #1,R3           ;SET WORD COUNT
                                        MOV     #OBUFF,R5       ;SET GOOD WORD POINTER
                                        MOV     #IBUFF,R4       ;SET TEST WORD POINTER
                        18$:            CMP     (R5),(R4)       ;CHECK WORD
                                        BEQ     19$             ;GOOD - SKIP
                                        CMP     MORECE,#10.     ;TEST IF COMPARE LIMIT REACHED
                                        BGT     20$             ;YES - SKIP
                                        PRINTB  #FMT15,#MWORD,R3,#RESE3,(R4),#RESE4,(R5)
                                        MOV     (R5),-(SP)
                                        MOV     #RESE4,-(SP)
                                        MOV     (R4),-(SP)
                                        MOV     #RESE3,-(SP)
                                        MOV     R3,-(SP)
                                        MOV     #MWORD,-(SP)
                                        MOV     #FMT15,-(SP)
```

```
                012570  012746  000007                  MOV     #7,-(SP)
                012574  010600                          MOV     SP,R0
                012576  104014                          EMT     C$PNTB
                012600  062706  000020                  ADD     #20,SP
    342         012604  005237  002716          20$:    INC     MORECE          ;BUMP ERROR COUNTER
    343         012610  022524                  19$:    CMP     (R5)+,(R4)+     ;BUMP POINTERS
    344         012612  005203                          INC     R3              ;BUMP COUNTER
    345         012614  005301                          DEC     R1              ;DEC LENGTH COUNT
    346         012616  001343                          BNE     18$             ;LOOP IF NOT DONE
    347         012620  005737  002716          25$:    TST     MORECE          ;TEST IF ANY COMPARE ERRORS
    348         012624  001421                          BEQ     27$             ;NO - SKIP
    349         012626  012701  000200                  MOV     #128.,R1        ;SET COMPARE LENGTH
    350         012632                                  PRINTB  #FMT27,#TCERR,MORECE,#RESE6,R1
                012632  010146                          MOV     R1,-(SP)
                012634  012746  010377                  MOV     #RESE6,-(SP)
                012640  013746  002716                  MOV     MORECE,-(SP)
                012644  012746  007540                  MOV     #TCERR,-(SP)
                012650  012746  011475                  MOV     #FMT27,-(SP)
                012654  012746  000005                  MOV     #5,-(SP)
                012660  010600                          MOV     SP,R0
                012662  104014                          EMT     C$PNTB
                012664  062706  000014                  ADD     #14,SP
    351         012670  012605          27$:    MOV     (SP)+,R5        ;RESTORE R5, 4, 3, 1
    352         012672  012604                  MOV     (SP)+,R4
    353         012674  012603                  MOV     (SP)+,R3
    354         012676  012601                  MOV     (SP)+,R1
    355         012700  004737  014750          JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
    356         012704                  ENDMSG
                012704                  L10005:
                012704  104023                  EMT     C$MSG
    357
    358         012706                  BGNMSG  ERR7
    359         012706  005277  170226          INC     @ERRPOINT       ;BUMP ERROR COUNT
    360         012712  010146                  MOV     R1,-(SP)        ;STORE R1
    361         012714  004737  023422          JSR     PC,RPTOP        ;REPORT OPERATION
    362         012720  012721  000003          MOV     #3,(R1)+        ;SET PARAM NUMBER
    363         012724  012721  007731          MOV     #MDRVST,(R1)+   ;INSERT NAME ADD POINTER
    364         012730  013721  002762          MOV     T.STAT,(R1)+    ;INSERT IS VALUE
    365         012734  010311                  MOV     R3,(R1) ;INSERT SB VALUE
    366         012736  004737  024210          JSR     PC,RPTRES       ;REPORT RESULTS
    367         012742  004737  024416          JSR     PC,RPTREM       ;REPORT REMAINDER
    368         012746  012601                  MOV     (SP)+,R1        ;RESTORE R1
    369         012750  004737  014750          JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
    370         012754                  ENDMSG
                012754                  L10006:
                012754  104023                  EMT     C$MSG
    371
    372         012756                  BGNMSG  ERR8
    373         012756  005277  170156          INC     @ERRPOINT       ;BUMP ERROR COUNT
    374         012762  010146                  MOV     R1,-(SP)        ;STORE R1
    375         012764  010346                  MOV     R3,-(SP)        ;STORE R3
    376         012766  004737  023422          JSR     PC,RPTOP        ;REPORT OPERATION
    377         012772  012721  000003          MOV     #3,(R1)+        ;SET PARAM NUMBER
    378         012776  012721  010134          MOV     #MCYLOC,(R1)+   ;INSERT NAME ADD POINTER
    379         013002  013711  002754          MOV     HDWRD1,(R1)     ;GET HEADER WORD
    380         013006  012703  000007          MOV     #7,R3           ;SET SHIFT COUNT
    381         013012  000241          3$:     CLC
```

```
 382 013014  006011                    ROR     (R1)            ;ALIGN CHAR FOR PRINTING
 383 013016  005303                    DEC     R3              ;  AS IS VALUE
 384 013020  001374                    BNE     3$
 385 013022  005721                    TST     (R1)+           ;BUMP PARAM POINTER
 386 013024  013711  003004            MOV     NEWCYL,(R1)     ;INSERT SB VALUE
 387 013030  004737  024210            JSR     PC,RPTRES       ;REPORT RESULTS
 388 013034  004737  024416            JSR     PC,RPTREM       ;REPORT REMAINDER
 389 013040  012603                    MOV     (SP)+,R3        ;RESTORE R3
 390 013042  012601                    MOV     (SP)+,R1        ;RESTORE R1
 391 013044  004737  014750            JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
 392 013050                    ENDMSG
     013050                    L10007:
     013050  104023                    EMT     C$MSG
 393
 394 013052                    BGNMSG  ERR9
 395 013052  005277  170062            INC     @ERRPOINT       ;BUMP ERROR COUNT
 396 013056  010146                    MOV     R1,-(SP)        ;STORE R1
 397 013060  004737  023422            JSR     PC,RPTOP        ;REPORT OPERATION
 398 013064  012721  000003            MOV     #3,(R1)+        ;SET PARAM NUMBER
 399 013070  010321                    MOV     R3,(R1)+        ;INSERT NAME ADD POINTER
 400 013072  010421                    MOV     R4,(R1)+        ;SET IS VALUE
 401 013074  010521                    MOV     R5,(R1)+        ;SET SB VALUE
 402 013076  004737  024210            JSR     PC,RPTRES       ;REPORT RESULTS
 403 013102  004737  024416            JSR     PC,RPTREM       ;REPORT REMAINDER
 404 013106  012601                    MOV     (SP)+,R1        ;RESTORE R1
 405 013110  004737  014750            JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
 406 013114                    ENDMSG
     013114                    L10010:
     013114  104023                    EMT     C$MSG
 407 013116                    BGNMSG  ERR10
 408 013116  010146                    MOV     R1,-(SP)        ;STORE R1
 409 013120  005737  002716            TST     MORECE          ;TEST IF 2ND BAD LINE
 410 013124  001051                    BNE     3$              ;YES - SKIP
 411 013126  005277  170006            INC     @ERRPOINT       ;BUMP ERROR COUNT
 412 013132  004737  023422            JSR     PC,RPTOP        ;REPORT OPERATION
 413 013136                            PRINTB  #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1> ;REPORT ID
     013136  005046                    CLR     -(SP)
     013140  153716  002735            BISB    RLDRV+1,(SP)
     013144  012746  006033            MOV     #DRVNAM,-(SP)
     013150  013746  002730            MOV     RLBAS,-(SP)
     013154  012746  006022            MOV     #BASADD,-(SP)
     013160  012746  010630            MOV     #FMT5,-(SP)
     013164  012746  000005            MOV     #5,-(SP)
     013170  010600                    MOV     SP,R0
     013172  104014                    EMT     C$PNTB
     013174  062706  000014            ADD     #14,SP
 414 013200                            PRINTB  #FMT14,#MRSLT,#MWORD,R3,#RESE3,(R4),#RESE4,(R5)
     013200  011546                    MOV     (R5),-(SP)
     013202  012746  010365            MOV     #RESE4,-(SP)
     013206  011446                    MOV     (R4),-(SP)
     013210  012746  010361            MOV     #RESE3,-(SP)
     013214  010346                    MOV     R3,-(SP)
     013216  012746  006171            MOV     #MWORD,-(SP)
     013222  012746  005405            MOV     #MRSLT,-(SP)
     013226  012746  011101            MOV     #FMT14,-(SP)
     013232  012746  000010            MOV     #10,-(SP)
     013236  010600                    MOV     SP,R0
```

```
                013240  104014                          EMT     C$PNTB
                013242  062706  000022                  ADD     #22,SP
          415   013246  000421                          BR      4$
          416   013250                          3$:     PRINTB  #FMT15,#MWORD,R3,#RESE3,(R4),#RESE4,(R5) ;REPORT DATA
                013250  011546                          MOV     (R5),-(SP)
                013252  012746  010365                  MOV     #RESE4,-(SP)
                013256  011446                          MOV     (R4),-(SP)
                013260  012746  010361                  MOV     #RESE3,-(SP)
                013264  010346                          MOV     R3,-(SP)
                013266  012746  006171                  MOV     #MWORD,-(SP)
                013272  012746  011133                  MOV     #FMT15,-(SP)
                013276  012746  000007                  MOV     #7,-(SP)
                013302  010600                          MOV     SP,R0
                013304  104014                          EMT     C$PNTB
                013306  062706  000020                  ADD     #20,SP
          417   013312  005237  002716          4$:     INC     MORECE          ;INC COMPARE ERROR COUNT
          418   013316  012601                          MOV     (SP)+,R1        ;RESTORE R1
          419   013320  004737  014750                  JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
          420   013324                          ENDMSG
                013324                          L10011:
                013324  104023                          EMT     C$MSG
          421   013326                          ENDMOD
          422
          423
          424   013326                          BGNMOD  HPTCODE
          425   013326                          BGNHW
                013326  000006                          .WORD   L10012-L$HW/2
          426   013330  174400                          .WORD   174400          ;CSR BASE ADDRESS DEFAULT
          427   013332  000160                          .WORD   160             ;VECTOR DEFAULT
          428   013334  000240                          .WORD   240             ;PRIORITY DEFAULT
          429   013336  000001                          .WORD   1               ;TYPE OF DRIVE
          430   013340  000000                          .WORD   0               ;DRIVE NUMBER DEFAULT
          431   013342  000001                          .WORD   1               ;RL11 CONTROLLER
          432   013344                          ENDHW
                013344                          L10012:
          433   013344                          ENDMOD
          434
          435   013344                          BGNMOD  SPTCODE
          436   013344                          BGNSW
                013344  000006                          .WORD   L10013-L$SW/2
          437   013346  000000          MISWIW: .WORD   0               ;BIT 0 = USE ALL CYLINDERS
          438                                                           ;BIT 1 = USE ALL SECTORS
          439                                                           ;BIT 2 = EXECUTE DRIVE SELECT TEST
          440                                                           ;BIT 3 = EXECUTE HEAD ALIGNMENT
          441                                                           ;BIT 4 = DROP DRIVE IF NO RESPUNSE
          442                                                           ;BIT 12 = HEAD SELECT SUPPLIED FLAG
          443                                                           ;BIT 13 = HILIMIT SPECIFIED FLAG
          444                                                           ;BIT 14 = LO LIMIT SPECIFIED FLAG
          445                                                           ;BIT 15 = DO MANUAL INTERVENTION
          446   013350  000000          LOLIMW: .WORD   0
          447   013352  000377          HILIMW: .WORD   255.
          448   013354  000000          HEADW:  .WORD   0
          449   013356  000024          ERLIMW: .WORD   20.              ;ERROR LIMIT
          450   013360  000012          DCLIMW: .WORD   10.              ;COMPARE ERROR LIMIT
          451   013362                          ENDSW
                013362                          L10013:
          452   013362                          ENDMOD
```

```
453
454 013362                          BGNMOD  DSPCODE
459 013362                          DISPATCH        17
    013362  000021                          .WORD   17
    013364  024702                          .WORD   T1
    013366  025114                          .WORD   T2
    013370  025324                          .WORD   T3
    013372  025534                          .WORD   T4
    013374  025760                          .WORD   T5
    013376  026160                          .WORD   T6
    013400  026406                          .WORD   T7
    013402  026716                          .WORD   T8
    013404  027214                          .WORD   T9
    013406  027512                          .WORD   T10
    013410  031350                          .WORD   T11
    013412  032060                          .WORD   T12
    013414  032274                          .WORD   T13
    013416  033020                          .WORD   T14
    013420  034130                          .WORD   T15
    013422  035122                          .WORD   T16
    013424  036336                          .WORD   T17
461 013426                          ENDMOD
462
463 013426                          BGNMOD  INITCODE
464 013426                          BGNINIT
465 013426                                  SETPRI  #340
    013426  012700  000340                  MOV     #340,R0
    013432  104041                          EMT     C$SPRI
466 013434                                  MANUAL                  ;CHECK IF MANUAL INTERVENTION ALLOWED
    013434  104051                          EMT     C$MANI
467 013436                                  BCOMPLETE       1$      ;YES - SKIP
    013436  103403                          BCS     1$
468 013440  042737  100014  013346        BIC     #MITEST!DRSELT!HDALIGN,MISWIW ;CLEAR ALL MANUAL
469                                                                 ; INTERVENTION FLAGS
470 013446  005037  002704          1$:     CLR     SSINDX          ;CLEAR SUBROUTINE STACK INDEX
471 013452                                  READEF  #EF.PWR         ;POWER FAILURE
    013452  012700  000034                  MOV     #EF.PWR,R0
    013456  104050                          EMT     C$REFG
472 013460                                  BNCOMPLETE  4$          ;NO, GO CHECK NEW PASS
    013460  103005                          BCC     4$
473 013462  013737  002012  003352          MOV     L$UNIT,PWRFLG   ;SET POWER FAIL FLAG
474 013470  000137  014102                  JMP     PWCON           ;GO SERVICE POWER FAIL
475 013474                          4$:     READEF  #EF.START       ;CHECK IF START
    013474  012700  000040                  MOV     #EF.START,R0
    013500  104050                          EMT     C$REFG
476 013502                                  BNCOMPLETE      RESTART ;NO - SKIP
    013502  103034                          BCC     RESTART
477                                 ;       ON START INITIALIZE TO START AT FIRST DRIVE, CLEAR INTERNAL
478                                 ;       PASS COUNT, AND ERROR COUNT.
479 013504  013737  002012  002776          MOV     L$UNIT,DRVCNT   ;SET UP UNIT COUNT
480 013512  005037  003342          RSTRT:  CLR     PASNUM          ;CLEAR PASS NUMBER
481 013516  012700  003142                  MOV     #ERRCNT,R0
482 013522  012701  000100                  MOV     #64.,R1         ;GET A COUNT
483 013526  005020                  1$:     CLR     (R0)+           ;CLEAR A ERROR COUNTER STORAGE AREA
484 013530  005301                          DEC     R1
485 013532  001375                          BNE     1$              ;LOOP TILL ALL CLEARED
486 013534  012737  003140  003140          MOV     #ERRCNT-2,ERRPOINT ;INIT ERROR POINTER
```

```
487 013542  012737  177777  003344          MOV     #-1,PSETNM      ;SET PARAM SELECT TO INITIAL VALUE
488 013550  012737  177777  002712          MOV     #-1,HADONE      ;PRESET HEAD ALIGN DONE FLAG
489 013556  032737  040000  013346  LAB:    BIT     #LOCYL,MISWIW   ;TEST IF LO LIMIT SET
490 013564  001002                          BNE     5$              ;YES - SKIP
491 013566  005037  013350                  CLR     LOLIMW          ;ELSE CLEAR LO LIMIT
492 013572  000432                  5$:     BR      SETDON
493 013574                           RESTART:
494 013574                                   READEF  #EF.RESTART     ;CHECK IF RESTART
    013574  012700  000037                  MOV     #EF.RESTART,R0
    013600  104050                          EMT     C$REFG
495 013602                                   BCOMPLETE       RSTRT   ;NO - SKIP
    013602  103743                          BCS     RSTRT
496 013604                           CONTINUE:
497 013604                                   READEF  #EF.CONTINUE    ;TEST IF CONTINUE
    013604  012700  000036                  MOV     #EF.CONTINUE,R0
    013610  104050                          EMT     C$REFG
498 013612                                   BCOMPLETE       PWCON
    013612  103533                          BCS     PWCON
499                                  ;       ON CONTINUE PICK UP UNIT LAST UNDER TEST.
500 013614                                   READEF  #EF.NEW         ;CHECK IF STARTING NEW PASS
    013614  012700  000035                  MOV     #EF.NEW,R0
    013620  104050                          EMT     C$REFG
501 013622                                   BCOMPLETE       PASNEW
    013622  103403                          BCS     PASNEW
502 013624                           NXTPAS:
503 013624  005737  002776                  TST     DRVCNT          ;TEST IF ALL UNITS CHECKED
504 013630  001013                          BNE     SETDON          ;NO - SKIP
505 013632  005237  003342          PASNEW: INC     PASNUM          ;ELSE BUMP PASS COUNT
506 013636  012737  003140  003140          MOV     #ERRCNT-2,ERRPOINT ;INIT ERROR POINTER
507 013644  013737  002012  002776          MOV     L$UNIT,DRVCNT   ;GET ALL DRIVES
508 013652  012737  177777  003344          MOV     #-1,PSETNM      ;SET PARAM SELECT TO INITIAL
509 013660  005237  003344          SETDON: INC     PSETNM          ;NEXT SET OF PARAMETERS
510 013664  005337  002776                  DEC     DRVCNT          ;DOWN COUNT DRIVE TOTAL
511 013670  062737  000002  003140          ADD     #2,ERRPOINT     ;UPDATE THE ERROR POINTER
512 013676  013700  003344                  MOV     PSETNM,R0       ;SET UP TO GET PARAMETERS
513 013702  012702  002730                  MOV     #RLBAS,R2
514 013706                                   GPHARD  R0,R1
    013706  104042                          EMT     C$GPHRD
    013710  010001                          MOV     R0,R1
515 013712                                   BCOMPLETE       7$      ;SKIP IF GOOD PARAM
    013712  103406                          BCS     7$
516 013714  005737  003352                  TST     PWRFLG          ;RECENT POWER FAILURE
517 013720  001741                          BEQ     NXTPAS  ;NO
518 013722  005337  003352                  DEC     PWRFLG          ;ACCOUNT FOR DRIVE
519 013726  000736                          BR      NXTPAS
520 013730  012122                  7$:     MOV     (R1)+,(R2)+     ;STORE PARAMETERS CSR
521 013732  012122                          MOV     (R1)+,(R2)+     ;               VECTOR
522 013734  005721                          TST     (R1)+           ;BUMP PAST PRIORITY
523 013736  012137  002200                  MOV     (R1)+,T.DRIVE
524 013742  012122                          MOV     (R1)+,(R2)+
525 013744  022737  000001  002200          CMP     #1,T.DRIVE
526 013752  001426                          BEQ     65$
527 013754  012737  000776  002210          MOV     #510.,NXTHL
528 013762  012737  000777  002204          MOV     #511.,HLMTW
529 013770  012737  001000  002212          MOV     #512.,GBND
530 013776  012737  177600  002214          MOV     #177600,CAMSK
531 014004  012737  177600  002216          MOV     #177600,DIRMSK
```

```
 532 014012  012737  177600  002220          MOV     #177600,HDCYL
 533 014020  012737  177000  002206          MOV     #177000,CLRBYT
 534 014026  000425                          BR      PWCON
 535
 536 014030  012737  000377  002204  65$:    MOV     #255.,HLMTW
 537 014036  012737  000400  002212          MOV     #256.,GBND
 538 014044  012737  077600  002214          MOV     #77600,CAMSK
 539 014052  012737  077600  002216          MOV     #77600,DIRMSK
 540 014060  012737  077600  002220          MOV     #77600,HDCYL
 541 014066  012737  000376  002210          MOV     #254.,NXTHL
 542 014074  012737  177400  002206          MOV     #177400,CLRBYT
 543
 544 014102  032737  020000  013346  PWCON:  BIT     #HICYL,MISWIW
 545 014110  001003                          BNE     1$
 546 014112  013737  002204  013352          MOV     HLMTW,HILIMW
 547 014120                      1$:         SETVEC  RLVEC,#INTHLR,#340      ;SET UP VECTOR
     014120  012746  000340                  MOV     #340,-(SP)
     014124  012746  014712                  MOV     #INTHLR,-(SP)
     014130  013746  002732                  MOV     RLVEC,-(SP)
     014134  012746  000003                  MOV     #3,-(SP)
     014140  104037                          EMT     C$SVEC
     014142  062706  000010                  ADD     #10,SP
 548 014146                                  SETPRI  #0              ;SET PRIORITY
     014146  012700  000000                  MOV     #0,R0
     014152  104041                          EMT     C$SPRI
 549 014154  013702  002730                  MOV     RLBAS,R2        ;SET RL BASE ADDRESS POINTER
 550
 551
 552
 553                             ;       CHECK IF DOING AUTO SIZE AND DROP DRIVE IF NOT READY AND
 554                             ;       ERROR SETS ON GET STATUS.
 555 014160  005737  003342                  TST     PASNUM          ;TEST IF PASS 0
 556 014164  001135                          BNE     22$             ;NO - SKIP
 557 014166  032737  000020  013346          BIT     #AUTOSZ,MISWIW  ;TEST IF DOING AUTO SIZE
 558 014174  001531                          BEQ     22$             ;NO - SKIP
 559                            ;CHECK IF UNIBUS ADDRESS IS THERE BEFORE WE CHECK DRIVE READY
 560 014176  005037  003350                  CLR     TRPFLG          ;TRAP OCCURANCE
 561 014202                                  SETVEC  ERRVEC,#TRPHAN,#340     ;SET TRAP VECTOR
     014202  012746  000340                  MOV     #340,-(SP)
     014206  012746  014704                  MOV     #TRPHAN,-(SP)
     014212  013746  003132                  MOV     ERRVEC,-(SP)
     014216  012746  000003                  MOV     #3,-(SP)
     014222  104037                          EMT     C$SVEC
     014224  062706  000010                  ADD     #10,SP
 562 014230  005762  000000                  TST     RLCS(R2)        ;ACCESS BUS
 563 014234  005737  003350                  TST     TRPFLG          ;TRAP OCCUR??
 564 014240  001032                          BNE     5$              ;YES, DON'T INVESTIGATE FURTHER
 565 014242  013705  002734                  MOV     RLDRV,R5        ;GET DRIVE NUMBER
 566 014246  052705  000200                  BIS     #CRDYMSK,R5     ;INSERT CONT READY
 567 014252  010562  000000                  MOV     R5,RLCS(R2)     ;LOAD IN DRIVE NUMBER
 568 014256  032762  000001  000000          BIT     #DRDYMSK,RLCS(R2) ;CHECK IF DRIVE IS READY
 569 014264  001072                          BNE     20$             ;YES - GO DO TEST
 570 014266  012762  000003  000004          MOV     #GETSTAT,RLDA(R2) ;ELSE INSERT GET STATUS
 571 014274  052705  000004                  BIS     #4,R5           ;LOAD R5 WITH GET STATUS FUNCTION
 572 014300  042705  000200                  BIC     #CRDYMSK,R5     ;CLEAR CONTROLLER READY
 573 014304  010562  000000                  MOV     R5,RLCS(R2)     ;LOAD CS REG
 574 014310                                  WAITMS  #4              ;WAIT 4 MS
```

```
     014310  012700  000004                    MOV     #4,R0
     014314  104026                            EMT     C$WTM
575  014316  032762  002000  000000            BIT     #OPIERR,RLCS(R2);TEST IF OPI SET
576  014324  001452                            BEQ     20$             ;NO - SKIP
577  014326                          5$:       CLRVEC  ERRVEC
     014326  013700  003132                    MOV     ERRVEC,R0
     014332  104036                            EMT     C$CVEC
578  014334                                    PRINTF  #FMT24,#DRVNAV
     014334  012746  006040                    MOV     #DRVNAV,-(SP)
     014340  012746  011434                    MOV     #FMT24,-(SP)
     014344  012746  000002                    MOV     #2,-(SP)
     014350  010600                            MOV     SP,R0
     014352  104017                            EMT     C$PNTF
     014354  062706  000006                    ADD     #6,SP
579  014360                          10$:      PRINTF  #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
     014360  005046                            CLR     -(SP)
     014362  153716  002735                    BISB    RLDRV+1,(SP)
     014366  012746  006033                    MOV     #DRVNAM,-(SP)
     014372  013746  002730                    MOV     RLBAS,-(SP)
     014376  012746  006022                    MOV     #BASADD,-(SP)
     014402  012746  010630                    MOV     #FMT5,-(SP)
     014406  012746  000005                    MOV     #5,-(SP)
     014412  010600                            MOV     SP,R0
     014414  104017                            EMT     C$PNTF
     014416  062706  000014                    ADD     #14,SP
580  014422                                    PRINTF  #FMT3
     014422  012746  010614                    MOV     #FMT3,-(SP)
     014426  012746  000001                    MOV     #1,-(SP)
     014432  010600                            MOV     SP,R0
     014434  104017                            EMT     C$PNTF
     014436  062706  000004                    ADD     #4,SP
581  014442                                    DODU    PSETNM          ;DROP DRIVE
     014442  013700  003344                    MOV     PSETNM,R0
     014446  104053                            EMT     C$DODU
582  014450                                    DOCLN
     014450  104044                            EMT     C$DCLN
583  014452                          20$:      CLRVEC  ERRVEC
     014452  013700  003132                    MOV     ERRVEC,R0
     014456  104036                            EMT     C$CVEC
584  014460                          22$:
595                                  ;CHECK IF POWER FAILURE WAIT IS NEEDED
596
597  014460  005737  003352          4$:       TST     PWRFLG          ;NEEDED???
598  014464  001434                            BEQ     8$              ;NO, SKIP
599
600  014466  013705  002734                    MOV     RLDRV,R5        ;DRIVE SELECT
601  014472  052705  000200                    BIS     #CRDYMSK,R5     ;SET CRDY
602  014476  010562  000000                    MOV     R5,RLCS(R2)     ;SELECT DRIVE
603  014502  012701  000074                    MOV     #60.,R1         ;SIXTY SECOND TIMER
604  014506  032762  000001  000000  9$:       BIT     #DRDYMSK,RLCS(R2)       ;DRIVE UP YET
605  014514  001020                            BNE     8$              ;YES START TEST
606
607  014516                                    WAITMS  #10.            ;WAIT A SECOND
     014516  012700  000012                    MOV     #10.,R0
     014522  104026                            EMT     C$WTM
608  014524  005301                            DEC     R1              ;SIXTY GONE BY
609  014526  001367                            BNE     9$              ;NO
```

```
610 014530                                      PRINTF  #FMT24,#NOPWR
    014530  012746  006057                     MOV     #NOPWR,-(SP)
    014534  012746  011434                     MOV     #FMT24,-(SP)
    014540  012746  000002                     MOV     #2,-(SP)
    014544  010600                             MOV     SP,R0
    014546  104017                             EMT     C$PNTF
    014550  062706  000006                     ADD     #6,SP
611 014554  000701                             BR      10$
612
613 014556                              8$:
614
615 014556                              ENDINIT
    014556                              L10014:
    014556  104011                             EMT     C$INIT
616 014560                              ENDMOD
617
618 014560                              BGNMOD  CLNCODE
619 014560                              BGNCLN
620
621 014560                                      SETVEC  ERRVEC,#TRPHAN,#340
    014560  012746  000340                     MOV     #340,-(SP)
    014564  012746  014704                     MOV     #TRPHAN,-(SP)
    014570  013746  003132                     MOV     ERRVEC,-(SP)
    014574  012746  000003                     MOV     #3,-(SP)
    014600  104037                             EMT     C$SVEC
    014602  062706  000010                     ADD     #10,SP
622
623 014606                                      SETPRI #7               ;SET PRORITY TO 7
    014606  012700  000007                     MOV     #7,R0
    014612  104041                             EMT     C$SPRI
624 014614  032762  000200  000000      2$:    BIT     #CRDYMSK,RLCS(R2)       ;TEST IF CONTROLLER READY
625 014622  001407                             BEQ     3$              ;NO LOOP UNTIL READY
626 014624  053762  002734  000000             BIS     RLDRV,RLCS(R2)  ;SET DRIVE NUMBER
627 014632  032762  000001  000000             BIT     #DRDYMSK,RLCS(R2)       ;TEST IF DRIVE BUSY
628 014640  001003                             BNE     5$              ;NO - SKIP
629 014642                              3$:    WAITMS  #3              ;WAIT 300 MS
    014642  012700  000003                     MOV     #3,R0
    014646  104026                             EMT     C$WTM
630 014650                              5$:    CLRVEC  RLVEC           ;RELEASE VEC
    014650  013700  002732                     MOV     RLVEC,R0
    014654  104036                             EMT     C$CVEC
631 014656  005737  003352                     TST     PWRFLG  ;PWR FAIL SET
632 014662  001402                             BEQ     7$              ;NO
633 014664  005337  003352                     DEC     PWRFLG
634 014670                              7$:    CLRVEC  ERRVEC
    014670  013700  003132                     MOV     ERRVEC,R0
    014674  104036                             EMT     C$CVEC
635 014676                              ENDCLN
    014676                              L10015:
    014676  104012                             EMT     C$CLEAN
636
637 014700                              BGNDU
638 014700  000240                             NOP
639 014702                              ENDDU
    014702                              L10016:
    014702  104055                             EMT     C$DU
640
```

```
    014704                                ENDMOD
    014704                                BGNMOD  GLBSUB

    014704  005237  003350                TRPHAN: INC     TRPFLG
    014710  000002                                RTI

    014712                                BGNSRV  INTHLR
                                          ;       INTERRUPT HANDLER. ABORTS WAIT TIMER AND STORES ALL RL11 REGS
    014712                                        ABORTWAIT
    014712  104021                                EMT     C$ABRT
    014714  012237  002746                        MOV     (R2)+,T.CS       ;STORE RL REGISTERS
    014720  012237  002750                        MOV     (R2)+,T.BA
    014724  012237  002752                        MOV     (R2)+,T.DA
    014730  011237  002754                        MOV     (R2),T.MP
    014734  012737  177777  002710                MOV     #-1,DONE         ;SET DONE FLAG
    014742  013702  002730                        MOV     RLBAS,R2         ;RESTORE R2
    014746                                ENDSRV
    014746                                L10017:
    014746  000002                                RTI

                                          ;       ERROR LIMIT CHECKING ROUTINE
                                          ;       DROPS DRIVE IF ERROR LIMIT EXCEEDED
    014750  027737  166164  013356        CKERLM: CMP     @ERRPOINT,ERLIMW         ;TEST IF ERROR LIMIT EXCEEDED
    014756  002453                                BLT     1$               ;NO - SKIP
    014760                                        INLOOP                   ;CHECK IF IN ERROR LOOP
    014760  104020                                EMT     C$INLP
    014762                                        BCOMPLETE       1$       ;YES - SKIP
    014762  103451                                BCS     1$
    014764                                        PRINTF  #FMT25,ERLIMW,#MEXERS
    014764  012746  010306                        MOV     #MEXERS,-(SP)
    014770  013746  013356                        MOV     ERLIMW,-(SP)
    014774  012746  011441                        MOV     #FMT25,-(SP)
    015000  012746  000003                        MOV     #3,-(SP)
    015004  010600                                MOV     SP,R0
    015006  104017                                EMT     C$PNTF
    015010  062706  000010                        ADD     #10,SP
    015014                                        PRINTF  #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
    015014  005046                                CLR     -(SP)
    015016  153716  002735                        BISB    RLDRV+1,(SP)
    015022  012746  006033                        MOV     #DRVNAM,-(SP)
    015026  013746  002730                        MOV     RLBAS,-(SP)
    015032  012746  006022                        MOV     #BASADD,-(SP)
    015036  012746  010630                        MOV     #FMT5,-(SP)
    015042  012746  000005                        MOV     #5,-(SP)
    015046  010600                                MOV     SP,R0
    015050  104017                                EMT     C$PNTF
    015052  062706  000014                        ADD     #14,SP
    015056                                        PRINTF  #FMT3
    015056  012746  010614                        MOV     #FMT3,-(SP)
    015062  012746  000001                        MOV     #1,-(SP)
    015066  010600                                MOV     SP,R0
    015070  104017                                EMT     C$PNTF
    015072  062706  000004                        ADD     #4,SP
    015076                                        DODU    PSETNM           ;DROP DRIVE
    015076  013700  003344                        MOV     PSETNM,R0
    015102  104053                                EMT     C$DODU
    015104                                        DOCLN                    ;GO TO CLEAN UP
```

```
     015104  104044                          EMT     C$DCLN
669  015106  000207              1$:         RTS     PC
670
671                              ;           READ AND STORE ALL RL11 REGISTERS
672  015110  016237  000000  002746 READRL:  MOV     RLCSR(R2),T.CS  ;GET CS REG
673  015116  016237  000002  002750          MOV     RLBA(R2),T.BA   ;GET BUS ADDRESS REG
674  015124  016237  000004  002752          MOV     RLDA(R2),T.DA   ;GET DISK ADDRESS
675  015132  016237  000006  002754          MOV     RLMP(R2),T.MP   ;GET MULTI-PURPOSE REG
676  015140  000207                          RTS     PC              ;RETURN
677
678                              ;           WAIT FOR CONTROLLER TIMEOUT TO FORCE INTERRUPT ROUTINE
679  015142  011646              WAITIN:     MOV     (SP),-(SP)      ;MAKE ROOM FOR ERROR POINTER
680  015144  005066  000002                  CLR     2(SP)           ;CLEAR FOR POINTER
681  015150  032762  000200  000000          BIT     #CRDYMSK,RLCSR(R2)      ;TEST IF CONTROLLER READY
682  015156  001420                          BEQ     4$              ;NO - SKIP TO WAIT
683  015160  004737  015110                  JSR     PC,READRL       ;READ ALL RL REGS
684  015164  005737  002710                  TST     DONE            ;TEST IF INTERRUPT OCCURRED
685  015170  001433                          BEQ     5$              ;NO - GO SET NO INTERRUPT ERR FLAG
686  015172  012766  006177  000002  1$:     MOV     #MTOSLOW,2(SP)  ;ELSE SET TO SLOW ERROR POINTER
687  015200  032737  002000  002746          BIT     #OPIERR,T.CS    ;TEST IF OPI SET
688  015206  001403                          BEQ     2$              ;NO - SKIP
689  015210  012766  006216  000002          MOV     #MDRRES,2(SP)   ;SET MESSAGE FOR NO DRIVE RESPONSE
690  015216  000207              2$:         RTS     PC              ;RETURN
691  015220                      4$:         WAITMS  #3              ;WAIT 300 MS FOR TIMEOUT
     015220  012700  000003                  MOV     #3,R0
     015224  104026                          EMT     C$WTM
692  015226  032762  000200  000000          BIT     #CRDYMSK,RLCS(R2)       ;TEST IF READY NOW SET
693  015234  001006                          BNE     3$              ;YES - SKIP
694  015236  004737  015110                  JSR     PC,READRL       ;READ RL REGS
695  015242  012766  006271  000002          MOV     #MCONHNG,2(SP)  ;SET MESSAGE FOR CONTROLLER HUNG
696  015250  000762                          BR      2$              ;SKIP
697  015252  005737  002710      3$:         TST     DONE            ;ELSE CHECK IF INTERRUPT OCCURRED
698  015256  001345                          BNE     1$              ;YES - SKIP TO SET TO SLOW
699  015260  004737  015110      5$:         JSR     PC,READRL       ;READ RL REGS
700  015264  012766  006236  000002          MOV     #MNOINT,2(SP)   ;ELSE SET NO INTERRUPT FLAG
701  015272  000751                          BR      2$              ;GO TO RETURN
702
703                              ;           OPERATION AND TEST INITIALIZE ROUTINE
704  015274  005037  002706      TSTINT:     CLR     OPFLAG          ;CLEAR OPERATION FLAGS
705  015300  105037  003347                  CLRB    NOERCT          ;RESET INHIBIT ERROR COUNTING
706  015304  005037  002716                  CLR     MORECE          ;RESET MORE COMPARE ERRORS
707  015310  000207                          RTS     PC
708
709                              ;           GET STATUS AND GET STATUS WITH RESET ROUTINE
710  015312  013746  003030      GSTATR:     MOV     TEMP4,-(SP)     ;STORE TEMP4
711  015316  012737  000013  003030          MOV     #GETSTAT!DRSET,TEMP4    ;SET FOR RESET
712  015324  000412                          BR      GSTATG
713  015326  013746  003030      GSTATC:     MOV     TEMP4,-(SP)     ;STORE TEMP4
714  015332  012737  000003  003030          MOV     #GETSTAT,TEMP4  ;SET FOR NO RESET
715  015340  000404                          BR      GSTATG
716  015342  013746  003030      GSTAT:      MOV     TEMP4,-(SP)     ;STORE TEMP4
717  015346  005037  003030                  CLR     TEMP4           ;SET FOR SAVE L. AND T. REGS
718  015352  010346              GSTATG:     MOV     R3,-(SP)        ;STORE R3
719  015354  013703  002704                  MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
720  015360  005723                          TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
721  015362  016663  000004  002306          MOV     4(SP),SUBSTK(R3) ;INSERT THIS CALL
722  015370  162763  000004  002306          SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
```

```
723 015376  010337  002704                  MOV     R3,SSINDX       ;STORE IT BACK
724 015402  010046                          MOV     R0,-(SP)        ;STORE R0
725 015404  010146                          MOV     R1,-(SP)        ;STORE R1
726 015406  012737  000002  002720          MOV     #2,ERRSWI       ;SET FOR NO ERROR RETURN
727 015414  032737  000010  003030          BIT     #DRSET,TEMP4    ;TEST IF DRIVE RESET
728 015422  001453                          BEQ     11$             ;NO - SKIP
729 015424  032762  040000  000000          BIT     #DRVERR,RLCS(R2) ;TEST IF DRIVE ERROR SET
730 015432  001403                          BEQ     49$             ;NO - SKIP
731 015434                                  WAITMS  #3              ;WAIT FOR 300 MS FOR DRIVE TO SETTLE
    015434  012700  000003                  MOV     #3,R0
    015440  104026                          EMT     C$WTM
732 015442  012701  000062          49$:    MOV     #50.,R1         ;SET WAIT FOR 5 SEC
733 015446  004737  015342          50$:    JSR     PC,GSTAT        ;GET DRIVE STATUS
734 015452  016106                          3$
735 015454  032737  000001  002746          BIT     #DRDYMSK,T.CS   ;TEST IF DRIVE READY
736 015462  001051                          BNE     5$              ;YES - GO DO CLEAR
737 015464  032737  000020  002754          BIT     #HOSTAT,T.MP    ;ELSE TEST IF HEADS OUT
738 015472  001010                          BNE     51$             ;YES - BYPASS RELOAD WAIT FLAG SETTING
739 015474  032737  144000  002754          BIT     #SPDSTAT!HCESTAT!WDESTAT,T.MP ;TEST IF DRIVE HAS ERROR
740                                                                 ;THAT CAUSED HEADS TO
741                                                                 ;UNLOAD
742 015502  001441                          BEQ     5$              ;NO - SKIP
743 015504  052737  040000  002706          BIS     #RELDWT,OPFLAG  ;ELSE SET WAIT FLAG
744 015512  000435                          BR      5$              ;SKIP TO CLEAR
745 015514  032737  040000  002746  51$:    BIT     #DRVERR,T.CS    ;TEST IF DRIVE ERROR NOW
746 015522  001031                          BNE     5$              ;YES - SKIP TO CLEAR
747 015524                                  WAITMS  #1              ;WAIT FOR DRIVE TO GET ERROR, RDY, OR HO
    015524  012700  000001                  MOV     #1,R0
    015530  104026                          EMT     C$WTM
748 015532  005301                          DEC     R1              ;DEC WAIT COUNTER
749 015534  001344                          BNE     50$             ;IF NOT DONE, LOOP
750 015536  012703  010164                  MOV     #MUNDEF,R3      ;MESSAGE FOR UNDEFINED STATE
751 015542                                  ERRHRD  10001.,,ERR1
    015542  104443                          TRAP    T$ERCODE
    015544  023421                          .WORD   10001
    015546  011526                          .WORD   ERR1
752 015550  000554                          BR      14$             ;EXIT
753 015552  005737  003030          11$:    TST     TEMP4           ;TEST IF SAVE REGISTERS
754 015556  001013                          BNE     5$              ;NO SKIP
755 015560  012701  000004                  MOV     #4,R1           ;SET SAVE COUNT
756 015564  012703  002746                  MOV     #L.MP+2,R3      ;SET ADDRESS OF FIRST SAVE
757 015570  014346                  8$:     MOV     -(R3),-(SP)     ;PUT REG ON STACK
758 015572  005301                          DEC     R1              ;DEC COUNT
759 015574  001375                          BNE     8$              ;LOOP UNTIL ALL SAVED
760 015576  012737  000003  002742          MOV     #GETSTAT,L.DA   ;SET FOR GET STATUS
761 015604  000403                          BR      6$              ;SKIP
762 015606  013737  003030  002742  5$:     MOV     TEMP4,L.DA      ;INSERT PRESET FOR STATUS
763 015614                          6$:
764 015614  005037  002710                  CLR     DONE            ;CLEAR INTERRUPT FLAG
765 015620  013737  002734  002736          MOV     RLDRV,L.CS      ;SET UP TO GET STATUS
766 015626  042737  002000  002736          BIC     #BIT10,L.CS     ;CLEAR FOR DRIVE 4 - 7 SPEC'D
767 015634  052737  000104  002736          BIS     #GTSTAT,L.CS
768 015642  013762  002742  000004          MOV     L.DA,RLDA(R2)   ;LOAD RL REGS
769 015650  013762  002736  000000          MOV     L.CS,RLCSR(R2)  ;LOAD CS REG
770 015656                                  WAITUS  #1              ;WAIT 100 US FOR INTERRUPT
    015656  012700  000001                  MOV     #1,R0
    015662  104027                          EMT     C$WTU
```

```
 771 015664  005737  002710                    TST     DONE            ;CHECK IF INTERRUPT OCCURRED
 772 015670  001476                            BEQ     1$              ;NO - SKIP
 773 015672  013737  002754  002762  4$:       MOV     T.MP,T.STAT     ;STORE MP REGISTER
 774 015700  042737  177770  002762            BIC     #^C<STAMSK>,T.STAT ;CLEAR ALL BUT STATE
 775 015706  032737  000010  002742            BIT     #DRSET,L.DA     ;TEST IF RESET WAS SPECIFIED
 776 015714  001474                            BEQ     3$              ;NO - SKIP TO EXIT
 777 015716  032737  040000  002706            BIT     #RELDWT,OPFLAG  ;TEST IF RELOAD WAIT FLAG SET
 778 015724  001424                            BEQ     12$             ;NO - SKIP
 779 015726  012701  001130                    MOV     #600.,R1        ;SET WAIT COUNT FOR 60 SECONDS
 780 015732  032762  000001  000000  13$:      BIT     #DRDYMSK,RLCS(R2) ;TEST IF DRIVE NOW READY
 781 015740  001016                            BNE     12$             ;YES - SKIP
 782 015742                                    WAITMS  #1              ;CALL WAIT
     015742  012700  000001                    MOV     #1,R0
     015746  104026                            EMT     C$WTM
 783 015750  005301                            DEC     R1              ;DEC COUNT
 784 015752  001367                            BNE     13$             ;LOOP IF NOT 0
 785 015754  004737  015342                    JSR     PC,GSTAT        ;GET DRIVE STATUS
 786 015760  016106                            3$                      ;ERROR RETURN
 787 015762  012703  010231                    MOV     #MRLFAL,R3      ;SET RESULT MESSAGE POINTER
 788 015766                                    ERRHRD  10003.,,ERR1
     015766  104443                            TRAP    T$ERCODE
     015770  023423                            .WORD   10003
     015772  011526                            .WORD   ERR1
 789 015774  000442                            BR      14$             ;GO TO EXIT
 790 015776                            12$:    WAITUS  #10.            ;WAIT FOR 1MS
     015776  012700  000012                    MOV     #10.,R0
     016002  104027                            EMT     C$WTU
 791 016004  004737  015342                    JSR     PC,GSTAT        ;GET DRIVE STATUS
 792 016010  016106                            3$
 793 016012  032737  100000  002746            BIT     #ANYERR,T.CS    ;TEST IF ANY ERROR
 794 016020  001432                            BEQ     3$              ;NO - SKIP
 795 016022  032737  001000  002754            BIT     #VCSTAT,T.MP    ;CHECK IF VOLUME CHECK RESET
 796 016030  001403                            BEQ     7$              ;YES SKIP
 797 016032  012703  006325                    MOV     #VCNRST,R3      ;SET REASON POINTER
 798 016036  000416                            BR      2$              ;EXIT
 799 016040  032737  040000  002746  7$:       BIT     #DRVERR,T.CS    ;CHECK IF DRIVE ERROR
 800 016046  001404                            BEQ     9$              ;NO - SKIP
 801 016050                                    ERRHRD  10004.,,ERR6
     016050  104443                            TRAP    T$ERCODE
     016052  023424                            .WORD   10004
     016054  012030                            .WORD   ERR6
 802 016056  000411                            BR      14$             ;EXIT
 803 016060  012703  006346            9$:     MOV     #UNXERR,R3      ;SET REASON POINTER
 804 016064  000403                            BR      2$              ;EXIT
 805 016066  004737  015142            1$:     JSR     PC,WAITIN       ;WAIT FOR INTERRUPT
 806 016072  012603                            MOV     (SP)+,R3        ;STORE REASON POINTER FOR RETURN
 807 016074                            2$:     ERRHRD  10002.,,ERR1
     016074  104443                            TRAP    T$ERCODE
     016076  023422                            .WORD   10002
     016100  011526                            .WORD   ERR1
 808 016102  005037  002720            14$:    CLR     ERRSWI          ;CLEAR FOR ERROR RETURN
 809 016106  005737  003030            3$:     TST     TEMP4           ;TEST IF REGISTERS WERE SAVED
 810 016112  001007                            BNE     22$             ;NO - SKIP
 811 016114  012703  002736                    MOV     #L.CS,R3        ;SET POINTER TO RESTORE
 812 016120  012701  000004                    MOV     #4,R1           ;SET REGISTER COUNT
 813 016124  012623                    20$:    MOV     (SP)+,(R3)+     ;RESTORE REG
 814 016126  005301                            DEC     R1              ;DEC COUNT
```

```
                                                BNE     20$             ;LOOP UNTIL ALL ARE RESTORED
016130  001375
016132  162737  000002  002704    22$:          SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
016140  012601                                  MOV     (SP)+,R1        ;RESTORE R1
016142  012600                                  MOV     (SP)+,R0
016144  012603                                  MOV     (SP)+,R3        ;RESTORE R3
016146  012637  003030                          MOV     (SP)+,TEMP4     ;RESTORE TEMP4
016152  005737  002720                          TST     ERRSWI          ;TEST IF ERROR RETURN
016156  001403                                  BEQ     99$             ;YES - SKIP
016160  063716  002720                          ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
016164  000207                                  RTS     PC
016166  017616  000000            99$:          MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
016172  000207                                  RTS     PC
                                                ;
                                                ;       SEEK ROUTINE
                                                ;
016174  012737  177777  003022    XSEEKT:       MOV     #-1,TEMP1       ;SET SPECIAL TIMING SEEK FLAG
016202  000402                                  BR      XSEEK1
016204  005037  003022            XSEEK:        CLR     TEMP1           ;CLEAR SPECIAL SEEK FOR TIMING FLAG
016210  010346                    XSEEK1:       MOV     R3,-(SP)        ;STORE R3
016212  013703  002704                          MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
016216  005723                                  TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
016220  016663  000002  002306                  MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
016226  162763  000004  002306                  SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
016234  010337  002704                          MOV     R3,SSINDX       ;STORE IT BACK
016240  010046                                  MOV     R0,-(SP)
016242  010146                                  MOV     R1,-(SP)
016244  010546                                  MOV     R5,-(SP)        ;STORE REG
016246  012737  000002  002720                  MOV     #2,ERRSWI       ;SET FOR NO ERROR RETURN
016254  005037  003000                          CLR     DIFAUG          ;CLEAR DIFFERENCE AUGMENT (FOR SEEKING
                                                ;         PAST GUARD BAND)
016260  004737  021274                          JSR     PC,GETPOS       ;GET PRESENT POSITION
016264  016706                                  65$
016266  013737  003006  003002                  MOV     CURCYL,OLDCYL   ;MOVE CURRENT TO OLD CYLINDER
016274  023737  003004  002204                  CMP     NEWCYL,HLMTW    ;TEST IF NEW IS GREATER THAN 255
016302  003427                                  BLE     3$              ;NO - SKIP
016304  163737  002204  003004                  SUB     HLMTW,NEWCYL    ;ELSE SUBTRACT 255.
016312  013737  003004  003000                  MOV     NEWCYL,DIFAUG   ;STORE DIFFERENCE AS AUGMENT
016320  013737  002204  003004                  MOV     HLMTW,NEWCYL    ;SET NEWCYL AS 255.
016326  022737  000001  002200                  CMP     #1,T.DRIVE
016334  001424                                  BEQ     6$
016336  162737  000001  003004                  SUB     #1,NEWCYL
016344  012737  000001  003012                  MOV     #1,DESSGN
016352  012737  000001  003010                  MOV     #1,DESDIF
016360  000451                                  BR      18$
016362  005737  003004            3$:           TST     NEWCYL          ;TEST IF NEWCYL HAS NEGATIVE VALUE
016366  100007                                  BPL     6$              ;NO - SKIP
016370  005437  003004                          NEG     NEWCYL          ;ELSE MAKE IT POSITIVE
016374  013737  003004  003000                  MOV     NEWCYL,DIFAUG   ;AND STORE IT AS AUGMENT
016402  005037  003004                          CLR     NEWCYL          ;AND SET NEWCYL TO 0
016406  013705  003006            6$:           MOV     CURCYL,R5       ;COMPUTE DIFFERENCE AND NEW CYLINDER
016412  163705  003004                          SUB     NEWCYL,R5       ;SUB NEWCYL FROM CURCYL
016416  100005                                  BPL     13$             ;IF DIFF IS POSITIVE - SKIP(REV SEEK)
016420  012737  000001  003012                  MOV     #1,DESSGN       ;ELSE SET SIGN FOR FORWARD
016426  005405                                  NEG     R5              ;MAKE DIFFERENCE POSITIVE
016430  000402                                  BR      14$             ;SKIP
016432  005037  003012            13$:          CLR     DESSGN          ;SET SIGN FOR REVERSE
016436  010537  003010            14$:          MOV     R5,DESDIF       ;STORE DIFFERENCE
```

```
 872 016442  005737  003000              TST     DIFAUG          ;IS THERE A DIFFERENCE AUGMENT
 873 016446  001416                      BEQ     18$             ;NO - SKIP
 874 016450  023737  003004  002204      CMP     NEWCYL,HLMTW    ;CHECK IF NEW CYL IS 255.
 875 016456  001007                      BNE     17$             ;NO - SKIP
 876 016460  012737  000001  003012      MOV     #1,DESSGN       ;ELSE FORCE SIGN FOR FORWARD
 877                                                             ;(INNER GUARD BAND)
 878 016466  022737  000001  002200      CMP     #1,T.DRIVE
 879 016474  001003                      BNE     18$
 880 016476  063737  003000  003010  17$: ADD     DIFAUG,DESDIF
 881 016504                          18$:
 882 016504  012705  002736              MOV     #L.CS,R5        ;GET L REG ADDRESS
 883 016510  012715  000106              MOV     #SEEK,(R5)      ;SET FOR SEEK
 884 016514  053715  002734              BIS     RLDRV,(R5)      ;INSERT DRIVE NUMBER
 885 016520  042725  002000              BIC     #BIT10,(R5)+    ;CLEAR IF DRIVE 4 - 7 SPEC'D
 886 016524  005025                      CLR     (R5)+           ;CLEAR BUS ADDRESS
 887 016526  013715  003010              MOV     DESDIF,(R5)     ;LOAD DIFFERENCE
 888 016532  012700  000007              MOV     #7,R0           ;SET TO SHIFT DIFFERENCE
 889 016536  006315                  21$: ASL     (R5)
 890 016540  005300                      DEC     R0
 891 016542  001375                      BNE     21$             ;LOOP UNTIL ALIGNED
 892 016544  005737  003012              TST     DESSGN          ;TEST SIGN
 893 016550  001402                      BEQ     23$             ;SKIP IF 0
 894 016552  052715  000004              BIS     #DIRBIT,(R5)    ;ELSE INSERT SIGN
 895 016556  005737  003014          23$: TST     DESHD           ;TEST IF HEAD 0
 896 016562  001402                      BEQ     25$             ;YES - SKIP
 897 016564  052715  000020              BIS     #HDSEL,(R5)     ;ELSE SET HEAD BIT
 898 016570  052725  000001          25$: BIS     #MBSET0,(R5)+   ;INSERT MARKER BIT
 899 016574  004737  017306              JSR     PC,RDYCHK       ;CHECK IF DRIVE READY
 900 016600  016706                      65$
 901 016602  005037  002710              CLR     DONE            ;CLEAR INTERRUPT FLAG
 902 016606  005737  003022              TST     TEMP1           ;CHECK IF SPECIAL SEEK FLAG SET
 903 016612  001035                      BNE     65$             ;YES - SKIP, DO NOT START SEEK
 904 016614  014562  000004              MOV     -(R5),RLDA(R2)  ;LOAD RL REGISTERS
 905 016620  014562  000002              MOV     -(R5),RLBA(R2)
 906 016624  014562  000000              MOV     -(R5),RLCS(R2)
 907 016630                          30$: WAITUS  #10.
     016630  012700  000012              MOV     #10.,R0
     016634  104027                      EMT     C$WTU
 908 016636  005737  002710              TST     DONE            ;TEST IF INTERRUPT DONE
 909 016642  001011                      BNE     32$             ;YES - SKIP
 910 016644  004737  015142              JSR     PC,WAITIN       ;GO WAIT FOR INTERRUPT
 911 016650  012603                      MOV     (SP)+,R3        ;GET RESULT MESSAGE POINTER
 912 016652                              ERRHRD  10005.,,ERR1
     016652  104443                      TRAP    T$ERCODE
     016654  023425                      .WORD   10005
     016656  011526                      .WORD   ERR1
 913 016660  005037  002720              CLR     ERRSWI          ;CLEAR FOR ERROR ERROR RETURN
 914 016664  000410                      BR      65$
 915 016666  005737  002746          32$: TST     T.CS            ;TEST IF ANY ERROR
 916 016672  100005                      BPL     65$             ;NO - SKIP
 917 016674                              ERRHRD  10006.,,ERR6
     016674  104443                      TRAP    T$ERCODE
     016676  023426                      .WORD   10006
     016700  012030                      .WORD   ERR6
 918 016702  005037  002720              CLR     ERRSWI          ;CLEAR FOR ERROR ERROR RETURN
 919 016706  162737  000002  002704  65$: SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
 920 016714  012605                      MOV     (SP)+,R5        ;RESTORE REGISTER
```

```
                                                MOV     (SP)+,R1
                                                MOV     (SP)+,R0
                                                MOV     (SP)+,R3        ;RESTORE R3
 924 016724  005737  002720                     TST     ERRSWI          ;TEST IF ERROR RETURN
 925 016730  001403                             BEQ     99$             ;YES - SKIP
 926 016732  063716  002720                     ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
 927 016736  000207                             RTS     PC
 928 016740  017616  000000             99$:    MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
 929 016744  000207                             RTS     PC
 930
 987
 989                                    ;       POSITION HEADS ROUTINE. POSITIONS HEADS USING 1 CYLINDER SEEKS
 990                                    ;       TO CYLINDER SPECIFIED IN R5 BY THE CALLING ROUTINE
 991 016746  010346                     POSHDS: MOV     R3,-(SP)        ;SAVE REGS
 992 016750  013703  002704                     MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
 993 016754  005723                             TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
 994 016756  016663  000002  002306             MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
 995 016764  162763  000004  002306             SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
 996 016772  010337  002704                     MOV     R3,SSINDX       ;STORE IT BACK
 997 016776  010346                             MOV     R3,-(SP)
 998 017000  010446                             MOV     R4,-(SP)
 999 017002  012737  000002  002720             MOV     #2,ERRSWI       ;SET FOR NO ERROR RETURN
1000 017010  004737  021274                     JSR     PC,GETPOS       ;GET CURRENT POSITION
1001 017014  017250                             PH65$
1002 017016  012704  000012                     MOV     #10.,R4         ;SET RETRY COUNT
1003 017022                             BGNSEG
     017022  104004                            EMT     C$BSEG
1004 017024                             1$:     INLOOP                  ;CHECK IF IN ERROR LOOP
     017024  104020                            EMT     C$INLP
1005 017026                                     BNCOMPLETE      5$      ;NO - SKIP
     017026  103012                            BCC     5$
1006 017030  004737  021274                     JSR     PC,GETPOS       ;ELSE GET POSITION
1007 017034  017246                             60$
1008 017036  023737  003006  003004             CMP     CURCYL,NEWCYL   ;CHECK IF AT INTENDED POSITION
1009 017044  001017                             BNE     8$              ;NO - SKIP
1010 017046  004737  017632                     JSR     PC,ONSWAP       ;SWAP OLDCYL AND NEWCYL
1011 017052  000414                             BR      8$              ;SKIP
1012 017054  013737  003006  003002     5$:     MOV     CURCYL,OLDCYL   ;IN NOT LOOPING, STORE CURCYL AS OLDCYL
1013 017062  023705  003006                     CMP     CURCYL,R5       ;CHECK IF HDS AT FINAL POSITION
1014 017066  001467                             BEQ     60$             ;YES - GO TO EXIT
1015 017070  003003                             BGT     7$              ;IF CURCYL > FINAL POSITION - SKIP
1016 017072  005237  003004                     INC     NEWCYL          ;ELSE BUMP NEWCYL (MOVE HDS IN)
1017 017076  000402                             BR      8$              ;SKIP
1018 017100  005337  003004             7$:     DEC     NEWCYL          ;DEC NEWCYL (MOVE HDS OUT)
1019 017104  004737  016204             8$:     JSR     PC,XSEEK        ;DO SEEK
1020 017110  017246                             60$
1021 017112  012701  005670                     MOV     #3000.,R1       ;SET WAIT COUNT 300 MS
1022 017116  004737  021026                     JSR     PC,RDYWAIT      ;WAIT FOR DRIVE READY
1023 017122  017246                             60$
1024 017124  005737  002746                     TST     T.CS            ;TEST IF ANY ERROR
1025 017130  100006                             BPL     10$             ;NO - SKIP
1026 017132                                     ERRHRD  10008.,,ERR6
     017132  104443                            TRAP    T$ERCODE
     017134  023430                            .WORD   10008
     017136  012030                            .WORD   ERR6
1027 017140  005037  002720                     CLR     ERRSWI          ;CLEAR FOR ERROR ERROR RETURN
1028 017144  000440                             BR      60$
```

```
1029 017146 004737 021274       10$:  JSR     PC,GETPOS       ;GET POSITION
1030 017152 017246                    60$
1031 017154 023737 003006 003004      CMP     CURCYL,NEWCYL   ;CHECK IF ARRIVED AT DESIRED PLACE
1032 017162 001003                    BNE     15$             ;NO - SKIP
1033 017164 012704 000012       14$:  MOV     #10.,R4         ;ELSE INIT RETRY COUNT
1034 017170 000715                    BR      1$              ;GO DO NEXT SEEK
1035 017172 005737 003012       15$:  TST     DESSGN          ;TEST IF GOING IN
1036 017176 001016                    BNE     17$             ;YES - SKIP
1037 017200 023737 003006 003004      CMP     CURCYL,NEWCYL   ;CHECK IF HEADS DID NOT MOVE IN
1038 017206 003366                    BGT     14$             ;YES - SKIP
1039 017210 005304              16$:  DEC     R4              ;DEC RETRY COUNT
1040 017212 001334                    BNE     8$              ;DO ANOTHER SEEK IF NOT 0
1041 017214 012703 007234             MOV     #HDMOVF,R3      ;ELSE SET RESULT MESSAGE POINTER
1042 017220                           ERRHRD  10009.,,ERR1
     017220 104443                    TRAP    T$ERCODE
     017222 023431                    .WORD   10009
     017224 011526                    .WORD   ERR1
1043 017226 005037 002720             CLR     ERRSWI          ;CLEAR FOR ERROR ERROR RETURN
1044 017232 000405                    BR      60$
1045 017234 023737 003006 003004 17$: CMP     CURCYL,NEWCYL   ;HDS SHOULD MOVE OUT, CHK THEY DID
1046 017242 002750                    BLT     14$             ;YES - SKIP
1047 017244 000761                    BR      16$             ;ELSE GO DEC AND RETRY
1048 017246                     20$:
1049 017246                     60$:
1050 017246                     ENDSEG
     017246                     10000$:
     017246 104005                    EMT     C$ESEG
1051 017250 162737 000002 002704 PH65$: SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
1052 017256 012604                    MOV     (SP)+,R4        ;RESTORE REGISTERS
1053 017260 012600                    MOV     (SP)+,R0
1054 017262 012603                    MOV     (SP)+,R3
1055 017264 005737 002720             TST     ERRSWI          ;TEST IF ERROR RETURN
1056 017270 001403                    BEQ     99$             ;YES - SKIP
1057 017272 063716 002720             ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
1058 017276 000207                    RTS     PC
1059 017300 017616 000000       99$:  MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
1060 017304 000207                    RTS     PC
1061
1063                            ;     DRIVE READY TEST ROUTINE. CHECKS DEIVE IS READY. IF NOT, WAIT
1064                            ;     500MS FOR READY TO SET.
1065 017306 010346              RDYCHK: MOV     R3,-(SP)        ;STORE REGS
1066 017310 013703 002704             MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
1067 017314 005723                    TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
1068 017316 016663 000002 002306      MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
1069 017324 162763 000004 002306      SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
1070 017332 010337 002704             MOV     R3,SSINDX       ;STORE IT BACK
1071 017336 010046                    MOV     R0,-(SP)
1072 017340 010146                    MOV     R1,-(SP)
1073 017342 010446                    MOV     R4,-(SP)
1074 017344 012737 000002 002720      MOV     #2,ERRSWI       ;SET FOR NO ERROR RETURN
1075 017352 012701 011610             MOV     #5000.,R1       ;SET WAIT COUNT
1076 017356 004737 015342       1$:   JSR     PC,GSTAT        ;GET DRIVE STATUS
1077 017362 017502                    4$
1078 017364 032737 000001 002746      BIT     #DRDYMSK,T.CS   ;TEST IF DRIVE READY
1079 017372 001045                    BNE     5$              ;YES - EXIT
1080 017374                           WAITUS  #1
     017374 012700 000001             MOV     #1,R0
```

```
                017400  104027                          EMT     C$WTU
1081            017402  005301                          DEC     R1              ;DEC WAIT COUNT
1082            017404  001364                          BNE     1$              ;LOOP IF NOT 0
1083            017406  012703  007561                  MOV     #MDRDY,R3       ;SET RESULT MESSAGE POINTER
1084            017412  012704  010445                  MOV     #C500MS,R4      ;SET CONDITION MESSAGE POINTER
1085            017416                                  ERRHRD  10010.,,ERR5
                017416  104443                          TRAP    T$ERCODE
                017420  023432                          .WORD   10010
                017422  011760                          .WORD   ERR5
1086            017424  012701  000062                  MOV     #50.,R1         ;SET WAIT COUNT FOR 5 SECONDS
1087            017430  004737  015342          2$:     JSR     PC,GSTAT        ;GET DRIVE STATUS
1088            017434  017502                          4$
1089            017436  032737  000001  002746          BIT     #DRDYMSK,T.CS   ;TEST IF DRIVE READY
1090            017444  001005                          BNE     3$              ;YES - SKIP
1091            017446                                  WAITMS  #1              ;WAIT FOR 100MS
                017446  012700  000001                  MOV     #1,R0
                017452  104026                          EMT     C$WTM
1092            017454  005301                          DEC     R1              ;DEC WAIT COUNTER
1093            017456  001364                          BNE     2$              ;LOOP UNTIL TIME DONE
1094            017460  032737  100000  002746  3$:     BIT     #ANYERR,T.CS    ;TEST IF ANYERR SET
1095            017466  001405                          BEQ     4$              ;NO - SKIP
1096            017470                                  ERRHRD  10011.,,ERR6    ;REPORT ALL ERRORS
                017470  104443                          TRAP    T$ERCODE
                017472  023433                          .WORD   10011
                017474  012030                          .WORD   ERR6
1097            017476  005337  003142                  DEC     ERRCNT          ;REDUCE ERROR COUNT FOR DUAL ERRORS
1098            017502  005037  002720          4$:     CLR     ERRSWI          ;CLEAR FOR ERROR RETURN
1099            017506  162737  000002  002704  5$:     SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
1100            017514  012604                          MOV     (SP)+,R4        ;RESTORE REGS
1101            017516  012601                          MOV     (SP)+,R1
1102            017520  012600                          MOV     (SP)+,R0
1103            017522  012603                          MOV     (SP)+,R3
1104            017524  005737  002720                  TST     ERRSWI          ;TEST IF ERROR RETURN
1105            017530  001403                          BEQ     99$             ;YES - SKIP
1106            017532  063716  002720                  ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
1107            017536  000207                          RTS     PC
1108            017540  017616  000000          99$:    MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
1109            017544  000207                          RTS     PC
1110
1111                                            ;       CHOSE HEAD ROUTINE. PICKS HEAD 0 UNLESS SPECIFIC HEAD IS
1112                                            ;       SELECTED BY SOFTWARE PARAMETER.
1113            017546  005037  003014          CHOSHD: CLR     DESHD           ;CLEAR TO HEAD 0
1114            017552  032737  010000  013346          BIT     #HEADLM,MISWIW  ;TEST IF HEAD SPECIFIED
1115            017560  001403                          BEQ     1$              ;NO - SKIP
1116            017562  013737  013354  003014          MOV     HEADW,DESHD     ;INSERT SPECIFIED HEAD
1117            017570  000207                  1$:     RTS     PC
1118
1119                                            ;       SWAP HEAD ROUTINE. CHANGES SELECTED HEAD TO HEAD 1
1120                                            ;       UNLESS HEAD 0 SPECIFICALLY SELECTED BY SOFTWARE PARAMETER.
1121            017572  032737  010000  013346  SWAPHD: BIT     #HEADLM,MISWIW  ;TEST IF HEAD SPECIFIED
1122            017600  001011                          BNE     2$              ;YES - TAKE ABORT EXIT
1123            017602  005737  003014                  TST     DESHD           ;TEST IF HEAD ONE USED
1124            017606  001006                          BNE     2$              ;YES - TAKE ABORT EXIT
1125            017610  012737  000001  003014          MOV     #1,DESHD        ;ELSE SET FOR HEAD ONE
1126            017616  062716  000002                  ADD     #2,(SP)         ;BUMP PAST ABORT RETURN
1127            017622  000207                          RTS     PC              ;RETURN
1128            017624  017616  000000          2$:     MOV     @(SP),(SP)      ;GET ABORT DESTINATION
```

```
1129 017630  000207                    3$:     RTS     PC
1130
1131                                    ;       SWAP OLD CYLINDER AND NEW CYLINDER ROUTINE.
1132 017632  010046                    ONSWAP: MOV     R0,-(SP)        ;STORE R0
1133 017634  013700  003002                    MOV     OLDCYL,R0       ;MOVE OLD TO R0
1134 017640  013737  003004  003002            MOV     NEWCYL,OLDCYL   ;MOVE NEW TO OLD
1135 017646  010037  003004                    MOV     R0,NEWCYL       ;PUT OLD IN NEW
1136 017652  012600                            MOV     (SP)+,R0        ;RESTORE R0
1137 017654  000207                            RTS     PC
1138
1140                                    ;       BAD SECTOR FILES VALID CHECK ROUTINE. CHECKS IF BAD SECTOR
1141                                    ;       FILES HAVE BEEN READ AND STORED. IF NOT, REPORT AND FORCE
1142                                    ;       FORCE FILES TO LOOK LIKE ALL SECTORS OK.
1143 017656  005737  003354            CKBSVD: TST     BSFVAL          ;TEST IF BAD SECTORS STORED
1144 017662  001051                            BNE     5$              ;YES - EXIT
1145 017664                                    PRINTF  #FMT9,#BSNSTR   ;REPORT
      017664  012746  007464                    MOV     #BSNSTR,-(SP)
      017670  012746  011014                    MOV     #FMT9,-(SP)
      017674  012746  000002                    MOV     #2,-(SP)
      017700  010600                            MOV     SP,R0
      017702  104017                            EMT     C$PNTF
      017704  062706  000006                    ADD     #6,SP
1146 017710                                    PRINTF  #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
      017710  005046                            CLR     -(SP)
      017712  153716  002735                    BISB    RLDRV+1,(SP)
      017716  012746  006033                    MOV     #DRVNAM,-(SP)
      017722  013746  002730                    MOV     RLBAS,-(SP)
      017726  012746  006022                    MOV     #BASADD,-(SP)
      017732  012746  010630                    MOV     #FMT5,-(SP)
      017736  012746  000005                    MOV     #5,-(SP)
      017742  010600                            MOV     SP,R0
      017744  104017                            EMT     C$PNTF
      017746  062706  000014                    ADD     #14,SP
1147 017752                                    PRINTF  #FMT3
      017752  012746  010614                    MOV     #FMT3,-(SP)
      017756  012746  000001                    MOV     #1,-(SP)
      017762  010600                            MOV     SP,R0
      017764  104017                            EMT     C$PNTF
      017766  062706  000004                    ADD     #4,SP
1148 017772  012737  177777  003356            MOV     #-1,SBSFIL      ;FORCE FILES TO NO ENTRIES
1149 020000  012737  177777  003552            MOV     #-1,FBSFIL
1150 020006  000207                    5$:     RTS     PC
1151
1153                                    ;       READ HEADERS ROUTINE.
1154 020010  012737  000001  003030  XRDHDC: MOV     #1,TEMP4        ;SET FLAG TO BYPASS REG STORAGE
1155 020016  000402                            BR      XRDHDG          ;GO DO IT
1156 020020  005037  003030            XRDHD:  CLR     TEMP4           ;SET FLAG TO SAVE T. AMD L. REGS
1157 020024  010346                    XRDHDG: MOV     R3,-(SP)        ;STORE REGISTERS
1158 020026  013703  002704                    MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
1159 020032  005723                            TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
1160 020034  016663  000002  002306            MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
1161 020042  162763  000004  002306            SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
1162 020050  010337  002704                    MOV     R3,SSINDX       ;STORE IT BACK
1163 020054  010046                            MOV     R0,-(SP)
1164 020056  010146                            MOV     R1,-(SP)
1165 020060  010446                            MOV     R4,-(SP)
1166 020062  012737  000002  002720            MOV     #2,ERRSWI       ;SET FOR NO ERROR RETURN
```

```
1167 020070  005737  003030                    TST     TEMP4           ;TEST IF REGISTERS TO BE SAVED
1168 020074  001007                            BNE     2$              ;NO - SKIP
1169 020076  012703  002746                    MOV     #L.MP+2,R3      ;SET POINTER FOR REGS
1170 020102  012701  000004                    MOV     #4,R1           ;SET COUNT
1171 020106  014346                    1$:     MOV     -(R3),-(SP)     ;SAVE REGISTER
1172 020110  005301                            DEC     R1              ;DEC COUNT
1173 020112  001375                            BNE     1$              ;LOOP UNTIL ALL ARE SAVED
1174 020114  004737  017306            2$:     JSR     PC,RDYCHK       ;CHECK DRIVE READY
1175 020120  020354                            65$
1176 020122  005037  002710                    CLR     DONE            ;CLEAR INTERRUPT FLAG
1177 020126  012701  002736                    MOV     #L.CS,R1        ;GET ADDRESS OF LOAD REGS
1178 020132  013711  002734                    MOV     RLDRV,(R1)      ;LOAD DRIVE NUMBER
1179 020136  042711  002000                    BIC     #BIT10,(R1)     ;CLEAR FOR DRIVE 4 - 7 SPEC'D
1180 020142  052721  000110                    BIS     #RDHEAD,(R1)+   ;INSERT COMMAND
1181 020146  005021                            CLR     (R1)+           ;CLEAR BA
1182 020150  005021                            CLR     (R1)+           ;CLEAR DA
1183 020152  014162  000004                    MOV     -(R1),RLDA(R2)  ;LOAD RL11 REGS
1184 020156  014162  000002                    MOV     -(R1),RLBA(R2)
1185 020162  014162  000000                    MOV     -(R1),RLCSR(R2)
1186 020166                            3$:     WAITUS  #10.            ;WAIT 1MS FOR INTERRUPT
     020166  012700  000012                    MOV     #10.,R0
     020172  104027                            EMT     C$WTU
1187 020174  005737  002710                    TST     DONE            ;TEST IN INTERRUPT FLAG SET
1188 020200  001455                            BEQ     14$             ;NO - SKIP
1189 020202  032737  000001  002746    5$:     BIT     #DRDYMSK,T.CS   ;TEST IF DRIVE READY
1190 020210  001033                            BNE     10$             ;YES - SKIP
1191 020212  012703  007561                    MOV     #MDRDY,R3       ;SET NO READY MESSAGE
1192 020216  012704  010464                    MOV     #CAFDT,R4       ;CONDITION OF AFTER DATA XFER
1193 020222                                    ERRHRD  10017.,,ERR5
     020222  104443                            TRAP    T$ERCODE
     020224  023441                            .WORD   10017
     020226  011760                            .WORD   ERR5
1194 020230  012701  000062                    MOV     #50.,R1         ;SET WAIT COUNT FOR 5 SECONDS
1195 020234  004737  015342            4$:     JSR     PC,GSTAT        ;GET STATUS
1196 020240  020350                            60$
1197 020242  032737  000001  002746            BIT     #DRDYMSK,T.CS   ;TEST IF DRIVE HAS COME READY
1198 020250  001403                            BEQ     11$             ;NO - SKIP
1199 020252  005037  002720                    CLR     ERRSWI          ;CLEAR ERROR SWITCH
1200 020256  000410                            BR      10$             ;SKIP
1201 020260  005301                    11$:    DEC     R1              ;DEC WAIT COUNT
1202 020262  001364                            BNE     4$              ;LOOP UNTIL TIME DONE
1203 020264  012704  010475                    MOV     #C5SEC,R4       ;SET CONDITION AFTER 5 SECONDS
1204 020270                                    ERRHRD  10014.,,ERR5
     020270  104443                            TRAP    T$ERCODE
     020272  023436                            .WORD   10014
     020274  011760                            .WORD   ERR5
1205 020276  000424                            BR      60$             ;EXIT
1206 020300  005737  002746            10$:    TST     T.CS            ;CHECK FOR ANY ERRORS
1207 020304  100004                            BPL     12$             ;NO - SKIP
1208 020306                                    ERRHRD  10016.,,ERR6    ;REPORT ALL ERRORS
     020306  104443                            TRAP    T$ERCODE
     020310  023440                            .WORD   10016
     020312  012030                            .WORD   ERR6
1209 020314  000415                            BR      60$
1210 020316  012701  002756            12$:    MOV     #HDWRD2,R1      ;GET POINTER
1211 020322  016221  000006                    MOV     RLMP(R2),(R1)+  ;STORE LAST TWO HEADER WORDS
1212 020326  016221  000006                    MOV     RLMP(R2),(R1)+
```

```
1213 020332  000410                        BR      65$             ;EXIT
1214 020334  004737  015142          14$:   JSR     PC,WAITIN       ;WAIT FOR INTERRUPT
1215 020340  012603                        MOV     (SP)+,R3        ;GET RESULTS
1216 020342                                ERRHRD  10015.,,ERR1    ;REPORT
     020342  104443                        TRAP    T$ERCODE
     020344  023437                        .WORD   10015
     020346  011526                        .WORD   ERR1
1217 020350  005037  002720          60$:   CLR     ERRSWI          ;CLEAR FOR ERROR ERROR RETURN
1218 020354  005737  003030          65$:   TST     TEMP4           ;TEST IF REGISTERS WERE SAVED
1219 020360  001007                        BNE     22$             ;NO - SKIP
1220 020362  012703  002736                MOV     #L.CS,R3        ;SET POINTER TO RESTORE REGS
1221 020366  012701  000004                MOV     #4,R1           ;SET COUNT
1222 020372  012623                  20$:   MOV     (SP)+,(R3)+     ;RESTORE REGISTER
1223 020374  005301                        DEC     R1              ;DEC COUNT
1224 020376  001375                        BNE     20$             ;LOOP UNTIL ALL ARE RESTORED
1225 020400  162737  000002  002704  22$:   SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
1226 020406  012604                        MOV     (SP)+,R4        ;RESTORE REGS
1227 020410  012601                        MOV     (SP)+,R1
1228 020412  012600                        MOV     (SP)+,R0
1229 020414  012603                        MOV     (SP)+,R3
1230 020416  005737  002720                TST     ERRSWI          ;TEST IF ERROR RETURN
1231 020422  001403                        BEQ     99$             ;YES - SKIP
1232 020424  063716  002720                ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
1233 020430  000207                        RTS     PC
1234 020432  017616  000000          99$:   MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
1235 020436  000207                        RTS     PC
1236
1238                                ;       VERIFY HEADERS ROUTINE. COMPARES 40 HEADERS FOR CONTENT AND
1239                                ;       SEQUENCE.
1240 020440  010346                VERHDR: MOV     R3,-(SP)        ;STORE REGS
1241 020442  013703  002704                MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
1242 020446  005723                        TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
1243 020450  016663  000002  002306        MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
1244 020456  162763  000004  002306        SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
1245 020464  010337  002704                MOV     R3,SSINDX       ;STORE IT BACK
1246 020470  010046                        MOV     R0,-(SP)
1247 020472  010146                        MOV     R1,-(SP)
1248 020474  010446                        MOV     R4,-(SP)
1249 020476  010546                        MOV     R5,-(SP)
1250 020500  012737  000002  002720        MOV     #2,ERRSWI       ;SET FOR NO ERROR RETURN
1251 020506  052737  000002  002706        BIS     #HDRCMP,OPFLAG  ;SET HEADER COMPARE FLAG
1252 020514  005037  002716                CLR     MORECE          ;CLEAR MORE ERRORS FLAG
1253 020520  012704  003746                MOV     #IBUFF,R4       ;SET POINTER TO HEADERS
1254 020524  012705  003020                MOV     #TEMP0,R5       ;SET POINTER TO WORK AREA
1255 020530  005003                        CLR     R3              ;CLEAR FOR WORD COUNTER
1256 020532  011415                        MOV     (R4),(R5)       ;MOVE HDR WORD TO WORK AREA
1257 020534  011401                        MOV     (R4),R1         ;PUT WORD IN REG 1
1258 020536  042701  000177                BIC     #177,R1 ;CLEAR ALL BUT CYLINDER
1259 020542  012700  000007                MOV     #7,R0           ;SET SHIFT COUNT
1260 020546  006201                  3$:   ASR     R1              ;SHIFT
1261 020550  005300                        DEC     R0              ;DEC
1262 020552  001375                        BNE     3$              ;LOOP
1263 020554  020137  003004                CMP     R1,NEWCYL       ;CHECK IF CYLINDER PART GOOD
1264 020560  001406                        BEQ     4$              ;YES - SKIP
1265 020562                                ERRHRD  10018.,,ERR10   ;REPORT ERROR
     020562  104443                        TRAP    T$ERCODE
     020564  023442                        .WORD   10018
```

```
                020566  013116                          .WORD   ERR10
1266            020570  005037  002720                  CLR     ERRSWI          ;CLEAR FOR ERROR ERROR RETURN
1267            020574  000454                          BR      65$
1268            020576  012701  000050         4$:      MOV     #40.,R1         ;SET HEADER COUNT
1269            020602  042715  000100                  BIC     #HDHSEL,(R5)    ;CLEAR HEAD SELECT AND 0 BIT
1270            020606  005737  003014                  TST     DESHD           ;ARE WE USING HD 0?
1271            020612  001402                          BEQ     5$              ;YES - SKIP
1272            020614  052715  000100                  BIS     #HDHSEL,(R5)    ;INSERT HEAD BIT
1273            020620  005065  000002         5$:      CLR     2(R5)           ;CLEAR 2ND WORD OF WORK AREA
1274            020624  021524                 6$:      CMP     (R5),(R4)+      ;TEST FIRST WORD OK
1275            020626  001407                          BEQ     8$              ;YES - SKIP
1276            020630  005744                          TST     -(R4)           ;ELSE SET POINTER FOR ERROR
1277            020632                                  ERRHRD  10018.,,ERR10   ;REPORT
                020632  104443                          TRAP    T$ERCODE
                020634  023442                          .WORD   10018
                020636  013116                          .WORD   ERR10
1278            020640  005037  002720                  CLR     ERRSWI          ;CLEAR FOR ERROR RETURN
1279            020644  005724                          TST     (R4)+           ;RESET POINTER
1280            020646  005203                 8$:      INC     R3              ;BUMP WORD COUNTER
1281            020650  005724                          TST     (R4)+           ;TEST 2ND WORD IS 0
1282            020652  001407                          BEQ     12$             ;YES - SKIP
1283            020654  022544                          CMP     (R5)+,-(R4)     ;ADJUST POINTERS FOR REPORT
1284            020656                                  ERRHRD  10018.,,ERR10   ;REPORT
                020656  104443                          TRAP    T$ERCODE
                020660  023442                          .WORD   10018
                020662  013116                          .WORD   ERR10
1285            020664  005037  002720                  CLR     ERRSWI          ;CLEAR FOR ERROR RETURN
1286            020670  024524                          CMP     -(R5),(R4)+     ;RESET POINTERS
1287            020672  005724                 12$:     TST     (R4)+           ;BUMP PAST ECC WORD
1288            020674  005203                          INC     R3              ;BUMP WORD COUNTER
1289            020676  005215                          INC     (R5)            ;BUMP SECTOR OF EXPECTED HEADER
1290            020700  011500                          MOV     (R5),R0         ;MOVE EXPECTED HDR TO R0
1291            020702  042700  177700                  BIC     #^CHDSEC,R0     ;CLEAR ALL BUT SECTOR
1292            020706  022700  000050                  CMP     #40.,R0         ;TEST IF AT SECTOR 40
1293            020712  001002                          BNE     15$             ;NO - SKIP
1294            020714  042715  000077                  BIC     #HDSEC,(R5)     ;CLEAR SECTOR TO 0
1295            020720  005203                 15$:     INC     R3              ;BUMP HDR WORD COUNTER
1296            020722  005301                          DEC     R1              ;DEC HEADER COUNT
1297            020724  001337                          BNE     6$              ;LOOP IF NOT YET DONE
1298            020726  162737  000002  002704 65$:     SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
1299            020734  012605                          MOV     (SP)+,R5        ;RESTORE REGISTERS
1300            020736  012604                          MOV     (SP)+,R4
1301            020740  012601                          MOV     (SP)+,R1
1302            020742  012600                          MOV     (SP)+,R0
1303            020744  012603                          MOV     (SP)+,R3
1304            020746  005737  002720                  TST     ERRSWI          ;TEST IF ERROR RETURN
1305            020752  001403                          BEQ     99$             ;YES - SKIP
1306            020754  063716  002720                  ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
1307            020760  000207                          RTS     PC
1308            020762  017616  000000         99$:     MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
1309            020766  000207                          RTS     PC
1310
1312                                            ;       POSITION HEAD BIT FROM HEADER OR MULTIPURPOSE REGISTER TO LSB.
1313            020770  013705  002754         POSHW1: MOV     HDWRD1,R5       ;START FOR POSITION HD BIT IN WD 1
1314            020774  000402                          BR      POSHD0          ;SKIP
1315            020776  013705  002754         POSHSB: MOV     I.MP,R5 ;START FOR POSITION HD BIT IN MP
1316            021002  010146                 POSHD0: MOV     R1,-(SP)        ;STORE R1
```

```
 1317 021004  042705  177677                BIC     #^CHSSTAT,R5    ;CLEAR ALL BUT HEAD SEL BIT
 1318 021010  012701  000006                MOV     #6,R1           ;SET SHIFT COUNT
 1319 021014  006205                1$:     ASR     R5              ;SHIFT FOR RIGHT JUSTIFY
 1320 021016  005301                        DEC     R1
 1321 021020  001375                        BNE     1$
 1322 021022  012601                        MOV     (SP)+,R1        ;RESTORE R1
 1323 021024  000207                        RTS     PC              ;RETURN
 1324
*1325                           ;       WAIT FOR READY ROUTINE. DURATION OF WAIT PASSED TO THE ROUTINE
 1326                           ;       FROM THE CALLING ROUTINE IN R1.
 1327 021026  010346            RDYWAIT:        MOV     R3,-(SP)        ;STORE R3
 1328 021030  013703  002704                MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
 1329 021034  005723                        TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
 1330 021036  016663  000002  002306        MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
 1331 021044  162763  000004  002306        SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
 1332 021052  010337  002704                MOV     R3,SSINDX       ;STORE IT BACK
 1333 021056  010046                        MOV     R0,-(SP)
 1334 021060  010146                        MOV     R1,-(SP)
 1335 021062  010446                        MOV     R4,-(SP)
 1336 021064  012737  000002  002720        MOV     #2,ERRSWI       ;SET FOR NO ERROR RETURN
 1337 021072  004737  015342        5$:     JSR     PC,GSTAT        ;GET DRIVE STATUS
 1338 021076  021230                        10$
 1339 021100  032737  000001  002746        BIT     #DRDYMSK,T.CS   ;CHECK IF READY
 1340 021106  001052                        BNE     9$              ;YES - SKIP
 1341 021110  005301                        DEC     R1              ;DEC WAIT COUNT
 1342 021112  001404                        BEQ     7$              ;SKIP IF 0
 1343 021114                                WAITUS  #1
      021114  012700  000001                MOV     #1,R0
      021120  104027                        EMT     C$WTU
 1344 021122  000763                        BR      5$
 1345 021124  012703  007561        7$:     MOV     #MDRDY,R3       ;SET NAME MESSAGE PTR
 1346 021130                                ERRHRD  10020.,,ERR3    ;REPORT READY ERROR
      021130  104443                        TRAP    T$ERCODE
      021132  023444                        .WORD   10020
      021134  011642                        .WORD   ERR3
 1347 021136  012701  000062                MOV     #50.,R1         ;SET WAIT COUNT FOR 5 SECONDS
 1348 021142  004737  015342        6$:     JSR     PC,GSTAT        ;GET DRIVE STATUS
 1349 021146  021230                        10$
 1350 021150  032737  000001  002746        BIT     #DRDYMSK,T.CS   ;TEST IF DRIVE READY
 1351 021156  001013                        BNE     8$              ;YES - SKIP
 1352 021160                                WAITMS  #1              ;WAIT 100 MS
      021160  012700  000001                MOV     #1,R0
      021164  104026                        EMT     C$WTM
 1353 021166  005301                        DEC     R1              ;DEC WAIT COUNT
 1354 021170  001364                        BNE     6$              ;LOOP UNTIL TIME DONE
 1355 021172  012704  010475                MOV     #C5SEC,R4       ;SET CONDITION AFTER 5 SECDS
 1356 021176                                ERRHRD  10021.,,ERR5
      021176  104443                        TRAP    T$ERCODE
      021200  023445                        .WORD   10021
      021202  011760                        .WORD   ERR5
 1357 021204  000407                        BR      11$             ;EXIT
 1358 021206  032737  100000  002746  8$:   BIT     #ANYERR,T.CS    ;TEST IF ANY ERROR SET
 1359 021214  001405                        BEQ     10$             ;NO - SKIP
 1360 021216                                ERRHRD  10022.,,ERR6    ;REPORT ALL ERRORS
      021216  104443                        TRAP    T$ERCODE
      021220  023446                        .WORD   10022
      021222  012030                        .WORD   ERR6
```

```
1361 021224  005337  003142          11$:   DEC   ERRCNT          ;DEC FOR DOUBLE ERROR REPORT
1362 021230  005037  002720          10$:   CLR   ERRSWI          ;CLEAR FOR ERROR ERROR RETURN
1363 021234  162737  000002  002704  9$:    SUB   #2,SSINDX        ;REMOVE ENTRY FROM SUBROUT STACK
1364 021242  012604                        MOV   (SP)+,R4         ;RESTORE REGISTERS
1365 021244  012601                        MOV   (SP)+,R1
1366 021246  012600                        MOV   (SP)+,R0
1367 021250  012603                        MOV   (SP)+,R3         ;RESTORE R3
1368 021252  005737  002720                TST   ERRSWI           ;TEST IF ERROR RETURN
1369 021256  001403                        BEQ   99$              ;YES - SKIP
1370 021260  063716  002720                ADD   ERRSWI,(SP)      ;ADD IN ERROR RETURN
1371 021264  000207                        RTS   PC
1372 021266  017616  000000          99$:   MOV   @(SP),(SP)       ;SET ERROR RETURN ADDRESS
1373 021272  000207                        RTS   PC
1374
1375                                 ;      GET POSITION ROUTINE. READS A HEADER FROM CURRENT CYLINDER
1376                                 ;      (WHERE IT IS PRESENTLY POSITIONED) AND STORES CYLINDER
1377                                 ;      NUMBER IN CURCYL.
1378 021274  010346                GETPOS: MOV   R3,-(SP)         ;STORE REGISTERS
1379 021276  013703  002704                MOV   SSINDX,R3        ;GET SUBROUTINE INDEX
1380 021302  005723                        TST   (R3)+            ;BUMP IT FOR NEXT ENTRY
1381 021304  016663  000002  002306        MOV   2(SP),SUBSTK(R3) ;INSERT THIS CALL
1382 021312  162763  000004  002306        SUB   #4,SUBSTK(R3)    ;ADJUST IT TO CALLING LOCATION
1383 021320  010337  002704                MOV   R3,SSINDX        ;STORE IT BACK
1384 021324  010046                        MOV   R0,-(SP)
1385 021326  010546                        MOV   R5,-(SP)
1386 021330  004737  020020                JSR   PC,XRDHD         ;DO READ HEADER
1387 021334  021364                        65$
1388 021336  013703  002754                MOV   HDWRD1,R3        ;GET HEADER WORD
1389 021342  012705  000007                MOV   #7,R5            ;SET SHIFT COUNT
1390 021346  006203                4$:     ASR   R3               ;SHIFT TO RIGHT JUSTIFY
1391 021350  005305                        DEC   R5
1392 021352  001375                        BNE   4$
1393 021354  042703  177000                BIC   #177000,R3
1394 021360  010337  003006                MOV   R3,CURCYL        ;STORE AS CURRENT CYLINDER
1395 021364  162737  000002  002704  65$:   SUB   #2,SSINDX        ;REMOVE ENTRY FROM SUBROUT STACK
1396 021372  012605                        MOV   (SP)+,R5         ;RESTORE REGISTERS
1397 021374  012600                        MOV   (SP)+,R0
1398 021376  012603                        MOV   (SP)+,R3
1399 021400  005737  002720                TST   ERRSWI           ;TEST IF ERROR RETURN
1400 021404  001403                        BEQ   99$              ;YES - SKIP
1401 021406  063716  002720                ADD   ERRSWI,(SP)      ;ADD IN ERROR RETURN
1402 021412  000207                        RTS   PC
1403 021414  017616  000000          99$:   MOV   @(SP),(SP)       ;SET ERROR RETURN ADDRESS
1404 021420  000207                        RTS   PC
1405
1407                                 ;      VERIFY POSITION ROUTINE. READS A HEADER (USING GETPOS) AND
1408                                 ;      CHECKS HEADS ARE POSITIONED AT NEW CYLINDER (CURCYL = NEWCYL).
1409 021422  010346                VERPOS: MOV   R3,-(SP)         ;STORE R3
1410 021424  013703  002704                MOV   SSINDX,R3        ;GET SUBROUTINE INDEX
1411 021430  005723                        TST   (R3)+            ;BUMP IT FOR NEXT ENTRY
1412 021432  016663  000002  002306        MOV   2(SP),SUBSTK(R3) ;INSERT THIS CALL
1413 021440  162763  000004  002306        SUB   #4,SUBSTK(R3)    ;ADJUST IT TO CALLING LOCATION
1414 021446  010337  002704                MOV   R3,SSINDX        ;STORE IT BACK
1415
1416 021452  012737  000002  002720        MOV   #2,ERRSWI        ;SET FOR NO ERROR RETURN
1417 021460  004737  021274                JSR   PC,GETPOS        ;GET POSITION
1418 021464  021510                        65$
```

```
1419 021466  023737  003004  003006          CMP     NEWCYL,CURCYL   ;CHECK IF CURRENT CYL IS NEW CYL
1420 021474  001405                          BEQ     1$              ;YES - SKIP
1421 021476                                  ERRHRD  10022.,ERR8
     021476  104443                          TRAP    T$ERCODE
     021500  023446                          .WORD   10022
     021502  012756                          .WORD   ERR8
1422 021504  005037  002720                  CLR     ERRSWI          ;CLEAR FOR ERROR ERROR RETURN
1423 021510                          1$:
1424 021510  162737  000002  002704  65$:    SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
1425 021516  012603                          MOV     (SP)+,R3        ;RESTORE R3
1426 021520  005737  002720                  TST     ERRSWI          ;TEST IF ERROR RETURN
1427 021524  001403                          BEQ     99$             ;YES - SKIP
1428 021526  063716  002720                  ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
1429 021532  000207                          RTS     PC
1430 021534  017616  000000          99$:    MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
1431 021540  000207                          RTS     PC
1432
1434                                 ;       READ ALL HEADERS ROUTINE. 40 HEADERS ARE READ AND STORED
1435                                 ;       IN IBUFF.
1436 021542  010346                  RDALHD: MOV     R3,-(SP)        ;STORE REGISTERS
1437 021544  013703  002704                  MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
1438 021550  005723                          TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
1439 021552  016663  000002  002306          MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
1440 021560  162763  000004  002306          SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
1441 021566  010337  002704                  MOV     R3,SSINDX       ;STORE IT BACK
1442 021572  010046                          MOV     R0,-(SP)
1443 021574  010146                          MOV     R1,-(SP)
1444 021576  010446                          MOV     R4,-(SP)
1445 021600  012737  000002  002720          MOV     #2,ERRSWI       ;SET FOR NO ERROR RETURN
1446 021606  012701  000050                  MOV     #40.,R1         ;SET HEADER COUNT
1447 021612  052737  100000  002706          BIS     #HDR40,OPFLAG   ;SET 40 HDR OP FLAG
1448 021620  012703  003746                  MOV     #IBUFF,R3       ;SET POINTER TO STORE HDRS
1449 021624  013704  002730                  MOV     RLBAS,R4        ;GET BASE ADDRESS
1450 021630  062704  000006                  ADD     #RLMP,R4        ;MAKE IT POINT TO MP REG
1451 021634  012737  000010  002736          MOV     #10,L.CS        ;LOAD FOR READ HEADER, NO INTERRUPT
1452 021642  053737  002734  002736          BIS     RLDRV,L.CS      ;INSERT DRIVE NUMBER
1453 021650  042737  002000  002736          BIC     #BIT10,L.CS     ;CLEAR FOR DRIVE 4 - 7 SPEC'D
1454 021656  005037  002740                  CLR     L.BA            ;CLEAR BA
1455 021662  005037  002742                  CLR     L.DA            ;CLEAR DA
1456 021666  005737  003014                  TST     DESHD           ;TEST IF HEAD 0
1457 021672  001403                          BEQ     3$              ;YES - SKIP
1458 021674  052737  000020  002742          BIS     #HDSEL,L.DA     ;ELSE INSERT HEAD 0
1459 021702  013762  002742  000004  3$:     MOV     L.DA,RLDA(R2)   ;LOAD RLDA REG
1460 021710  013762  002740  000002          MOV     L.BA,RLBA(R2)   ;LOAD RLBA
1461 021716  032762  000200  000000          BIT     #CRDYMSK,RLCS(R2)       ;TEST IF CONTROLLER READY
1462 021724  001003                          BNE     6$              ;YES - SKIP
1463 021726  004737  017306                  JSR     PC,RDYCHK       ;ELSE CHECK READY
1464 021732  022044                          65$
1465 021734  013762  002736  000000  6$:     MOV     L.CS,RLCS(R2)   ;LOAD RLCS REG
1466 021742  012700  077777                  MOV     #77777,R0       ;SET COUNT FOR WAIT
1467 021746  032762  000200  000000  7$:     BIT     #CRDYMSK,RLCS(R2) ;CHECK THAT OPERATION COMPLETED
1468 021754  001015                          BNE     8$              ;YES - SKIP
1469 021756  005300                          DEC     R0              ;DEC COUNT
1470 021760  001372                          BNE     7$              ;SKIP IF NOT YET 0
1471 021762  004737  015110                  JSR     PC,READRL       ;ELSE GET ALL REGISTERS
1472 021766  004737  015142                  JSR     PC,WAITIN       ;ELSE WAIT FOR TIMEOUT
1473 021772  012603                          MOV     (SP)+,R3        ;GET RESULT MESSAGE POINTER
```

```
1474 021774                                    ERRHRD  10025.,,ERR1
     021774  104443                            TRAP    T$ERCODE
     021776  023451                            .WORD   10025
     022000  011526                            .WORD   ERR1
1475 022002  005037  002720                    CLR     ERRSWI          ;CLEAR FOR ERROR RETURN
1476 022006  000416                            BR      65$
1477 022010  005737  002746          8$:       TST     T.CS            ;TEST FOR ANY ERRORS
1478 022014  100006                            BPL     12$             ;NO - SKIP
1479 022016                                    ERRHRD  10026.,,ERR6
     022016  104443                            TRAP    T$ERCODE
     022020  023452                            .WORD   10026
     022022  012030                            .WORD   ERR6
1480 022024  005037  002720                    CLR     ERRSWI          ;CLEAR FOR ERROR RETURN
1481 022030  000405                            BR      65$
1482 022032  011423                  12$:      MOV     (R4),(R3)+      ;STORE HEADER WORDS
1483 022034  011423                            MOV     (R4),(R3)+
1484 022036  011423                            MOV     (R4),(R3)+
1485 022040  005301                            DEC     R1              ;DEC HEADER COUNT
1486 022042  001334                            BNE     6$
1487 022044  162737  000002  002704  65$:      SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
1488 022052  012604                            MOV     (SP)+,R4        ;RESTORE REGISTERS
1489 022054  012601                            MOV     (SP)+,R1
1490 022056  012600                            MOV     (SP)+,R0
1491 022060  012603                            MOV     (SP)+,R3
1492 022062  005737  002720                    TST     ERRSWI          ;TEST IF ERROR RETURN
1493 022066  001403                            BEQ     99$             ;YES - SKIP
1494 022070  063716  002720                    ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
1495 022074  000207                            RTS     PC
1496 022076  017616  000000          99$:      MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
1497 022102  000207                            RTS     PC
1498
1499
1501                                 ;         GENERATE DATA ROUTINE. PATTERN TO BE GENERATED IS GIVEN
1502                                 ;         IN THE WORD FOLLOWING THE CALL. 128 WORDS ARE GENERATED
1503                                 ;         IN OBUFF.
1504 022104  010146                  DATGEN:   MOV     R1,-(SP)        ;STORE REGISTERS
1505 022106  010346                            MOV     R3,-(SP)
1506 022110  010446                            MOV     R4,-(SP)
1507 022112  012701  004346                    MOV     #OBUFF,R1       ;SET POINTER TO OBUFF
1508 022116  012504                            MOV     (R5)+,R4        ;GET DATA PATTERN SELECTOR
1509 022120  006304                            ASL     R4              ;ADJUST IT FOR INDEXING
1510 022122  016403  002262                    MOV     PATTBL(R4),R3   ;GET ADDRESS OF PATTERN
1511 022126  011321                            MOV     (R3),(R1)+      ;MOVE FIRST PATTERN WORD
1512 022130  001421                            BEQ     5$              ;SKIP IF PATTERN IS 0
1513 022132  021327  177777                    CMP     (R3),#-1        ;CHECK IF PATTERN IS ALL 1'S
1514 022136  001416                            BEQ     5$              ;YES - SKIP
1515 022140  020427  000010                    CMP     R4,#8.          ;TEST IF PATTERN 5
1516 022144  001403                            BEQ     3$              ;YES - SKIP
1517 022146  020427  000020                    CMP     R4,#16.         ;CHECK IF PATTERN 9 OR 10
1518 022152  002413                            BLT     6$              ;NO - SKIP
1519 022154  005723                  3$:       TST     (R3)+           ;BUMP SOURCE POINTER
1520 022156  012321                            MOV     (R3)+,(R1)+     ;MOVE TWO MORE WORDS FORM SOURCE
1521 022160  012321                            MOV     (R3)+,(R1)+
1522 022162  012704  000015                    MOV     #13.,R4         ;SET COUNT
1523 022166  012703  004346                    MOV     #OBUFF,R3       ;RESET POINTER
1524 022172  000406                            BR      8$
1525 022174  012703  004346          5$:       MOV     #OBUFF,R3       ;ELSE SET OBUFF AS PATTERN SOURCE
```

```
1526 022200  000401                          BR      7$              ;GO TO FILL
1527 022202  005723                  6$:     TST     (R3)+           ;BUMP SOURCE POINTER
1528 022204  012704  000017          7$:     MOV     #15.,R4         ;SET MOVE COUNT
1529 022210  012321                  8$:     MOV     (R3)+,(R1)+     ;MOVE 15 WORDS INTO BUFFER
1530 022212  005304                          DEC     R4
1531 022214  001375                          BNE     8$
1532 022216  012703  004346                  MOV     #OBUFF,R3       ;SET SOURCE TO TOP OF OBUFF
1533 022222  012704  000160                  MOV     #112.,R4        ;SET COUNT FOR REST OF BUFFER
1534 022226  012321                  10$:    MOV     (R3)+,(R1)+     ;REPEAT PATTERN IN BUFFER
1535 022230  005304                          DEC     R4
1536 022232  001375                          BNE     10$
1537 022234  012604                          MOV     (SP)+,R4        ;RESTORE REGISTERS
1538 022236  012603                          MOV     (SP)+,R3
1539 022240  012601                          MOV     (SP)+,R1
1540 022242  000205                          RTS     R5              ;RETURN
1541
1542                                 ;       DATA COMPARE ROUTINE. COMPARES THE CONTENTS OF IBUFF AND OBUFF.
1543                                 ;       ERROR REPORTING IS LIMITED BY SOFTWARE PARAMETER.
1544 022244  010346                  DATCOM: MOV     R3,-(SP)        ;STORE R3
1545 022246  013703  002704                  MOV     SSINDX,R3       ;GET SUBROUTINE STACK INDEX
1546 022252  005723                          TST     (R3)+           ;BUMP INDEX TO NEXT ENTRY
1547 022254  016663  000002  002306          MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
1548 022262  162763  000004  002306          SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
1549 022270  010337  002704                  MOV     R3,SSINDX       ;STORE IT BACK
1550 022274  010146                          MOV     R1,-(SP)        ;STORE OTHER REGISTERS
1551 022276  010446                          MOV     R4,-(SP)
1552 022300  010546                          MOV     R5,-(SP)
1553 022302  052737  000001  002706          BIS     #DATACMP,OPFLAG ;SET DATA COMPARE FLAG
1554 022310  005037  002716                  CLR     MORECE          ;CLEAR MORE ERROR FLAG
1555 022314  012705  004346                  MOV     #OBUFF,R5       ;SET POINTERS TO DATA FOR COMPARE
1556 022320  012704  003746                  MOV     #IBUFF,R4
1557 022324  012703  000001                  MOV     #1,R3           ;SET WORD COUNTER
1558 022330  012701  000200                  MOV     #128.,R1        ;SET COMPARE COUNT
1559 022334  022425                  5$:     CMP     (R4)+,(R5)+     ;COMPARE DATA
1560 022336  001052                          BNE     10$             ;ERROR - SKIP TO REPORT
1561 022340  005203                  7$:     INC     R3              ;BUMP WORD COUNT
1562 022342  005301                          DEC     R1              ;DEC COMPARE COUNT
1563 022344  001373                          BNE     5$              ;LOOP IF NOT 0
1564 022346  042737  000001  002706  9$:     BIC     #DATACMP,OPFLAG ;CLEAR DATA COMPARE FLAG
1565 022354  005737  002720                  TST     ERRSWI          ;TEST IF ANY COMPARE ERRORS
1566 022360  001021                          BNE     15$             ;NO - SKIP
1567 022362  012701  000200                  MOV     #128.,R1        ;SET REPORT VALUE
1568 022366                                  PRINTB  #FMT27,#TCERR,MORECE,#RESE6,R1
     022366  010146                          MOV     R1,-(SP)
     022370  012746  010377                  MOV     #RESE6,-(SP)
     022374  013746  002716                  MOV     MORECE,-(SP)
     022400  012746  007540                  MOV     #TCERR,-(SP)
     022404  012746  011475                  MOV     #FMT27,-(SP)
     022410  012746  000005                  MOV     #5,-(SP)
     022414  010600                          MOV     SP,R0
     022416  104014                          EMT     C$PNTB
     022420  062706  000014                  ADD     #14,SP
1569 022424  162737  000002  002704  15$:    SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
1570 022432  012605                          MOV     (SP)+,R5        ;RESTORE REGS
1571 022434  012604                          MOV     (SP)+,R4
1572 022436  012601                          MOV     (SP)+,R1
1573 022440  012603                          MOV     (SP)+,R3
```

```
1574 022442  005737  002720                    TST     ERRSWI          ;TEST IF ERROR RETURN
1575 022446  001403                            BEQ     99$             ;YES - SKIP
1576 022450  063716  002720                    ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
1577 022454  000207                            RTS     PC
1578 022456  017616  000000          99$:      MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
1579 022462  000207                            RTS     PC
1580 022464  023737  002716  013360  10$:      CMP     MORECE,DCLIMW   ;TEST IF COMPARE ERRORS LIMIT EXCEEDED
1581 022472  002010                            BGE     13$             ;YES - SKIP
1582 022474  024445                            CMP     -(R4),-(R5)     ;SET PTRS BACK TO ERROR WORDS
1583 022476                                    ERRHRD  10035.,,ERR10   ;REPORT ERROR
     022476  104443                            TRAP    T$ERCODE
     022500  023463                            .WORD   10035
     022502  013116                            .WORD   ERR10
1584 022504  005037  002720                    CLR     ERRSWI          ;CLEAR ERROR SWITCH
1585 022510  022425                            CMP     (R4)+,(R5)+     ;BUMP PTRS PAST ERROR WORDS
1586 022512  000712                            BR      7$              ;DO NEXT COMPARE
1587 022514  005237  002716          13$:      INC     MORECE          ;BUMP ERROR COUNTER
1588 022520  000707                            BR      7$              ;DO NEXT COMPARE
1589
1590                                    ;       WRITE AND READ DATA ROUTINE.
1591 022522  012737  177777  003022  XWRITT:   MOV     #-1,TEMP1       ;SET SPECIAL WRITE FOR TIMING FLAG
1592 022530  000402                            BR      XWRIT1
1593 022532  005037  003022          XWRITE:   CLR     TEMP1           ;CLEAR SPECIAL WRITE FLAG
1594 022536  012737  000112  003036  XWRIT1:   MOV     #WTDATA,TEMP7   ;SET FOR WRITE
1595 022544  023737  002204  003006            CMP     HLMTW,CURCYL    ;TEST IF CYLINDER 255 (BAD SEC)
1596 022552  001006                            BNE     1$              ;NO - SKIP
1597 022554  005737  003014                    TST     DESHD           ;TEST IF HEAD 1 (BAD SECTOR FILES)
1598 022560  001403                            BEQ     1$              ;NO - SKIP
1599 022562  052737  004000  002706            BIS     #BADADD,OPFLAG  ;SET BAD ADDRESS FLAG
1600 022570  000403                    1$:     BR      XREADG          ;SKIP TO EXECUTE
1601 022572  012737  000114  003036  XREAD:    MOV     #RDDATA,TEMP7   ;SET FOR READ
1602 022600  010346                    XREADG:   MOV     R3,-(SP)        ;STORE R3
1603 022602  013703  002704                    MOV     SSINDX,R3       ;SET SUBROUTINE INDEX
1604 022606  005723                            TST     (R3)+           ;BUMP TO NEXT STACK ENTRY
1605 022610  016663  000002  002306            MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
1606 022616  162763  000004  002306            SUB     #4,SUBSTK(R3)   ;ADJUST TO POINT TO CALL
1607 022624  010337  002704                    MOV     R3,SSINDX       ;STORE IT BACK
1608 022630  010046                            MOV     R0,-(SP)
1609 022632  010146                            MOV     R1,-(SP)        ;STORE OTHER REGISTERS
1610 022634  010446                            MOV     R4,-(SP)
1611 022636  004737  017306                    JSR     PC,RDYCHK       ;CHECK IF DRIVE READY
1612 022642  023212                            65$
1613 022644  012703  002736                    MOV     #L.CS,R3        ;GET ADDRESS OF LOAD REGS
1614 022650  013713  003036                    MOV     TEMP7,(R3)      ;SET COMMAND
1615 022654  053713  002734                    BIS     RLDRV,(R3)      ;INSERT DRIVE NUMBER
1616 022660  042713  002000                    BIC     #BIT10,(R3)     ;CLEAR FOR DRIVE 4 - 7 SPEC'D
1617 022664  032723  000004                    BIT     #BIT2,(R3)+     ;TEST IF WRITE DATA
1618 022670  001403                            BEQ     3$              ;YES - SKIP
1619 022672  012723  003746                    MOV     #IBUFF,(R3)+    ;ELSE SET BA FOR READ
1620 022676  000402                            BR      4$
1621 022700  012723  004346          3$:       MOV     #OBUFF,(R3)+    ;SET BA FOR WRITE
1622 022704  013713  003006          4$:       MOV     CURCYL,(R3)     ;GET CURRENT CYLINDER
1623 022710  012704  000007                    MOV     #7,R4           ;ALIGN IT IN DA
1624 022714  006313                    5$:     ASL     (R3)
1625 022716  005304                            DEC     R4
1626 022720  001375                            BNE     5$
1627 022722  005737  003014                    TST     DESHD           ;TEST IF HEAD 0
```

```
1628 022726  001402                        BEQ     7$              ;YES - SKIP
1629 022730  052713  000100                BIS     #HSMSK,(R3)     ;SET FOR HEAD 1
1630 022734  053723  003016          7$:   BIS     DESSEC,(R3)+    ;INSERT DESIRED SECTOR
1631 022740  012713  177600                MOV     #177600,(R3)    ;INSERT WORD COUNT
1632 022744  005737  003022                TST     TEMP1           ;CHECK IF SPECIAL WRITE FOR TIMING
1633 022750  001402                        BEQ     8$              ;NO - SKIP
1634 022752  012713  177777                MOV     #177777,(R3)    ;ELSE SET FOR 1 WORD TRANSFER
1635 022756  032737  004000  002706  8$:   BIT     #BADADD,OPFLAG  ;TEST IF BAD ADDRESS FLAG SET
1636 022764  001413                        BEQ     2$              ;NO - SKIP
1637 022766  042737  173777  002706        BIC     #^CBADADD,OPFLAG  ;CLEAR ALL BUT THIS FLAG
1638 022774  012703  010272                MOV     #MWRTAB,R3      ;SET RESULT MESSAGE POINTER
1639 023000                                ERRHRD  10032.,,ERR1
     023000  104443                        TRAP    T$ERCODE
     023002  023460                        .WORD   10032
     023004  011526                        .WORD   ERR1
1640 023006  005037  002706                CLR     OPFLAG          ;CLEAR ALL FLAGS
1641 023012  000475                        BR      64$
1642 023014  005037  002710          2$:   CLR     DONE            ;CLEAR INTERRUPT FLAG
1643 023020  005737  003022                TST     TEMP1           ;CHECK IF SPECIAL WRITE FLAG SET
1644 023024  001072                        BNE     65$             ;YES - DO NOT START WRITE
1645 023026  011362  000006                MOV     (R3),RLMP(R2)   ;LOAD RL REGS
1646 023032  014362  000004                MOV     -(R3),RLDA(R2)
1647 023036  014362  000002                MOV     -(R3),RLBA(R2)
1648 023042  014362  000000                MOV     -(R3),RLCS(R2)
1649 023046                          10$:  WAITUS  #3000.          ;WAIT 300MS FOR INTERRUPT
     023046  012700  005670                MOV     #3000.,R0
     023052  104027                        EMT     C$WTU
1650 023054  005737  002710                TST     DONE            ;CHECK IF INTERRUPT
1651 023060  001007                        BNE     14$             ;YES - SKIP
1652 023062  004737  015142                JSR     PC,WAITIN       ;WAIT FOR INTERRUPT
1653 023066  012603                        MOV     (SP)+,R3        ;GET RESULT MESSAGE
1654 023070                                ERRHRD  10030.,,ERR1
     023070  104443                        TRAP    T$ERCODE
     023072  023456                        .WORD   10030
     023074  011526                        .WORD   ERR1
1655 023076  000443                        BR      64$
1656 023100  032737  000001  002746  14$:  BIT     #DRDYMSK,T.CS   ;TEST IF DRIVE READY
1657 023106  001031                        BNE     20$             ;YES - SKIP
1658 023110  012703  007561                MOV     #MDRDY,R3       ;SET RESULT MESSAGE
1659 023114  012704  010464                MOV     #CAFDT,R4       ;CONDITION AFTER DATA XFER
1660 023120                                ERRHRD  10032.,,ERR5
     023120  104443                        TRAP    T$ERCODE
     023122  023460                        .WORD   10032
     023124  011760                        .WORD   ERR5
1661 023126  012701  000062                MOV     #50.,R1         ;SET WAIT COUNT FOR 5 SECDS
1662 023132  004737  015342          17$:  JSR     PC,GSTAT        ;GET DRIVE STATUS
1663 023136  023206                        64$
1664 023140  032737  000001  002746        BIT     #DRDYMSK,T.CS   ;TEST IF DRIVE READY NOW
1665 023146  001011                        BNE     20$             ;YES - SKIP
1666 023150  005301                        DEC     R1              ;DEC WAIT COUNT
1667 023152  001367                        BNE     17$             ;LOOP IF NOT TIME DONE
1668 023154  012704  010475                MOV     #C5SEC,R4       ;SET CONDITION 5 SECONDS
1669 023160                                ERRHRD  10033.,,ERR5
     023160  104443                        TRAP    T$ERCODE
     023162  023461                        .WORD   10033
     023164  011760                        .WORD   ERR5
1670 023166  005037  002720                CLR     ERRSWI          ;CLEAR ERROR SWITCH
```

```
1671 023172  005737  002746          20$:   TST     T.CS            ;CHECK IF ANY ERROR
1672 023176  100005                         BPL     65$             ;NO - SKIP
1673 023200                                  ERRHRD  10031.,,ERR6
     023200  104443                         TRAP    T$ERCODE
     023202  023457                         .WORD   10031
     023204  012030                         .WORD   ERR6
1674 023206  005037  002720          64$:   CLR     ERRSWI          ;CLEAR ERROR SWITCH
1675 023212  162737  000002  002704  65$:   SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
1676 023220  012604                         MOV     (SP)+,R4        ;RESTORE REGISTERS
1677 023222  012601                         MOV     (SP)+,R1
1678 023224  012600                         MOV     (SP)+,R0
1679 023226  012603                         MOV     (SP)+,R3
1680 023230  005737  002720                 TST     ERRSWI          ;TEST IF ERROR RETURN
1681 023234  001403                         BEQ     99$             ;YES - SKIP
1682 023236  063716  002720                 ADD     ERRSWI,(SP)     ;ELSE ADD IN ERROR RETURN
1683 023242  000207                         RTS     PC
1684 023244  017616  000000          99$:   MOV     @(SP),(SP)      ;ADJUST FOR ERROR RETURN
1685 023250  000207                         RTS     PC
1686
1687                                 ;       BAD SECTOR CHECK ROUTINE. CHECKS IF SECTOR SPECIFIED IN CURCYL,
1688                                 ;       DESHD, AND DESSEC IS LISTED AS BAD IN THE BAD SECTOR FILES.
1689 023252  010046                 BSCHK:  MOV     R0,-(SP)        ;STORE REGISTERS
1690 023254  010146                         MOV     R1,-(SP)
1691 023256  010346                         MOV     R3,-(SP)
1692 023260  005037  002722                 CLR     BSFLAG          ;CLEAR FLAG
1693 023264  012703  003552                 MOV     #FBSFIL,R3      ;GET POINTER TO FACTORY FILE
1694 023270  022713  177777                 CMP     #-1,(R3)        ;CHECK IF ALL ONES
1695 023274  001005                         BNE     4$              ;NO SKIP TO TEST
1696 023276  012703  003356          2$:    MOV     #SBSFIL,R3      ;ELSE SET POINTER TO SOFTWARE FILE
1697 023302  022713  177777                 CMP     #-1,(R3)        ;CHECK IF ALL ONES
1698 023306  001431                         BEQ     20$             ;YES - EXIT
1699 023310  013700  003004          4$:    MOV     NEWCYL,R0       ;BUILD HEADER OF ADDRESS IN QUESTION
1700 023314  012701  000007                 MOV     #7,R1           ;POSITION CYLINDER
1701 023320  006300                 5$:    ASL     R0
1702 023322  005301                         DEC     R1
1703 023324  001375                         BNE     5$
1704 023326  005737  003014                 TST     DESHD           ;CHECK IF HEAD 0
1705 023332  001402                         BEQ     7$              ;YES - SKIP
1706 023334  052700  000100                 BIS     #BIT6,R0        ;INSERT HEAD 1
1707 023340  053700  003016          7$:    BIS     DESSEC,R0       ;INSERT SECTOR
1708 023344  022300                 8$:    CMP     (R3)+,R0        ;CHECK THIS WORD IN FILE
1709 023346  001402                         BEQ     12$             ;YES - EXIT,ERROR
1710 023350  101005                         BHI     15$             ;EXIT- NO ERROR
1711 023352  000774                         BR      8$
1712 023354  012737  000001  002722  12$:   MOV     #1,BSFLAG       ;SET ERROR FLAG
1713 023362  000403                         BR      20$             ;GO TO EXIT
1714 023364  020327  003552          15$:   CMP     R3,#FBSFIL      ;DONE BOTH FILES?
1715 023370  003342                         BGT     2$              ;NO GO DO SOFTWARE FILE
1716 023372  012603                 20$:   MOV     (SP)+,R3        ;ELSE RESTORE REGISTERS
1717 023374  012601                         MOV     (SP)+,R1
1718 023376  012600                         MOV     (SP)+,R0
1719 023400  005737  002722                 TST     BSFLAG          ;CHECK IF ERROR
1720 023404  001003                         BNE     99$             ;YES - SKIP
1721 023406  062716  000002                 ADD     #2,(SP)         ;ELSE BUMP ERROR RETURN
1722 023412  000207                         RTS     PC
1723 023414  017616  000000          99$:   MOV     @(SP),(SP)      ;SET FOR ERROR RETURN
1724 023420  000207                         RTS     PC
```

```
 1725
 1727                                   ;       REPORT OPERATION ROUTINE. PRINTS SUBROUTINE TRACE SEQUENCE AND
 1728                                   ;       OPERATION BEING PERFORMED PORTION OF ALL
 1729                                   ;       ERROR MESSAGES.
 1730 023422  010446                   RPTOP:  MOV     R4,-(SP)
 1731 023424  005737  002704                   TST     SSINDX          ;TEST SUBROUTINE INDEX 0
 1732 023430  001433                           BEQ     1$              ;SKIP IF 0
 1733 023432  012704  000002                   MOV     #2,R4           ;SET INDEXER TO FIRST ENTRY
 1734 023436                                   PRINTB  #FMT9,#SEQMES   ;PRINT "SUBROUTINE CALL SEQ"
      023436  012746  007430                   MOV     #SEQMES,-(SP)
      023442  012746  011014                   MOV     #FMT9,-(SP)
      023446  012746  000002                   MOV     #2,-(SP)
      023452  010600                           MOV     SP,R0
      023454  104014                           EMT     C$PNTB
      023456  062706  000006                   ADD     #6,SP
 1735 023462                           3$:     PRINTB  #FMT16,SUBSTK(R4)       ;PRINT CALLING LOCATION
      023462  016446  002306                   MOV     SUBSTK(R4),-(SP)
      023466  012746  011167                   MOV     #FMT16,-(SP)
      023472  012746  000002                   MOV     #2,-(SP)
      023476  010600                           MOV     SP,R0
      023500  104014                           EMT     C$PNTB
      023502  062706  000006                   ADD     #6,SP
 1736 023506  062704  000002                   ADD     #2,R4           ;BUMP INDEX
 1737 023512  020437  002704                   CMP     R4,SSINDX       ;CHECK IF ALL PRINTED
 1738 023516  003761                           BLE     3$              ;LOOP IF NOT ALL PRINTED YET
 1739 023520                           1$:     PRINTB  #FMT4,ERHEAD,#TSTLAB    ;PRINT ERROR HEADER
      023520  012746  006363                   MOV     #TSTLAB,-(SP)
      023524  013746  002714                   MOV     ERHEAD,-(SP)
      023530  012746  010617                   MOV     #FMT4,-(SP)
      023534  012746  000003                   MOV     #3,-(SP)
      023540  010600                           MOV     SP,R0
      023542  104014                           EMT     C$PNTB
      023544  062706  000010                   ADD     #10,SP
 1740 023550  042737  030000  002706           BIC     #SEEKOP!RORWOP,OPFLAG   ;CLEAR SK & RD OR WRT FLAG
 1741 023556  013701  002736                   MOV     L.CS,R1         ;GET COMMAND EXECUTED
 1742 023562  042701  177741                   BIC     #177741,R1      ;STRIP ALL BUT FUNCTION CODE
 1743 023566  022701  000006                   CMP     #6,R1   ;TEST IF SEEK OPERATION
 1744 023572  001003                           BNE     2$              ;NO - SKIP
 1745 023574  052737  010000  002706           BIS     #SEEKOP,OPFLAG  ;ELSE SET SEEK FLAG
 1746 023602  022701  000012           2$:     CMP     #12,R1          ;TEST IF WRITE
 1747 023606  001003                           BNE     20$             ;NO - SKIP
 1748 023610  052737  020000  002706           BIS     #RORWOP,OPFLAG  ;SET RD OR WRT FLAG
 1749 023616  022701  000014           20$:    CMP     #14,R1          ;TEST IF READ
 1750 023622  001003                           BNE     22$             ;NO - SKIP
 1751 023624  052737  020000  002706           BIS     #RORWOP,OPFLAG  ;SET RD OR WRT FLAG
 1752 023632                           22$:    PRINTB  #FMT1,#MOPER,OPMSGS(R1) ;PRINT OPERATION
      023632  016146  002126                   MOV     OPMSGS(R1),-(SP)
      023636  012746  005376                   MOV     #MOPER,-(SP)
      023642  012746  010575                   MOV     #FMT1,-(SP)
      023646  012746  000003                   MOV     #3,-(SP)
      023652  010600                           MOV     SP,R0
      023654  104014                           EMT     C$PNTB
      023656  062706  000010                   ADD     #10,SP
 1753 023662  020127  000004                   CMP     R1,#4           ;CHECK IF GET STATUS
 1754 023666  001007                           BNE     4$              ;NO - SKIP
 1755 023670  032737  000010  002742           BIT     #DRSET,L.DA     ;TEST IF RESET INCLUDED
 1756 023676  001403                           BEQ     4$              ;NO - SKIP
```

```
1757 023700  012701  000016                      MOV     #16,R1              ;SET TO PRINT WITH RESET
1758 023704  000436                              BR      9$
1759 023706  032737  007777  002706  4$:         BIT     #COMPOP,OPFLAG      ;TEST IF ANY OTHER OPERATION
1760 023714  001424                              BEQ     8$                  ;NO - SKIP
1761 023716  013704  002706                      MOV     OPFLAG,R4           ;SET UP TO DETERMINE WHICH ONE
1762 023722  012701  000020                      MOV     #20,R1              ;PRESET THE POINTER
1763 023726  032704  000001          5$:         BIT     #BIT00,R4           ;CHECK THE BIT
1764 023732  001003                              BNE     6$                  ;IF SET - SKIP
1765 023734  005721                              TST     (R1)+               ;BUMP POINTER
1766 023736  006204                              ASR     R4
1767 023740  000772                              BR      5$
1768 023742                          6$:         PRINTB  #FMT2,OPMSGS(R1)
     023742  016146  002126                      MOV     OPMSGS(R1),-(SP)
     023746  012746  010611                      MOV     #FMT2,-(SP)
     023752  012746  000002                      MOV     #2,-(SP)
     023756  010600                              MOV     SP,R0
     023760  104014                              EMT     C$PNTB
     023762  062706  000006                      ADD     #6,SP
1769 023766  032737  100000  002706  8$:         BIT     #HDR40,OPFLAG       ;TEST IF 40 HEADER OPERATION
1770 023774  001415                              BEQ     10$                 ;NO - SKIP
1771 023776  012701  000050                      MOV     #50,R1              ;ELSE PRINT IT
1772 024002                          9$:         PRINTB  #FMT2,OPMSGS(R1)
     024002  016146  002126                      MOV     OPMSGS(R1),-(SP)
     024006  012746  010611                      MOV     #FMT2,-(SP)
     024012  012746  000002                      MOV     #2,-(SP)
     024016  010600                              MOV     SP,R0
     024020  104014                              EMT     C$PNTB
     024022  062706  000006                      ADD     #6,SP
1773 024026  000434                              BR      15$                 ;SKIP
1774 024030  032737  010000  002706  10$:        BIT     #SEEKOP,OPFLAG      ;TEST IF SEEK
1775 024036  001430                              BEQ     15$                 ;NO - SKIP
1776 024040                                      PRINTB  #FMT13,#FRMWD,OLDCYL,#DIFWD,DESDIF,#SGNWD,DESSGN,#HDWD,DESHD
     024040  013746  003014                      MOV     DESHD,-(SP)
     024044  012746  007371                      MOV     #HDWD,-(SP)
     024050  013746  003012                      MOV     DESSGN,-(SP)
     024054  012746  007364                      MOV     #SGNWD,-(SP)
     024060  013746  003010                      MOV     DESDIF,-(SP)
     024064  012746  007356                      MOV     #DIFWD,-(SP)
     024070  013746  003002                      MOV     OLDCYL,-(SP)
     024074  012746  007407                      MOV     #FRMWD,-(SP)
     024100  012746  011035                      MOV     #FMT13,-(SP)
     024104  012746  000011                      MOV     #11,-(SP)
     024110  010600                              MOV     SP,R0
     024112  104014                              EMT     C$PNTB
     024114  062706  000024                      ADD     #24,SP
1777 024120  032737  020000  002706  15$:        BIT     #RORWOP,OPFLAG      ;TEST IF READ OR WRITE SET
1778 024126  001424                              BEQ     17$                 ;NO - SKIP
1779 024130                                      PRINTB  #FMT22,#CYLWD,CURCYL,#HDWD,DESHD,#SECWD,DESSEC
     024130  013746  003016                      MOV     DESSEC,-(SP)
     024134  012746  007375                      MOV     #SECWD,-(SP)
     024140  013746  003014                      MOV     DESHD,-(SP)
     024144  012746  007371                      MOV     #HDWD,-(SP)
     024150  013746  003006                      MOV     CURCYL,-(SP)
     024154  012746  007402                      MOV     #CYLWD,-(SP)
     024160  012746  011364                      MOV     #FMT22,-(SP)
     024164  012746  000007                      MOV     #7,-(SP)
     024170  010600                              MOV     SP,R0
```

```
                024172  104014                          EMT     C$PNTB
                024174  062706  000020                  ADD     #20,SP
1780            024200  004737  024652          17$:    JSR     PC,CLRPARM      ;CLEAR PARAM TABLE
1781            024204  012604                          MOV     (SP)+,R4        ;RESTORE R4
1782            024206  000207                          RTS     PC
1783
1784                                            ;       REPORT REASON ROUTINE
1785                                            ;       PRINTS REASON PORTION FOR ALL ERROR REPORTS.
1786            024210  010146                  RPTRES: MOV     R1,-(SP)        ;STORE R1
1787            024212  010346                          MOV     R3,-(SP)        ;STORE R3
1788            024214  010446                          MOV     R4,-(SP)        ;STORE R4
1789            024216  012701  002764                  MOV     #RESPARM,R1     ;GET START OF PARAM
1790            024222  012103                          MOV     (R1)+,R3        ;GET NUMBER OF PARAM
1791            024224                                  PRINTB  #FMT1.1,#MRSLT,(R1)     ;PRINT NAME
                024224  011146                          MOV     (R1),-(SP)
                024226  012746  005405                  MOV     #MRSLT,-(SP)
                024232  012746  010602                  MOV     #FMT1.1,-(SP)
                024236  012746  000003                  MOV     #3,-(SP)
                024242  010600                          MOV     SP,R0
                024244  104014                          EMT     C$PNTB
                024246  062706  000010                  ADD     #10,SP
1792            024252  021127  010141                  CMP     (R1),#MNDRST    ;TEST IF MESSAGE IS NO DRV STATUS
1793            024256  001453                          BEQ     6$              ;YES - SKIP REST OF REPORT
1794            024260  012704  011021                  MOV     #FMT11,R4       ;PRISET FOR FORMAT 11
1795            024264  022127  010134                  CMP     (R1)+,#MCYLOC   ;CHECK IF REPORTING CYLINDER LOC
1796            024270  001002                          BNE     3$              ;NO - SKIP
1797            024272  012704  011027                  MOV     #FMT12,R4       ;ELSE CHANGE TO FORMAT 12
1798            024276  005303                  3$:     DEC     R3              ;DEC PARAM COUNT
1799            024300  001442                          BEQ     6$              ;IF 0 - EXIT
1800            024302                                  PRINTB  R4,#RESE3,(R1)+  ;REPORT IS VALUE
                024302  012146                          MOV     (R1)+,-(SP)
                024304  012746  010361                  MOV     #RESE3,-(SP)
                024310  010446                          MOV     R4,-(SP)
                024312  012746  000003                  MOV     #3,-(SP)
                024316  010600                          MOV     SP,R0
                024320  104014                          EMT     C$PNTB
                024322  062706  000010                  ADD     #10,SP
1801            024326                                  PRINTB  R4,#RESE4,(R1)+  ;REPORT SB VALUE
                024326  012146                          MOV     (R1)+,-(SP)
                024330  012746  010365                  MOV     #RESE4,-(SP)
                024334  010446                          MOV     R4,-(SP)
                024336  012746  000003                  MOV     #3,-(SP)
                024342  010600                          MOV     SP,R0
                024344  104014                          EMT     C$PNTB
                024346  062706  000010                  ADD     #10,SP
1802            024352  162703  000002                  SUB     #2,R3           ;DEC PARAM COUNT
1803            024356  001413                          BEQ     6$              ;IF 0 - EXIT
1804            024360                                  PRINTB  #FMT1,#RESE5,(R1)+  ;REPORT CONDITION
                024360  012146                          MOV     (R1)+,-(SP)
                024362  012746  010372                  MOV     #RESE5,-(SP)
                024366  012746  010575                  MOV     #FMT1,-(SP)
                024372  012746  000003                  MOV     #3,-(SP)
                024376  010600                          MOV     SP,R0
                024400  104014                          EMT     C$PNTB
                024402  062706  000010                  ADD     #10,SP
1805            024406  012604                  6$:     MOV     (SP)+,R4        ;RESTORE REGS
1806            024410  012603                          MOV     (SP)+,R3
```

```
                    .
 1807 024412 012601                         MOV     (SP)+,R1
 1808 024414 000207                         RTS     PC                  ;RETURN
 1809
 1810                               ;       REPORT PHYSICAL ADDRESS OF DEVICE UNDER TEST
 1811                               ;       AND ALL REGISTER CONTENTS.
 1812 024416                        RPTREM: PRINTB  #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
      024416 005046                         CLR     -(SP)
      024420 153716 002735                  BISB    RLDRV+1,(SP)
      024424 012746 006033                  MOV     #DRVNAM,-(SP)
      024430 013746 002730                  MOV     RLBAS,-(SP)
      024434 012746 006022                  MOV     #BASADD,-(SP)
      024440 012746 010630                  MOV     #FMT5,-(SP)
      024444 012746 000005                  MOV     #5,-(SP)
      024450 010600                         MOV     SP,R0
      024452 104014                         EMT     C$PNTB
      024454 062706 000014                  ADD     #14,SP
 1813                               ;       REPORT RL11 REGISTERS
 1814 024460                                PRINTB  #FMT6,#CSNAM,#DANAM,#BANAM,#MPNAM,#CYLWD,#HDWD
      024460 012746 007371                  MOV     #HDWD,-(SP)
      024464 012746 007402                  MOV     #CYLWD,-(SP)
      024470 012746 006136                  MOV     #MPNAM,-(SP)
      024474 012746 006124                  MOV     #BANAM,-(SP)
      024500 012746 006131                  MOV     #DANAM,-(SP)
      024504 012746 006117                  MOV     #CSNAM,-(SP)
      024510 012746 010650                  MOV     #FMT6,-(SP)
      024514 012746 000007                  MOV     #7,-(SP)
      024520 010600                         MOV     SP,R0
      024522 104014                         EMT     C$PNTB
      024524 062706 000020                  ADD     #20,SP
 1815 024530                                PRINTB  #FMT8,#LAB1,L.CS,L.DA,L.BA,L.MP
      024530 013746 002744                  MOV     L.MP,-(SP)
      024534 013746 002740                  MOV     L.BA,-(SP)
      024540 013746 002742                  MOV     L.DA,-(SP)
      024544 013746 002736                  MOV     L.CS,-(SP)
      024550 012746 006143                  MOV     #LAB1,-(SP)
      024554 012746 010762                  MOV     #FMT8,-(SP)
      024560 012746 000006                  MOV     #6,-(SP)
      024564 010600                         MOV     SP,R0
      024566 104014                         EMT     C$PNTB
      024570 062706 000016                  ADD     #16,SP
 1816 024574                                PRINTB  #FMT7,#LAB2,T.CS,T.DA,T.BA,T.MP,CURCYL,DESHD
      024574 013746 003014                  MOV     DESHD,-(SP)
      024600 013746 003006                  MOV     CURCYL,-(SP)
      024604 013746 002754                  MOV     T.MP,-(SP)
      024610 013746 002750                  MOV     T.BA,-(SP)
      024614 013746 002752                  MOV     T.DA,-(SP)
      024620 013746 002746                  MOV     T.CS,-(SP)
      024624 012746 006156                  MOV     #LAB2,-(SP)
      024630 012746 010712                  MOV     #FMT7,-(SP)
      024634 012746 000010                  MOV     #10,-(SP)
      024640 010600                         MOV     SP,R0
      024642 104014                         EMT     C$PNTB
      024644 062706 000022                  ADD     #22,SP
 1817 024650 000207                         RTS     PC
 1818
 1819                               ;       CLEAR PARAMETER BLOCK FOR REPORTING
 1820 024652 010546                 CLRPARM:        MOV     R5,-(SP)        ;STORE R5
```

```
1821 024654  012701  002764                MOV     #RESPARM,R1     ;GET ADDRESS OF BLOCK
1822 024660  012705  000005                MOV     #5,R5           ;SET COUNT
1823 024664  005021                2$:     CLR     (R1)+           ;CLEAR WORD
1824 024666  005305                        DEC     R5              ;DEC COUNT
1825 024670  001375                        BNE     2$              ;LOOP UNTIL 0
1826 024672  012701  002764                MOV     #RESPARM,R1     ;RESET POINTER
1827 024676  012605                        MOV     (SP)+,R5        ;RESTORE R5
1828 024700  000207                        RTS     PC
1829
1830 024702                        ENDMOD
```

```
1                       .TITLE  CZRLJA0 RL01/2 DR TST 2
```

```
  1 024702                              BGNMOD  HRDWTST
  2                                     .SBTTL  *TEST 1         **OUTER GUARD BAND DETECTION
  3 024702                              BGNTST                  ;TEST 1
    024702                                                                                          T1::
  4 024702  012737  006371  002714              MOV     #P2T03E,ERHEAD  ;SET ERROR HEADER
  5 024710  004737  015274                      JSR     PC,TSTINT       ;INITIALIZE TEST
  6 024714  004737  015312                      JSR     PC,GSTATR       ;CLEAR DRIVE
  7 024720  025112                              T1965$
  8 024722  004737  017546                      JSR     PC,CHOSHD       ;GO CHOSE HEAD
  9 024726  005005                      T197$:  CLR     R5              ;CLEAR FOR POSITION TO 0
 10 024730  004737  016746                      JSR     PC,POSHDS       ;POSITION HEADS
 11 024734  025112                              T1965$
 12 024736                              BGNSUB
    024736                                                                                          T1.1:
    024736  104002                              EMT     C$BSUB
 13 024740  012737  177777  003004              MOV     #-1,NEWCYL      ;SET FOR GUARD BAND SEEK
 14 024746  004737  016204                      JSR     PC,XSEEK        ;DO SEEK
 15 024752  025066                              60$
 16 024754  012701  000003                      MOV     #3.,R1          ;SET WAIT COUNT FOR 3MS
 17 024760  032762  000001  000000  8$:         BIT     #DRDYMSK,RLCS(R2) ;TEST IF DRIVE READY
 18 024766  001413                              BEQ     9$              ;NO-SKIP
 19 024770  004737  015342                      JSR     PC,GSTAT        ;GET DRIVE STATUS
 20 024774  025066                              60$
 21 024776  012703  007561                      MOV     #MDRDY,R3       ;SET NAME MESSAGE PTR
 22 025002  012704  010434                      MOV     #C10MS,R4       ;SET CONDITION MESSAGE PTR
 23 025006                                      ERRHRD  301.,,ERR4      ;REPORT READY ERROR
    025006  104443                              TRAP    T$ERCODE
    025010  000455                              .WORD   301
    025012  011710                              .WORD   ERR4
 24 025014  000424                              BR      60$             ;EXIT TEST
 25 025016  005301                      9$:     DEC     R1              ;DEC WAIT COUNT
 26 025020  001404                              BEQ     12$             ;SKIP IF 0
 27 025022                                      WAITUS  #10.
    025022  012700  000012                      MOV     #10.,R0
    025026  104027                              EMT     C$WTU
 28 025030  000753                              BR      8$              ;LOOP
 29 025032  012701  000226          12$:        MOV     #150.,R1        ;SET WAIT COUNT FOR 15 MS
 30 025036  004737  021026                      JSR     PC,RDYWAIT      ;WAIT FOR READY & REPORT IF NOT READY
 31 025042  025066                              60$
 32
 33 025044  004737  021274                      JSR     PC,GETPOS       ;GET POSITION
 34 025050  025066                              60$
 35 025052  005737  003006                      TST     CURCYL          ;CHECK IF HEADS STILL AT 0
 36 025056  001403                              BEQ     15$             ;YES-SKIP
 37 025060                                      ERRHRD  302.,,ERR8      ;ELSE REPORT CYLINDER ERROR
    025060  104443                              TRAP    T$ERCODE
    025062  000456                              .WORD   302
    025064  012756                              .WORD   ERR8
 38 025066                              15$:
 39 025066  012737  000002  002720  60$:        MOV     #2,ERRSWI       ;INIT ERROR SWITCH
 40 025074                              ENDSUB
    025074                              L10021:
    025074  104003                              EMT     C$ESUB
 41 025076                                      ESCAPE  TST             ;EXIT TEST IF ERROR
    025076  104010                              EMT     C$ESCAPE
    025100  000012                              .WORD   L10020-.
 42 025102  004737  017572                      JSR     PC,SWAPHD       ;GO SWAP TO HEAD 1 OR END TEST
```

```
     43 025106  025112                          17$                ;ABORT RETURN
     44 025110  000706                          BR       T197$     ;REDO TEST
     45 025112                          17$:
     46 025112                          T1965$:
     47 025112                          ENDTST
        025112                          L10020:
        025112  104001                          EMT      C$ETST
```

```
                                        .SBTTL  *TEST 2          **INCREMENTAL FORWARD SEEK HEAD 0
025114                                  BGNTST                   ;TEST 2
025114                                                                                    T2::
025114  012737  006407  002714                  MOV     #P2T04E,ERHEAD  ;SET ERROR HEADER
025122  004737  015274                          JSR     PC,TSTINT       ;INITIALIZE TEST
025126  004737  015312                          JSR     PC,GSTATR       ;CLEAR DRIVE
025132  025322                                  T2065$
025134  004737  017546                          JSR     PC,CHOSHD       ;GO CHOSE HEAD
025140  005737  003014                          TST     DESHD           ;TEST IF THIS IS HEAD 0
025144  001402                                  BEQ     2$              ;YES - SKIP
025146                                          EXIT    TST             ;ELSE EXIT TEST
025146  104032                                  EMT     C$EXIT
025150  000152                                  .WORD   L10022-.
025152  013705  013350          2$:             MOV     LOLIMW,R5       ;CLEAR TO POSITION HEADS TO LOLIMIT
025156  004737  016746                          JSR     PC,POSHDS       ;POSITION HEADS
025162  025322                                  T2065$
025164                                  BGNSUB
025164                                                                                    T2.1:
025164  104002                                  EMT     C$BSUB
025166  004737  021274          T206$:  JSR     PC,GETPOS       ;GET POSITION
025172  025312                                  60$
025174                                          INLOOP                  ;CHECK IF IN ERROR LOOP
025174  104020                                  EMT     C$INLP
025176                                          BNCOMPLETE      5$      ;NO - SKIP
025176  103007                                  BCC     5$
025200  023737  003006  003004                  CMP     CURCYL,NEWCYL   ;CHECK IF POSITIONED AT DESIRED LOC
025206  001003                                  BNE     5$              ;NO - SKIP
025210  004737  017632                          JSR     PC,ONSWAP       ;ELSE SWAP NEW AND OLD CYLINDERS
025214  000405                                  BR      7$              ;SKIP
025216  013737  003006  003004  5$:             MOV     CURCYL,NEWCYL   ;PLACE CURRENT INTO NEW
025224  005237  003004                          INC     NEWCYL          ;BUMP FOR ONE CYLINDER SEEK
025230                          7$:
025230  004737  016204                          JSR     PC,XSEEK        ;DO SEEK
025234  025312                                  60$
025236  012701  000226                          MOV     #150.,R1        ;SET WAIT TIME 15 MS
025242  004737  021026                          JSR     PC,RDYWAIT      ;WAIT FOR READY
025246  025312                                  60$

025250  004737  021422                          JSR     PC,VERPOS       ;GO VERIFY POSITON
025254  025312                                  60$

025256  032737  000002  013346                  BIT     #ALLSEC,MISWIW  ;TEST IF CHECK ALL SECTORS
025264  001406                                  BEQ     11$             ;NO-SKIP
025266  004737  021542                          JSR     PC,RDALHD       ;GO READ ALL HEADERS
025272  025312                                  60$
025274  004737  020440                          JSR     PC,VERHDR       ;GO VERIFY HEADER
025300  025312                                  60$
025302                          11$:
025302  023737  013352  003004                  CMP     HILIMW,NEWCYL   ;CHECK IF HILIMIT REACHED
025310  103726                                  BLO     T206$           ;NO-LOOP
025312  012737  000002  002720  60$:             MOV     #2,ERRSWI       ;INIT ERROR SWITCH
025320                                  ENDSUB
025320                          L10023:
025320  104003                                  EMT     C$ESUB
025322                          T2065$:
025322                                  ENDTST
025322                          L10022:
```

```
CZRLJA0 RL01/2 DR TST 2 MACRO V03.01 9-FEB-79 19:23:55 PAGE 6-1
•TEST 2          ••INCREMENTAL FORWARD SEEK HEAD 0                    SEQ 0100

        025322  104001                          EMT     C$ETST
```

```
                                        .SBTTL  *TEST 3         **INCREMENTAL REVERSE SEEK HEAD 0
                                        BGNTST                  ;TEST 3
                                                                                        T3::
                        012737  006431  002714                 MOV     #P2TO5E,ERHEAD  ;SET ERROR HEADER
                        004737  015274                         JSR     PC,TSTINT       ;INITIALIZE TEST
                        004737  015312                         JSR     PC,GSTATR       ;CLEAR DRIVE
                                                                T2165$
                        004737  017546                         JSR     PC,CHOSHD       ;GO CHOSE HEAD
                        005737  003014                         TST     DESHD           ;TEST IF HEAD 0 SELECTED
                        001402                                 BEQ     2$              ;YES - SKIP
                                                               EXIT    TST             ;ELSE EXIT TEST
                        104032                                 EMT     C$EXIT
                        000152                                 .WORD   L10024-.
                        013705  013352          2$:           MOV     HILIMW,R5       ;SET TO POSITION HDS TO HILIMIT
                        004737  016746                         JSR     PC,POSHDS       ;POSITION HEADS
                                                               T2165$
                                        BGNSUB
                                                                                        T3.1:
                        104002                                 EMT     C$BSUB
                        004737  021274          T216$:        JSR     PC,GETPOS       ;GET POSITION
                                                               60$
                                                               INLOOP                  ;CHECK IF IN ERROR LOOP
                        104020                                 EMT     C$INLP
                                                               BNCOMPLETE      5$      ;NO - SKIP
                        103007                                 BCC     5$
                        023737  003006  003004                 CMP     CURCYL,NEWCYL   ;CHECK IF POSITIONED AT DES LOC
                        001003                                 BNE     5$              ;NO - SKIP
                        004737  017632                         JSR     PC,ONSWAP       ;ELSE SWAP OLD AND NEW CYLINDERS
                        000405                                 BR      7$              ;SKIP
                        013737  003006  003004  5$:           MOV     CURCYL,NEWCYL   ;PUT CURRENT INTO NEW
                        005337  003004                         DEC     NEWCYL          ;DEC FOR ONE CYLINDER REVERSE SEEK
                        004737  016204          7$:           JSR     PC,XSEEK        ;SEEK TO IT
                                                               60$
                        012701  000226                         MOV     #150.,R1        ;SET WAIT FOR 15 MS
                        004737  021026                         JSR     PC,RDYWAIT      ;WAIT FOR READY
                                                               60$

                        004737  021422                         JSR     PC,VERPOS       ;VERIFY POSITION
                                                               60$

                        032737  000002  002706                 BIT     #ALLSEC,OPFLAG  ;TEST IF USE ALL SECTORS
                        001406                                 BEQ     11$             ;NO-SKIP
                        004737  021542                         JSR     PC,RDALHD       ;ELSE READ ALL THE HDRS
                                                               60$
                        004737  020440                         JSR     PC,VERHDR       ;VERIFY THE HEADERS
                                                               60$
                                                11$:
                        023737  013350  003004                 CMP     LOLIMW,NEWCYL   ;CHECK IF REACHED LOLIMIT
                        103726                                 BLO     T216$           ;NO - LOOP
                        012737  000002  002720  60$:           MOV     #2,ERRSWI       ;INIT ERROR SWITCH
                                                ENDSUB
                                                L10025:
                        104003                                 EMT     C$ESUB
                                                T2165$:
                                                ENDTST
                                                L10024:
                        104001                                 EMT     C$ETST
```

```
                                        .SBTTL  *TEST 4          **INCREMENTAL FORWARD SEEK HEAD 1
                                        BGNTST                   ;TEST 4
025534                                                                                          T4::
025534  012737  006453  002714          MOV     #P2106E,ERHEAD   ;SET ERROR HEADER
025542  004737  015274                  JSR     PC,TSTINT        ;INITIALIZE TEST
025546  004737  015312                  JSR     PC,GSTATR        ;CLEAR DRIVE
025552  025756                          T2265$
025554  005037  003014                  CLR     DESHD            ;SET HEAD TO 0
025560  013705  013350                  MOV     LOLIMW,R5        ;CLEAR FOR POSITION HDS TO LOLIMIT
025564  004737  016746                  JSR     PC,POSHDS        ;POSITION HDS
025570  025756                          T2265$
025572  012737  000001  003014          MOV     #1,DESHD         ;SET TO HEAD 1
025600  032737  010000  013346          BIT     #HEADLM,MISWIW   ;TEST IF HEAD SPECIFIED
025606  001405                          BEQ     2$               ;NO - SKIP
025610  005737  013354                  TST     HEADW            ;TEST IF IT IS HEAD 0
025614  001002                          BNE     2$               ;NO - SKIP
025616                                  EXIT    TST              ;ELSE EXIT TEST
025616  104032                          EMT     C$EXIT
025620  000136                          .WORD   L10026-.
025622                          2$:
025622                          BGNSUB
025622                                                                                          T4.1:
025622  104002                          EMT     C$BSUB
025624  004737  021274          T227$:  JSR     PC,GETPOS        ;GET CURRENT POSITION
025630                                  INLOOP                   ;CHECK IF IN ERROR LOOP
025630  104020                          EMT     C$INLP
025632                                  BNCOMPLETE 5$            ;NO - SKIP
025632  103007                          BCC     5$
025634  023737  003006  003004          CMP     CURCYL,NEWCYL    ;CHECK IF AT DESIRED LOCATION
025642  001003                          BNE     5$               ;NO - SKIP
025644  004737  017632                  JSR     PC,ONSWAP        ;SWAP OLD AND NEW CYLINDER
025650  000405                          BR      7$               ;SKIP
025652  013737  003006  003004  5$:     MOV     CURCYL,NEWCYL    ;MOVE CURRENT INTO NEW
025660  005237  003004                  INC     NEWCYL           ;BUMP NEWCYL FOR ONE CYL FWRD SEEK
025664                          7$:
025664  004737  016204                  JSR     PC,XSEEK         ;DO SEEK
025670  025746                          60$
025672  012701  000226                  MOV     #150.,R1         ;SET WAIT COUNT 15 MS
025676  004737  021026                  JSR     PC,RDYWAIT       ;WAIT FOR READY
025702  025746                          60$
025704  004737  021422                  JSR     PC,VERPOS        ;VERIFY POSITION IS CORRECT
025710  025746                          60$

025712  032737  000002  013346          BIT     #ALLSEC,MISWIW   ;CHECK IF USE ALL SECTORS
025720  001406                          BEQ     9$               ;NO-SKIP
025722  004737  021542                  JSR     PC,RDALHD        ;ELSE READ ALL HEADERS
025726  025746                          60$
025730  004737  020440                  JSR     PC,VERHDR        ;VERIFY HEADERS
025734  025746                          60$
025736                          9$:
025736  023737  013352  003004          CMP     HILIMW,NEWCYL    ;CHECK IF DONE
025744  101327                          BH!     T227$            ;NO - LOOP
025746  012737  000002  002720  60$:     MOV     #2,ERRSWI        ;INIT ERROR SWITCH
025754                                  ENDSUB
025754                          L10027:
025754  104003                          EMT     C$ESUB
025756                          T2265$:
```

```
     49 025756                          ENDTST
        025756                          L10026:
        025756  104001                          EMT     C$ETST
```

```
                                        .SBTTL  *TEST 5          **INNER GUARD BAND DETECTION
025760                                  BGNTST                   ;TEST 5
025760                                                                                  T5::
025760  012737  006475  002714                  MOV     #P2TO7E,ERHEAD  ;SET ERROR HEADER
025766  004737  015274                          JSR     PC,TSTINT       ;INITIALIZE TEST
025772  004737  015312                          JSR     PC,GSTATR       ;CLEAR DRIVE
026776  026156                                  T2365$
026000  004737  017546                          JSR     PC,CHOSHD       ;GO CHOSE HEAD
026004  013705  002204          T233$:  MOV     HLMTW,R5        ;SET FOR POSITION TO 255.
026010  004737  016746                          JSR     PC,POSHDS       ;POSITION HEADS
026014  026156                                  T2365$
026016                                  BGNSUB
026016                                                                                  T5.1:
026016  104002                                  EMT     C$BSUB
026020  013737  002212  003004                  MOV     GBND,NEWCYL     ;SET FOR INNER GUARD BAND SEEK
026026  004737  016204                          JSR     PC,XSEEK        ;DO IT
026032  026132                                  60$
026034  012701  000003                          MOV     #3.,R1          ;SET WAIT COUNT 3 MS

026040  032762  000001  000000  7$:     BIT     #DRDYMSK,RLCS(R2) ;CHECK IF READY
026046  001413                          BEQ     9$              ;NO-SKIP
026050  004737  015342                          JSR     PC,GSTAT        ;GET DRIVE STATUS
026054  026132                                  60$
026056  012703  007561                          MOV     #MDRDY,R3       ;SET NAME MESSAGE PTR
026062  012704  010434                          MOV     #C10MS,R4       ;SET CONDITION MESSAGE PTR
026066                                  ERRHRD  701.,,ERR4      ;REPORT READY ERROR
026066  104443                          TRAP    T$ERCODE
026070  001275                          .WORD   701
026072  011710                          .WORD   ERR4
026074  000416                          BR      60$             ;EXIT TEST
026076  005301                  9$:     DEC     R1              ;DEC WAIT COUNT
026100  001404                          BEQ     11$             ;SKIP IF 0
026102                                  WAITUS  #10.            ;WAIT 100 US
026102  012700  000012                  MOV     #10.,R0
026106  104027                          EMT     C$WTU
026110  000753                          BR      7$              ;LOOP
026112  012701  000226          11$:    MOV     #150.,R1                ;SET WAIT COUNT 15 MS
026116  004737  021026                  JSR     PC,RDYWAIT      ;GO WAIT FOR READY
026122  026132                          60$

026124  004737  021422                  JSR     PC,VERPOS       ;GO VERIFY POSITION IS 255
026130  026132                          60$
026132  012737  000002  002720  60$:    MOV     #2,ERRSWI       ;INIT ERROR SWITCH
026140                                  ENDSUB
026140                                  L10031:
026140  104003                          EMT     C$ESUB
026142                                  ESCAPE  TST             ;EXIT TEST IF ERROR
026142  104010                          EMT     C$ESCAPE
026144  000012                          .WORD   L10030-.
026146  004737  017572                  JSR     PC,SWAPHD       ;GO SWAP TO HEAD 1 OR END TEST
026152  026156                          15$                     ;ABORT RETURN
026154  000713                          BR      T233$           ;REPEAT THE TESTS
026156                                  15$:
026156                                  T2365$:
026156                                  ENDTST
026156                                  L10030:
026156  104001                          EMT     C$ETST
```

```
                                        .SBTTL  *TEST 6          **INCREMENTAL REVERSE SEEK HEAD 1
                                        BGNTST                   ;TEST 6
026160                                                                                  T6::
026160  012737  006513  002714                  MOV     #P2T08E,ERHEAD  ;SET ERROR HEADER
026166  004737  015274                          JSR     PC,TSTINT       ;INITIALIZE TEST
026172  004737  015312                          JSR     PC,GSTATR       ;GET STATUS & CLEAR
026176  026404                                  T2465$
026200  005037  003014                          CLR     DESHD           ;SET TO HEAD 0
026204  013705  013352                          MOV     HILIMW,R5       ;SET TO POSITION HDS AT HILIMIT
026210  004737  016746                          JSR     PC,POSHDS       ;POSITION HDS
026214  026404                                  T2465$
026216  012737  000001  003014                  MOV     #1,DESHD        ;SET TO SELECT HD 1
026224  032737  010000  013346                  BIT     #HEADLM,MISWIW  ;TEST IF HEAD SPECIFIED
026232  001405                                  BEQ     2$              ;NO - SKIP
026234  005737  013354                          TST     HEADW           ;TEST IF HEAD SPECIFIED IS 0
026240  001002                                  BNE     2$              ;NO - SKIP
026242                                          EXIT    TST             ;ESLE EXIT TEST
026242  104032                                  EMT     C$EXIT
026244  000140                                  .WORD   L10032-.
                                        2$:
                                        BGNSUB
026246                                                                                  T6.1:
026246  104002                                  EMT     C$BSUB
026250  00473'  021274          T247$:          JSR     PC,GETPOS       ;GET CURRENT POSITION
026254  02637.                                  60$
026256                                          INLOOP                  ;CHECK IF IN ERROR LOOP
026256  10402)                                  EMT     C$INLP
026260                                          BNCOMPLETE      5$      ;NO - SKIP
026260  103007                                  BCC     5$
026262  023737  003006  003004                  CMP     CURCYL,NEWCYL   ;CHECK IF POSITIONED AT DESIRED LOC
026270  001003                                  BNE     5$              ;NO - SKIP
026272  004737  017632                          JSR     PC,ONSWAP       ;ELSE SWAP OLD AND NEW CYLINDER
026276  000405                                  BR      7$              ;SKIP
026300  013737  003006  003004  5$:             MOV     CURCYL,NEWCYL   ;MOV CUR TO NEW
026306  005337  003004                          DEC     NEWCYL          ;DEC NEWCYL FOR 1 CYL REV SEEK
026312  004737  016204          7$:             JSR     PC,XSEEK        ;DO SEEK
026316  026374                                  60$
026320  012701  000226                          MOV     #150.,R1                ;SET WAIT FOR 15 MS
026324  004737  021026                          JSR     PC,RDYWAIT      ;WAIT FOR READY
026330  026374                                  60$
026332  004737  021422                          JSR     PC,VERPOS       ;VERIFY POSITION
026336  026374                                  60$
026340  032737  000002  013346                  BIT     #ALLSEC,MISWIW  ;TEST IF ALL SECTORS
026346  001406                                  BEQ     9$              ;NO-EXIT
026350  004737  021542                          JSR     PC,RDALHD       ;READ ALL HEADERS
026354  026374                                  60$
026356  004737  020440                          JSR     PC,VERHDR       ;VERIFY HEADER
026362  026374                                  60$
026364                          9$:
026364  023737  013350  003004                  CMP     LOLIMW,NEWCYL   ;CHECK IF AT LOLIMIT
026372  103726                                  BLO     T247$           ;NO - LOOP
026374  012737  000002  002720  60$:            MOV     #2,ERRSWI       ;INIT ERROR SWITCH
026402                                  ENDSUB
026402                                  L10033:
026402  104003                                  EMT     C$ESUB
026404                          T2465$:
026404                          ENDTST
```

```
026404
026404  104001                  L10032:
                                        EMT     C$ETST
```

```
                                        .SBTTL  *TEST 7          **SEEK TESTS
026406                                  BGNTST          ;TEST 7
026406                                                                          T7::
026406  012737  006535  002714                MOV     #P2T09E,ERHEAD  ;SET ERROR HEADER
026414  004737  015274                        JSR     PC,TSTINT       ;INITIALIZE TEST
026420  004737  015312                        JSR     PC,GSTATR       ;CLEAR DRIVE
026424  026714                                T2565$
026426  004737  017546                        JSR     PC,CHOSHD       ;GO CHOSE HEAD
026432  013705  013350                        MOV     LOLIMW,R5       ;SET TO POSTION HEADS TO LOLIMIT
026436  004737  016746                        JSR     PC,POSHDS       ;POSITION HDS TO LOWLIMIT
026442  026714                                T2565$
026444  004737  021274          T256$:        JSR     PC,GETPOS       ;GET CURRENT POSITION
026450  026714                                T2565$
026452  013737  003006  003004                MOV     CURCYL,NEWCYL   ;PUT CURRENT INTO NEW
026460  012704  002332                        MOV     #T25TBL,R4      ;SET POINTER TO TABLE OF SEEK DIFF FOR RL01
026464  022737  000001  002200  T258$:        CMP     #1,T.DRIVE      ;CHECK TYPE OF DRIVE
026472  001402                                BEQ     T2588$          ;BRANCH IF RL01
026474  012704  00236C                        MOV     #T25TB2,R4      ;POINT TO THE RL02 TABLE OF CYLINDERS

026500  012405          T2588$:               MOV     (R4)+,R5        ;PUT FIRST IN R5
026502  013701  013352                        MOV     HILIMW,R1       ;GET HILIMIT
026506  163701  013350                        SUB     LOLIMW,R1       ;SUBTRACT LOLIMIT
026512  021401                                CMP     (R4),R1         ;CHECK IF NEW DIFFERENCE IS IN BOUNDS
026514  101073                                BHI     T2517$$         ;NO - SKIP TEST
026516  060537  003004          T257$:        ADD     R5,NEWCYL       ;ADD TO PRESENT POSITION
026522  023737  003004  013350                CMP     NEWCYL,LOLIMW   ;CHECK IF AT OR PAST LOLIMIT
026530  002004                                BGE     9$              ;NO - SKIP
026532  013737  013350  003004                MOV     LOLIMW,NEWCYL   ;ELSE SET TO LOLIMIT
026540  000407                                BR      11$
026542  023737  003004  013352  9$:           CMP     NEWCYL,HILIMW   ;CHECK IF AT HILIMIT OR GREATER
026550  003403                                BLE     11$             ;NO - SKIP
026552  013737  013352  003004                MOV     HILIMW,NEWCYL   ;ELSE SET FOR HILIMIT
026560                                  11$:
026560                                  BGNSUB
026560                                                                          T7.1:
026560  104002                                EMT     C$BSUB
026562                                        INLOOP                  ;CHECK IF IN ERROR LOOP
026562  104020                                EMT     C$INLP
026564                                        BNCOMPLETE      13$     ;NO - SKIP
026564  103011                                BCC     13$
026566  004737  021274                        JSR     PC,GETPOS       ;GET CURRENT POSITION
026572  026636                                60$
026574  023737  003006  003004                CMP     CURCYL,NEWCYL   ;CHECK IF HEADS AT DESIRED POSITION
026602  001002                                BNE     13$             ;NO - SKIP
026604  004737  017632                        JSR     PC,ONSWAP       ;ELSE SWAP CURRENT AND NEW CYLINDERS
026610  004737  016204          13$:          JSR     PC,XSEEK        ;DO SEEK
026614  026636                                60$
026616  012701  005670                        MOV     #3000.,R1       ;SET WAIT COUNT
026622  004737  021026                        JSR     PC,RDYWAIT      ;WAIT FOR READY
026626  026636                                60$
026630  004737  021422                        JSR     PC,VERPOS       ;VERIFY POSITION
026634  026636                                60$
026636  012737  000002  002720  60$:          MOV     #2,ERRSWI       ;INITIALIZE ERROR SWITCH
026644                                  ENDSUB
026644                                  L10035:
026644  104003                                EMT     C$ESUB
026646                                        ESCAPE  TST             ;EXIT TEST IF ERROR
```

```
        026646  104010                          EMT     C$ESCAPE
        026650  000044                          .WORD   L10034-.
    51  026652  023737  013352  003004          CMP     HILIMW,NEWCYL   ;CHECK IF SEEK WAS TO HILIMIT
    52  026660  001002                          BNE     15$             ;NO - SKIP
    53  026662  005405                          NEG     R5              ;ELSE SET R5 TO REPEAT DIFF IN REVERSE
    54  026664  000714                          BR      T257$
    55  026666  023737  013350  003004  15$:    CMP     LOLIMW,NEWCYL   ;TEST IF LAST SEEK WAS TO LOLIMIT
    56  026674  001310                          BNE     T257$           ;NO - GO DO SEEK TEST
    57  026676  021437  002204                  CMP     (R4),HLMTW      ;CHECK IF ALL TABLE DIFF USED
    58  026702  001276                          BNE     T2588$          ;NO - SKIP
    59  026704  004737  017572          T2517$: JSR     PC,SWAPHD       ;GO SWAP TO HEAD 1 OR END TEST
    60  026710  026714                          T2565$                  ;ABORT RETURN
    61  026712  000654                          BR      T256$           ;REPEAT TEST HEAD 1
    62  026714                          T2565$:
    63  026714                          ENDTST
        026714                          L10034:
        026714  104001                          EMT     C$ETST
```

```
                                        .SBTTL  *TEST 8          **FORWARD OSCILLATING SEEK
026716                                  BGNTST          ;TEST 8
026716                                                                                  T8::
026716  012737  006542  002714                  MOV     #P2T10E,ERHEAD  ;SET ERROR HEADER
026724  004737  015274                  JSR     PC,TSTINT       ;INITIALIZE TEST
026730  004737  015312                  JSR     PC,GSTATR       ;CLEAR DRIVE
026734  027212                          T2665$
026736  004737  017546                  JSR     PC,CHOSHD       ;GO CHOSE HEAD
026742  012705  000001          T266$:  MOV     #1,R5           ;LOAD R5 FOR FIRST SEEK
026746  032737  020000  013346          BIT     #HICYL,MISWIW   ;TEST IF HI CYLINDER SPECED
026754  001402                          BEQ     2$              ;NO - SKIP
026756  013705  013352                  MOV     HILIMW,R5       ;ELSE SET UPPER LIMIT
026762  005037  003004          2$:     CLR     NEWCYL          ;SET TO SEEK TO CYL 0
026766  032737  040000  013346          BIT     #LOCYL,MISWIW   ;CHECK IF LO CYL SPEC'D
026774  001403                          BEQ     5$              ;NO - SKIP
026776  013737  013350  003004          MOV     LOLIMW,NEWCYL   ;ELSE SET LOWER LIMIT
027004  004737  016204          5$:     JSR     PC,XSEEK        ;DO SEEK
027010  027212                          T2665$
027012  012701  005670                  MOV     #3000.,R1       ;SET WAIT COUNT FOR 120 MS
027016  004737  021026                  JSR     PC,RDYWAIT      ;WAIT FOR READY
027022  027212                          T2665$
027024  004737  021274          T267$:  JSR     PC,GETPOS       ;GET HEAD POSITION
027030  027212                          T2665$
027032  010537  003004                  MOV     R5,NEWCYL       ;LOAD NEW CYLINDER INTO NEWCYL
027036                                  BGNSUB
027036                                                                                  T8.1:
027036  104002                          EMT     C$BSUB
027040                                  INLOOP                  ;CHECK IF IN ERROR LOOP
027040  104020                          EMT     C$INLP
027042                                  BNCOMPLETE      18$     ;NO - SKIP
027042  103011                          BCC     18$
027044  004737  021274                  JSR     PC,GETPOS       ;GET POSITION
027050  027146                          60$
027052  023737  003006  003004          CMP     CURCYL,NEWCYL   ;CHECK IF HEADS AT DESIRED LOC
027060  001002                          BNE     18$             ;NO - SKIP
027062  004737  017632                  JSR     PC,ONSWAP               ;SWAP OLD AND NEW
027066  004737  016204          18$:    JSR     PC,XSEEK        ;DO SEEK
027072  027146                          60$
027074  012701  005670                  MOV     #3000.,R1       ;SET WAIT COUNT 120 MS
027100  004737  021026                  JSR     PC,RDYWAIT      ;WAIT FOR READY
027104  027146                          60$
027106  004737  021422                  JSR     PC,VERPOS       ;VERIFY HEAD POSITION
027112  027146                          60$
027114  005737  003012                  TST     DESSGN          ;TEST IF JUST SEEK REV
027120  001412                          BEQ     60$             ;YES - SKIP
027122  005037  003004                  CLR     NEWCYL          ;ELSE SET TO SEEK TO 0
027126  032737  040000  013346          BIT     #LOCYL,MISWIW   ;CHECK IF LO LIMIT SPEC'D
027134  001754                          BEQ     18$             ;NO - SKIP
027136  013737  013350  003004          MOV     LOLIMW,NEWCYL   ;ELSE SET LOW LIMIT FOR SEEK
027144  000750                          BR      18$
027146  012737  000002  002720  60$:    MOV     #2,ERRSWI       ;INIT ERROR SWITCH
027154                                  ENDSUB
027154                                  L10037:
027154  104003                          EMT     C$ESUB
027156                                  ESCAPE  TST             ;EXIT TEST IF ERROR
027156  104010                          EMT     C$ESCAPE
027160  000032                          .WORD   L10036-.
```

```
     49 027162  032737  020000  013346          BIT     #HICYL,MISWIW   ;TEST IF UPPER LIMIT SPEC'D
     50 027170  001004                          BNE     20$             ;YES - SKIP
     51 027172  005205                          INC     R5              ;BUMP R5
     52 027174  020537  002212                  CMP     R5,GBND ;ALL CYLINDERS DONE
     53 027200  001311                          BNE     T267$           ;NO - GO DO NEXT CYLINDER
     54 027202  004737  017572          20$:    JSR     PC,SWAPHD       ;GO SWAP TO HEAD 1 OR END TEST
     55 027206  027212                          T2665$                  ;ABORT RETURN
     56 027210  000654                          BR      T266$           ;GO DO TESTS
     57 027212                          T2665$:
     58 027212                          ENDTST
        027212                          L10036:
        027212  104001                          EMT     C$ETST
```

```
                                        .SBTTL  *TEST 9          **REVERSE OSCILLATING SEEK
027214                                  BGNTST                   ;TEST 9
027214                                                                                    T9::
027214  012737  006557  002714          MOV     #P2T11E,ERHEAD   ;SET ERROR HEADER
027222  004737  015274                  JSR     PC,TSTINT        ;INITIALIZE TEST
027226  004737  015312                  JSR     PC,GSTATR        ;CLEAR DRIVE
027232  027510                          T2765$
027234  004737  017546                  JSR     PC,CHOSHD        ;GO CHOSE HEAD
027240  013737  002204  003004  T275$:  MOV     HLMTW,NEWCYL     ;SEEK OUT TO 255.
027246  032737  020000  013346          BIT     #HICYL,MISWIW    ;TEST IF UPPER LIMIT SPEC'D
027254  001403                          BEQ     2$               ;NO - SKIP
027256  013737  013352  003004          MOV     HILIMW,NEWCYL    ;ELSE SET UPPER LIMIT
027264  013705  002210          2$:     MOV     NXTHL,R5         ;SET R5 FOR FIRST SEEKS
027270  032737  040000  013346          BIT     #LOCYL,MISWIW    ;CHECK IF LO LIMIT SPEC'D
027276  001402                          BEQ     5$               ;NO - SKIP
027300  013705  013350                  MOV     LOLIMW,R5        ;SET LOWER LIMIT
027304  004737  016204          5$:     JSR     PC,XSEEK         ;DO SEEK
027310  027510                          T2765$
027312  012701  005670                  MOV     #3000.,R1        ;SET WAIT TO 120 MS
027316  004737  021026                  JSR     PC,RDYWAIT       ;WAIT FOR DRIVE READY
027322  027510                          T2765$
027324  004737  021274          T276$:  JSR     PC,GETPOS        ;GET POSITION
027330  027510                          T2765$
027332  010537  003004                  MOV     R5,NEWCYL        ;SET FOR NEXT SEEK
027336                                  BGNSUB
027336                                                                                    T9.1:
027336  104002                          EMT     C$BSUB
027340                                  INLOOP                   ;CHECK IF IN ERROR LOOP
027340  104020                          EMT     C$INLP
027342                                  BNCOMPLETE      18$      ;NO - SKIP
027342  103011                          BCC     18$
027344  004737  021274                  JSR     PC,GETPOS        ;ELSE GET POSITION
027350  027450                          60$
027352  023737  003006  003004          CMP     CURCYL,NEWCYL    ;CHECK IF AT DESIRED CYL
027360  001002                          BNE     18$              ;NO - SKIP
027362  004737  017632                  JSR     PC,ONSWAP        ;ELSE SWAP OLD AND NEW CYL
027366  004737  016204          18$:    JSR     PC,XSEEK         ;DO SEEK
027372  027450                          60$
027374  012701  005670                  MOV     #3000.,R1        ;SET WAIT FOR 120 MS
027400  004737  021026                  JSR     PC,RDYWAIT       ;WAIT FOR READY
027404  027450                          60$
027406  004737  021422                  JSR     PC,VERPOS        ;VERIFY POSITION
027412  027450                          60$
027414  005737  003012                  TST     DESSGN           ;CHECK IF JUST SEEK FWD
027420  001013                          BNE     60$              ;YES - SKIP
027422  013737  002204  003004          MOV     HLMTW,NEWCYL     ;ELSE SEEK TO TO 255
027430  032737  020000  013346          BIT     #HICYL,MISWIW    ;TEST IF HILIMIT SPEC'D
027436  001753                          BEQ     18$              ;NO - SKIP
027440  013737  013352  003004          MOV     HILIMW,NEWCYL    ;SET TO UPPER LIMIT
027446  000747                          BR      18$
027450  012737  000002  002720  60$:    MOV     #2,ERRSWI        ;INIT ERROR SWITCH
027456                                  ENDSUB
027456                                  L10041:
027456  104003                          EMT     C$ESUB
027460                                  ESCAPE  TST              ;EXIT TEST IF ERROR
027460  104010                          EMT     C$ESCAPE
027462  000026                          .WORD   L10040-.
```

```
     49 027464  032737  040000  013346          BIT     #LOCYL,MISWIW   ;TEST IF LOLIMIT SPEC'D
     50 027472  001002                          BNE     20$             ;YES - SKIP
     51 027474  005305                          DEC     R5              ;DEC CYLINDER COUNT
     52 027476  100312                          BPL     T276$           ;IF STILL POSITIVE, DO SEEKS AGAIN
     53 027500  004737  017572          20$:    JSR     PC,SWAPHD       ;GO SWAP TO HEAD 1 OR END TEST
     54 027504  027510                          T276S$                  ;ABORT RETURN
     55 027506  000654                          BR      T275$           ;LOOP AGAIN
     56 027510                          T276S$:
     57 027510                          ENDTST
        027510                          L10040:
        027510  104001                          EMT     C$ETST
```

```
                                        .SBTTL  *TEST 10         **SEEK TIMING
027512                                  BGNTST                   ;TEST 10
027512                                                                          T10::
027512  012737  006574  002714                MOV     #P2T12E,ERHEAD  ;SET ERROR HEADER
027520  005737  003342                        TST     PASNUM          ;TEST IF PASS 0
027524  001003                              BNE     2$              ;NO - SKIP
027526  005737  013346                        TST     MISWIW          ;TEST IF MANUAL TESTS WERE RUN
027532  100402                              BMI     1$              ;YES - SKIP
027534  000137  031346                2$:     JMP     65$             ;ELSE EXIT TEST
027540  004737  015274                1$:     JSR     PC,TSTINT       ;INITIALIZE TEST
027544  004737  015312                        JSR     PC,GSTATR       ;CLEAR DRIVE
027550  031346                              65$
027552  012700  003042                        MOV     #OFIN,R0        ;GET ADDRESS OF 1ST TIME VALUE
027556  012701  000030                        MOV     #24.,R1         ;SET COUNT FOR CLEAR
027562  005020                        4$:     CLR     (R0)+           ;CLEAR TIMER STORAGE
027564  005301                              DEC     R1
027566  001375                              BNE     4$
027570  005037  003134                        CLR     PASCNT          ;CLEAR PASS COUNTER
027574  005037  003004                        CLR     NEWCYL          ;POSITION HEADS AT 0
027600  004737  016204                        JSR     PC,XSEEK        ;DO SEEK
027604  031346                              65$
027606  012701  005670                        MOV     #3000.,R1       ;SET WAIT FOR 300 MS
027612  004737  021026                        JSR     PC,RDYWAIT      ;WAIT FOR READY
027616  031346                              65$
027620  004737  021422                        JSR     PC,VERPOS       ;VERIFY POSITION
027624  031346                              65$
027626  004737  017546                        JSR     PC,CHOSHD       ;GO CHOSE HEAD
027632  012700  003052                        MOV     #OFOUT,R0       ;SET PTRS FOR 1 CYL FWD OUTER TIMER
027636  012701  003054                        MOV     #OFOUTU,R1
027642  012703  003066                        MOV     #OROUT,R3
027646  012704  003070                        MOV     #OROUTU,R4
027652  012737  000001  003004                MOV     #1,NEWCYL       ;SET NEWCYL TO CYL 1
027660  012737  000200  003136        8$:     MOV     #128.,COUNT     ;SET COUNTER FOR SEEK LOOP
027666  012737  000110  003040                MOV     #RDHEAD,TEMP8   ;BUILD READ HEADER COMMAND
027674  053737  002734  003040                BIS     RLDRV,TEMP8
027702  042737  002000  003040                BIC     #BIT10,TEMP8
027710  004737  016174                9$:     JSR     PC,XSEEKT       ;DO SEEK BUILD BUT DO NOT START
027714  031346                              65$
027716  013762  002742  000004                MOV     L.DA,RLDA(R2)   ;LOAD RL REGISTERS
027724  013762  002736  000000                MOV     L.CS,RLCS(R2)
027732  010046                              MOV     R0,-(SP)        ;STORE R0
027734                                      WAITUS  #10.            ;WAIT FOR INTERRUPT
027734  012700  000012                        MOV     #10.,R0
027740  104027                              EMT     C$WTU
027742  005737  002710                        TST     DONE            ;TEST IF INTERRUPT
027746  001010                              BNE     17$             ;YES - SKIP
027750  004737  015142                        JSR     PC,WAITIN       ;WAIT FOR INTERRUPT
027754  012603                              MOV     (SP)+,R3        ;GET MESSAGE POINTER
027756                                      ERRHRD  1201.,,ERR1
027756  104443                              TRAP    T$ERCODE
027760  002261                              .WORD   1201
027762  011526                              .WORD   ERR1
027764  000137  031346                      JMP     65$
027770  005737  002746                17$:    TST     T.CS            ;CHECK IF ANY ERRORS
027774  100005                              BPL     14$             ;NO - SKIP
027776                                      ERRHRD  1202.,,ERR6
027776  104443                              TRAP    T$ERCODE
```

```
                030000  002262                          .WORD   1202
                030002  012030                          .WORD   ERR6
     51         030004  000137  031346                  JMP     65$
     52         030010  005037  002710          14$:    CLR     DONE            ;CLEAR INTERRUPT FLAG
     53         030014  013762  003040  000000          MOV     TEMP8,RLCS(R2)  ;LOAD RL REGISTER
     54         030022                                  WAITUS  #2000.          ;WAIT FOR INTERRUPT
                030022  012700  003720                  MOV     #2000.,R0
                030026  104027                          EMT     C$WTU
     55         030030                                  GETTIM  R5              ;GET TIME USED
                030030  104052                          EMT     C$GTIM
                030032  010005                          MOV     R0,R5
     56         030034  012600                          MOV     (SP)+,R0        ;RESTORE R0
     57         030036  013737  003040  002736          MOV     TEMP8,L.CS      ;SET IF ERROR TO REPORT
     58         030044  004737  021422                  JSR     PC,VERPOS       ;VERIFY POSITION
     59         030050  031346                          65$
     60         030052  005737  003012                  TST     DESSGN          ;CHECK WHICH SEEK DIRECTION
     61         030056  001403                          BEQ     15$             ;REVERSE - SKIP
     62         030060  060510                          ADD     R5,(R0)         ;ADD TO FORWARD TOTAL
     63         030062  005511                          ADC     (R1)            ;ADD IN OVERFLOW
     64         030064  000402                          BR      16$             ;SKIP
     65         030066  060513          15$:    ADD     R5,(R3)         ;ADD TO REVERSE TOTAL
     66         030070  005514                          ADC     (R4)            ;ADD IN OVERFLOW
     67         030072  005337  003136          16$:    DEC     COUNT           ;DEC SEEK COUNT
     68         030076  001403                          BEQ     18$             ;SKIP IF 0
     69         030100  004737  017632                  JSR     PC,ONSWAP       ;ELSE SWAP OLD AND NEW CYL
     70         030104  000701                          BR      9$              ;REDO SEEK LOOP
     71         030106  162710  000470          18$:    SUB     #312.,(R0)      ;SUB CONSTANT FOR READ HEADER TIME
     72         030112  162713  000470                  SUB     #312.,(R3)
     73         030116  012705  000006                  MOV     #6,R5           ;SET SHIFT COUNT TO DIVIDE BY 64
     74         030122  000241          10$:    CLC                     ;DIVIDE BOTH TOTALS BY 64
     75         030124  006011                          ROR     (R1)
     76         030126  006010                          ROR     (R0)
     77         030130  000241                          CLC
     78         030132  006014                          ROR     (R4)
     79         030134  006013                          ROR     (R3)
     80         030136  005305                          DEC     R5
     81         030140  001370                          BNE     10$
     82         030142  005237  003134                  INC     PASCNT          ;BUMP PASS COUNT
     83         030146  022737  000001  003134          CMP     #1,PASCNT       ;TEST IF PASS 1
     84         030154  001051                          BNE     24$             ;NO - SKIP
     85         030156  012737  000177  003004          MOV     #127.,NEWCYL    ;ELSE SET TO POSITION HDS TO 127
     86         030164  022737  000001  002200          CMP     #1,T.DRIVE      ;DRIVE = RL01?
     87         030172  001403                          BEQ     101$            ;YUP
     88         030174  012737  000377  003004          MOV     #255.,NEWCYL    ;NO - SET FOR A MID POS SEEK RL02
     89         030202  004737  016204          101$:   JSR     PC,XSEEK        ;DO SEEK
     90         030206  031346                          65$
     91         030210  012701  005670                  MOV     #3000.,R1       ;SET WAIT COUNT FOR 300 MS
     92         030214  004737  021026                  JSR     PC,RDYWAIT      ;WAIT FOR READY
     93         030220  031346                          65$
     94         030222  004737  021422                  JSR     PC,VERPOS       ;VERIFY POSITION
     95         030226  031346                          65$
     96         030230  012700  003046                  MOV     #OFMID,R0       ;SET PTRS FOR TIMING 1 CYL SK AT 127
     97         030234  012701  003050                  MOV     #OFMIDU,R1
     98         030240  012703  003062                  MOV     #ORMID,R3
     99         030244  012704  003064                  MOV     #ORMIDU,R4
    100         030250  012737  000200  003004          MOV     #128.,NEWCYL    ;SET NEWCYL TO 128
    101         030256  022737  000001  002200          CMP     #1,T.DRIVE      ;RL01?
```

```
102 030264  001403                          BEQ     102$            ;YUP
103 030266  012737  000400  003004          MOV     #256.,NEWCYL    ;SET FOR RL02
104 030274  000137  027660          102$:   JMP     8$              ;DO SEEK LOOP
105 030300  022737  000002  003134  24$:    CMP     #2,PASCNT       ;TEST IF PASS 2
106 030306  001033                          BNE     28$             ;NO - SKIP
107 030310  013737  002210  003004          MOV     NXTHL,NEWCYL    ;SET UP TO TIME 1 CYL SEEK AT INNER
108 030316  004737  016204                  JSR     PC,XSEEK        ;  LIMIT
109 030322  031346                          65$
110 030324  012701  005670                  MOV     #3000.,R1       ;SET WAIT COUNT FOR 300 MS
111 030330  004737  021026                  JSR     PC,RDYWAIT      ;WAIT FOR READY
112 030334  031346                          65$
113 030336  004737  021422                  JSR     PC,VERPOS       ;VERIFY POSITION
114 030342  031346                          65$
115 030344  012700  003042                  MOV     #OFIN,R0        ;SET POINTERS
116 030350  012701  003044                  MOV     #OFINU,R1
117 030354  012703  003056                  MOV     #ORIN,R3
118 030360  012704  003060                  MOV     #ORINU,R4
119 030364  013737  002204  003004          MOV     HLMTW,NEWCYL    ;LOAD NEW CYLINDER
120 030372  000137  027660                  JMP     8$              ;DO SEEK LOOP
121 030376  022737  000003  003134  28$:    CMP     #3,PASCNT       ;TEST IF PASS 3
122 030404  001040                          BNE     32$             ;NO - SKIP
123 030406  005037  003004                  CLR     NEWCYL          ;ELSE SET UP TO TIME 128 CYL SEEK
124 030412  004737  016204                  JSR     PC,XSEEK        ;  AT OUTER LIMIT
125 030416  031346                          65$
126 030420  012701  005670                  MOV     #3000.,R1       ;SET WAIT COUNT FOR 300 MS
127 030424  004737  021026                  JSR     PC,RDYWAIT      ;WAIT FOR DRIVE READY
128 030430  031346                          65$
129 030432  004737  021422                  JSR     PC,VERPOS       ;VERIFY POSITION
130 030436  031346                          65$
131 030440  012700  003076                  MOV     #HFOUT,R0       ;SET POINTERS
132 030444  012701  003100                  MOV     #HFOUTU,R1
133 030450  012703  003106                  MOV     #HROUT,R3
134 030454  012704  003100                  MOV     #HFOUTU,R4
135 030460  012737  000177  003004          MOV     #127.,NEWCYL    ;LOAD NEWCYL FOR 128 CYL SEEK
136 030466  022737  000001  002200          CMP     #1,T.DRIVE      ;RL01?
137 030474  001505                          BEQ     39$             ;YUP
138 030476  012737  000377  003004          MOV     #255.,NEWCYL    ;NO - SET FOR RL02
139 030504  000501                          BR      39$
140 030506  022737  000004  003134  32$:    CMP     #4,PASCNT       ;TEST IF PASS 4
141 030514  001041                          BNE     36$             ;NO - SKIP
142 030516  012737  000200  003004          MOV     #128.,NEWCYL    ;ELSE SET UP TO TIME 128 CYL SEEK
143 030524  022737  000001  002200          CMP     #1,T.DRIVE      ;RL01?
144 030532  001403                          BEQ     321$            ;YES
145 030534  012737  000400  003004          MOV     #256.,NEWCYL    ;NO - SET FOR RL02
146 030542  004737  016204          321$:   JSR     PC,XSEEK        ; AT INNER LIMIT
147 030546  031346                          65$
148 030550  012701  005670                  MOV     #3000.,R1       ;SET WAIT COUNT FOR 300 MS
149 030554  004737  021026                  JSR     PC,RDYWAIT      ;WAIT FOR READY
150 030560  031346                          65$
151 030562  004737  021422                  JSR     PC,VERPOS       ;VERIFY POSITION
152 030566  031346                          65$
153 030570  012700  003072                  MOV     #HFIN,R0        ;SET POINTERS
154 030574  012701  003074                  MOV     #HFINU,R1
155 030600  012703  003102                  MOV     #HRIN,R3
156 030604  012704  003104                  MOV     #HRINU,R4
157 030610  013737  002204  003004          MOV     HLMTW,NEWCYL    ;SET NEWCYL TO 255 FOR 128 CYL SEEK
158 030616  000434                          BR      39$             ;DO TIMING LOOP
```

```
    159 030620  022737  000005  003134  36$:    CMP     #5,PASCNT       ;TEST IF PASS 5
    160 030626  001032                           BNE     40$             ;NO - SKIP
    161 030630  005037  003004                   CLR     NEWCYL          ;ELSE SET UP TO TIME 256 CYL SEEK
    162 030634  004737  016204                   JSR     PC,XSEEK        ;  OVER ALL SURFACE
    163 030640  031346                           65$
    164 030642  012701  005670                   MOV     #3000.,R1       ;SET WAIT COUNT FOR 300 MS
    165 030646  004737  021026                   JSR     PC,RDYWAIT      ;WAIT FOR DRIVE READY
    166 030652  031346                           65$
    167 030654  004737  021422                   JSR     PC,VERPOS       ;VERIFY POSITION
    168 030660  031346                           65$
    169 030662  012700  003112                   MOV     #AFMID,R0       ;SET POINTERS
    170 030666  012701  003114                   MOV     #AFMIDU,R1
    171 030672  012703  003116                   MOV     #ARMID,R3
    172 030676  012704  003120                   MOV     #ARMIDU,R4
    173 030702  013737  002204  003004           MOV     HLMTW,NEWCYL    ;SET NEWCYL
    174 030710  000137  027660          39$:     JMP     8$
    175 030714                          40$:     PRINTF  #FMT1.1,#SKTMES,#VALDES
        030714  012746  007040                   MOV     #VALDES,-(SP)
        030720  012746  007002                   MOV     #SKTMES,-(SP)
        030724  012746  010602                   MOV     #FMT1.1,-(SP)
        030730  012746  000003                   MOV     #3,-(SP)
        030734  010600                           MOV     SP,R0
        030736  104017                           EMT     C$PNTF
        030740  062706  000010                   ADD     #10,SP
    176 030744                                   PRINTF  #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
        030744  005046                           CLR     -(SP)
        030746  153716  002735                   BISB    RLDRV+1,(SP)
        030752  012746  006033                   MOV     #DRVNAM,-(SP)
        030756  013746  002730                   MOV     RLBAS,-(SP)
        030762  012746  006022                   MOV     #BASADD,-(SP)
        030766  012746  010630                   MOV     #FMT5,-(SP)
        030772  012746  000005                   MOV     #5,-(SP)
        030776  010600                           MOV     SP,R0
        031000  104017                           EMT     C$PNTF
        031002  062706  000014                   ADD     #14,SP
    177 031006                                   PRINTF  #FMT18,#LABIN,#LABMID,#LABOUT,#LABEXP
        031006  012746  007117                   MOV     #LABEXP,-(SP)
        031012  012746  007111                   MOV     #LABOUT,-(SP)
        031016  012746  007102                   MOV     #LABMID,-(SP)
        031022  012746  007074                   MOV     #LABIN,-(SP)
        031026  012746  011222                   MOV     #FMT18,-(SP)
        031032  012746  000005                   MOV     #5,-(SP)
        031036  010600                           MOV     SP,R0
        031040  104017                           EMT     C$PNTF
        031042  062706  000014                   ADD     #14,SP
    178 031046                                   PRINTF  #FMT19,#LABOCF,OFIN,OFMID,OFOUT,EXOCYL
        031046  013746  003122                   MOV     EXOCYL,-(SP)
        031052  013746  003052                   MOV     OFOUT,-(SP)
        031056  013746  003046                   MOV     OFMID,-(SP)
        031062  013746  003042                   MOV     OFIN,-(SP)
        031066  012746  007130                   MOV     #LABOCF,-(SP)
        031072  012746  011254                   MOV     #FMT19,-(SP)
        031076  012746  000006                   MOV     #6,-(SP)
        031102  010600                           MOV     SP,R0
        031104  104017                           EMT     C$PNTF
        031106  062706  000016                   ADD     #16,SP
    179 031112                                   PRINTF  #FMT19,#LABOCR,ORIN,ORMID,OROUT,EXOCYL
```

```
        031112  013746  003122                 MOV     EXOCYL,-(SP)
        031116  013746  003066                 MOV     OROUT,-(SP)
        031122  013746  003062                 MOV     ORMID,-(SP)
        031126  013746  003056                 MOV     ORIN,-(SP)
        031132  012746  007142                 MOV     #LABOCR,-(SP)
        031136  012746  011254                 MOV     #FMT19,-(SP)
        031142  012746  000006                 MOV     #6,-(SP)
        031146  010600                         MOV     SP,R0
        031150  104017                         EMT     C$PNTF
        031152  062706  000016                 ADD     #16,SP
    180 031156                                 PRINTF  #FMT20,#LABHCF,HFIN,HFOUT,EXHCYL
        031156  013746  003124                 MOV     EXHCYL,-(SP)
        031162  013746  003076                 MOV     HFOUT,-(SP)
        031166  013746  003072                 MOV     HFIN,-(SP)
        031172  012746  007154                 MOV     #LABHCF,-(SP)
        031176  012746  011311                 MOV     #FMT20,-(SP)
        031202  012746  000005                 MOV     #5,-(SP)
        031206  010600                         MOV     SP,R0
        031210  104017                         EMT     C$PNTF
        031212  062706  000014                 ADD     #14,SP
    181 031216                                 PRINTF  #FMT20,#LABHCR,HRIN,HROUT,EXHCYL
        031216  013746  003124                 MOV     EXHCYL,-(SP)
        031222  013746  003106                 MOV     HROUT,-(SP)
        031226  013746  003102                 MOV     HRIN,-(SP)
        031232  012746  007170                 MOV     #LABHCR,-(SP)
        031236  012746  011311                 MOV     #FMT20,-(SP)
        031242  012746  000005                 MOV     #5,-(SP)
        031246  010600                         MOV     SP,R0
        031250  104017                         EMT     C$PNTF
        031252  062706  000014                 ADD     #14,SP
    182 031256                                 PRINTF  #FMT21,#LABACF,AFMID,EXACYL
        031256  013746  003126                 MOV     EXACYL,-(SP)
        031262  013746  003112                 MOV     AFMID,-(SP)
        031266  012746  007204                 MOV     #LABACF,-(SP)
        031272  012746  011341                 MOV     #FMT21,-(SP)
        031276  012746  000004                 MOV     #4,-(SP)
        031302  010600                         MOV     SP,R0
        031304  104017                         EMT     C$PNTF
        031306  062706  000012                 ADD     #12,SP
    183 031312                                 PRINTF  #FMT21,#LABACR,ARMID,EXACYL
        031312  013746  003126                 MOV     EXACYL,-(SP)
        031316  013746  003116                 MOV     ARMID,-(SP)
        031322  012746  007220                 MOV     #LABACR,-(SP)
        031326  012746  011341                 MOV     #FMT21,-(SP)
        031332  012746  000004                 MOV     #4,-(SP)
        031336  010600                         MOV     SP,R0
        031340  104017                         EMT     C$PNTF
        031342  062706  000012                 ADD     #12,SP
    184 031346                             65$:
    185 031346                             ENDTST
        031346                             L10042:
        031346  104001                         EMT     C$ETST
```

```
                                .SBTTL  *TEST 11        **BASIC READ DATA (BAD SECTOR FILE)
031350                          BGNTST          ;TEST 11
031350                                                                          T11::
031350  012737  006610  002714          MOV     #P2T13E,ERHEAD  ;SET ERROR HEADER
031356  004737  015274                  JSR     PC,TSTINT       ;INITIALIZE TEST
031362  004737  015312                  JSR     PC,GSTATR       ;CLEAR DRIVE
031366  032030                          65$
031370  012737  000001  003014          MOV     #1,DESHD        ;SET TO HEAD 1
031376  032737  010000  013346          BIT     #HEADLM,MISWIW  ;TEST IF HEAD SPEC'D
031404  001405                          BEQ     2$              ;NO - SKIP
031406  005737  013354                  TST     HEADW           ;TEST IF HEAD 0
031412  001002                          BNE     2$              ;NO - SKIP
031414                                  EXIT    TST             ;ELSE EXIT TEST
031414  104032                          EMT     C$EXIT
031416  000440                          .WORD   L10043-.
031420  013737  002204  003004  2$:     MOV     HLMTW,NEWCYL    ;POSITION HEADS AT 255
031426  004737  016204                  JSR     PC,XSEEK        ;DO SEEK
031432  032030                          65$
031434  012701  005670                  MOV     #3000.,R1       ;SET WAIT COUNT FOR 300 MS
031440  004737  021026                  JSR     PC,RDYWAIT      ;WAIT FOR INTERRUPT
031444  032030                          65$
031446  004737  021422                  JSR     PC,VERPOS       ;VERIFY POSITION
031452  032030                          65$
031454  005037  003016                  CLR     DESSEC          ;SET FOR SECTOR 0
031460  012737  003552  003032          MOV     #FBSFIL,TEMP5   ;SET TEMP STORAGE FOR FACTORY BS FILE
031466  012737  000020  003034          MOV     #16.,TEMP6      ;SET MAX SECTOR COUNT
031474  112737  000001  003347          MOVB    #1,NOERCT       ;SET FOR NO ERROR COUNTING
031502  105037  003346                  CLRB    LOCERR          ;CLEAR LOCAL ERROR COUNTER
031506  005037  003026          4$:     CLR     TEMP3           ;CLEAR ONES DETECTED FLAG
031512  013701  003032                  MOV     TEMP5,R1        ;INIT POINTERS
031516  013700  003034                  MOV     TEMP6,R0
031522  012703  003746                  MOV     #IBUFF,R3
031526  012737  000002  002720          MOV     #2,ERRSWI       ;INIT ERROR SWITCH
031534  004737  022572                  JSR     PC,XREAD        ;DO READ
031540  031712                          39$
031542  005723                          TST     (R3)+           ;TEST IF WORD 0 NOT NEG
031544  100515                          BMI     45$             ;YES, BAD FMT ERROR
031546  005723                          TST     (R3)+           ;ELSE TEST WORD 1 NOT NEG
031550  100513                          BMI     45$             ;YES - BAD FMT ERROR REPORT
031552  005723                  7$:     TST     (R3)+           ;TEST WORD 2 IS 0
031554  001111                          BNE     45$             ;NO - SKIP TO FMT ERROR RPT
031556  005723                          TST     (R3)+           ;TEST WORD 3 IS 0
031560  001107                          BNE     45$             ;NO - SKIP TO FMT ERROR RPT
031562  021327  177777          8$:     CMP     (R3),#-1        ;TEST IF NEXT WORD IS ALL 1'S
031566  001004                          BNE     10$             ;NO  - SKIP
031570  012737  000001  003026          MOV     #1,TEMP3        ;ELSE SET 1'S DETECTED FLAG
031576  000403                          BR      11$             ;SKIP
031600  005737  003026          10$:    TST     TEMP3           ;TEST IF ONES HAVE BEEN DETECTED
031604  001075                          BNE     45$             ;YES - SKIP TO FMT ERROR RPT
031606  012311                  11$:    MOV     (R3)+,(R1)      ;STORE CYLINDER WORD
031610  012705  000007                  MOV     #7,R5           ;ALIGN IT TO LOOK LIKE HEADER
031614  006311                  12$:    ASL     (R1)
031616  005305                          DEC     R5
031620  001375                          BNE     12$
031622  032713  000400                  BIT     #BIT8,(R3)      ;TEST IF HEAD 1
031626  001402                          BEQ     15$             ;NO - SKIP
031630  052711  000100                  BIS     #BIT6,(R1)      ;INSERT HEAD BIT
```

```
 55 031634  042713  177400          15$:   BIC     #177400,(R3)     ;CLEAR ALL BUT SECTOR
 56 031640  052321                         BIS     (R3)+,(R1)+      ;INSERT SECTOR NUMBER
 57 031642  020327  004346                 CMP     R3,#IBUFF+256.   ;CHECK IF IBUFF EMPTY
 58 031646  001345                         BNE     8$               ;NO GET NEXT CYLINDER
 59 031650  005737  003026                 TST     TEMP3            ;ELSE TEST IF 1'S DETECTED
 60 031654  001457                         BEQ     48$              ;TO MANY ERRORS - REPORT
 61 031656  022737  000044  003034         CMP     #36.,TEMP6       ;CHECK IF SOFTWARE BAD READ
 62 031664  001461                         BEQ     65$              ;YES - SKIP
 63 031666  012737  003356  003032  37$:   MOV     #SBSFIL,TEMP5    ;ELSE CHANGE POINTERS
 64 031674  012737  000044  003034         MOV     #36.,TEMP6       ;       MAX SECTOR NUMBER
 65 031702  012737  000024  003016         MOV     #20.,DESSEC      ;       SECTOR NUMBER START
 66 031710  000676                         BR      4$               ;DO READ
 67 031712  005237  003346          39$:   INC     LOCERR           ;BUMP LOCAL ERROR COUNTER
 68 031716  012777  177777  151106  40$:   MOV     #-1,@TEMP5       ;MOV    1'S INTO FILE STORAGE
 69 031724                                 INLOOP                   ;CHECK IF IN ERROR LOOP
    031724  104020                         EMT     C$INLP
 70 031726                                 BCOMPLETE       4$               ;YES - GO DO READ
    031726  103667                         BCS     4$
 71 031730  023737  003016  003034  41$:   CMP     DESSEC,TEMP6     ;CHECK IF ALL SECTORS READ
 72 031736  001014                         BNE     43$              ;NO - SKIP
 73 031740  012703  005724                 MOV     #MBADSF,R3       ;SET RESULT MESSAGE POINTER
 74 031744  005237  003346                 INC     LOCERR           ;BUMP LOCAL ERROR COUNTER
 75 031750                                 ERRHRD  1301.,,ERR1
    031750  104443                         TRAP    T$ERCODE
    031752  002425                         .WORD   1301
    031754  011526                         .WORD   ERR1
 76 031756  022737  003356  003032         CMP     #SBSFIL,TEMP5    ;TEST IF SOFTWARE FILES CHECKED
 77 031764  001421                         BEQ     65$              ;YES - EXIT
 78 031766  000737                         BR      37$              ;ELSE GO CHECK SOFTWARE FILES
 79 031770  062737  000004  003016  43$:   ADD     #4,DESSEC        ;BUMP TO NEXT SECTOR
 80 031776  000643                         BR      4$               ;GO DO READ
 81 032000  012703  005754          45$:   MOV     #MFMTER,R3       ;SET RESULT MESSAGE POINTER
 82 032004                                 ERRHRD  1302.,,ERR1
    032004  104443                         TRAP    T$ERCODE
    032006  002426                         .WORD   1302
    032010  011526                         .WORD   ERR1
 83 032012  000737                         BR      39$              ;GO CHECK FOR LOOP
 84 032014  012703  006001          48$:   MOV     #MTMBS,R3        ;SET RESULT MESSAGE PTR
 85 032020                                 ERRHRD  1303.,,ERR1
    032020  104443                         TRAP    T$ERCODE
    032022  002427                         .WORD   1303
    032024  011526                         .WORD   ERR1
 86 032026  000733                         BR      40$              ;GO CHECK FOR LOOP
 87 032030  012737  000002  002720  65$:   MOV     #2,ERRSWI        ;INIT ERROR SWITCH
 88 032036  012737  000001  003354         MOV     #1,BSFVAL        ;SET BAD SECTOR FILES VALID FLAG
 89 032044  105737  003346                 TSTB    LOCERR           ;TEST IF LOCAL ERRORS
 90 032050  001402                         BEQ     66$              ;NO - SKIP
 91 032052  005237  003142                 INC     ERRCNT           ;ELSE BUMP ERROR COUNT
 92 032056                           66$:
 93 032056                           ENDTST
    032056                           L10043:
    032056  104001                         EMT     C$ETST
```

```
                                        .SBTTL  *TEST 12        **WRITE/READ DATA (PART 1)
032060                                  BGNTST.         ;TEST 12
032060                                                                                  T12::
032060  012737  006626  002714                 MOV     #P2T14E,ERHEAD  ;SET ERROR HEADER
032066  004737  017656                        JSR     PC,CKBSVD       ;GO CHECK IF BAD SECTOR FILES VALID
032072  004737  015274                        JSR     PC,TSTINT       ;INITIALIZE TEST
032076  004737  015312                        JSR     PC,GSTATR       ;CLEAR DRIVE
032102  032272                                T3065$
032104  004737  017546                        JSR     PC,CHOSHD       ;GO CHOSE HEAD
032110  005037  003016                        CLR     DESSEC          ;       SECTOR 0
032114  005037  003004                        CLR     NEWCYL          ;       CYLINDER 0
032120  005037  032164                        CLR     T310$           ;CLEAR PATTERN SELECT
032124  004737  016204          T306$:        JSR     PC,XSEEK        ;POSITION HEADS
032130  032272                                T3065$
032132  012701  005670                        MOV     #3000.,R1       ;SET WAIT COUNT FOR 300 MS
032136  004737  021026                        JSR     PC,RDYWAIT      ;WAIT FOR READY
032142  032272                                T3065$
032144  004737  021422                        JSR     PC,VERPOS       ;VERIFY POSITION
032150  032272                                T3065$
032152  005037  032164                        CLR     T310$           ;CLEAR PATTERN SELECTOR
032156                          T307$:
032156                                  BGNSUB
032156                                                                                  T12:1:
032156  104002                                  EMT     C$BSUB
032160  004537  022104                        JSR     R5,DATGEN       ;GENERATE DATA
032164  000000                  T310$:          .WORD   0               ;PATTERN SELECT WORD
032166  004737  022532                        JSR     PC,XWRITE       ;DO WRITE DATA
032172  032210                                60$
032174  004737  022572                        JSR     PC,XREAD        ;DO READ DATA
032200  032210                                60$
032202  004737  022244                        JSR     PC,DATCOM       ;COMPARE DATA
032206  032210                                60$
032210  012737  000002  002720    60$:        MOV     #2,ERRSWI       ;INIT ERROR SWITCH
032216                                  ENDSUB
032216                                  L10045:
032216  104003                                  EMT     C$ESUB
032220                                          ESCAPE  TST             ;EXIT TEST IF ERROR
032220  104010                                  EMT     C$ESCAPE
032222  000050                                  .WORD   L10044-.
032224  022737  000010  032164                 CMP     #8.,T310$       ;WAS DATA PAT 8 USED?
032232  001403                                  BEQ     10$             ;YES - SKIP
032234  005237  032164                        INC     T310$           ;ELSE BUMP TO NEXT PATTERN
032240  000746                                  BR      T307$           ;DO TEST WITH NEW PATTERN
032242  004737  017572          10$:          JSR     PC,SWAPHD       ;GO SWAP TO HEAD 1 OR END TEST
032246  032272                                T3065$                  ;ABORT RETURN
032250  005037  032164                        CLR     T310$           ;SET PATTERN SELECT TO 0
032254  004737  023252          11$:          JSR     PC,BSCHK        ;CHECK IF SECTOR BAD
032260  032264                                13$                     ;YES RETURN - SKIP TO 13$
032262  000720                                BR      T306$           ;NO RETURN - DO TEST THIS SECTOR
032264  005237  003004          13$:          INC     NEWCYL          ;BUMP TO NEXT CYLINDER
032270  000771                                BR      11$             ;CHECK IF THIS ONE BAD
032272                          T3065$:
032272                                  ENDTST
032272                                  L10044:
032272  104001                                  EMT     C$ETST
```

```
                                        .SBTTL  *TEST 13        **SPINDLE TIMING TEST
032274                                  BGNTST          ;TEST 13
032274                                                                          T13::
032274  012737  006647  002714                  MOV     #P2T15E,ERHEAD  ;SET ERROR HEADER
032302  005737  003342                          TST     PASNUM          ;TEST IF PASS 0
032306  001003                                  BNE     2$              ;NO - SKIP
032310  005737  013346                          TST     MISWIW          ;TEST IF MANUAL TESTS WERE RUN
032314  100402                                  BMI     1$              ;YES - SKIP
032316                                  2$:     EXIT    TST             ;ELSE SKIP TEST
032316  104032                                  EMT     C$EXIT
032320  000476                                  .WORD   L10046-.
032322  005003                          1$:     CLR     R3              ;CLEAR FOR TIMING STORAGE
032324  005004                                  CLR     R4
032326  004737  015274                          JSR     PC,TSTINT       ;INITIALIZE TEST
032332  004737  015312                          JSR     PC,GSTATR       ;CLEAR DRIVE
032336  033010                                  60$
032340  004537  022104                          JSR     R5,DATGEN       ;GENERATE DATA
032344  000000                                  0                       ;PATTERN 0
032346  005037  003016                          CLR     DESSEC          ;CLEAR TO SECTOR 0
032352  004737  017546                          JSR     PC,CHOSHD       ;GO SELECT HEAD
032356  013737  013350  003004                  MOV     LOLIMW,NEWCYL   ;SET FOR CYLINDER
032364  004737  016204                          JSR     PC,XSEEK        ;DO SEEK
032370  033010                                  60$
032372  012701  005670                          MOV     #3000.,R1       ;SET WAIT FOR 300 MS
032376  004737  021026                          JSR     PC,RDYWAIT      ;WAIT FOR READY
032402  033010                                  60$
032404  004737  021422                          JSR     PC,VERPOS       ;VERIFY POSITION
032410  033010                                  60$
032412  012701  000100                          MOV     #64.,R1 ;SET LOOP COUNTER
032416  012705  002744                  5$:     MOV     #L.MP,R5        ;SET A POINTER
032422  004737  022522                          JSR     PC,XWRITT       ;DO FIRST WRITE
032426  033010                                  60$
032430  011562  000006                          MOV     (R5),RLMP(R2)   ;LOAD RL REGISTERS
032434  014562  000004                          MOV     -(R5),RLDA(R2)
032440  014562  000002                          MOV     -(R5),RLBA(R2)
032444  014562  000000                          MOV     -(R5),RLCS(R2)
032450                                          WAITUS  #3000.
032450  012700  005670                          MOV     #3000.,R0
032454  104027                                  EMT     C$WTU
032456  005737  002710                          TST     DONE            ;TEST IF INTERRUPT
032462  001010                                  BNE     6$              ;YES - SKIP
032464  004737  015142                          JSR     PC,WAITIN       ;ELSE WAIT FOR TIMEOUT
032470  012603                                  MOV     (SP)+,R3        ;GET MESSAGE POINTER
032472                                          ERRHRD  1501.,,ERR1
032472  104443                                  TRAP    T$ERCODE
032474  002735                                  .WORD   1501
032476  011526                                  .WORD   ERR1
032500  000137  033010                          JMP     60$
032504  005737  002746                  6$:     TST     T.CS            ;TEST IF ANY ERRORS
032510  100005                                  BPL     4$              ;NO - SKIP
032512                                          ERRHRD  1502.,,ERR6
032512  104443                                  TRAP    T$ERCODE
032514  002736                                  .WORD   1502
032516  012030                                  .WORD   ERR6
032520  000137  033010                          JMP     60$
032524  012705  002744                  4$:     MOV     #L.MP,R5        ;SET POINTER TO RL LOAD REGS
032530  005037  002710                          CLR     DONE            ;CLEAR INTERRUPT INDICATOR
```

```
 47 032534  011562  000006                MOV     (R5),RLMP(R2)   ;LOAD RL REGISTERS FOR 2ND WRITE
 48 032540  014562  000004                MOV     -(R5),RLDA(R2)
 49 032544  014562  000002                MOV     -(R5),RLBA(R2)
 50 032550  014562  000000                MOV     -(R5),RLCS(R2)
 51 032554                                WAITUS  #3000.          ;WAIT FOR INTERRUPT
    032554  012700  005670                MOV     #3000.,R0
    032560  104027                       EMT     C$WTU
 52 032562                                GETTIM  R0              ;GET TIME WAITED
    032562  104052                       EMT     C$GTIM
 53 032564  005737  002710                TST     DONE            ;TEST IN INTERRUPT OCCURRED
 54 032570  001007                       BNE     7$              ;YES - SKIP
 55 032572  004737  015142                JSR     PC,WAITIN       ;GO WAIT FOR INTERRUPT
 56 032576  012603                       MOV     (SP)+,R3        ;GET MESSAGE POINTER
 57 032600                                ERRHRD  1503.,,ERR1     ;REPORT
    032600  104443                       TRAP    T$ERCODE
    032602  002737                       .WORD   1503
    032604  011526                       .WORD   ERR1
 58 032606  000500                       BR      60$
 59 032610  005737  002746        7$:     TST     T.CS            ;TEST IN ANY ERROR
 60 032614  100004                       BPL     8$              ;NO - SKIP
 61 032616                                ERRHRD  1504.,,ERR6     ;REPORT ERRORS
    032616  104443                       TRAP    T$ERCODE
    032620  002740                       .WORD   1504
    032622  012030                       .WORD   ERR6
 62 032624  000471                       BR      60$
 63 032626  060003        8$:     ADD     R0,R3           ;ADD IN TIME USED
 64 032630  005504                       ADC     R4              ;DOUBLE PRECISSION
 65 032632  005301                       DEC     R1              ;DEC LOOP COUNTER
 66 032634  001270                       BNE     5$              ;LOOP UNTIL 0
 67 032636  012701  000006                MOV     #6,R1           ;SET DIVIDE COUNT
 68 032642  000241        10$:    CLC                     ;CLEAR CARRY FOR DIVIDE
 69 032644  006004                       ROR     R4              ;DIVIDE SUM BY 100(8)
 70 032646  006003                       ROR     R3
 71 032650  005301                       DEC     R1              ;DEC DIVIDE COUNT
 72 032652  001373                       BNE     10$             ;LOOP UNTIL DONE
 73 032654                                PRINTF  #FMT1.1,#SRTMES,#VALDES
    032654  012746  007040                MOV     #VALDES,-(SP)
    032660  012746  007016                MOV     #SRTMES,-(SP)
    032664  012746  010602                MOV     #FMT1.1,-(SP)
    032670  012746  000003                MOV     #3,-(SP)
    032674  010600                       MOV     SP,R0
    032676  104017                       EMT     C$PNTF
    032700  062706  000010                ADD     #10,SP
 74 032704                                PRINTF  #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
    032704  005046                       CLR     -(SP)
    032706  153716  002735                BISB    RLDRV+1,(SP)
    032712  012746  006033                MOV     #DRVNAM,-(SP)
    032716  013746  002730                MOV     RLBAS,-(SP)
    032722  012746  006022                MOV     #BASADD,-(SP)
    032726  012746  010630                MOV     #FMT5,-(SP)
    032732  012746  000005                MOV     #5,-(SP)
    032736  010600                       MOV     SP,R0
    032740  104017                       EMT     C$PNTF
    032742  062706  000014                ADD     #14,SP
 75 032746                                PRINTF  #FMT26,#RESE3,R3,#RESE4,#MAPROX,EXROT
    032746  013746  003130                MOV     EXROT,-(SP)
    032752  012746  007064                MOV     #MAPROX,-(SP)
```

```
        032756  012746  010365                 MOV     #RESE4,-(SP)
        032762  010346                         MOV     R3,-(SP)
        032764  012746  010361                 MOV     #RESE3,-(SP)
        032770  012746  011451                 MOV     #FMT26,-(SP)
        032774  012746  000006                 MOV     #6,-(SP)
        033000  010600                         MOV     SP,R0
        033002  104017                         EMT     C$PNTF
        033004  062706  000016                 ADD     #16,SP
     76 033010  012737  000002  002720  60$:   MOV     #2,ERRSWI       ;INITIALIZE ERROR SWITCH
     77 033016                          ENDTST
        033016                          L10046:
        033016  104001                         EMT     C$ETST
     78
```

```
                                        .SBTTL  *TEST 14         **WRITE/READ DATA (PART 2)
033020                                  BGNTST                   ;TEST 14
033020                                                                                  T14::
033020  012737  006672  002714                  MOV     #P2T16E,ERHEAD  ;SET ERROR HEADER
033026  004737  017656                          JSR     PC,CKBSVD       ;GO CHECK IF BAD SECTOR FILES VALID
033032  004737  015274                          JSR     PC,TSTINT       ;INITIALIZE TEST
033036  004737  015312                          JSR     PC,GSTATR       ;CLEAR DRIVE
033042  034126                                  T3165$
033044  005037  003134                          CLR     PASCNT          ;CLEAR PASS TO 0
033050  012705  177776                          MOV     #-2,R5          ;SET
033054  005737  003342                          TST     PASNUM          ;TEST IF FIRST PASS (QUICK VERIFY)
033060  001006                                  BNE     1$              ;NO - SKIP
033062  032737  000001  013346                  BIT     #ALLCYL,MISWIW  ;TEST IF USE ALL CYLINDERS
033070  001002                                  BNE     1$              ;YES - SKIP
033072  012705  177760                          MOV     #-16.,R5                ;ELSE SET PEOPLE TO NEG 8
033076                                  1$:
033076  012701  002406                          MOV     #T33TBL,R1      ;GET ADDRESS OF WORK TABLE
033102  012737  00001C  002202                  MOV     #10,JJJ         ;SET CLEAR COUNT
033110  013721  013350          2$:     MOV     LOLIMW,(R1)+    ;CLEAR LOCATIONS TO LO LIMIT
033114  005337  002202                          DEC     JJJ             ;DEC COUNT
033120  001373                                  BNE     2$              ;LOOP UNTIL 0
033122  013737  013352  002412                  MOV     HILIMW,T33TBL+4 ;INSERT HILIMIT
033130  013737  013352  002414                  MOV     HILIMW,T33TBL+6 ;INTO APPROPRIATE LOCATIONS
033136  013737  013352  002416                  MOV     HILIMW,T33TBL+10
033144  062705  000002          T3100$: ADD     #2,R5   ;BUMP R5 BY 2
033150  032737  000001  013346                  BIT     #ALLCYL,MISWIW  ;TEST IF USE ALL CYLINDERS
033156  001031                                  BNE     5$              ;YES - SKIP
033160  005737  003342                          TST     PASNUM          ;TEST IF FIRST PASS (QUICK VERIFY)
033164  001002                                  BNE     3$              ;NO - SKIP
033166  062705  000016                          ADD     #16,R5          ;ELSE BUMP CYLINDER POINTER BY 7
033172  022737  000001  002200  3$:     CMP     #1,T.DRIVE      ;RL01 OR RL02?THAT IS THE Q
033200  001404                                  BEQ     44$             ;ANS IS RL01
033202  020527  000244                          CMP     R5,#164.
033206  103013                                  BHIS    4$
033210  000403                                  BR      69$             ;TEST PAST TABLE-YES EXIT
033212  020527  000122          44$:    CMP     R5,#82.
033216  103007                                  BHIS    4$              ;TES PAST THE TABLE

033220  016537  002506  002202  69$:    MOV     CYLTBL(R5),JJJ  ;GET NEXT TABLE ENTRY
033226  043737  002206  002202                  BIC     CLRBYT,JJJ      ;CLEAR UPPER BYTE
033234  001007                                  BNE     8$
033236  000137  034126          4$:     JMP     T3165$          ;EXIT TEST
033242  023705  013352          5$:     CMP     HILIMW,R5       ;TEST IF ALL CYLINDERS USED
033246  001773                                  BEQ     4$              ;YES - EXIT TEST
033250  010537  002202                          MOV     R5,JJJ          ;USE R5 AS NEXT CYLINDER
033254  023737  002202  013350  8$:     CMP     JJJ,LOLIMW      ;CHECK IF LOWER THAN LOLIMIT
033262  103730                                  BLO     T3100$          ;YES - SKIP
033264  023737  002202  013352                  CMP     JJJ,HILIMW      ;CHECK IF HIGHER THAN HILIMIT
033272  101324                                  BHI     T3100$          ;YES - SKIP
033274  012703  002446                          MOV     #TBT,R3
033300  013713  002202                          MOV     JJJ,(R3)
033304  013763  002202  000002                  MOV     JJJ,2(R3)
033312  013763  002202  000004                  MOV     JJJ,4(R3)
033320  013763  002202  000006                  MOV     JJJ,6(R3)
033326  013763  002202  000010                  MOV     JJJ,10(R3)
033334  013763  002202  000012                  MOV     JJJ,12(R3)
033342  010337  002726                          MOV     R3,TBLSTR       ;STORE TABLE ADDRESS
```

```
CZRLJA0 RL01/2 DR TST 2 MACRO V03.01 9-FEB-79 19:23:55 PAGE 18-1
*TEST 14        **WRITE/READ DATA (PART 2)                                    SEQ 0125

     57 033346  004737  017546                    JSR     PC,CHOSHD       ;GO CHOSE HEAD
     58 033352                          T3101$:
     59 033352                          BGNSUB
        033352                                                              T14.1:
        033352  104002                          EMT     C$BSUB
     60 033354  042737  003760  002706           BIC     #MQUALS,OPFLAG  ;CLEAR ALL MESSAGE QUALIFIERS
     61 033362  005737  003134                   TST     PASCNT          ;TEST IF PASS 0
     62 033366  001414                           BEQ     11$             ;YES - SKIP
     63 033370  023727  003134  000003           CMP     PASCNT,#3       ;TEST IF PASS 3
     64 033376  001404                           BEQ     10$             ;YES - SKIP
     65 033400  002407                           BLT     11$             ;CHECK IF LESS THAN 3, IF YES CLEAR TO 0
     66 033402  012737  000003  003134           MOV     #3,PASCNT       ;ELSE SET TO 3
     67 033410  052737  000020  002706  10$:     BIS     #INOUTS,OPFLAG  ;SET MESSAGE QUAL
     68 033416  000405                           BR      12$             ;SKIP
     69 033420  005037  003134          11$:     CLR     PASCNT          ;SET PASS COUNT TO 0
     70 033424  052737  000040  002706           BIS     #OUTINS,OPFLAG  ;SET MESSAGE QUAL
     71 033432  012737  000003  002724  12$:     MOV     #3,WRTSWI       ;SET READ AND WRITE SWITCH
     72 033440  013703  002726                   MOV     TBLSTR,R3       ;GET STORED TABLE ADDRESS
     73 033444  012701  002406                   MOV     #T33TBL,R1
     74 033450  012703  002446                   MOV     #TBT,R3
     75 033454  005037  003016          15$:     CLR     DESSEC          ;CLEAR TO SECTOR 0
     76 033460  012137  003004                   MOV     (R1)+,NEWCYL    ;GET NEXT TABLE ENTRY
     77 033464  004737  016204                   JSR     PC,XSEEK        ;DO SEEK
     78 033470  034034                           60$
     79 033472  012701  005670                   MOV     #3000.,R1       ;SET WAIT COUNT FOR 300 MS
     80 033476  004737  021026                   JSR     PC,RDYWAIT      ;WAIT FOR READY
     81 033502  034034                           60$
     82 033504  012337  003004                   MOV     (R3)+,NEWCYL    ;GET NEXT TABLE ENTRY
     83 033510  004737  016204                   JSR     PC,XSEEK        ;DO SEEK
     84 033514  034034                           60$
     85 033516  012701  005670                   MOV     #3000.,R1       ;SET WAIT COUNT FOR 300 MS
     86 033522  004737  021026                   JSR     PC,RDYWAIT      ;WAIT FOR READY
     87 033526  034034                           60$
     88 033530  004737  021422                   JSR     PC,VERPOS       ;VERIFY POSITION
     89 033534  034034                           60$
     90 033536  004737  023252          16$:     JSR     PC,BSCHK        ;CHECK FOR BAD SECTOR
     91 033542  033674                           32$                     ;"YES" RETURN
     92 033544  013737  003016  033564           MOV     DESSEC,25$      ;SET DATA PATTERN = TO SECTOR NUMBER
     93 033552  042737  177770  033564           BIC     #177770,25$     ;CLEAR ALL BUT LSD
     94 033560  004537  022104                   JSR     R5,DATGEN       ;GO GENERATE DATA
     95 033564  000000                  25$:     .WORD   0
     96 033566  032737  000001  002724           BIT     #BIT0,WRTSWI    ;TEST IF WRITE THIS PASS
     97 033574  001425                           BEQ     29$             ;NO - SKIP
     98 033576  004737  022532                   JSR     PC,XWRITE       ;DO WRITE
     99 033602  034034                           60$
    100 033604  005237  003016                   INC     DESSEC          ;INC SECTOR
    101 033610  022737  000050  003016           CMP     #40.,DESSEC     ;TEST IF ALL SECTORS USED
    102 033616  001347                           BNE     16$             ;NO - SKIP
    103 033620  042737  000060  002706           BIC     #INOUTS!OUTINS,OPFLAG ;CLEAR QUALIFIERS
    104 033626  042737  000001  002724           BIC     #BIT0,WRTSWI    ;CLEAR WRITE REQUIRED SWITCH
    105 033634  052737  000100  002706           BIS     #FOLWRT,OPFLAG  ;SET FOLLOWING WRITE QUALIFIER
    106 033642  005037  003016                   CLR     DESSEC          ;CLEAR TO SECTOR 0
    107 033646  000733                           BR      16$             ;SKIP
    108 033650  032737  000002  002724  29$:     BIT     #BIT1,WRTSWI    ;TEST IF READ THIS PASS
    109 033656  001414                           BEQ     33$             ;NO - SKIP
    110 033660  004737  022572          31$:     JSR     PC,XREAD        ;ELSE DO READ
    111 033664  034034                           60$
```

```
    112 033666  004737  022244                      JSR     PC,DATCOM       ;COMPARE DATA
    113 033672  034034                              60$
    114 033674  005237  003016              32$:    INC     DESSEC          ;BUMP SECTOR
    115 033700  022737  000050  003016              CMP     #40.,DESSEC     ;TEST IF ALL SECTORS USED
    116 033706  001313                              BNE     16$             ;NO - LOOP
    117 033710  005037  003016              33$:    CLR     DESSEC          ;CLEAR DESIRED SECTOR
    118 033714  005037  002724                      CLR     WRTSWI          ;CLEAR WRITE/READ SWITCH
    119 033720  005237  003134                      INC     PASCNT          ;BUMP PASS COUNT
    120 033724  042737  003760  002706              BIC     #MQUALS,OPFLAG  ;CLEAR ALL QUALIFIERS
    121 033732  023727  003134  000003              CMP     PASCNT,#3       ;TEST IS PASS 3
    122 033740  001435                              BEQ     60$             ;YES - SKIP
    123 033742  023727  003134  000006              CMP     PASCNT,#6       ;TEST IF PASS 6
    124 033750  001431                              BEQ     60$             ;YES - SKIP
    125 033752  012737  000002  002724              MOV     #BIT1,WRTSWI    ;SET READ REQUIRED BIT
    126 033760  023727  003134  000001              CMP     PASCNT,#1       ;TEST IF PASS 1
    127 033766  001415                              BEQ     40$             ;YES - SKIP
    128 033770  023727  003134  000005              CMP     PASCNT,#5       ;TEST IF PASS 4
    129 033776  001411                              BEQ     40$             ;YES - SKIP
    130 034000  000404                              BR      39$             ;SKIP
    131 034002  052737  002000  002706  37$:        BIS     #FWDSKO,OPFLAG  ;SET FWD QUALIFIER
    132 034010  000407                              BR      36$             ;GO DO NEXT PASS
    133 034012  052737  000020  002706  39$:        BIS     #INOUTS,OPFLAG  ;SET QUALIFIER
    134 034020  000403                              BR      36$             ;SKIP
    135 034022  052737  000040  002706  40$:        BIS     #OUTINS,OPFLAG  ;SET MESSAGE QUALIFIER
    136 034030  000137  033454          36$:        JMP     15$             ;GO DO NEXT PASS
    137 034034  012737  000002  002720  60$:        MOV     #2,ERRSWI       ;INIT ERROR SWITCH
    138 034042                          ENDSUB
        034042                          L10050:
        034042  104003                              EMT     C$ESUB
    139 034044                                      ESCAPE  TST             ;EXIT TEST IF ERROR
        034044  104010                              EMT     C$ESCAPE
        034046  000060                              .WORD   L10047-.
    140 034050  012737  000003  002724              MOV     #3,WRTSWI       ;SET FOR READ AND WRITE REQ.
    141 034056  023727  003134  000003              CMP     PASCNT,#3       ;TEST IF PASS 3
    142 034064  001004                              BNE     45$             ;NO - SKIP
    143 034066  012737  002414  002726              MOV     #T33TBL+6,TBLSTR ;STORE MID POINT IN TABLE
    144 034074  000410                              BR      48$             ;GO START PASS 4
    145 034076  005037  003134          45$:        CLR     PASCNT          ;CLEAR TO PASS 0
    146 034102  004737  017572                      JSR     PC,SWAPHD       ;GO SWAP TO HEAD 1 OR END TEST
    147 034106  033144                              T3100$                  ;ABORT RETURN
    148 034110  012737  002406  002726              MOV     #T33TBL,TBLSTR  ;STORE START OF TABLE
    149 034116  062703  000006          48$:        ADD     #6,R3
    150 034122  000137  033352                      JMP     T3101$
    151 034126                          T3165$:
    152 034126                          ENDTST
        034126                          L10047:
        034126  104001                              EMT     C$ETST
```

```
      1                                 .SBTTL  *TEST 15         **WRITE LOCK ERROR AND DATA PROTECTION
      2 034130                          BGNTST                   ;TEST 15
        034130                                                                                  T15::
      3 034130  005737  003342                       TST     PASNUM          ;TEST IF FIRST PASS
      4 034134  001003                                BNE     2$              ;NO - SKIP
      5 034136  005737  013346                       TST     MISWIW          ;TEST IF RUN MANUAL INTERVENTION
      6 034142  100402                                BMI     3$              ;YES - SKIP
      7 034144  000137  035120          2$:           JMP     T3265$          ;EXIT TST
      8 034150                          3$:
      9 034150                                        BGNSUB
        034150                                                                                  T15.1:
        034150  104002                                EMT     C$BSUB
     10 034152  012737  006713  002714                MOV     #P2T17E,ERHEAD  ;SET ERROR HEADER
     11 034160  004737  015274                       JSR     PC,TSTINT       ;INITIALIZE TEST
     12 034164  004737  015312                       JSR     PC,GSTATR       ;CLEAR DRIVE
     13 034170  034772                                60$
     14 034172  005037  003014                       CLR     DESHD           ;SET TO HEAD 0
     15 034176  005037  003016                       CLR     DESSEC          ;SET TO SECTOR 0
     16 034202  005037  003004                       CLR     NEWCYL          ;CLEAR TO CYLINDER 0
     17 034206  004737  016204                       JSR     PC,XSEEK        ;DO SEEK
     18 034212  034772                                60$
     19 034214  012701  005670                       MOV     #3000.,R1       ;SET WAIT FOR 300 MS
     20 034220  004737  021026                       JSR     PC,RDYWAIT      ;WAIT FOR READY
     21 034224  034772                                60$
     22 034226  004737  021422                       JSR     PC,VERPOS       ;VERIFY POSITION
     23 034232  034772                                60$
     24 034234  032737  020000  002754                BIT     #WLSTAT,T.MP    ;TEST IF WRITE LOCK SET
     25 034242  001114                                BNE     7$              ;YES - SKIP
     26 034244  004537  022104                       JSR     R5,DATGEN       ;GENERATE DATA
     27 034250  000007                                7                       ;PATTERN 7
     28 034252  004737  022532                       JSR     PC,XWRITE       ;WRITE DATA
     29 034256  034772                                60$
     30 034260  004737  022572                       JSR     PC,XREAD        ;READ DATA
     31 034264  034772                                60$
     32 034266  004737  022244                       JSR     PC,DATCOM       ;CHECK DATA
     33 034272  034772                                60$
     34 034274                                        PRINTF  #FMTOP1,#OPR004,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1> ;REQUEST SET WRT LC
        034274  005046                                CLR     -(SP)
        034276  153716  002735                       BISB    RLDRV+1,(SP)
        034302  012746  006033                       MOV     #DRVNAM,-(SP)
        034306  013746  002730                       MOV     RLBAS,-(SP)
        034312  012746  006022                       MOV     #BASADD,-(SP)
        034316  012746  007312                       MOV     #OPR1A,-(SP)
        034322  012746  007341                       MOV     #OPR004,-(SP)
        034326  012746  010503                       MOV     #FMTOP1,-(SP)
        034332  012746  000007                       MOV     #7,-(SP)
        034336  010600                               MOV     SP,R0
        034340  104017                               EMT     C$PNTF
        034342  062706  000020                       ADD     #20,SP
     35 034346  012701  000006                       MOV     #6.,R1          ;SET WAIT COUNT FOR 30 SECONDS
     36 034352                          5$:          WAITMS  #50.            ;CALL WAIT
        034352  012700  000062                       MOV     #50.,R0
        034356  104026                               EMT     C$WTM
     37 034360  004737  015312                       JSR     PC,GSTATR       ;GET STATUS
     38 034364  034772                                60$
     39 034366  032737  020000  002754                BIT     #WLSTAT,T.MP    ;CHECK IF WRITE LOCK SET
     40 034374  001037                                BNE     7$              ;YES - SKIP
```

```
     41 034376                                  PRINTF  #FMT2,#BELL     ;RING BELL
        034376  012746  010355                  MOV     #BELL,-(SP)
        034402  012746  010611                  MOV     #FMT2,-(SP)
        034406  012746  000002                  MOV     #2,-(SP)
        034412  010600                          MOV     SP,R0
        034414  104017                          EMT     C$PNTF
        034416  062706  000006                  ADD     #6,SP
     42 034422  005301                          DEC     R1              ;DEC COUNT
     43 034424  001352                          BNE     5$              ;SKIP IF NOT 0
     44 034426                                  PRINTF  #FMT23,#P2T17E,#BYPSNM,#OPR1A,<B,RLDRV+1> ;RPT BYPASSED
        034426  005046                          CLR     -(SP)
        034430  153716  002735                  BISB    RLDRV+1,(SP)
        034434  012746  007312                  MOV     #OPR1A,-(SP)
        034440  012746  007415                  MOV     #BYPSNM,-(SP)
        034444  012746  006713                  MOV     #P2T17E,-(SP)
        034450  012746  011420                  MOV     #FMT23,-(SP)
        034454  012746  000005                  MOV     #5,-(SP)
        034460  010600                          MOV     SP,R0
        034462  104017                          EMT     C$PNTF
        034464  062706  000014                  ADD     #14,SP
     45 034470                                  EXIT    TST
        034470  104032                          EMT     C$EXIT
        034472  000426                          .WORD   L10051-.
     46 034474  004537  022104          7$:     JSR     R5,DATGEN       ;GENERATE DATA
     47 034500  000001                          1                       ;PATTERN 1
     48 034502  012705  002736                  MOV     #L.CS,R5        ;GET ADDRESS OF L REGS
     49 034506  012715  000112                  MOV     #WTDATA,(R5)    ;LOAD WRITE COMMAND
     50 034512  053715  002734                  BIS     RLDRV,(R5)      ;INSERT DRIVE NUMBER
     51 034516  042725  002000                  BIC     #BIT10,(R5)+    ;CLEAR FOR DRIVE 4 - 7 SPEC'D
     52 034522  012725  004346                  MOV     #OBUFF,(R5)+    ;LOAD BUS ADDRESS
     53 034526  005025                          CLR     (R5)+           ;CYL 0, HD 0, SECTOR 0
     54 034530  012725  177600                  MOV     #177600,(R5)+   ;128 WORDS
     55 034534  012701  000454                  MOV     #300.,R1        ;SET WAIT COUNT FOR 30 MS
     56 034540  005037  002710                  CLR     DONE            ;CLEAR INTERRUPT FLAG
     57 034544  014562  000006                  MOV     -(R5),RLMP(R2)  ;LOAD RL REGS
     58 034550  014562  000004                  MOV     -(R5),RLDA(R2)
     59 034554  014562  000002                  MOV     -(R5),RLBA(R2)
     60 034560  014562  000000                  MOV     -(R5),RLCS(R2)
     61 034564                          10$:    WAITUS  #1
        034564  012700  000001                  MOV     #1,R0
        034570  104027                          EMT     C$WTU
     62 034572  005737  002710                  TST     DONE            ;CHECK IF INTERRUPT
     63 034576  001012                          BNE     14$             ;YES - SKIP
     64 034600  005301                          DEC     R1              ;DEC WAIT COUNT
     65 034602  001370                          BNE     10$             ;LOOP IF NOT 0
     66 034604  004737  015142                  JSR     PC,WAITIN       ;WAIT FOR INTERRUPT
     67 034610  012603                          MOV     (SP)+,R3        ;GET RESULT MESSAGE
     68 034612                                  ERRHRD  1701.,,ERR1
        034612  104443                          TRAP    T$ERCODE
        034614  003245                          .WORD   1701
        034616  011526                          .WORD   ERR1
     69 034620                                  EXIT    SUB
        034620  104032                          EMT     C$EXIT
        034622  000156                          .WORD   L10052-.
     70 034624  004737  015342          14$:    JSR     PC,GSTAT        ;GET STATUS
     71 034630  034772                          60$
     72 034632  032737  040000  002746          BIT     #DRVERR,T.CS    ;TEST IF ANY ERROR SET
```

```
     034640  001005                          BNE     15$             ;YES - SKIP
     034642  012703  007703                  MOV     #MDRERR,R3      ;SET RESULT MESSAGE POINTER
     034646                                  ERRHRD  1702.,,ERR3     ;REPORT ERROR NOT SET
     034646  104443                          TRAP    T$ERCODE
     034650  003246                          .WORD   1702
     034652  011642                          .WORD   ERR3
     034654  032737  002000  002754  15$:    BIT     #WGESTAT,T.MP   ;TEST IF WGE SET
     034662  001005                          BNE     18$             ;YES - SKIP
     034664  012703  007762                  MOV     #MWGERR,R3      ;SET MESSAGE FOR WGE NOT SET
     034670                                  ERRHRD  1704.,,ERR3
     034670  104443                          TRAP    T$ERCODE
     034672  003250                          .WORD   1704
     034674  011642                          .WORD   ERR3
     034676  042737  040000  002746  18$:    BIC     #DRVERR,T.CS    ;CLEAR DRIVE ERROR BIT
     034704  042737  002000  002754          BIC     #WGESTAT,T.MP   ;CLEAR WGE BIT
     034712  032737  157400  002754          BIT     #157400,T.MP    ;TEST IF ANY OTHER ERRORS
     034720  001004                          BNE     16$             ;YES - GO REPORT
     034722  032737  036000  002746          BIT     #36000,T.CS     ;TEST ANY ERRORS IN CS REG
     034730  001404                          BEQ     17$             ;NO - SKIP
     034732                          16$:    ERRHRD  1703.,,ERR6     ;REPORT ERRORS
     034732  104443                          TRAP    T$ERCODE
     034734  003247                          .WORD   1703
     034736  012030                          .WORD   ERR6
     034740  000414                          BR      60$             ;EXIT TEST
     034742  004737  015312          17$:    JSR     PC,GSTATR       ;GET STATUS AND RESET ERROR
     034746  034772                          60$
     034750  004537  022104                  JSR     R5,DATGEN       ;GO GENERATE DATA
     034754  000007                          7                       ;PATTERN 7
     034756  004737  022572                  JSR     PC,XREAD        ;READ DATA
     034762  034772                          60$
     034764  004737  022244                  JSR     PC,DATCOM       ;COMPARE DATA
     034770  034772                          60$
     034772  012737  000002  002720  60$:    MOV     #2,ERRSWI       ;INIT ERROR SWITCH
     035000                          ENDSUB
     035000                          L10052:
     035000  104003                          EMT     C$ESUB
     035002  012737  000002  002720  T3204$: MOV     #2,ERRSWI       ;INIT ERROR SWITCH
     035010                                  PRINTF  #FMTOP1,#OPR12,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1> ;REQ RESET WRT LCK
     035010  005046                          CLR     -(SP)
     035012  153716  002735                  BISB    RLDRV+1,(SP)
     035016  012746  006033                  MOV     #DRVNAM,-(SP)
     035022  013746  002730                  MOV     RLBAS,-(SP)
     035026  012746  006022                  MOV     #BASADD,-(SP)
     035032  012746  007312                  MOV     #OPR1A,-(SP)
     035036  012746  007273                  MOV     #OPR12,-(SP)
     035042  012746  010503                  MOV     #FMTOP1,-(SP)
     035046  012746  000007                  MOV     #7,-(SP)
     035052  010600                          MOV     SP,R0
     035054  104017                          EMT     C$PNTF
     035056  062706  000020                  ADD     #20,SP
     035062  012701  000454                  MOV     #300.,R1        ;SET WAIT FOR 30 SEC
     035066                          16$:    WAITMS  #1
     035066  012700  000001                  MOV     #1,R0
     035072  104026                          EMT     C$WTM
     035074  004737  015312                  JSR     PC,GSTATR       ;GET STATUS
     035100  035002                          T3204$
     035102  032737  020000  002754          BIT     #WLSTAT,T.MP    ;CHECK IF WRITE LOCK RESET
```

```
    105 035110  001403                        BEQ     T3265$
    106 035112  005301                        DEC     R1              ;DEC WAIT COUNT
    107 035114  001364                        BNE     16$             ;LOOP IF NOT 0
    108 035116  000731                        BR      T3204$          ;ELSE REPEAT MESSAGE
    109 035120                          T3265$:
    110 035120                          ENDTST
        035120                          L10051:
        035120  104001                        EMT     C$ETST
    111
```

```
                                .SBTTL  *TEST 16         **ADJACENT CYLINDER INTERFERENCE
035122                          BGNTST                   ;TEST 16
035122                                                                  T16::
035122  012737  006745  002714          MOV     #P2T18E,ERHEAD  ;SET ERROR HEADER
035130  004737  017656                  JSR     PC,CKBSVD       ;GO CHECK IF BAD SECTOR FILES VALID
035134  004737  015274                  JSR     PC,TSTINT       ;INITIALIZE TEST
035140  004737  015312 .                JSR     PC,GSTATR       ;CLEAR DRIVE
035144  036334                          T3365$
035146  005037  003134                  CLR     PASCNT          ;CLEAR PASS TO 0
035152  012705  177776                  MOV     #-2,R5          ;SET R5
035156  005737  003342                  TST     PASNUM          ;TEST IF FIRST PASS (QUICK VERIFY)
035162  001007                          BNE     1$              ;NO - SKIP
035164  032737  000001  013346          BIT     #ALLCYL,MISWIW  ;TEST IF USE ALL CYLINDERS
035172  001003                          BNE     1$              ;YES - SKIP
035174  012705  177730                  MOV     #-40.,R5                ;ELSE SET R5 TO NEG 20
035200  000402                          BR      9$              ;SKIP
035202  012705  177770          1$:     MOV     #-10,R5         ;ELSE SET FOR NEG 4
035206  01 701  002406          9$:     MOV     #T33TBL,R1      ;GET ADDRESS OF WORK TABLE
035212  012737  000010  002202          MOV     #10,JJJ         ;SET CLEAR COUNT
035220  013721  013350          2$:     MOV     LOLIMW,(R1)+    ;CLEAR LOCATIONS TO LOLIMIT
035224  005337  002202                  DEC     JJJ             ;DEC COUNT
035230  001373                                  BNE     2$              ;LOOP UNTIL 0
035232  004537  022104                  JSR     R5,DATGEN       ;GO GENERATE DATA
035236  000011                          9.                      ;PATTERN 9
035240  013737  013352  002410          MOV     HILIMW,T33TBL+2 ;INSERT HILIMIT
035246  013737  013352  002412          MOV     HILIMW,T33TBL+4 ;INTO APPROPRIATE LOCATIONS
035254  013737  013352  002416          MOV     HILIMW,T33TBL+10
035262  013737  013352  002424          MOV     HILIMW,T33TBL+16
035270  062705  000002          T3300$: ADD     #2,R5

035274  032737  000001  013346          BIT     #ALLCYL,MISWIW  ;TEST IF USE ALL CYLINDERS
035302  001034                          BNE     5$              ;YES - SKIP
035304  005737  003342                  TST     PASNUM          ;TEST IF FIRST PASS (QUICK VERIFY)
035310  001403                          BEQ     3$              ;NO - SKIP
035312  062705  000006                  ADD     #6,R5           ;ELSE BUMP CYLINDER POINTER BY 3
035316  000402                          BR      6$              ;SKIP
035320  062705  000044          3$:     ADD     #36.,R5         ;BUMP TO NEXT ENTRY
035324  022737  000001  002200  6$:     CMP     #1,T.DRIVE
035332  001404                          BEQ     44$
035334  020537  000244                  CMP     R5,164.
035340  103013                          BHIS    4$
035342  000403                          BR      69$

035344  020527  000122          44$:    CMP     R5,#82.
035350  103007                          BHIS    4$

035352  016537  002506  002202  69$:    MOV     CYLTBL(R5),JJJ
035360  043737  002206  002202          BIC     CLRBYT,JJJ
035366  001013                          BNE     8$
035370  000137  034126          4$:     JMP     T3165$
035374  005705                  5$:     TST     R5              ;TEST IF R5 0
035376  001002                          BNE     7$              ;NO - SKIP
035400  062705  000002                  ADD     #2,R5
035404  023705  002204          7$:     CMP     HLMTW,R5        ;TEST IF ALL CYLINDERS USED
035410  001767                          BEQ     4$              ;YES - EXIT TEST
035412  010537  002202                  MOV     R5,JJJ          ;USE R5 AS NEXT CYLINDER
035416  023737  002202  013350  8$:     CMP     JJJ,LOLIMW      ;CHECK IF LOWER THAN LOLIMIT
```

```
     57 035424  103721                          BLO     T3300$          ;YES - SKIP
     58 035426  023737  002202  013352          CMP     JJJ,HILIMW      ;CHECK IF HIGHER THAN HILIMIT
     59 035434  101315                          BHI     T3300$          ;YES - SKIP
     60 035436  012703  002446                  MOV     #TBT,R3
     61 035442  013713  002202                  MOV     JJJ,(R3)
     62 035446  013763  002202  000006          MOV     JJJ,6(R3)
     63 035454  013763  002202  000010          MOV     JJJ,10(R3)
     64 035462  013763  002202  000012          MOV     JJJ,12(R3)
     65 035470  013763  002202  000016          MOV     JJJ,16(R3)
     66 035476  162737  000001  002202          SUB     #1,JJJ
     67 035504  013763  002202  000002          MOV     JJJ,2(R3)
     68 035512  013763  002202  000012          MOV     JJJ,12(R3)
     69 035520  062737  000002  002202          ADD     #2,JJJ
     70 035526  013763  002202  000004          MOV     JJJ,4(R3)
     71 035534  013763  002202  000014          MOV     JJJ,14(R3)
     72 035542  010337  002726                  MOV     R3,TBLSTR
     73 035546  004737  017546                  JSR     PC,CHOSHD       ;GO CHOSE HEAD
     74 035552                          T3301$:
     75 035552                          BGNSUB
        035552                                                                  T16.1:
        035552  104002                          EMT     C$BSUB
     76 035554  042737  003760  002706          BIC     #MQUALS,OPFLAG  ;CLEAR ALL MESSAGE QUALIFIERS
     77 035562  005737  003134                  TST     PASCNT          ;TEST IF PASS 0
     78 035566  001414                          BEQ     11$             ;YES - SKIP
     79 035570  023727  003134  000004          CMP     PASCNT,#4       ;TEST IF PASS 4
     80 035576  001404                          BEQ     10$             ;YES - SKIP
     81 035600  002407                          BLT     11$             ;CHECK IF LESS THAN 4, IF YES CLEAR TO 0
     82 035602  012737  000004  003134          MOV     #4,PASCNT       ;ELSE SET TO 4
     83 035610  052737  000020  002706  10$:     BIS     #INOUTS,OPFLAG  ;SET MESSAGE QUAL
     84 035616  000405                          BR      12$             ;SKIP
     85 035620  005037  003134          11$:     CLR     PASCNT          ;SET PASS COUNT TO 0
     86 035624  052737  000040  002706          BIS     #OUTINS,OPFLAG  ;SET MESSAGE QUAL
     87 035632  012737  000003  002724  12$:     MOV     #3,WRTSWI       ;SET READ AND WRITE SWITCH
     88 035640  012701  002406                  MOV     #T33TBL,R1
     89 035644  012703  002446                  MOV     #TBT,R3
     90 035650  005037  003016          15$:     CLR     DESSEC          ;CLEAR TO SECTOR 0
     91 035654  012137  003004                  MOV     (R1)+,NEWCYL    ;GET NEXT TABLE ENTRY
     92 035660  004737  016204                  JSR     PC,XSEEK        ;DO SEEK
     93 035664  036242                          60$
     94 035666  012701  005670                  MOV     #3000.,R1       ;SET WAIT COUNT FOR 300 MS
     95 035672  004737  021026                  JSR     PC,RDYWAIT      ;WAIT FOR READY
     96 035676  036242                          60$
     97 035700  012337  003004                  MOV     (R3)+,NEWCYL    ;GET NEXT TABLE ENTRY
     98 035704  004737  016204                  JSR     PC,XSEEK        ;DO SEEK
     99 035710  036242                          60$
    100 035712  012701  005670                  MOV     #3000.,R1       ;SET WAIT COUNT FOR 300 MS
    101 035716  004737  021026                  JSR     PC,RDYWAIT      ;WAIT FOR READY
    102 035722  036242                          60$
    103 035724  004737  021422                  JSR     PC,VERPOS       ;VERIFY POSITION
    104 035730  036242                          60$
    105 035732  004737  023252          16$:     JSR     PC,BSCHK        ;CHECK FOR BAD SECTOR
    106 035736  036046                          32$                     ;"YES" RETURN
    107 035740  032737  000001  002724          BIT     #BIT0,WRTSWI    ;TEST IF WRITE THIS PASS
    108 035746  001425                          BEQ     29$             ;NO - SKIP
    109 035750  004737  022532                  JSR     PC,XWRITE       ;DO WRITE
    110 035754  036242                          60$
    111 035756  005237  003016                  INC     DESSEC          ;INC SECTOR
```

```
    112 035762  022737  000050  003016          CMP     #40.,DESSEC     ;TEST IF ALL SECTORS USED
    113 035770  001360                          BNE     16$             ;NO - SKIP
    114 035772  042737  000060  002706          BIC     #INOUTS!OUTINS,OPFLAG ;CLEAR QUALIFIERS
    115 036000  042737  000001  002724          BIC     #BIT0,WRTSWI    ;CLEAR WRITE REQUIRED SWITCH
    116 036006  052737  000100  002706          BIS     #FOLWRT,OPFLAG  ;SET FOLLOWING WRITE QUALIFIER
    117 036014  005037  003016                  CLR     DESSEC          ;CLEAR TO SECTOR 0
    118 036020  000744                          BR      16$             ;SKIP
    119 036022  032737  000002  002724  29$:    BIT     #BIT1,WRTSWI    ;TEST IF READ THIS PASS
    120 036030  001414                          BEQ     33$             ;NO - SKIP
    121 036032  004737  022572          31$:    JSR     PC,XREAD        ;ELSE DO READ
    122 036036  036242                          60$
    123 036040  004737  022244                  JSR     PC,DATCOM       ;COMPARE DATA
    124 036044  036242                          60$
    125 036046  005237  003016          32$:    INC     DESSEC          ;BUMP SECTOR
    126 036052  022737  000050  003016          CMP     #40.,DESSEC     ;TEST IF ALL SECTORS USED
    127 036060  001324                          BNE     16$             ;NO - LOOP
    128 036062  005037  003016          33$:    CLR     DESSEC          ;CLEAR DESIRED SECTOR
    129 036066  005037  002724                  CLR     WRTSWI          ;CLEAR WRITE/READ SWITCH
    130 036072  005237  003134                  INC     PASCNT          ;BUMP PASS COUNT
    131 036076  042737  003760  002706          BIC     #MQUALS,OPFLAG  ;CLEAR ALL QUALIFIERS
    132 036104  023727  003134  000004          CMP     PASCNT,#4       ;TEST IS PASS 4
    133 036112  001453                          BEQ     60$             ;YES - SKIP
    134 036114  023727  003134  000010          CMP     PASCNT,#8.      ;TEST IF PASS 8.
    135 036122  001447                          BEQ     60$             ;YES - SKIP
    136 036124  023727  003134  000003          CMP     PASCNT,#3       ;TEST IF PASS 3
    137 036132  001430                          BEQ     39$             ;YES - SKIP
    138 036134  023727  003134  000007          CMP     PASCNT,#7       ;TEST IF PASS 7
    139 036142  001430                          BEQ     40$             ;YES - SKIP
    140 036144  012737  000001  002724          MOV     #BIT0,WRTSWI    ;SET WRITE REQUIRED
    141 036152  023727  003134  000001          CMP     PASCNT,#1       ;TEST IF PASS 1
    142 036160  001411                          BEQ     37$             ;YES - SKIP
    143 036162  023727  003134  000002          CMP     PASCNT,#2       ;TEST IF PASS 2
    144 036170  001405                          BEQ     37$             ;YES - SKIP
    145 036172  052737  000040  002706          BIS     #OUTINS,OPFLAG  ;SET MESSAGE QUALIFIER
    146 036200  000137  035650          36$:    JMP     15$             ;GO DO NEXT PASS
    147 036204  052737  000020  002706  37$:    BIS     #INOUTS,OPFLAG  ;SET MESSAGE QUALIFIER
    148 036212  000772                          BR      36$
    149 036214  052737  000200  002706  39$:    BIS     #REVSKS,OPFLAG  ;SET MESSAGE QUALIFIER
    150 036222  000403                          BR      41$
    151 036224  052737  000400  002706  40$:    BIS     #FWDSKS,OPFLAG  ;SET MESSAGE QUALIFIER
    152 036232  012737  000002  002724  41$:    MOV     #BIT1,WRTSWI    ;SET READ REQUIRED
    153 036240  000757                          BR      36$
    154 036242  012737  000002  002720  60$:    MOV     #2,ERRSWI       ;INIT ERROR SWITCH
    155 036250                          ENDSUB
        036250                          L10054:
        036250  104003                          EMT     C$ESUB
    156 036252                                  ESCAPE  TST             ;EXIT TEST IF ERROR
        036252  104010                          EMT     C$ESCAPE
        036254  000060                          .WORD   L10053-.
    157 036256  012737  000003  002724          MOV     #3,WRTSWI       ;SET FOR READ AND WRITE REQ.
    158 036264  023727  003134  000004          CMP     PASCNT,#4       ;TEST IF PASS 4
    159 036272  001004                          BNE     45$             ;NO - SKIP
    160 036274  012737  002416  002726          MOV     #T33TBL+10,TBLSTR ;STORE MID POINT IN TABLE
    161 036302  000410                          BR      48$             ;GO START PASS 4
    162 036304  005037  003134          45$:    CLR     PASCNT          ;CLEAR TO PASS 0
    163 036310  004737  017572                  JSR     PC,SWAPHD       ;GO SWAP TO HEAD 1 OR END TEST
    164 036314  035270                          T3300$                  ;ABORT RETURN
```

```
    165 036316  012737  002406  002726          MOV     #T33TBL,TBLSTR  ;STORE START OF TABLE
    166
    167 036324  062703  000010          48$:    ADD     #10,R3
    168 036330  000137  035552                  JMP     T3301$
    169 036334                          T3365$:
    170 036334                          ENDTST
        036334                          L10053:
        036334  104001                          EMT     C$ETST
```

```
                                        .SBTTL  *TEST 17         **OVERWRITE
036336                                  BGNTST                   ;TEST 17
036336                                                                        T17::
036336  012737  006772  002714                MOV     #P2T19E,ERHEAD  ;SET ERROR HEADER
036344  004737  017656                        JSR     PC,CKBSVD       ;GO CHECK IF BAD SECTOR FILES VALID
036350  004737  015274                        JSR     PC,TSTINT       ;INITIALIZE TEST
036354  004737  015312                        JSR     PC,GSTATR       ;CLEAR DRIVE
036360  037526                              T3465$
036362  005037  003134                        CLR     PASCNT          ;CLEAR PASS TO 0
036366  012705  177776                        MOV     #-2,R5          ;SET R5
036372  005737  003342                        TST     PASNUM          ;TEST IF FIRST PASS (QUICK VERIFY)
036376  001007                              BNE     1$              ;NO - SKIP
036400  032737  000001  013346                BIT     #ALLCYL,MISWIW  ;TEST IF USE ALL CYLINDERS
036406  001003                              BNE     1$              ;YES - SKIP
036410  012705  177730                        MOV     #-40.,R5        ;ELSE SET R5 TO NEG 20
036414  000402                              BR      9$              ;SKIP
036416  012705  177770                1$:     MOV     #-10,R5         ;SET FOR NEXT ENTRY
036422  012701  002406                9$:     MOV     #T33TBL,R1      ;GET ADDRESS OF WORK TABLE
036426  012737  000010  002202                MOV     #10,JJJ         ;SET CLEAR COUNT
036434  013721  013350                2$:     MOV     LOLIMW,(R1)+    ;CLEAR LOCATIONS TO LOLIMIT
036440  005337  002202                        DEC     JJJ             ;DEC COUNT
036444  001373                              BNE     2$              ;LOOP UNTIL 0
036446  013737  013352  002410                MOV     HILIMW,T33TBL+2 ;INSERT HILIMIT
036454  013737  013352  002414                MOV     HILIMW,T33TBL+6 ;INTO APPROPRIATE LOCATIONS
036462  013737  013352  002420                MOV     HILIMW,T33TBL+12
036470  062705  000002              T3400$:   ADD     #2,R5
036474  032737  000001  013346                BIT     #ALLCYL,MISWIW  ;TEST IF USE ALL CYLINDERS
036502  001034                              BNE     5$              ;YES - SKIP
036504  005737  003342                        TST     PASNUM          ;TEST IF FIRST PASS (QUICK VERIFY)
036510  001003                              BNE     3$              ;NO - SKIP
036512  062705  000046                        ADD     #38.,R5         ;ELSE BUMP CYLINDER POINTER BY 19
036516  000402                              BR      6$              ;SKIP
036520  062705  000006                3$:     ADD     #6,R5           ;BUMP CYLINDER POINTER BY 3
036524  022737  000001  002200        6$:     CMP     #1,T.DRIVE
036532  001404                              BEQ     444$
036534  020527  000244                        CMP     R5,#164.
036540  103013                              BHIS    4$              ;FOR INF. SEE TEST 18
036542  000403                              BR      669$
036544  020527  000122                444$:   CMP     R5,#82.
036550  103007                              BHIS    4$
036552  016537  002506  002202        669$:   MOV     CYLTBL(R5),JJJ
036560  043737  002206  002202                BIC     CLRBYT,JJJ
036566  001013                              BNE     8$
036570  000137  037526                4$:     JMP     T3465$          ;EXIT TEST
036574  005705                          5$:     TST     R5              ;TEST IF R5 0
036576  001002                              BNE     7$              ;NO - SKIP
036600  062705  000002                        ADD     #2,R5
036604  022705  002204                7$:     CMP     #HLMTW,R5       ;TEST IF ALL CYLINDERS USED
036610  001767                              BEQ     4$              ;YES - EXIT TEST
036612  010537  002202                        MOV     R5,JJJ          ;USE R5 AS NEXT CYLINDER
036616  023737  002202  013350        8$:     CMP     JJJ,LOLIMW      ;TEST IF PAST LO LIMIT
036624  103721                              BLO     T3400$          ;YES - SKIP
036626  023737  002202  013352              CMP     JJJ,HILIMW      ;TEST IF PAST HILIMIT
036634  101315                              BHI     T3400$          ;YES - SKIP
036636  012703  002446                        MOV     #TBT,R3
036642  013713  002202                        MOV     JJJ,(R3)
036646  013763  002202  000002              MOV     JJJ,2(R3)
```

```
     57 036654  013763  002202  000004          MOV     JJJ,4(R3)
     58 036662  013763  002202  000006          MOV     JJJ,6(R3)
     59 036670  013763  002202  000010          MOV     JJJ,10(R3)
     60 036676  013763  002202  000012          MOV     JJJ,12(R3)
     61 036704  010337  002726                   MOV     R3,TBLSTR
     62 036710  004737  017546                   JSR     PC,CHOSHD       ;GO CHOSE HEAD
     63 036714                          T3401$:
     64 036714                          BGNSUB
        036714                                                                  T17.1:
        036714  104002                          EMT     C$BSUB
     65 036716  042737  003760  002706          BIC     #MQUALS,OPFLAG  ;CLEAR ALL MESSAGE QUALIFIERS
     66 036724  005737  003134                   TST     PASCNT          ;TEST IF PASS 0
     67 036730  001414                           BEQ     11$             ;YES - SKIP
     68 036732  023727  003134  000003          CMP     PASCNT,#3       ;TEST IF PASS 3
     69 036740  001404                           BEQ     10$             ;YES - SKIP
     70 036742  002407                           BLT     11$             ;CHECK IF LESS THAN 3, IF YES CLEAR TO 0
     71 036744  012737  000003  003134          MOV     #3,PASCNT       ;ELSE SET TO 3
     72 036752  052737  000020  002706  10$:     BIS     #INOUTS,OPFLAG  ;SET MESSAGE QUAL
     73 036760  000405                           BR      12$             ;SKIP
     74 036762  005037  003134          11$:     CLR     PASCNT          ;SET PASS COUNT TO 0
     75 036766  052737  000040  002706          BIS     #OUTINS,OPFLAG  ;SET MESSAGE QUAL
     76 036774  012737  000003  002724  12$:     MOV     #3,WRTSWI       ;SET READ AND WRITE SWITCH
     77 037002  012701  002406                   MOV     #T33TBL,R1
     78 037006  012703  002446                   MOV     #TBT,R3
     79 037012  005037  003016          15$:     CLR     DESSEC
     80 037016  012137  003004                   MOV     (R1)+,NEWCYL    ;GET NEXT TABLE ENTRY
     81 037022  004737  016204                   JSR     PC,XSEEK        ;DO SEEK
     82 037026  037434                           60$
     83 037030  012701  005670                   MOV     #3000.,R1       ;SET WAIT COUNT FOR 300 MS
     84 037034  004737  021026                   JSR     PC,RDYWAIT      ;WAIT FOR READY
     85 037040  037434                           60$
     86 037042  012337  003004                   MOV     (R3)+,NEWCYL    ;GET NEXT TABLE ENTRY
     87 037046  004737  016204                   JSR     PC,XSEEK        ;DO SEEK
     88 037052  037434                           60$
     89 037054  012701  005670                   MOV     #3000.,R1       ;SET WAIT COUNT FOR 300 MS
     90 037060  004737  021026                   JSR     PC,RDYWAIT      ;WAIT FOR READY
     91 037064  037434                           60$
     92 037066  004737  021422                   JSR     PC,VERPOS       ;VERIFY POSITION
     93 037072  037434                           60$
     94 037074  004737  023252          16$:     JSR     PC,BSCHK        ;CHECK FOR BAD SECTOR
     95 037100  037250                           32$                     ;"YES" RETURN
     96 037102  005737  003134                   TST     PASCNT          ;TEST IF PASS 0
     97 037106  001407                           BEQ     17$             ;YES - SKIP
     98 037110  022737  000003  003134          CMP     #3,PASCNT       ;TEST IF PASS 3
     99 037116  001403                           BEQ     17$             ;YES - SKIP
    100 037120  005037  037140                   CLR     25$             ;ELSE CLEAR DATA PATTERN SELECTOR
    101 037124  000403                           BR      18$
    102 037126  012737  000010  037140  17$:     MOV     #8.,25$         ;SET DATA PATTERN SELECTOR TO 8
    103 037134  004537  022104          18$:     JSR     R5,DATGEN       ;GO GENERATE DATA
    104 037140  000000                  25$:     .WORD   0
    105 037142  032737  000001  002724          BIT     #BIT0,WRTSWI    ;TEST IF WRITE THIS PASS
    106 037150  001425                           BEQ     29$             ;NO - SKIP
    107 037152  004737  022532                   JSR     PC,XWRITE       ;DO WRITE
    108 037156  037434                           60$
    109 037160  005237  003016                   INC     DESSEC          ;INC SECTOR
    110 037164  022737  000050  003016          CMP     #40.,DESSEC     ;TEST IF ALL SECTORS USED
    111 037172  001340                           BNE     16$             ;NO - SKIP
```

```
    112 037174  042737  000060  002706          BIC     #INOUTS!OUTINS,OPFLAG ;CLEAR QUALIFIERS
    113 037202  042737  000001  002724          BIC     #BIT0,WRTSWI    ;CLEAR WRITE REQUIRED SWITCH
    114 037210  052737  000100  002706          BIS     #FOLWRT,OPFLAG  ;SET FOLLOWING WRITE QUALIFIER
    115 037216  005037  003016                  CLR     DESSEC          ;CLEAR TO SECTOR 0
    116 037222  000724                          BR      16$             ;SKIP
    117 037224  032737  000002  002724  29$:    BIT     #BIT1,WRTSWI    ;TEST IF READ THIS PASS
    118 037232  001414                          BEQ     33$             ;NO - SKIP
    119 037234  004737  022572          31$:    JSR     PC,XREAD        ;ELSE DO READ
    120 037240  037434                          60$
    121 037242  004737  022244                  JSR     PC,DATCOM       ;COMPARE DATA
    122 037246  037434                          60$
    123 037250  005237  003016          32$:    INC     DESSEC          ;BUMP SECTOR
    124 037254  022737  000050  003016          CMP     #40.,DESSEC     ;TEST IF ALL SECTORS USED
    125 037262  001304                          BNE     16$             ;NO - LOOP
    126 037264  005037  003016          33$:    CLR     DESSEC          ;CLEAR DESIRED SECTOR
    127 037270  005037  002724                  CLR     WRTSWI          ;CLEAR WRITE/READ SWITCH
    128 037274  005237  003134                  INC     PASCNT          ;BUMP PASS COUNT
    129 037300  042737  00376C  002706          BIC     #MQUALS,OPFLAG  ;CLEAR ALL QUALIFIERS
    130 037306  023727  003134  000003          CMP     PASCNT,#3       ;TEST IS PASS 3
    131 037314  001447                          BEQ     60$             ;YES - SKIP
    132 037316  023727  003134  000006          CMP     PASCNT,#6       ;TEST IF PASS 6
    133 037324  001443                          BEQ     60$             ;YES - SKIP
    134 037326  023727  003134  000001          CMP     PASCNT,#1       ;TEST IF PASS 1
    135 037334  001424                          BEQ     39$             ;YES - SKIP
    136 037336  023727  003134  000004          CMP     PASCNT,#4       ;TEST IF PASS 4
    137 037344  001424                          BEQ     40$             ;YES - SKIP
    138 037346  012737  000002  002724          MOV     #BIT1,WRTSWI    ;SET WRITE REQUIRED BIT
    139 037354  023727  003134  000002          CMP     PASCNT,#2       ;TEST IF PASS 2
    140 037362  001405                          BEQ     37$             ;YES - SKIP
    141 037364  052737  001000  002706          BIS     #REVSKO,OPFLAG  ;SET REVERSE QUALIFIER
    142 037372  000137  037012          36$:    JMP     15$             ;GO DO NEXT PASS
    143 037376  052737  002000  002706  37$:    BIS     #FWDSKO,OPFLAG  ;SET FWD QUALIFIER
    144 037404  000772                          BR      36$             ;GO DO NEXT PASS
    145 037406  052737  000020  002706  39$:    BIS     #INOUTS,OPFLAG  ;SET QUALIFIER
    146 037414  000403                          BR      41$             ;SKIP
    147 037416  052737  000040  002706  40$:    BIS     #OUTINS,OPFLAG  ;SET MESSAGE QUALIFIER
    148 037424  012737  000001  002724  41$:    MOV     #BIT0,WRTSWI    ;SET WRITE REQUIRED BIT
    149 037432  000757                          BR      36$             ;GO DO NEXT PASS
    150 037434  012737  000002  002720  60$:    MOV     #2,ERRSWI       ;INIT ERROR SWITCH
    151 037442                          ENDSUB
        037442                          L10056:
        037442  104003                          EMT     C$ESUB
    152 037444                          ESCAPE  TST                     ;EXIT TEST IF ERROR
        037444  104010                          EMT     C$ESCAPE
        037446  000060                          .WORD   L10055-.
    153 037450  012737  000003  002724          MOV     #3,WRTSWI       ;SET FOR READ AND WRITE REQ.
    154 037456  023727  003134  000003          CMP     PASCNT,#3       ;TEST IF PASS 3
    155 037464  001004                          BNE     45$             ;NO - SKIP
    156 037466  012737  002414  002726          MOV     #T33TBL+6,TBLSTR ;STORE MID POINT IN TABLE
    157 037474  000410                          BR      48$             ;GO START PASS 4
    158 037476  005037  003134          45$:    CLR     PASCNT          ;CLEAR TO PASS 0
    159 037502  004737  017572                  JSR     PC,SWAPHD       ;GO SWAP TO HEAD ONE OR ABORT TEST
    160 037506  036470                          T3400$                  ;ABORT RETURN
    161 037510  012737  002406  002726          MOV     #T33TBL,TBLSTR  ;STORE START OF TABLE
    162 037516  062703  000006          48$:    ADD     #6,R3
    163 037522  000137  036714                  JMP     T3401$
    164 037526                          T3465$:
```

```
    165 037526                          ENDTST
        037526                          L10055:
        037526  104001                          EMT     C$ETST
    166 037530                          ENDMOD
```

```
                                .SBTTL  PARAMETER CODING
037530                          BGNMOD  HRDPRM
037530                          BGNHRD
037530  000030                          .WORD L10057-L$HARD/2
037532                          GPRML   CNTYPE,CNT,1,YES
037532  005130                          .WORD   T$CODE
037534  037666                          .WORD   CNTYPE
037536  000001                          .WORD   1
037540                          GPRMA   CSRMSG,CSR,0,160000,177776,YES
037540  000031                          .WORD   T$CODE
037542  037612                          .WORD   CSRMSG
037544  160000                          .WORD   T$LOLIM
037546  177776                          .WORD   T$HILIM
037550                          GPRMA   VECMSG,VECT,0,0,776,YES
037550  001031                          .WORD   T$CODE
037552  037623                          .WORD   VECMSG
037554  000000                          .WORD   T$LOLIM
037556  000776                          .WORD   T$HILIM
037560                          GPRMD   DRMSG,DRSB,0,3400,0,7,YES
037560  004032                          .WORD   T$CODE
037562  037660                          .WORD   DRMSG
037564  003400                          .WORD   3400
037566  000000                          .WORD   T$LOLIM
037570  000007                          .WORD   T$HILIM
037572                          GPRML   DRTYPE,TYPDR,1,YES
037572  003130                          .WORD   T$CODE
037574  037643                          .WORD   DRTYPE
037576  000001                          .WORD   1
037600                          GPRMD   BRMSG,PRIOR,0,340,0,7,YES
037600  002032                          .WORD   T$CODE
037602  037632                          .WORD   BRMSG
037604  000340                          .WORD   340
037606  000000                          .WORD   T$LOLIM
037610  000007                          .WORD   T$HILIM

037612                          ENDHRD
                                .EVEN
037612                          L10057:

037612  102  125  123          CSRMSG: .ASCIZ  /BUS ADDR/
037615  040  101  104
037620  104  122  000
037623  126  105  103          VECMSG: .ASCIZ  /VECTOR/
037626  124  117  122
037631  000
037632  102  122  040          BRMSG:  .ASCIZ  /BR LEVEL/
037635  114  105  126
037640  105  114  000
037643  104  122  111          DRTYPE: .ASCIZ  /DRIVE = RL01/
037646  126  105  040
037651  075  040  122
037654  114  060  061
037657  000
037660  104  122  111          DRMSG:  .ASCIZ  /DRIVE/
037663  126  105  000
037666  122  114  061          CNTYPE: .ASCIZ  /RL11/
037671  061  000
```

```
     19 037673                          ENDMOD
     20                                         .EVEN
     21
     22 037674                          BGNMOD  SFTPRM
     23 037674                          BGNSFT
        037674  000061                          .WORD L10060-L$SOFT/2
     24
     26 037676                          GPRML   CYLQ,MISWI,1,YES
        037676  000130                          .WORD   T$CODE
        037700  040040                          .WORD   CYLQ
        037702  000001                          .WORD   1
     27 037704                          GPRML   SECQ,MISWI,2,YES
        037704  000130                          .WORD   T$CODE
        037706  040054                          .WORD   SECQ
        037710  000002                          .WORD   2
     33 037712                          GPRML   MANQ,MISWI,100000,YES
        037712  000130                          .WORD   T$CODE
        037714  040071                          .WORD   MANQ
        037716  100000                          .WORD   100000
     34
     36 037720                          GPRML   LOLIMQ,MISWI,40000,YES
        037720  000130                          .WORD   T$CODE
        037722  040126                          .WORD   LOLIMQ
        037724  040000                          .WORD   40000
     37 037726                                  XFERF   1$
        037726  006044                          .WORD   T$CODE
     38 037730                                  GPRMD   LIMVAL,LOLIM,D,255.,0,253.,YES
        037730  001052                          .WORD   T$CODE
        037732  040145                          .WORD   LIMVAL
        037734  000377                          .WORD   255.
        037736  000000                          .WORD   T$LOLIM
        037740  000375                          .WORD   T$HILIM
     39 037742                          1$:     GPRML   HILIMQ,MISWI,20000,YES
        037742  000130                          .WORD   T$CODE
        037744  040153                          .WORD   HILIMQ
        037746  020000                          .WORD   20000
     40 037750                                  XFERF   2$
        037750  006044                          .WORD   T$CODE
     41 037752                                  GPRMD   LIMVAL,HILIM,D,255.,0,255.,YES
        037752  002052                          .WORD   T$CODE
        037754  040145                          .WORD   LIMVAL
        037756  000377                          .WORD   255.
        037760  000000                          .WORD   T$LOLIM
        037762  000377                          .WORD   T$HILIM
     42 037764                          2$:     GPRML   HEADQ,MISWI,10000,YES
        037764  000130                          .WORD   T$CODE
        037766  040174                          .WORD   HEADQ
        037770  010000                          .WORD   10000
     43 037772                                  XFERF   3$
        037772  006044                          .WORD   T$CODE
     44 037774                                  GPRMD   HEADV,HEAD,D,17,0,1,YES
        037774  003052                          .WORD   T$CODE
        037776  040216                          .WORD   HEADV
        040000  000017                          .WORD   17
        040002  000000                          .WORD   T$LOLIM
        040004  000001                          .WORD   T$HILIM
     46 040006                          3$:     GPRMD   ERLIMQ,ERLIM,D,377,0,377,YES
```

```
            040006  004052                          .WORD   T$CODE
            040010  040241                          .WORD   ERLIMQ
            040012  000377                          .WORD   377
            040014  000000                          .WORD   T$LOLIM
            040016  000377                          .WORD   T$HILIM
     48     040020                                  GPRMD   DCLIMQ,DCLIM,D,377,1,377,YES
            040020  005052                          .WORD   T$CODE
            040022  040313                          .WORD   DCLIMQ
            040024  000377                          .WORD   377
            040026  000001                          .WORD   T$LOLIM
            040030  000377                          .WORD   T$HILIM
     50     040032                          GPRML   AUTOQ,MISWI,20,YES
            040032  000130                          .WORD   T$CODE
            040034  040263                          .WORD   AUTOQ
            040036  000020                          .WORD   20
     51     040040                          ENDSFT
                                                    .EVEN
            040040                          L10060:
     52
     54     040040      125     123     105 CYLQ:   .ASCIZ  /USE ALL CYL/
            040043      040     101     114
            040046      114     040     103
            040051      131     114     000
     55     040054      125     123     105 SECQ:   .ASCIZ  /USE ALL SECT/
            040057      040     101     114
            040062      114     040     123
            040065      105     103     124
            040070      000
     61     040071      104     117     040 MANQ:   .ASCIZ  /DO MANUAL INTERVENTION TESTS/
            040074      115     101     116
            040077      125     101     114
            040102      040     111     116
            040105      124     105     122
            040110      126     105     116
            040113      124     111     117
            040116      116     040     124
            040121      105     123     124
            040124      123     000
     63     040126      114     117     127 LOLIMQ: .ASCIZ  /LOW SEEK LIMIT/
            040131      040     123     105
            040134      105     113     040
            040137      114     111     115
            040142      111     124     000
     64     040145      126     101     114 LIMVAL: .ASCIZ  /VALUE/
            040150      125     105     000
     65     040153      125     120     120 HILIMQ: .ASCIZ  /UPPER SEEK LIMIT/
            040156      105     122     040
            040161      123     105     105
            040164      113     040     114
            040167      111     115     111
            040172      124     000
     66     040174      125     123     105 HEADQ:  .ASCIZ  /USE ONLY ONE SURF/
            040177      040     117     116
            040202      114     131     040
            040205      117     116     105
            040210      040     123     125
            040213      122     106     000
```

```
     67 040216  127  110  101  HEADV:  .ASCIZ  /WHAT SURF (0 OR 1)/
        040221  124  040  123
        040224  125  122  106
        040227  040  050  060
        040232  040  117  122
        040235  040  061  051
        040240  000
     69 040241  111  116  120  ERLIMQ: .ASCIZ  /INPUT ERROR LIMIT/
        040244  125  124  040
        040247  105  122  122
        040252  117  122  040
        040255  114  111  115
        040260  111  124  000
     70 040263  104  122  117  AUTOQ:  .ASCIZ  /DROP DRV IF NO RESPONSE/
        040266  120  040  104
        040271  122  126  040
        040274  111  106  040
        040277  116  117  040
        040302  122  105  123
        040305  120  117  116
        040310  123  105  000
     72 040313  104  101  124  DCLIMQ: .ASCIZ  /DATA CMP ERR LIMIT/
        040316  101  040  103
        040321  115  120  040
        040324  105  122  122
        040327  040  114  111
        040332  115  111  124
        040335  000
     74                            .EVEN
     75 040336                  ENDMOD
     76 040336                  LASTAD
                                   .EVEN
        040336                  L$LAST::
```

```
1                       .SBTTL  DIAGNOSTIC SUPERVISOR -- LOW CORE SET UP
```

```
     35 071132  000000                    .WORD   0               ;SPACE FOR USER POOL POINTER
     36 071134  000000                    .WORD   0               ;SIZE
     37 071136  000000                    .WORD   0               ;CHECKSUM (NOT CURRENTLY USED)
     38 071140  000000                    .WORD   0               ;SIZE OF H.W. PTAB. ALLOCATION
     39         071144                    END.SUPV=.+2
     40         000200                    .END 200
```

ASSEMBLY ROUTINES MACRO V03.01 9-FEB-79 19:23:55 PAGE 172-1
SYMBOL TABLE SEQ 0145

```
ABOFLA  040662 G    BLOCK   063436      CSNAM   006117      C$REFG= 000050      D$AAJ   057360
ABOPAS  040600 G    BRMSG   037632      CSR     = 000000      C$REQT= 000045      D$AAK   057376
ABO.FM  043142      BSCHK   023252      CSRMSG  037612      C$RESE= 000033      D$AAL   057414
AFMID   003112      BSFLAG  002722      CURCYL  003006      C$REVI= 000002      D$AAM   057424
AFMIDU  003114      BSFVAL  003354      CURR.S  040344 G    C$RPT = 000025      EF.CON= 000036 G
AFSI    040370 G    BSNSTR  007464      CURR.T  040346 G    C$SEFG= 000047      EF.NEW= 000035 G
ALLCYL= 000001      BYPSNM  007415      CYLQ    040040      C$SPRI= 000041      EF.PWR= 000034 G
ALLOC   061302      B$AAB   047426      CYLTBL  002506      C$SVEC= 000037      EF.RES= 000037 G
ALLSEC= 000002      B$AAF   047340      CYLUP   = 000004    C$TPRI= 000013      EF.STA= 000040 G
ANYERR= 100000      CAFDT   010464      CYLWD   007402      C$UNBU= 000031      EF01    = 000001 G
APT.ER  042272      CALLPC= 000022      C$AAD   052704      C$WTM = 000026      EF02    = 000002 G
ARMID   003116      CALLPS= 000024      C$AAE   052716      C$WTU = 000027      EF03    = 000003 G
ARMIDU  003120      CALLSP= 000026      C$AAK   053714      C10MS   010434      EF04    = 000004 G
ASSEMB= 000011      CALLTC= 000030      C$AAL   054060      C5SEC   010475      EF05    = 000005 G
AUTOQ   040263      CAL.CL  066024      C$ABRT= 000021      C500MS  010445      EF06    = 000006 G
AUTOSZ= 000020      CAL.TI  066062 G    C$ADR = 000020      DANAM   006131      EF07    = 000007 G
A$AAV   045140      CAMSK   002214      C$AU  = 000054      DATACM= 000001      EF08    = 000010 G
A$AAW   045154      CCYLUP  010453      C$BRK = 000022      DATCOM  022244      EF09    = 000011 G
A$AAX   045166      CHKLUP  047442      C$BSEG= 000004      DATGEN  022104      EF10    = 000012 G
A$AAY   045174      CHKSTR  061644      C$BSUB= 000002      DCKERR= 004000      EF11    = 000013 G
A$AAZ   045210      CHKTTY  057732      C$BUFF= 000030      DCLIM = 000012      EF12    = 000014 G
A$ABA   045220      CHK.MA  045602      C$CEFG= 000046      DCLIMQ  040313      EF13    = 000015 G
BADADD= 004000      CHK.PC  052732      C$CLEA= 000012      DCLIMW  013360      EF14    = 000016 G
BAMSK = 000060      CHK.SW  041772      C$CLP1= 000006      DECMSG  057626      EF15    = 000017 G
BANAM   006124      CHOSHD  017546      C$CVEC= 000036      DESDIF  003010      EF16    = 000020 G
BASADD  006022      CHRCNT  061164      C$DCLN= 000044      DESHD   003014      EMT.TR  040666 G
BELL    010355      CH.FLA  045310      C$DODU= 000053      DESSEC  003016      END.OF  046712
BGN.SU= 040336      CH.PAS  045326      C$DRPT= 000024      DESSGN  003012      END.SU= 071144
BHSTAT= 000010      CKBSVD  017656      C$DU  = 000055      DEV.CO  040350 G    ENVIRO  040410 G
BINMSG  057612      CKDATA= 000102      C$EDIT= 000002      DIAGMC= 000000      EOP.CH  067612 G
BIT0  = 000001 G    CKERLM  014750      C$ERDF= 000002      DIAG.T  040670 G    EOP.FM  043156
BIT00 = 000001 G    CLEAR.  046724      C$ERHR= 000003      DIFAUG  003000      EOP.IN  045322
BIT01 = 000002 G    CLKACC  040576 G    C$ERSF= 000001      DIFWD   007356      ERHEAD  002714
BIT02 = 000004 G    CLKBFR  066026      C$ERSU= 000004      DIRBIT= 000004      ERLIM = 000010
BIT03 = 000010 G    CLKCNT  040574 G    C$ESCA= 000010      DIRMSK  002216      ERLIMQ  040241
BIT04 = 000020 G    CLKJUM  066432 G    C$ESEG= 000005      DLTERR= 010000      ERLIMW  013356
BIT05 = 000040 G    CLKRES  067434 G    C$ESUB= 000003      DONE    002710      ERRCNT  003142
BIT06 = 000100 G    CLKSER  067570 G    C$ETST= 000001      DPDVD   070300 G    ERRFOR  054136
BIT07 = 000200 G    CLKSON  040634 G    C$EXIT= 000032      DPMUL   070166 G    ERRHAN  052736
BIT08 = 000400 G    CLK.SE  045404      C$GMAN= 000043      DRDYMS= 000001      ERRPOI  003140
BIT09 = 001000 G    CLNCOD  014560 G    C$GPHR= 000042      DRMSG   037660      ERRSWI  002720
BIT1  = 000002 G    CLRBYT  002206      C$GPRI= 000040      DRSB  = 000010      ERRVEC  003132
BIT10 = 002000 G    CLRPAR  024652      C$GTIM= 000052      DRSELT= 000004      ERR.HR  053674
BIT11 = 004000 G    CLR.MA  045660      C$INIT= 000011      DRSET = 000010      ERR.NU  040340 G
BIT12 = 010000 G    CNT   = 000012      C$INLP= 000020      DRTYPE  037643      ERR.SF  053700
BIT13 = 020000 G    CNTYPE  037666      C$KWOF= 000035      DRVCNT  002776      ERR1    011526 G
BIT14 = 040000 G    CNVT    064102      C$KWON= 000034      DRVERR= 040000      ERR1FO  054222
BIT15 = 100000 G    COMMAN  040406 G    C$LOOP= 000100      DRVNAM  006033      ERR10   013116 G
BIT2  = 000004 G    COMMTA  063716      C$MANI= 000051      DRVNAV  006040      ERR2    011574 G
BIT3  = 000010 G    COMPOP= 007777      C$MSG = 000023      DSESTA= 000400      ERR3    011642 G
BIT4  = 000020 G    CONHNG= 000004      C$PNTB= 000014      DSMSK = 001400      ERR4    011710 G
BIT5  = 000040 G    CONTCL  067514 G    C$PNTF= 000017      DSPCOD  013362 G    ERR5    011760 G
BIT6  = 000100 G    CONTIN  013604      C$PNTS= 000016      DUNIT.  040604 G    ERR6    012030 G
BIT7  = 000200 G    COSTAT= 000040      C$PNTX= 000015      DVC.FT  053664      ERR7    012706 G
BIT8  = 000400 G    COUNT   003136      C$POIN= 000040      D$AAG   054570      ERR8    012756 G
BIT9  = 001000 G    CRDYMS= 000200      C$Q10 = 000377      D$AAH   054606      ERR9    013052 G
BLD.MW  046024      CRLF    060014      C$RDBU= 000007      D$AAI   057354      ESC.PC  052730
```

```
EV.COU  040342 G    F$HW  = 000013    HC.DEF  040366 G    IN.SUF  046676      L$DR    002112 G
EXACYL  003126      F$INIT= 000006    HC.DIA  040364 G    I$AU  = 000041      L$DRCT  002070 G
EXHCYL  003124      F$JMP = 000050    HDALIG= 000010      I$CLN = 000041      L$DRS   002072 G
EXOCYL  003122      F$MOD = 000000    HDCYL   002220      I$DU  = 000041      L$DRST  002112 G
EXROT   003130      F$MSG = 000011    HDHSEL= 000100      I$HRD = 000041      L$DTP   002040 G
FBSFIL  003552      F$PWR = 000017    HDMOVF  007234      I$INIT= 000041      L$DU    014700 G
FILL    060462      F$RPT = 000012    HDRCMP= 000002      I$MOD = 000041      L$DUT   002076 G
FILL.C  000204 G    F$SEG = 000003    HDR40 = 100000      I$MSG = 000041      L$DVTY  002114 G
FLAGS   040402 G    F$SOFT= 000005    HDSEC = 000077      I$PWR = 000041      L$EF    002056 G
FLAGS1  040404 G    F$SRV = 000010    HDSEL = 000020      I$RPT = 000041      L$EFLG  002034 G
FLAGTA  063634      F$SUB = 000002    HDWD    007371      I$SEG = 000041      L$EXP1  002042 G
FLAG.I  045370      F$SW  = 000014    HDWRD1  002754      I$SFT = 000041      L$EXP2  002044 G
FLA.SE  063602      F$TEST= 000001    HDWRD2  002756      I$SRV = 000041      L$EXP3  002046 G
FLG.MA  045330      GARBAG  061166     HDWRD3  002760      I$SUB = 000041      L$HARD  037532 G
FMTOP1  010503      GBND    002212     HEAD  = 000006      I$TST = 000041      L$HPCP  002016 G
FMTOP2  010532      GETCHR  057672     HEADLM= 010000      JJJ     002202      L$HPTP  002022 G
FMTOP3  010554      GETCMN  063256     HEADQ   040174      J$JMP = 000167      L$HW    013330 G
FMT1    010575      GETPAR  054750     HEADV   040216      KBPTR   040446 G    L$ICP   002104 G
FMT1.1  010602      GETPOS  021274     HEAD[illegible] 013354      KBUF    040450 G    L$INIT  013426 G
FMT11   011021      GETSTA= 000003     HERTZ.  044750      LAB     013556      L$LADP  002026 G
FMT12   011027      GETSWI  062252     HFIN    003072      LABACF  007204      L$LAST  040336 G
FMT13   011035      GET.TW  062022     HFINU   003074      LABACR  007220      L$MREV  002050 G
FMT14   011101      GLBDAT  002126 G   HFOUT   003076      LABEXP  007117      L$NAME  002000 G
FMT15   011133      GLBEQA  002126 G   HFOUTU  003100      LABHCF  007154      L$REPP  002066 G
FMT16   011167      GLBERR  011526 G   HICYL = 020000      LABHCR  007170      L$REV   002010 G
FMT17   011200      GLBSUB  014704 G   HILIM = 000004      LABIN   007074      L$SOFT  037676 G
FMT18   011222      GLBTXT  005224 G   HILIMQ  040153      LABMID  007102      L$SPC   002062 G
FMT19   011254      GSTAT   015342     HILIMW  013352      LABOCF  007130      L$SPCP  002020 G
FMT2    010611      GSTATC  015326     HLMTW   002204      LABOCR  007142      L$SPTP  002024 G
FMT20   011311      GSTATG  015352     HNFERR= 010000      LABOUT  007111      L$STA   002030 G
FMT21   011341      GSTATR  015312     HOLDSP= 000020      LAB1    006143      L$SW    013346 G
FMT22   011364      GTSTAT= 000104     HOSTAT= 000020      LAB2    006156      L$TIML  002014 G
FMT23   011420      G$EXCP= 000400     HPTCOD  013326 G    LIMVAL  040145      L$TIMU  002054 G
FMT24   011434      G$HILI= 000002     HRDPRM  037530 G    LINE.F  040664 G    L$TIM1  002052 G
FMT25   011441      G$LOLI= 000001     HRDWTS  024702 G    LOAD.F  045324      L$TST1  002100 G
FMT26   011451      G$NO  = 000000     HRIN    003102      LOCERR  003346      L$UNIT  002012 G
FMT27   011475      G$OFFS= 000400     HRINU   003104      LOCYL = 040000      L.BA    002740
FMT28   011514      G$OFSI= 000376     HROUT   003106      LOGMSG  057634      L.CLK.  044734
FMT3    010614      G$PRMA= 000001     HROUTU  003110      LOLIM = 000002      L.CS    002736
FMT4    010617      G$PRMD= 000002     HSMSK = 000100      LOLIMQ  040126      L.DA    002742
FMT5    010630      G$PRML= 000000     HSSTAT= 000100      LOLIMW  013350      L.MP    002744
FMT6    010650      G$RADA= 000140     HW.ADR  040372 G    LPBFR   040444 G    L10000  011572
FMT7    010712      G$RADB= 000000     H$AAB   064430      LPCNTR  040442 G    L10001  011640
FMT8    010762      G$RADD= 000040     IBUFF   003746      LPT.AD  044766      L10002  011706
FMT9    011014      G$RADF= 000200     ININIT  040614 G    LPT.RE  044762      L10003  011756
FOLWRT= 000100      G$RADL= 000120     INITCO  013426 G    LSI.RE  044756      L10004  012026
FORM.T  054232      G$RADO= 000020     INITIA  057642      LUP     065730      L10005  012704
FREE    061540      G$RADT= 000100     INIT.M  045726      LUP.AD  052734      L10006  012754
FRMWD   007407      G$XFER= 000004     INIT.R  040430 G    L$APT   002036 G    L10007  013050
FWDSKO= 002000      G$YES = 000010     INOUTS= 000020      L$AUT   002074 G    L10010  013114
FWDSKS= 000400      HADONE  002712     INPUTA  060570      L$CCP   002106 G    L10011  013324
F$AU  = 000015      HCESTA= 040000     INTEBL= 000100      L$CLEA  014560 G    L10012  013344
F$BGN = 000040      HCORED  045100     INTFOR  054066      L$CO    002032 G    L10013  013362
F$CLEA= 000007      HCOREQ  045010     INTHLR  014712 G    L$DEPO  002011 G    L10014  014556
F$DU  = 000016      HCORET  040624 G   INVAL.  044674      L$DESC  002102 G    L10015  014676
F$END = 000041      HCRCER= 004000     INVINT  053724      L$DEVP  002064 G    L10016  014702
F$HARD= 000004      HC.ADR  040374 G   INV.SW  041726      L$DISP  013364 G    L10017  014746
```

```
L10020  025112          MDSERR  007714          NEWCYL  003004          PASNEW  013632          P2T17E  006713
L10021  025074          MEM.SI  044776          NEWPRI  067560 G        PASNUM  003342          P2T18E  006745
L10022  025322          MERRS   010346          NEXTAR  064020          PASS.C  040352 G        P2T19E  006772
L10023  025320          MEXERS  010306          NOCLR =  000010         PATTBL  002262          RDALHD  021542
L10024  025532          MFLERR  010075          NOERCT  003347          PAT1    004746          RDDATA= 000114
L10025  025530          MFMTER  005754          NOIRPT= 000002          PAT10   005222          RDHEAD= 000110
L10026  025756          MFOLWR  005503          NOOP  =  000100         PAT2    004750          RDNOHR= 000116
L10027  025754          MFWDSK  005560          NOPWR   006057          PAT3    005010          RDYCHK  017306
L10030  026156          MFWSKO  005613          NO.CLK  044724          PAT4    005050          RDYWAI  021026
L10031  026140          MGTSTA  005277          NO.FLA  063614          PAT5    005110          READRL  015110
L10032  026404          MHCERR  010015          NO.LPT  061132          PAT6    005116          READ.P  066032 G
L10033  026402          MHCRC   007604          NO.PTA  045130          PAT7    005156          REGBAC  070522 G
L10034  026714          MHDERR  010060          NR    =  000000         PAT8    005160          REGSAV  070506 G
L10035  026644          MHDRCP  005330          NUMBIN  054256          PAT9    005220          RELDWT= 040000
L10036  027212          MHFCRC  007653          NUM.LA  054424          PH65$   017250          REQN.P  040412 G
L10037  027154          MHNF    007625          NUM.NO  040400 G        POSHDS  016746          REQN.T  045304
L10040  027510          MINOUT  005460          NUM.UN  041006          POSHDO  021002          RESE3   010361
L10041  027456          MIN.IN  040414 G        NUNITS  047414          POSHSB  020776          RESE4   010365
L10042  031346          MIN.US  040416 G        NXMERR= 020000          POSHW1  020770          RESE5   010372
L10043  032056          MISWI =  000000         NXTFOR  064074          PRINTC  061142          RESE6   010377
L10044  032272          MISWIW  013346          NXTHL   002210          PRINTF  064450          RESPAR  002764
L10045  032216          MITEST= 100000          NXTPAS  013624          PRIOR =  000004         RESTAR  013574
L10046  033016          MNDRST  010141          OBUFF   004346          PRIO0 =  000000 G       RESTBL  002222
L10047  034126          MNEERR  010123          OCTMSG  057620          PRIO1 =  000040 G       REVSKO= 001000
L10050  034042          MNOCLR  006305          OFIN    003042          PRIO2 =  000100 G       REVSKS= 000200
L10051  035120          MNOINT  006236          OFINU   003044          PRIO3 =  000140 G       RE.SET  042074
L10052  035000          MODR    070100 G        OFMID   003046          PRIO4 =  000200 G       RLBA  =  000002
L10053  036334          MOPER   005376          OFMIDU  003050          PRIO5 =  000240 G       RLBAS   002730
L10054  036250          MOPERR  010050          OFOUT   003052          PRIO6 =  000300 G       RLCS  =  000000
L10055  037526          MORECE  002716          OFOUTU  003054          PRIO7 =  000340 G       RLCSR =  000000
L10056  037442          MOUTIN  005437          OLDCYL  003002          PRNTST  061032          RLDA  =  000004
L10057  037612          MPNAM   006136          ONSWAP  017632          PRO.CM  045302          RLDRV   002734
L10060  040040          MQUALS= 003760          OPFLAG  002706          PSETNM  003344          RLMP  =  000006
MAJ.IN  040420 G        MREAD   005232          OPIERR= 002000          PTAB.S  040622 G        RLVEC   002732
MAJ.LO  066030          MREADH  005243          OPMSGS  002126          PUTCHR  057646          RORWOP= 020000
MAJ.US  040422 G        MRESKO  005647          OPR004  007341          PWCON   014102          RPTOP   023422
MANQ    040071          MREVSK  005525          OPR1A   007312          PWRFLG  003352          RPTREM  024416
MAN.TI  001244          MRLFAL  010231          OPR1B   007316          PWR.FA  070772 G        RPTRES  024210
MAPROX  007064          MRSLT   005405          OPR12   007273          PWR.FL  040426 G        RSTACK  067762 G
MAP16   070536 G        MSEEK   005224          ORIN    003056          PWR.MS  071120          RSTRT   013512
MASK.B  047440          MSG.AD  040362 G        ORINU   003060          PWR.SA  071114          SAMSK =  000077
MASK.W  047436          MSG.TY  040336 G        ORMID   003062          PWR.UP  071116          SAVEDO= 042272
MBADAD  005703          MSPERR  007744          ORMIDU  003064          P.CLK.  044742          SBSFIL  003356
MBADSF  005724          MSTERR  007777          OROUT   003066          P2T03E  006371          SEARCH  061770
MBSETO= 000001          MTMBS   006001          OROUTU  003070          P2T04E  006407          SECQ    040054
MCEPR   007572          MTOSLO  006177          OUTINS= 000040          P2T05E  006431          SECWD   007375
MCONHN  006271          MUL     070034 G        O$APTS= 000000          P2T06E  006453          SEEK  =  000106
MCYLOC  010134          MULOAD  005416          O$AU  =  000000         P2T07E  006475          SEEKOP= 010000
MCYLUP  005427          MUNDEF  010164          O$BGNR= 000000          P2T08E  006513          SEGSTA  040636 G
MDATCP  005311          MWDERR  010033          O$BGNS= 000001          P2T09E  006535          SEQMES  007430
MDCRC   007614          MWGERR  007762          O$DU  =  000001         P2T10E  006542          SETDON  013660
MDHEDR  002000 G        MWORD   006171          O$GNSW= 000001          P2T11E  006557          SET.MA  045514
MDLT    007641          MWRCHK  005253          O$POIN= 000001          P2T12E  006574          SFTPRM  037674 G
MDRDY   007561          MWRITE  005265          PARSES  063330          P2T13E  006610          SGNWD   007364
MDRERR  007703          MWRSET  005362          PART2 =  000001 G       P2T14E  006626          SHIFT   070620 G
MDRRES  006216          MWRTAB  010272          PAR.LA  057316          P2T15E  006647          SKTMES  007002
MDRVST  007731          M40HDR  005346          PASCNT  003134          P2T16E  006672          SPDSTA= 004000
```

ASSEMBLY ROUTINES MACRO V03.01 9-FEB-79 19:23:55 PAGE 172-4
SYMBOL TABLE SEQ 0148

```
SPEC.U  045230          TERMTA  057604          T$$INI= 010014          T258$   026464          VCNRST  006325
SPTCOD  013344 G        TEST.M  045242          T$$MSG= 010011          T2588$  026500          VCSTAT= 001000
SPV.SE  000400          TIMFLG  040572 G        T$$SEG= 010000          T266$   026742          VECMSG  037623
SRTMES  007016          TIM.CO  040424 G        T$$SOF= 010060          T2665$  027212          VECT  = 000002
SSINDX  002704          TIM.CP  054230          T$$SRV= 010017          T267$   027024          VERHDR  020440
STAMES  007453          TOO.MA  057564          T$$SUB= 010056          T275$   027240          VERPOS  021422
STAMSK= 000007          TOSLOW= 000001          T$$SW = 010013          T276$   027324          WAITIN  015142
STARTC  067510 G        TRPFLG  003350          T$$TES= 010055          T2765$  027510          WCMSK = 017777
STATE2  010404          TRPHAN  014704          T.BA    002750          T3      025324 G        WCRNG = 160000
STATE3  010414          TSTINT  015274          T.CS    002746          T3.1    025374          WDESTA= 100000
STATE5  010424          TSTLAB  006363          T.DA    002752          T306$   032124          WGESTA= 002000
STOSTA= 010000          TST.AB  047552          T.DRIV  002200          T3065$  032272          WIDTH   054624
STRCHR  060522          TST.TO  041754          T.MP    002754          T307$   032156          WLSTAT= 020000
STRT.T  045306          TYPDR = 000006          T.STAT  002762          T310$   032164          WRTSWI  002724
ST.SET  042140          TYPEC   060160          T1      024702 G        T3100$  033144          WTDATA= 000112
SUBSTK  002306          TYPEPC  054054          T1.1    024736          T3101$  033352          XEQDIA  067646 G
SUNIT.  045312          TYPFLA  063476          T10     027512 G        T3165$  034126          XEQSUB  067634 G
SUPERV  043174          TYPLIN  060056          T11     031350 G        T3204$  035002          XEQ.CL  047356
SUPFLA  040602 G        TYPNUM  057440          T12     032060 G        T3265$  035120          XEQ.CM  044666
SUPV.T  040754 G        TYPSTR  060076          T12.1   032156          T33TBL  002406          XEQ.IN  047040
SUP.PR  041712          TYP.ER  053704          T13     032274 G        T3300$  035270          XEQ.LA  043130
SVCBGL= 000001          TY.UNI  046716          T14     033020 G        T3301$  035552          XEQ.OP  047132
SVCGBL= 000000          T$ARGC= 000007          T14.1   033352          T3365$  036334          XEQ.PR  042332
SVCHAN  047630          T$CODE= 000130          T15     034130 G        T3400$  036470          XEQ.TE  047176
SVCINS= 000000          T$ERCO= 000043          T15.1   034150          T3401$  036714          XRDHD   020020
SVCSUB= 000001          T$ERRN= 003247          T16     035122 G        T3465$  037526          XRDHDC  020010
SVCTAG= 000000          T$EXCP= 000000          T16.1   035552          T4      025534 G        XRDHDG  020024
SVCTST= 000001          T$FLAG= 000040          T17     036336 G        T4.1    025622          XREAD   022572
SWAPHD  017572          T$HILI= 000377          T17.1   036714          T5      025760 G        XREADG  022600
SWCHAN  045122          T$LOLI= 000001          T1965$  025112          T5.1    026016          XSEEK   016204
SWITCH  063774          T$LSYM= 010000          T197$   024726          T6      026160 G        XSEEKT  016174
SW.ADR  040376 G        T$NEST= 177777          T2      025114 G        T6.1    026246          XSEEK1  016210
SW.PTA  045106          T$NSK0= 000000          T2.1    025164          T7      026406 G        XTIME   066520 G
SYS.FT  053654          T$NSK1= 000005          T206$   025166          T7.1    026560          XTIMEN  067344
S$LSYM= 010000          T$NSK2= 000002          T2065$  025322          T8      026716 G        XTIMST  066542
TBLSTR  002726          T$SAVL= 177777          T216$   025376          T8.1    027036          XWRITE  022532
TBT     002446          T$SEGL= 177777          T2165$  025532          T9      027214 G        XWRITT  022522
TCERR   007540          T$SEK0= 010000          T2265$  025756          T9.1    027336          XWRIT1  022536
TEMP0   003020          T$SUBN= 000001          T227$   025624          ULOAD = 000010          XXDP.D  044706
TEMP1   003022          T$TAGL= 177777          T233$   026004          UNDTST  007326          X$ALWA= 000000
TEMP2   003024          T$TAGN= 010061          T2365$  026156          UNIT.D  040354 G        X$FALS= 000040
TEMP3   003026          T$TEMP= 000000          T2465$  026404          UNI.MA  045232          X$OFFS= 000400
TEMP4   003030          T$TEST= 000021          T247$   026250          UNXERR  006346          X$TRUE= 000020
TEMP5   003032          T$TSTM= 177777          T25TBL  002332          USER.P  040616 G        $BREG   045402
TEMP6   003034          T$TSTS= 000001          T25TB2  002360          USER.T  040620 G        $ENDAD  067620 G
TEMP7   003036          T$$CLE= 010015          T2517$  026704          VALDES  007040          $SAV2   070664 G
TEMP8   003040          T$$DU = 010016          T256$   026444          VALID.  041056          $SAV3   070700 G
TERMI   066020          T$$HAR= 010057          T2565$  026714          VAL.LA  041676          $SAV4   070716 G
TERMLI  063622          T$$HW = 010012          T257$   026516          VAL.SW  045342          $SAV5   070736 G

. ABS.  071142      000
        000000      001
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  20928 WORDS  ( 82 PAGES)
DYNAMIC MEMORY AVAILABLE FOR  70 PAGES
C2RLJA.BIN,C2RLJA=#SVCRT/M,PART2,C2HEAD,C2RLJA,C2TAIL,DOCTOR
```