IDENTIFICATION

PRODUCT CODE: AC-E048B-MC

PRODUCT NAME: CZRLDB0 RL01 DRIVE TEST PART 2

DATE CREATED: 11-OCT-78

MAINTAINER: DIAGNOSTIC ENGINEERING

AUTHOR: D. DEKNIS

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 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 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.)
RL11/RLV11 CONTROLLER(S)
1 - 8 RL01 DRIVES
1 - 8 RL01K CARTRIDGES WITH BAD SECTOR FILE
KW11P, KW11L (OPTIONAL)
LINEPRINTER(OPTIONAL)

1.2.2 SOFTWARE REQUIREMENTS

CXRLDB0 RL01 DRIVE TEST PART 2
(FORMERLY MD-11-DZRLD-A)

1.3 RELATED DOCUMENTS AND STANDARDS

RL01 USERS MANUAL (EK-RL01-UG-PRE)
XXDP USERS MANUAL

1.4 DIAGNOSTIC HIERARCY PREREQUISITES

THE RL01 SUBSYSTEM SHOULD HAVE SUCCESSFULLY RUN THE FOLLOWING PROGRAMS:

CZRLAB0	RL11/RLV11 RL01 CONTROLLER TEST (PART 1)
CZRLBB0	RL11/RLV11 RL01 CONTROLLER TEST (PART 2)
CVRLAA0	RLV11 RL01 DISKLESS TEST (RLV11 ONLY)
CZRLCB0	RL01 DRIVE TEST (PART 1)

1.5 ASSUMPTIONS

THE HARDWARE OTHER THAN THE RL01 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 (AS IN 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-B>" 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 ONE 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 ENLESSLY 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 REISSUED.

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 AGAINS:

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 ON ERROR).

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

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

THIS WILL DO THE FOLLOWING:

1. TURN ON THE IER (INHIBIT ERROR PRINTOUT) FLAG
2. TURN ON THE LOE FLAG
3. TURN OFF THE HOE FLAG
4. RESUME EXECUTION AT INSTRUCTION AFTER ERROR REPORT

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

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

1. START
2. RESTART
3. CONTINUE

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

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

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

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

1. START
2. RESTART

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

THE FULL PRINT-OUT FROM THE ABOVE DIALOGUE MIGHT LOOK LIKE THIS:

```
                                                    BY
                                                    WHOM
                                                    ENTERED:
.R DZRKXX                                           O
DZRKXX                                              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
CSR (O) ?                                           D,O
VECTOR (O) ?                                        D,O
BR LEVEL (O) ?                                      D,O
DRIVE (O) ? 0                                       D,O
UNIT 2                                              D
CSR (O) ?                                           D,O
VECTOR (O) ?                                        D,O
BR LEVEL (O) ?                                      D,O
DRIVE (O) ? 1                                       D,O
CHANGE SW (L) ? N                                   D,O
DZRKXX HARD ERR 00004 TST 003 SUB 002 PC:004130     D
ERR HLT                                             D
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                                                  O
DS-B>CON/FLAGS:HOE:IER:LOE=0                        D,O
CHANGE SW (L) ? N                                   D,O
DZRKXX EOP 1                                        D
DS-B>RESTART/PASS:1                                 D,O
CHANGE SW (L) ? N                                   D,O
--------------------
--------------------
--------------------
--------------------
--------------------
```

2.2 HOW TO CREATE A CHAINABLE FILE

THE DIAGNOSTIC AS RECEIVED FROM RELEASE ENGINEERING CANNOT BE RUN IN CHAIN MODE. THA 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 PASSED	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 EXECUTING C) ERROR WAS ENCOUNTERED WITH HOE FLAG SET D) OPERATOR ENTERED CONTROL/C.

AFTER THE OPERATOR RESPONDS TO "# UNITS?", THE HARDWARE DIALOGUE IS INITIATED. WHEN IT IS COMPLETED, THE QUESTIONS "CHANGE SW?" IS ISSUED, AND THE ANSWERS, IF GIVEN, BECOME THE NEW DEFAULTS. THEREFORE IT IS NECESSARY TO RELOAD THE PROGRAM IN ORDER TO RETURN TO THE LOAD DEFAULTS.

THE SWITCH ARGUMENTS ARE AS FOLLOWS:

"TEST-LIST" IS A SEQUENCE OF DECIMAL NUMBERS (1:2 ETC.) OR RANGES OF DECIMAL NUMBERS (1-5:8-10 ETC.) THAT SPECIFY THE TESTS TO BE EXECUTED. THE NUMBERS ARE SEPARATED BY COLONS. THE NUMBERS RANGE FROM 1 TO THE LARGEST TEST NUMBER IN THE DIAGNOSTIC. THEY MAY BE SPECIFIED IN ANY ORDER. TESTS WILL BE EXECUTED IN NUMBERICAL 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 SUMBITTED. THE DEFAULT IS NON-ENDING EXECUTION..B "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 TES 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 ANSERS IF 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 REEXECUTED. 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, SAY) 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.

ON THE FIRST TRIP THRU THE QUESTIONS, ALL OF THE SLOTS IN ALL OF THE P-TABLES ARE FILLED. IF THE OPERATOR TYPES IN LESS THAN N EXPLICIT VALUES IN RESPONSE TO A PARTICULAR QUESTION, THESE VALUES ARE PLACED IN THE P-TABLES (ONE VALUE GOING INTO THE PROPER SLOT OF EACH P-TABLE BEGINNING WITH THE FIRST P-TABLE) UNTIL THE STRING OF VALUES IS EXHAUSTED. THE LAST VALUE THAT SLOT IN THE REMAINING P-TABLES.

ON SUBSEQUENT TRIPS THRU THE QUESTIONS, THE SAME PROCESS IS CARRIED OUT, EXCEPT THAT THE EARLIEST P-TABLE NOT TO HAVE RECEIVED AN EXPLICIT VALUE IN ANY OF ITS SLOTS NOW ASSUMES THE ROLE THAT TABLE NUMBER ONE PLAYED IN THE FIRST TRIP.

THE SERIES OF QUESTIONS IS REISSUED UNTIL AT LEAST ONE QUESTION HAS RECEIVED N EXPLICIT VALUES FROM THE OPERATOR.

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 64 UNITS, AND THAT THERE ARE THREE HARDWARE PARAMETERS FOR EACH (THREE SLOTS IN THE P-TABLE, THREE HARDWARE QUESTIONS IN THE DIALOGUE). LET THE DESIRED VALUE FOR THE FIRST PARAMETER BE THE NUMBER 75 FOR ALL 64 TABLES. LET THE DESIRED VALUE FOR THE SECOND PARAMETER BE EQUAL TO THE UNIT NUMBER (1,2,3,...,64) EXCEPT FOR UNIT 50, WHICH SOULD RECEIVE THE VALUE 49. LET THE DESIRED VALUE FOR THE THIRD PARAMETER BE THE NUMBER 76 FOR THE FIRST 20 UNITS AND THE NUMBER 77 FOR THE LAST 44 UNITS.

THE FOLLOWING DIALOGUE WOULD ACCOMPLISH THIS GOAL:

```
# UNITS (D) ? 64

UNIT 1
<QUESTION 1> ? 75
<QUESTION 2> ? 1-20
<QUESTION 3> ? 76

UNIT 21
<QUESTION 1> ?
<QUESTION 2> ? 21-49,,51-64
<QUESTION 3> ? 77
```

THE FIRST TIME THE SERIES IS ASKED, SLOT ONE RECEIVES A 75 IN ALL 64 TABLES. SLOT TWO RECEIVES THE VALUES 1,2,3,...,20 IN TABLES 1 THRU 20 AND A CONSTANT 20 IN TABLES 21 THRU 64. SLOT THREE RECEIVES A CONSTANT 76 IN ALL 64 TABLES.

THE SECOND TIME THRU THE SERIES, TABLES 21 THRU THE END ARE GOING TO BE AFFECTED (NOTE THAT THIS PIECE OF INFORMATION IS PRINTED OUT FOR THE OPERATOR IN THE FORM "UNIT XX" AT THE BEGINNING OF EACH SERIES). QUESTION 1 IS RESPONDED TO BY A <CR>, SO SLOT ONE STAYS A CONSTANT 75 IN TABLES 21 THRU 64, SINCE NO NEW EXLICIT VALUES ARE TYPED IN. SLOT TWO GETS THE VALUES 21,22,23,...,49 IN TABLES 21 THRU 49, AND GETS A 49 IN SLOT 50, AND GETS THE VALUES 51,52,53,...,64 IN TABLES 51 THRU 64. SLOT THREE GETS THE VALUE 77 IN TABLES 21 THRU 64.

THE DIALOGUE IS TERMINATED WHEN THE SOFTWARE RECOGNIZES THAT 64 EXPLICIT VALUES HAVE BEEN GIVEN FOR AT LEAST ON QUESTION (NAMELY QUESTION 2).

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) 330?

ANSWER WITH THE INTERRUPT VECTOR OF THE CONTROLLER.

BR LEVEL (O) 5?

ANSWER WITH THE INTERRUPT PRIORITY OF THE CONTROLLER.

DRIVE (O) 0?

ANSWER WITH THE DRIVE(S) CONNECTED TO 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 DETECTED IN THE ACTUAL OPERATION WHILE THE READ HEADER FOR 40 HEADERS WITH HEADER COMPARE INDICATES NO ERROR IN THE ACTUAL OPERATION BUT THE HEADER DATA ITSELF WAS IN ERROR.

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

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

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

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

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

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

PERFORMED.

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

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

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

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

3.1.2 SPECIFIC RESULT MESSAGES

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

THE GENERAL FORMAT FOR THE RESULT LINE IS —

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	(HEADER CRC OR DATA CRC ERROR BIT 11 OF CS REGISTER)
HDR NOT FND/DAT LATE	(HEADER 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" 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" IS REPORTED WHEN THE PROGRAM CANNOT DETERMINE THE DRIVE STATE OR STATUS.

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

"COULD NOT RETRIEVE DRIVE STATUS" 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" 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" IS REPORTED WHEN THE COMMAND SUCCESSFULLY COMPLETES BUT THE CONTROLLER HAS NOT GENERATED AN INTERRUPT.

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

"UNEXPECTED ERR" IS REPORTED WHEN THE CONTROLLER SENSES AN ERROR BUT NO ERROR BITS ARE SET.

"BAD SEC FILE FMT ERR" 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" 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 - MUST BE ZERO(0)
BIT 14-7 - CYLINDER ADDRESS FOR TRANSFER
BIT 6 - SURFACE FOR TRANSFER
BIT 5-0 - SECTOR FOR TRANSFER (0-47)

FOR SEEK FUNCTION
-----------------

BIT 15 - MUST BE ZERO(0)
BIT 14-7 - DIFFERENCE TO NEW CYLINDER
BIT 6-5 - MUST BE ZERO(0)
BIT 4 - SURFACE
BIT 3 - MUST BE ZERO
BIT 2 - SEEK DIRECTION( 1 - IN / 0 - OUT )
BIT 1 - MUST BE ZERO
BIT 0 - MUST BE ONE(1)

FOR GET STATUS FUNCTION
-----------------------
```

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

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 STATUS

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 - RESERVED(0)
BIT 6 - SURFACE
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 DIFFERENCE OF 1 SEEK TEST (PART 1)

DO READ HEADER, WAIT FOR INTERRUPT. STORE WORD 1 OF HEADER.

DO SEEK WITH DIFFERENCE OF 1, HEAD 0. IF CYLINDER OF STORED HEADER WORD IS NOT 255 THEN SIGN BIT 1, ELSE SIGN BIT 0. WAIT FOR INTERRUPT.

DO GET STATUS, WAIT FOR INTERRUPT. CHECK STATE IS 4. IF NOT:

DRIVE COMMAND SHIFT REGISTER BAD
DIFFERENCE REGISTER DROPPED BIT
STATE ROM FAILED

WAIT APPROX 5 MS. DO GET STATUS, WAIT FOR INTERRUPT. CHECK STATE IS 5. IF NOT:

DIFFERENCE REGISTER NOT COUNTING
COUNT PULSE NOT GENERATED (COUNT LOGIC)
SEEK ROM FAILED
FAILURE IN DC SERVO
NO TACH FEEDBACK

WAIT APPROX 5 MS LONGER. TEST DRIVE READY. IF SET:

FAILURE IN READY LATCH OR INTEGRATOR

WAIT APPROX 5 MS LONGER. TEST READY. IF RESET:

FAILURE IN INTEGRATOR
UNEXPECTED GUARD BAND DETECTED

DO SEEK WITH DIFFERENCE 1, OPPOSITE SIGN, HEAD 0. REPEAT ALL TESTS AS ABOVE.

REPEAT TEST USING HEAD 1.

NOTE: THIS TEST IS PERFORMED AT THE CYLINDER POSITION FOUND IN THE DRIVE WHEN THE TEST EXECUTES. CHOOSING A SINGLE SURFACE WILL LIMIT TESTING TO THAT SURFACE.

TEST 2 DIFFERENCE OF 1 SEEK TEST (PART 2)

DO READ HEADER, WAIT FOR INTERRUPT. STORE WORD 1 OF HEADER.

DO SEEK WITH DIFFERENCE OF 1, HEAD 0. IF CYLINDER OF STORED HEADER WORD IS NOT 255 THEN SIGN BIT 1, ELSE SIGN BIT 0. WAIT FOR INTERRUPT, WAIT FOR DRIVE READY.

DO READ HEADER, WAIT FOR INTERRUPT. COMPARE CYLINDER OF THIS HEADER WITH CYLINDER OF STORED HEADER FOR DIFFERENCE OF ONE. IF NOT:

COUNT LOGIC BAD
INTERGRATOR FAILED

CHECK THAT HEADS MOVED FORWARD OR REVERSE AS EXPECTED. IF

NOT:

SEEK ROM FAILED

DO SEEK WITH DIFFERENCE OF 1, OPPOSITE SIGN, HEAD 0. REPEAT ALL TESTS AS ABOVE.

REPEAT TEST USING HEAD 1.

NOTE: THIS TEST IS PERFORMED AT THE CYLINDER POSITION FOUND IN THE DRIVE WHEN THE TEST EXECUTES. CHOOSING A SINGLE SURFACE WILL LIMIT TESTING TO THAT SURFACE.

TEST 3 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 4 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

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 1 IS CHOSEN.

TEST 5 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 6 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 7 INNER GUARD BAND DETECTION TEST

POSITION HEADS AT CYLINDER 255 USING SEEK WITH DIFFERENCE OF 1, HEAD 0.

WHEN AT CYLINDER 255, 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 CYLINDER 255. IF NOT:

FAILED TO SEEK BACK TO CYLINDER 255

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 8 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 9 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 > 255. POSITION AT 255.

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: 6, 9, 12, 17, 22, 27, 34, 41, 128, 256. 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 10 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 255 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 11 REVERSE OSCILLATING SEEK TEST

POSITION HEADS AT CYLINDER 255. DO OSCILLATING SEEK USING HEAD 0. (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 12 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.

REPEAT ABOVE SEEKING BETWEEN CYLINDER 0 TO 127 AND 128 TO 256.

REPEAT ABOVE SEEKING BETWEEN CYLINDER 0 AND 255.

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
128 CYL FWD	X		X	X
128 CYL REV	X		X	X
256 CYL FWD		X		X
256 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 13 BASIC READ DATA TEST

POSITION HEADS AT CYLINDER 255.

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 14 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 15 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 16 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 17 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 18 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 255) 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 19 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.

```
                                        .NLIST  CND,MD,ME
                                        .ENABL  ABS,AMA
                 002000                 .=2000

 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  CZRLD,B,0,30000,30000,300,RL01
(4) 002000     103                     .ASCII /C/
(4) 002001     132                     .ASCII /Z/
(4) 002002     122                     .ASCII /R/
(4) 002003     114                     .ASCII /L/
(4) 002004     104                     .ASCII /D/
(6) 002005     000                     .BYTE   0
(6) 002006     000                     .BYTE   0
(5) 002007     000                     .BYTE   0
(4) 002010     102                     .ASCII  /B/
(4) 002011     060                     .ASCII  /0/
(4) 002012  000000                     .WORD   0
(4) 002014  000300                     .WORD   300
(4) 002016  037604                     .WORD   L$HARD
(4) 002020  037730                     .WORD   L$SOFT
(4) 002022  013356                     .WORD   L$HW
(4) 002024  013372                     .WORD   L$SW
(4) 002026  040514                     .WORD   L$LAST
(4) 002030  000000                     .WORD   0
(4) 002032  000000                     .WORD   0
(4) 002034  000000                     .WORD   0
(4) 002036  000000                     .WORD   0
(4) 002040  013410                     .WORD   L$DISPATCH
(4) 002042  000000                     .WORD   0
(4) 002044  000000                     .WORD   0
(4) 002046  000000                     .WORD   0
(4) 002050     002                     .BYTE   C$REVISION
(3) 002051     002                     .BYTE   C$EDIT
(4) 002052  030000                     .WORD   30000
(4) 002054  030000                     .WORD   30000
(4) 002056  000000                     .WORD   0
(5) 002060  000000                     .WORD   0
(4) 002062  000000                     .WORD   0
(4) 002064  002114                     .WORD   L$DVTYP
(4) 002066  000000                     .WORD   0
(4) 002070  002112                     .WORD   L$DR
(4) 002072  002112                     .WORD   L$DRST
(4) 002074  000000                     .WORD   0
(4) 002076  014564                     .WORD   L$DU
(5) 002100  000014                     .WORD   14
(4) 002102  000000                     .WORD   0
(4) 002104  013456                     .WORD   L$INIT
(4) 002106  014444                     .WORD   L$CLEAN
```

```
                                ENDMOD
                                        DEVREG
(5)  002110  000000                     .WORD   0
(2)  002112  000001                     .BLKW
     002114                             DEVTYP  <RL01>
(3)  002114  046122  030460  000        .ASCIZ  /RL01/
(2)  002122                             .EVEN

                                        ;COPYRIGHT (C) 1977, 1978
                                        ;THIS SOFTWARE IS FURNISHED UNDER LICENSE FOR USE ONLY
                                        ;ON A SINGLE COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH
                                        ;THE INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS
                                        ;SOFTWARE, OR ANY COPIES THEREOF, MAY NOT BE PROVIDED
                                        ;OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT
                                        ;FOR USE ON SUCH SYSTEM, AND TO ONE WHO AGREES TO THESE
                                        ;LICENSE TERMS. TITLE TO OWNERSHIP OF THE SOFTWARE SHALL
                                        ;AT ALL TIMES REMAIN IN DEC.
                                        ;
                                        ;THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE
                                        ;WITHOUT NOTICE AND SHALL NOT BE CONSTRUED AS A COMMITMENT
                                        ;BY DIGITAL EQUIPMENT CORPORATION.
                                        ;
                                        ;DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY
                                        ;OF ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.

002122                          BGNMOD  GLBEQAT

002122                          EQUALS
                                ;       OFFSETS FOR HARDWARE P-TABLE
000000                          CSR     =0                      ;BUS ADDRESS
000002                          VECT    =2                      ;VECTOR ADDRESS
000004                          PRIOR   =4                      ;PRIORITY
000006                          DRSB    =6                      ;DRIVE SELECT BIT
000010                          CNT     =10                     ;CONTROLLER TYPE

                                ;       OFFSET FOR SOFTWARE P-TABLE
000000                          MISWI   =0                      ;SOFTWARE PARAMETERS SWITCHES
000002                          LOLIM   =2                      ;CYLINDER LOWER LIMIT
000004                          HILIM   =4                      ;CYLINDER HIGH LIMIT
000006                          HEAD    =6                      ;SELECTED HEAD FOR RUNNING TESTS
000010                          ERLIM   =10                     ;ERROR LIMIT
000012                          DCLIM   =12                     ;DATA COMPARE ERROR LIMIT

                                ;       BIT ASSIGNMENT FOR SOFTWARE P-TABLE SWITCHES
000001                          ALLCYL  =BIT00                  ;USE ALL CYLINDERS
000002                          ALLSEC  =BIT01                  ;USE ALL SECTORS
000004                          DRSELT  =BIT02                  ;EXECUTE DRIVE SELECT TEST
000010                          HDALIGN =BIT03                  ;EXECUTE HEAD ALIGNMENT TEST
000020                          AUTOSZ  =BIT04                  ;AUTO SIZE FOR DRIVE-DROP IF NO RESPONSE
010000                          HEADLM  =BIT12                  ;HEAD LIMIT SPECIFIED FLAG
020000                          HICYL   =BIT13                  ;HI LIMIT SPECIFIED FLAG
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                          INTERL  =100                    ;INTERRUPT ENABLE MASK
000060                          BAMSK   =60                     ;BUS ADDRESS UPPER MASK
000001                          DRDYMSK =1                      ;DRIVE READY MASK
```

```
                                ;       REGISTER BIT DEFINITIONS - DISK ADDRESS FOR DATA XFER
000077                          SAMSK   =77                     ;SECTOR ADDRESS MASK
000100                          HSMSK   =100                    ;HEAD SELECT MASK
077600                          CAMSK   =77600                  ;CYLINDER ADDRESS MASK

                                ;       REGISTER BIT DEFINITIONS - DISK ADDRESS FOR SEEK
000001                          MBSET0  =1                      ;MUST BE SET, BIT 0
000004                          DIRBIT  =4                      ;DIRECTION BIT
000020                          HDSEL   =20                     ;HEAD SELECT BIT
077600                          DIRMSK  =77600                  ;CYLINDER DIFFERENCE MASK

                                ;       REGISTER BIT DEFINITIONS - DISK ADDRESS FOR GET STATUS
000003                          GETSTAT =3                      ;GET STATUS SETUP
000010                          DRSET   =10                     ;DRIVE RESET MASK

                                ;       REGISTER BIT DEFINITIONS - MP FOR DATA XFER
017777                          WCMSK   =17777                  ;WORD COUNT MASK
160000                          WCRNG   =160000                 ;WORD COUNT RANGE MASK

                                ;       REGISTER BIT DEFINITIONS - MP FOR READ HEADER
077600                          HDCYL   =077600                 ;CYLINDER MASK
000077                          HDSEC   =77                     ;SECTOR MASK
000100                          HDHSEL  =100                    ;HEAD SELECT MASK

                                ;       REGISTER BIT DEFINITIONS - MP FOR GET STATUS
000007                          STAMSK  =7                      ;STATE MASK
000010                          BRSTAT  =10                     ;BRUSH HOME STATUS
000020                          HDSTAT  =20                     ;HEADS OUT STATUS
000040                          COSTAT  =40                     ;COVER OPEN STATUS
000100                          HSSTAT  =100                    ;HEAD SELECT STATUS
000400                          DSESTAT =400                    ;DRIVE SELECT ERROR STATUS
001000                          VCSTAT  =1000                   ;VOLUME CHECK STATUS
002000                          WGESTAT =2000                   ;WRITE GATE ERROR STATUS
004000                          SPDSTAT =4000                   ;SPIN ERROR STATUS
010000                          STOSTAT =10000                  ;SEEK TIMEOUT ERROR STATUS
020000                          WLSTAT  =20000                  ;WRITE LOCK STATUS
040000                          HCESTAT =40000                  ;HEAD CURRENT ERROR STATUS
100000                          WDESTAT =100000                 ;WRITE DATA ERROR STATUS

002122                          EWDMOD
002122                          BGNMOD  GLBDAT

                                ;       TABLE OF OPERATION MESSAGES
002122  000000                  OPMSGS: .WORD   0               ;FILLER
002124  005002                          .WORD   MWRCHK          ;MESSAGE FOR WRITE CHECK
002126  005032                          .WORD   MGTSTA          ;       GET STATUS
002130  004744                          .WORD   MSEEK           ;       SEEK
002132  004765                          .WORD   MREADH          ;       READ HEADER
002134  005016                          .WORD   MWRITE          ;       WRITE DATA
002136  004752                          .WORD   MREAD           ;       READ DATA
002140  005127                          .WORD   MWRSET          ;       WITH RESET
002142  005046                          .WORD   MDATCP          ;       WITH DATA COMPARE
002144  005071                          .WORD   MHDRCP          ;       WITH HEADER COMPARE
002146  005204                          .WORD   MCYLUP          ;       LOAD HEADS
002150  005173                          .WORD   MULOAD          ;       UNLOAD HEADS
002152  005235                          .WORD   MINOUT          ;          IN-OUT SEQ
```



```
002154  005214                          .WORD   MOUTIN          ;          OUT-IN SEQ
002156  005260                          .WORD   MFOLWRT         ;          FOLLOWING WRITE
002160  005304                          .WORD   MREVSK          ;          REV SEEK
002162  005337                          .WORD   MFWDSK          ;          FWD SEEK
002164  005426                          .WORD   MRESKO          ;          REV SEEK
002166  005372                          .WORD   MFWSKO          ;          FWD SEEK
002170  005462                          .WORD   MBADAD          ;          BAD DISK ADD FOR WRITE
002172  005113                          .WORD   M40HDR          ;       40 HEADER OPERATION

                                ;       TABLE OF RESULT NAME MESSAGE ADDRESSES
002174  007554                  RESTBL: .WORD   MCERR            ;CONTROLLER ERROR
002176  007665                          .WORD   MDRERR          ;DRIVE ERROR
002200  010114                          .WORD   MNEERR          ;NON-EXISTANT MEMORY ERROR
002202  010066                          .WORD   MFLERR          ;HEADER NOT FOUND-DATA LATE
002204  010051                          .WORD   MHDERR          ;HEADER OR DATA ERROR
002206  010032                          .WORD   MOPERR          ;OPERATION INCOMPLETE
002210  010141                          .WORD   MNDRST          ;NO DRIVE STATUS AVAILABLE
002212  000000                          .WORD   0
002214  010015                          .WORD   MWDERR          ;WRITE DATA ERROR
002216  007777                          .WORD   MHCERR          ;HEAD CURRENT ERROR
002220  000000                          .WORD   0
002222  007761                          .WORD   MSTERR          ;SEEK TIMEOUT ERROR
002224  007726                          .WORD   MSPERR          ;SPINDLE ERROR
002226  007744                          .WORD   MWGERR          ;WRITE GATE ERROR
002230  000000                          .WORD   0
002232  007676                          .WORD   MDSERR          ;DRIVE SELECT ERROR

                                ;       PATTERN TABLE
002234  004466                  PATTBL: .WORD   PAT1
002236  004470                          .WORD   PAT2
002240  004530                          .WORD   PAT3
002242  004570                          .WORD   PAT4
002244  004630                          .WORD   PAT5
002246  004636                          .WORD   PAT6
002250  004676                          .WORD   PAT7
002252  004700                          .WORD   PAT8
002254  004740                          .WORD   PAT9
002256  004742                          .WORD   PAT10

                                ;       SUBROUTINE CALLING STACK
002260  000000                  SUBSTK: .WORD   0               ;STACK IS 12 WORDS LONG
002262  000000                          .WORD   0
002264  000000                          .WORD   0
002266  000000                          .WORD   0
002270  000000                          .WORD   0
002272  000000                          .WORD   0
002274  000000                          .WORD   0
002276  000000                          .WORD   0
002300  000000                          .WORD   0
002302  000000                          .WORD   0

002304  000002                  T25TBL: .WORD   2               ;TABLE OF DIFFERENCES TO BE USED
002306  000006                          .WORD   6               ;IN TEST 25
002310  000011                          .WORD   9.
002312  000014                          .WORD   12.
```

```
                                .WORD   17.
                                .WORD   22.
                                .WORD   27.
                                .WORD   34.
                                .WORD   41.
                                .WORD   128.
                                .WORD   255.

                        ;       TABLE TO BE USED IN TEST 33 AND 34 TO BUILD AND STORE THE
                        ;       CYLINDERS TO BE USED IN THE TEST.
002332  000010          T33TBL: .BLKW   10

002352  002             CYLTBL: .BYTE   2               ;TABLE OF DEFAULT CYLINDERS
002353  007                     .BYTE   7.
002354  016                     .BYTE   14.
002355  024                     .BYTE   20.
002356  033                     .BYTE   27.
002357  041                     .BYTE   33.
002360  046                     .BYTE   38.
002361  055                     .BYTE   45.
002362  064                     .BYTE   52.
002363  072                     .BYTE   58.
002364  101                     .BYTE   65.
002365  110                     .BYTE   72.
002366  115                     .BYTE   77.
002367  124                     .BYTE   84.
002370  133                     .BYTE   91.
002371  141                     .BYTE   97.
002372  146                     .BYTE   102.
002373  154                     .BYTE   108.
002374  161                     .BYTE   113.
002375  170                     .BYTE   120.
002376  177                     .BYTE   127.
002377  206                     .BYTE   134.
002400  213                     .BYTE   139.
002401  222                     .BYTE   146.
002402  230                     .BYTE   152.
002403  235                     .BYTE   157.
002404  244                     .BYTE   164.
002405  252                     .BYTE   170.
002406  261                     .BYTE   177.
002407  270                     .BYTE   184.
002410  275                     .BYTE   189.
002411  303                     .BYTE   195.
002412  312                     .BYTE   202.
002413  317                     .BYTE   207.
002414  326                     .BYTE   214.
002415  334                     .BYTE   220.
002416  343                     .BYTE   227.
002417  352                     .BYTE   234.
002420  361                     .BYTE   241.
002421  367                     .BYTE   247.
002422  375                     .BYTE   253.
002423  000                     .BYTE   0

002424  000000          SSINDX: .WORD   0               ;SUBROUTINE STACK INDEX POINTER
```



```
                        ;       OPERATIONAL FLAGS
002426  000000          OPFLAG: .WORD   0               ;OPERATION FLAGS
002430  000000          DONE:   .WORD   0               ;OPERATION COMPLETE FLAG
002432  000000          HADONE: .WORD   0               ;HEAD ALIGNMENT DONE FLAG
002434  000000          ERHEAD: .WORD   0               ;ADDRESS OF ERROR HEADER
002436  000000          MORECE: .WORD   0               ;MORE THAN 1 COMPARE ERROR
002440  000000          ERRSWI: .WORD   0               ;ERROR RETURN SWITCH
002442  000000          BSFLAG: .WORD   0               ;BAD SECTOR FLAGS
002444  000000          WRTSWI: .WORD   0               ;WRITE SWITCH
002446  000000          TBLSTR: .WORD   0               ;TABLE STORAGE

002450  000000          RLBAS:  .WORD   0               ;RL11 BASE ADDRESS
002452  000000          RLVEC:  .WORD   0               ;RL11 VECTOR ADDRESS
002454  000000          RLDRV:  .WORD   0               ;DRIVE NUMBER UNDER TEST

002456  000000          L.CS:   .WORD   0               ;CONTROLLER REGISTER STORAGE
002460  000000          L.BA:   .WORD   0               ;BEFORE OPERATION
002462  000000          L.DA:   .WORD   0
002464  000000          L.MP:   .WORD   0
002466  000000          T.CS:   .WORD   0               ;CONTROLLER REGISTER STORAGE
002470  000000          T.BA:   .WORD   0               ;    AFTER OPERATION
002472  000000          T.DA:   .WORD   0
002474                  T.MP:
002474  000000          HDWRD1: .WORD   0               ;HEADER WORD STORAGE
002476  000000          HDWRD2: .WORD   0
002500  000000          HDWRD3: .WORD   0

002502  000000          T.STAT: .WORD   0               ;DRIVE STATE STORAGE

002504  000000          RESPARM:        .WORD   0       ;PARAM BLOCK FOR REASON REPORT
002506  000000                  .WORD   0
002510  000000                  .WORD   0
002512  000000                  .WORD   0
002514  000000                  .WORD   0

002516  000000          DRVCNT: .WORD   0               ;DRIVE COUNT FOR DRIVES UNDER TEST
002520  000000          DIFAUG: .WORD   0               ;DIFFERENCE AUGMENT FOR SEEK
002522  000000          OLDCYL: .WORD   0               ;OLD CYLINDER
002524  000000          NEWCYL: .WORD   0               ;NEW CYLINDER
002526  000000          CURCYL: .WORD   0               ;CURRENT CYLINDER
002530  000000          DESDIF: .WORD   0               ;DESIRED DIFFERENCE
002532  000000          DESSGN: .WORD   0               ;DESIRED SIGN
002534  000000          DESHD:  .WORD   0               ;DESIRED HEAD
002536  000000          DESSEC: .WORD   0               ;DESIRED SECTOR
002540  000000          TEMP0:  .WORD   0               ;TEMPORARY STORAGE
002542  000000          TEMP1:  .WORD   0               ;TEMPORARY STARAGE
002544  000000          TEMP2:  .WORD   0               ;TEMPORARY STORAGE
002546  000000          TEMP3:  .WORD   0               ;TEMPORARY STORAGE
002550  000000          TEMP4:  .WORD   0               ;TEMPORARY STORAGE
002552  000000          TEMP5:  .WORD   0               ;TEMPORARY STORAGE
002554  000000          TEMP6:  .WORD   0               ;TEMPORARY STORAGE
002556  000000          TEMP7:  .WORD   0               ;TEMPORARY STORAGE
002560  000000          TEMP8:  .WORD   0               ;TEMPORARY STORAGE

                        ;       TIMER STORAGE
```

```
353  002562  000000                    OFIN:   .WORD   0               ;ONE CYLINDER FORWARD INNER
354  002564  000000                    OFINU:  .WORD   0               ;       UPPER
355  002566  000000                    OFMID:  .WORD   0               ;ONE CYLINDER FORWARD MIDDLE
356  002570  000000                    OFMIDU: .WORD   0               ;       UPPER
357  002572  000000                    OFOUT:  .WORD   0               ;ONE CYLINDER FORWARD OUTER
358  002574  000000                    OFOUTU: .WORD   0               ;       UPPER
359  002576  000000                    ORIN:   .WORD   0               ;ONE CYLINDER REVERSE INNER
360  002600  000000                    ORINU:  .WORD   0               ;       UPPER
361  002602  000000                    ORMID:  .WORD   0               ;ONE CYLINDER REVERSE MIDDLE
362  002604  000000                    ORMIDU: .WORD   0               ;       UPPER
363  002606  000000                    OROUT:  .WORD   0               ;ONE CYLINDER REVERSE OUTER
364  002610  000000                    OROUTU: .WORD   0               ;       UPPER
365  002612  000000                    HFIN:   .WORD   0               ;128 CYLINDER FORWARD INNER
366  002614  000000                    HFINU:  .WORD   0               ;       UPPER
367  002616  000000                    HFOUT:  .WORD   0               ;128 CYLINDER FORWARD OUTER
368  002620  000000                    HFOUTU: .WORD   0               ;       UPPER
369  002622  000000                    HRIN:   .WORD   0               ;128 CYLINDER REVERSE INNER
370  002624  000000                    HRINU:  .WORD   0               ;       UPPER
371  002626  000000                    HROUT:  .WORD   0               ;128 CYLINDER REVERSE OUTER
372  002630  000000                    HROUTU: .WORD   0               ;       UPPER
373  002632  000000                    AFMID:  .WORD   0               ;256 CYLINDER FORWARD
374  002634  000000                    AFMIDU: .WORD   0               ;       UPPER
375  002636  000000                    ARMID:  .WORD   0               ;256 CYLINDER REVERSE
376  002640  000000                    ARMIDU: .WORD   0               ;       UPPER
377
378  002642  000226                    EXOCYL: .WORD   150.            ;EXPECTED TIME ONE CYLINDER
379  002644  001046                    EXHCYL: .WORD   550.            ;EXPECTED TIME 128 CYLINDER
380  002646  001750                    EXACYL: .WORD   1000.           ;EXPECTED TIME 256 CYLINDER
381  002650  000372                    EXROT:  .WORD   250.            ;EXPECTED ROTATION TIME
382  002652  000004                    ERRVEC: .WORD   4               ;ERROR VECTOR USED WHEN AUTO SIZING
383
384                                    ;       MISCELLANEOUS COUNTERS
385  002654  000000                    PASCNT: .WORD   0               ;PASS COUNTER (LOCAL TO A TEST)
386  002656  000000                    COUNT:  .WORD   0               ;A COUNTER (LOCAL TO A TEST)
387  002660  000000                    ERRPOINT: .WORD 0               ;ERROR POINTER
388  002662  000100                    ERRCNT: .BLKW   64.             ;STORAGE FOR ERROR COUNTERS
389  003062  000000                    PASNUM: .WORD   0               ;PASS NUMBER FOR PROGRAM
390  003064  000000                    PSETNM: .WORD   0               ;COUNTER FOR PARAMETER SET NUMBER IN USE
391  003066      000                   LOCERR: .BYTE   0               ;LOCAL ERROR COUNTER
392  003067      000                   NOERCT: .BYTE   0               ;INHIBIT ERROR COUNTING FLAG
393  003070  000000                    TRPFLG: .WORD   0               ;HARDWARE TRAP OCCURANCE
394  003072  000000                    PWRFLG: .WORD   0               ;POWER FAILURE OCCURANCE
395
396                                    ;       BAD SECTOR TABLES AND POINTERS
397  003074  000000                    BSFVAL: .WORD   0               ;BAD SECTORS FILES VALID FLAG
398
399  003076  000076                    SBSFIL: .BLKW   76              ;SOFTWARE BAD SECTOR FILE
400  003272  000076                    FBSFIL: .BLKW   76              ;FACTORY BAD SECTOR FILE
401
402  003466  000200                    IBUFF:  .BLKW   200             ;INPUT BUFFER
403  004066  000200                    OBUFF:  .BLKW   200             ;OUTPUT BUFFER
404
405  004466  000000                    PAT1:   .WORD   0               ;PATTERN 1 (ALL ZEROS)
406  004470  177772                    PAT2:   .WORD   177772
407  004472  177777                            .WORD   177777
408  004474  177777                            .WORD   177777
```



```
409  004476  052525                            .WORD   052525
410  004500  052525                            .WORD   052525
411  004502  052525                            .WORD   052525
412  004504  177777                            .WORD   177777
413  004506  177777                            .WORD   177777
414  004510  052525                            .WORD   052525
415  004512  052525                            .WORD   052525
416  004514  177777                            .WORD   177777
417  004516  052525                            .WORD   052525
418  004520  177252                            .WORD   177252
419  004522  177252                            .WORD   177252
420  004524  172765                            .WORD   172765
421  004526  172765                            .WORD   172765
422
423  004530  000003                    PAT3:   .WORD   000003
424  004532  000000                            .WORD   000000
425  004534  000000                            .WORD   000000
426  004536  177777                            .WORD   177777
427  004540  177777                            .WORD   177777
428  004542  177777                            .WORD   177777
429  004544  000000                            .WORD   000000
430  004546  000000                            .WORD   000000
431  004550  177777                            .WORD   177777
432  004552  177777                            .WORD   177777
433  004554  000000                            .WORD   000000
434  004556  177777                            .WORD   177777
435  004560  000000                            .WORD   000000
436  004562  177777                            .WORD   177777
437  004564  000000                            .WORD   000000
438  004566  177777                            .WORD   177777
439
440  004570  025252                    PAT4:   .WORD   025252
441  004572  052525                            .WORD   052525
442  004574  052525                            .WORD   052525
443  004576  125252                            .WORD   125252
444  004600  125252                            .WORD   125252
445  004602  125252                            .WORD   125252
446  004604  052525                            .WORD   052525
447  004606  052525                            .WORD   052525
448  004610  125252                            .WORD   125252
449  004612  125252                            .WORD   125252
450  004614  052525                            .WORD   052525
451  004616  125252                            .WORD   125252
452  004620  052525                            .WORD   052525
453  004622  125252                            .WORD   125252
454  004624  052525                            .WORD   052525
455  004626  125252                            .WORD   125252
456
457  004630  155555                    PAT5:   .WORD   155555
458  004632  133333                            .WORD   133333
459  004634  066666                            .WORD   066666
460
461  004636  121105                    PAT6:   .WORD   121105
462  004640  150442                            .WORD   150442
463  004642  064221                            .WORD   064221
464  004644  132110                            .WORD   132110
```

```
                                        .WORD   055044
                                        .WORD   026442
                                        .WORD   013211
                                        .WORD   105504
                                        .WORD   042642
                                        .WORD   021321
                                        .WORD   110550
                                        .WORD   044264
                                        .WORD   022132
                                        .WORD   011055
                                        .WORD   104426
004674  042213                          .WORD   042213

004676  177777                  PAT7:   .WORD   177777

004700  045513                  PAT8:   .WORD   045513
004702  122645                          .WORD   122645
004704  151322                          .WORD   151322
004706  064551                          .WORD   064551
004710  132264                          .WORD   132264
004712  055132                          .WORD   055132
004714  026459                          .WORD   026459
004716  113226                          .WORD   113226
004720  045513                          .WORD   045513
004722  122645                          .WORD   122645
004724  151322                          .WORD   151322
004726  064551                          .WORD   064551
004730  132264                          .WORD   132264
004732  055132                          .WORD   055132
004734  026459                          .WORD   026459
004736  113226                          .WORD   113226

004740  125252                  PAT9:   .WORD   125252

004742  155555                  PAT10:  .WORD   155555

004744                          ENDMOD

004744                          BGNMOD  GLBTXT
004744  042523  045505  000040  MSEEK:  .ASCIZ  /SEEK /
004752  042542  042101  042040  MREAD:  .ASCIZ  /READ DATA /
004765      144  040505  020104  MRADHR: .ASCIZ  /READ HEADER /
005002  051127  052111  020105  MWRITE: .ASCIZ  /WRITE CHECK/
005016  051127  052111  020105  MWRITR: .ASCIZ  /WRITE DATA /
005032  042507  020124  051123  MGTSTA: .ASCIZ  /GET STATUS /
005046  044527  044124  042040  MDATCP: .ASCIZ  /WITH DATA COMPARE /
005071      127  052111  020110  MHDRCP: .ASCIZ  /WITH HDR COMPARE /
005113      106  051117  032040  M40HDR: .ASCIZ  /FOR 40 HDRS/
005127      127  052111  020110  MWRSET: .ASCIZ  /WITH RESET /
005143      117  042520  040522  MOPER:  .ASCIZ  /OPERATION: /
005157      122  051505  046125  MRSLT:  .ASCIZ  /RESULT:    /
005173      125  046116  020104  MULOAD: .ASCIZ  /UNLD DRV/
005204  042114  042040  053122  MCYLUP: .ASCIZ  /LD DRV /
005214  047506  020114  020060  MOUTIN: .ASCIZ  /FOL 0 TO CC SEEK/
005235      106  046117  031040  MINOUT: .ASCIZ  /FOL 255 TO CC SEEK/
005260  047506  020114  051127  MFOLWRT:        .ASCIZ  /FOL WRITE (NO SEEK)/
```



```
005304  042101  020112  054503  MREVSK: .ASCIZ  /ADJ CYL WRITTN AFTER REV SK/
005337      101  045104  041440  MFWDSK: .ASCIZ  /ADJ CYL WRITTN AFTER FWD SK/
005372  045623  043040  042127  MFWSKO: .ASCIZ  /SK FWD,WRT - SK REV,OVERWRT/
005426  045523  051040  053105  MRESKO: .ASCIZ  /SK REV,WRT - SK FWD,OVERWRT/
005462  047117  041040  042101  MBADAD: .ASCIZ  /ON BAD SEC FILES/
005503      103  047101  052047  MBADSF: .ASCIZ  /CAN'T GET BAD SEC FILES/
005533      102  042101  051440  MFMTER: .ASCIZ  /BAD SEC FILE FMT ERR/
005560  047524  046440  047101  MTMBS:  .ASCIZ  /TO MANY BAD SEC FOR PROG CAPACITY/
005622  052502  020123  042101  BASADD: .ASCIZ  /BUS ADD=/
005633      104  053117  000075  DRV=:  .ASCIZ  /DRV=/
005647  051104  053117  020105  DRVNAV: .ASCIZ  /DRIVE UNAVAILABLE FOR TEST/
005673      104  053122  042040  NOPWR:  .ASCIZ  /DRV DID NOT REC'R FROM PWR FAIL/
005733      122  041514  000123  CSNAM:  .ASCIZ  /RLCS/
005740  046122  040502      000  BANAM:  .ASCIZ  /RLBA/
005745      122  042114  000101  DANAM:  .ASCIZ  /RLDA/
005752  046122  050115      000  MPNAM:  .ASCIZ  /RLMP/
005757      117  020120  047111  LAB1:   .ASCIZ  /OP INIT = /
005772  050117  042040  047117  LAB2:   .ASCIZ  /OP DONE = /
006005      127  051117  020104  MWORD:  .ASCIZ  /WORD /
006013      111  052116  050122  MTOSLOW:        .ASCIZ  /INTRPT TO LATE/
006032  050117  020111  042523  MDRRES: .ASCIZ  /OPI SET-NO DRV RESPONSE/
006062  047516  044440  052116  MNOINT: .ASCIZ  /NO INTRPT ON CMND COMPLETE/
006115      103  052116  051114  MCONHNG:        .ASCIZ  /CNTLR HUNG (NO RDY)/
006141      105  051122  042040  MNOCLR: .ASCIZ  /ERR DID NOT CLR/
006161      126  046117  041440  VCNRST: .ASCIZ  /VOL CHK NOT RSET/
006207  047125  050130  052103  UNXERR: .ASCIZ  /UNXPCTED ERR/
006217      040  042524  052123  ISTLAB: .ASCIZ  / TEST/
006225                          P2T01E:
006244      104  043111  020106  P2T02E: .ASCIZ  /DIFF OF 1 SEEK/
006244  052517  020124  051107  P2T03E: .ASCIZ  /OUT GRD BAND DETECT/
006270  047111  020103  042523  P2T04E: .ASCIZ  /INC SEEK FWD HD 0/
006312  047111  020103  042523  P2T05E: .ASCIZ  /INC SEEK REV HD 0/
006334  047111  020103  042523  P2T06E: .ASCIZ  /INC SEEK FWD HD 1/
006356  047111  020116  051107  P2T07E: .ASCIZ  /INN GRD BAND DETECT/
006402  047111  020103  042523  P2T08E: .ASCIZ  /INC SEEK REV HD 1/
006424  042523  045505      000  P2T09E: .ASCIZ  /SEEK/
006431      106  042127  047440  P2T10E: .ASCIZ  /FWD OSC SEEK/
006446  042522  020126  051517  P2T11E: .ASCIZ  /REV OSC SEEK/
006463      123  042505  020113  P2T12E: .ASCIZ  /SEEK TIMING/
006477      102  051501  041511  P2T13E: .ASCIZ  /BASIC READ DATA/
006517      127  052122  051057  P2T14E: .ASCIZ  &WRT/READ DATA (P1)&
006542  050123  047111  046104  P2T15E: .ASCIZ  /SPINDLE ROTATION TIMING/
006572  051127  027524  042522  P2T16E: .ASCIZ  &WRT/READ DATA (P2)&
006615      127  052124  046040  P2T17E: .ASCIZ  /WRT LCK ERR AND DATA PROTECTION/
006665      101  045104  041440  P2T18E: .ASCIZ  /ADJ CYL INTERFERENCE/
006702  053117  051105  052127  P2T19E: .ASCIZ  /OVERWRITE/
006714  042523  045505  052040  SKTMES: .ASCIZ  /SEEK TIMES /
006730  050123  047111  046104  SRTMES: .ASCIZ  /SPINDLE ROTATION TIME /
006757      050  052123  052101  VALDES: .ASCIZ  /(STATED IN 100'S OF MICRO SEC)/
007016  050101  051120  054117  MAPROX: .ASCIZ  /APPROX /
007026  047111  042516  000122  LABIN:  .ASCIZ  /INNER/
007034  044515  042104  042514  LABMID: .ASCIZ  /MIDDLE/
007043      117  052125  051105  LABOUT: .ASCIZ  /OUTER/
007051      105  050130  041505  LABEXP: .ASCIZ  /EXPECTED/
007062  030060  020061  054503  LABOCF: .ASCIZ  /001 CYL FWD/
007076  030060  020061  054503  LABOCR: .ASCIZ  /001 CYL REV/
```

```
   594  007112  031061  020070  054503  LABHCF: .ASCIZ  /128 CYL FWD/
   595  007126  031061  020070  054503  LABHCR: .ASCIZ  /128 CYL REV/
   596  007142  032462  020065  054503  LABACF: .ASCIZ  /255 CYL FWD/
   597  007156  032462  020065  054503  LABACR: .ASCIZ  /255 CYL REV/
   598  007172  042110  020123  040506  HDMOVF: .ASCIZ  /HDS FAILED TO MOVE IN 10 TRIES/
   616  007231     122  051505  052105  OPR12:  .ASCIZ  /RESET WRT LCK /
   617  007250  047117  000040          OPR1A:  .ASCIZ  /ON /
   618  007254  047117  042040  053122  OPR1B:  .ASCIZ  /ON DRV /
   619  007264  047125  042504  020122  UNDTST: .ASCIZ  /UNDER TEST/
   620  007277     123  052105  053440  OPR004: .ASCIZ  /SET WRT LCK /
   621  007314  044504  043106  000040  DIFWD:  .ASCIZ  /DIFF /
   622  007327  043523  020116     000  SGNWD:  .ASCIZ  /SGN /
   623  007337     110  020104  000040  HDWD:   .ASCIZ  /HD /
   624  007353     123  041505  000040  SECWD:  .ASCIZ  /SEC /
   625  007340  054503  020114     000  CYLWD:  .ASCIZ  /CYL /
   626  007345     106  047522  020115  FRMWD:  .ASCIZ  /FROM /
   627  007353     040  054502  040520  BYPSWM: .ASCIZ  / BYPASSED /
   628  007366  047522  052125  047111  SEQMES: .ASCIZ  /ROUTINE TRACE SEQ (IN SEQ CALLED):/
   629  007431     104  053122  051440  STAMES: .ASCIZ  /DRV STAT/
   630  007442  040502  020104  042523  BSNSTR: .ASCIZ  /BAD SEC FILES NOT STRD. ALL SEC ASSUMED GOOD./
   631  007520  047524  020124  047503  TCERR:  .ASCIZ  /TOT COMPARE ERRS: /
   632
   633                                   ;       RESULT NAMES
   634  007543     104  053122  051040  MDRDY:  .ASCIZ  /DRV RDY /
   635  007554  047503  052116  042440  MCERR:  .ASCIZ  /COMT ERR /
   636  007566  042110  020122  051103  MHCRC:  .ASCIZ  /HDR CRC/
   637  007576  040504  040524  041440  MDCRC:  .ASCIZ  /DATA CRC/
   638  007607     110  051104  047040  MHNF:   .ASCIZ  /HDR NOT FND/
   639  007623     104  052101  020101  MDLT:   .ASCIZ  /DATA LATE/
   640  007635     110  051104  047040  MHFCRC: .ASCIZ  &HDR NOT FND/HDR CRC/OPI&
   641  007665     104  052122  042440  MDRERR: .ASCIZ  /DRV ERR /
   650  007676  051104  020122  042523  MDSERR: .ASCIZ  /DRV SEL ERR /
   651  007713     104  053122  051440  MDRVST: .ASCIZ  /DRV STATE /
   652  007726  050123  047111  052040  MSPERR: .ASCIZ  /SPIN TIMEOUT /
   653  007744  051127  020124  040507  MWGERR: .ASCIZ  /WRT GAT ERR /
   654  007761     123  042505  020113  MSTERR: .ASCIZ  /SEEK TIMEOUT /
   655  007777     110  040505  020104  MHCERR: .ASCIZ  /HEAD CUR ERR /
   656  010015     127  052122  042040  MWDERR: .ASCIZ  /WRT DAT ERR /
   657  010032  050117  044440  041516  MOPERR: .ASCIZ  /OP INCOMPLETE /
   658  010051     110  051104  042057  MHDERR: .ASCIZ  &HDR/DAT ERR &
   659  010066  042110  020122  047516  MFLERR: .ASCIZ  &HDR NOT FND/DAT LATE &
   660  010114  047516  026516  054105  MNEERR: .ASCIZ  /NON-EXSTNT MEM /
   661  010134  054503  020114     000  MCYLOC: .ASCIZ  /CYL /
   662  010141     103  052117  042114  MNDRST: .ASCIZ  /COULD NOT RETRIEVE DRIVE STATUS/
   663  010201     125  046516  020116  MUNDEF: .ASCIZ  /UNKN DRV STATE-NO RDY,NO ERR,HDS OUT/
   664  010246  040506  046111  052040  MRLFAL: .ASCIZ  /FAIL TO RELD HDS AFTER ERR CLEAR/
   665  010307     127  044522  042524  MWRTAB: .ASCIZ  /WRITE ABORTED/
   666  010335     040  051105  020122  MEXERS: .ASCIZ  / ERR LIMIT EXCEEDED - UNIT DROPPED/
   667  010370  042440  051122  051105  MERRS:  .ASCIZ  / ERROR/
   668  010377     207  177777     000  BELL:   .ASCIZ  <207><377><377>
   669
   670                                   ;       RESULT SETTINGS
   671  010403     111  020123     000  RESE3:  .ASCIZ  /IS /
   672  010407     040  041123  000040  RESE4:  .ASCIZ  / SB /
   673
   674                                   ;       RESULT CONDITIONS
```



```
   675  010414  044440  020116     000  RESE5:  .ASCIZ  / IN /
   676  010421     040  043117  000040  RESE6:  .ASCIZ  / OF /
   677  010426  052123  052101  020105  STATE2: .ASCIZ  /STATE 2/
   678  010436  052123  052101  020105  STATE3: .ASCIZ  /STATE 3/
   679  010446  052123  052101  020105  STATE5: .ASCIZ  /STATE 5/
   683  010456  044506  051522  020124  C10MS:  .ASCIZ  /FIRST 3 MS/
   684  010471     065  030060  051515  C500MS: .ASCIZ  /500MS/
   685  010477     103  041531  042514  CCYLUP: .ASCIZ  /CYCLE UP/
   686  010510  040504  040524  054040  CAFDT:  .ASCIZ  /DATA XFER/
   687  010522  020065  042523  042103  C5SEC:  .ASCIZ  /5 SECDS/
   688
   689  010532  047045  052045  047045  FMTOP1: .ASCIZ  /%N%T%N%T%T%O6%S%T%O1%N/
   690  010561     045  022516  022524  FMTOP2: .ASCIZ  /%N%T%O1%S1%T%O1%N/
   691  010603     045  022516  022524  FMTOP3: .ASCIZ  /%N%T%O1%S1%T%T%N/
   692  010624  052045  052045     000  FMT1:   .ASCIZ  /%T%T/
   693  010631     045  022516  022524  FMT1.1: .ASCIZ  /%N%T%T/
   694  010640  052045  052045     000  FMT2:   .ASCIZ  /%T/
   695  010643     045  000116          FMT3:   .ASCIZ  /%N/
   696  010646  047045  052045  052045  FMT4:   .ASCIZ  /%N%T%T%N/
   697  010657     045  022516  022524  FMT5:   .ASCIZ  /%N%T%O6%S1%T%O1/
   698  010677     045  022516  030523  FMT6:   .ASCIZ  /%N%S11%T%S4%T%S4%T%S4%T%S4%T%S2%T/
   699  010741     045  022516  022524  FMT7:   .ASCIZ  /%N%T%O6%S2%O6%S2%O6%S2%O6%S3%O3%S2%O1%N/
   700  011011     045  022516  022524  FMT8:   .ASCIZ  /%N%T%O6%S2%O6%S2%O6%S2%O6/
   701  011043     045  022516  000124  FMT9:   .ASCIZ  /%N%T/
   702  011050  052045  047445  000061  FMT11:  .ASCIZ  /%T%O1/
   703  011056  052045  047445  000063  FMT12:  .ASCIZ  /%T%O3/
   704  011064  047045  051445  030461  FMT13:  .ASCIZ  /%N%S11%T%O3%S1%T%O3%S1%T%O1%S1%T%O1/
   705  011130  047045  052045  052045  FMT14:  .ASCIZ  /%N%T%T%D3%S1%T%O6%S1%T%O6/
   706  011162  047045  051445  030461  FMT15:  .ASCIZ  /%N%S11%T%D3%S1%T%O6%S1%T%O6/
   707  011216  047045  051445  022465  FMT16:  .ASCIZ  /%N%S5%O6/
   708  011227     045  030523  022460  FMT17:  .ASCIZ  /%S10%T%N%S11%O6%N/
   709  011251     045  022516  030523  FMT18:  .ASCIZ  /%N%S13%T%S5%T%S4%T%S5%T%N/
   710  011303     045  022524  031123  FMT19:  .ASCIZ  /%T%S2%D6%S4%D6%S4%D6%S4%D6%N/
   711  011340  052045  051445  022461  FMT20:  .ASCIZ  /%T%S2%D6%S14%D6%S4%D6%N/
   712  011370  052045  051445  031041  FMT21:  .ASCIZ  /%T%S12%D6%S14%D6%N/
   713  011413     045  022516  030523  FMT22:  .ASCIZ  /%N%S11%T%O3%S1%T%O1%S1%T%O2/
   714  011447     045  022524  022524  FMT23:  .ASCIZ  /%T%T%T%O1%N/
   715  011463     045  022516  000124  FMT24:  .ASCIZ  /%N%T/
   716  011470  047045  042045  022462  FMT25:  .ASCIZ  /%N%D2%T/
   717  011500  047045  051445  022461  FMT26:  .ASCIZ  /%N%S1%T%D4%T%T%D3%N/
   718  011524  047045  052045  042045  FMT27:  .ASCIZ  /%N%T%D3%T%D3%N/
   719  011543     045  022516  022524  FMT28:  .ASCIZ  /%N%T%T%T/
   720  011554                          ENDMOD
   725
   726  011554                          BGNMOD  GLBERR
   727                                   ;       ERR1    R3 POINTS TO RESULT MESSAGE
   728                                   ;               RESULT: (R3)
   729
   730                                   ;       ERR2    R3 POINTS TO RESULT NAME
   731                                   ;               RESULT: (R3) IS 1 SB 0
   732
   733                                   ;       ERR3    R3 POINTS TO RESULT NAME
   734                                   ;               RESULT: (R3) IS 0 SB 1
   735
   736                                   ;       ERR4    R3 POINTS TO RESULT NAME
   737                                   ;               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)
                                
011554                          BGNMSG  ERR1
011554  105737  003067                TSTB    NOERCT          ;TEST IF ERROR COUNTING INHIBITED
011560  001002                  BNE     1$              ;YES - SKIP
011562  005277  171072          INC     @ERRPOINT       ;ELSE BUMP ERROR COUNT
011566  010146                1$:     MOV     R1,-(SP)        ;STORE R1
011570  004737  023244                JSR     PC,RPTOP        ;REPORT OPERATION
011574  012721  000001                MOV     #1,(R1)+        ;SET PARAM NUMBER
011600  010321                  MOV     R3,(R1)+        ;INSERT MESSAGE ADDRESS POINTER
011602  004737  024032                JSR     PC,RPTRES       ;REPORT RESULTS
011606  004737  024240                JSR     PC,RPTREM       ;REPORT REMAINDER
011612  012601                  MOV     (SP)+,R1        ;RESTORE R1
011614  004737  014634                JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
011620                          ENDMSG
(3) 011620                      L10000:
(3) 011620  104023                      EMT     C$MSG
                                
011622                          BGNMSG  ERR2
011622  005277  171032                INC     @ERRPOINT       ;BUMP ERROR COUNT
011626  010146                  MOV     R1,-(SP)        ;STORE R1
011630  004737  023244                JSR     PC,RPTOP        ;REPORT OPERATION
011634  012721  000003                MOV     #3,(R1)+        ;SET PARAM NUMBER
011640  010321                  MOV     R3,(R1)+        ;INSERT NAME ADD POINTER
011642  012721  000001                MOV     #1,(R1)+        ;SET IS VALUE
```



```
011646  005021                  CLR     (R1)+           ;SET SB VALUE
011650  004737  024032                JSR     PC,RPTRES       ;REPORT RESULTS
011654  004737  024240                JSR     PC,RPTREM       ;REPORT REMAINDER
011660  012601                  MOV     (SP)+,R1        ;RESTORE R1
011662  004737  014634                JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
011666                          ENDMSG
(3) 011666                      L10001:
(3) 011666  104023                      EMT     C$MSG
                                
011670                          BGNMSG  ERR3
011670  005277  170764                INC     @ERRPOINT       ;BUMP ERROR COUNT
011674  010146                  MOV     R1,-(SP)        ;STORE R1
011676  004737  023244                JSR     PC,RPTOP        ;REPORT OPERATION
011702  012721  000003                MOV     #3,(R1)+        ;SET PARAM NUMBER
011706  010321                  MOV     R3,(R1)+        ;INSERT NAME ADD POINTER
011710  005021                  CLR     (R1)+           ;SET IS VALUE
011712  012721  000001                MOV     #1,(R1)+        ;SET SB VALUE
011716  004737  024032                JSR     PC,RPTRES       ;REPORT RESULTS
011722  004737  024240                JSR     PC,RPTREM       ;REPORT REMAINDER
011726  012601                  MOV     (SP)+,R1        ;RESTORE R1
011730  004737  014634                JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
011734                          ENDMSG
(3) 011734                      L10002:
(3) 011734  104023                      EMT     C$MSG
                                
011736                          BGNMSG  ERR4
011736  005277  170716                INC     @ERRPOINT       ;BUMP ERROR COUNT
011742  010146                  MOV     R1,-(SP)        ;STORE R1
011744  004737  023244                JSR     PC,RPTOP        ;REPORT OPERATION
011750  012721  000004                MOV     #4,(R1)+        ;SET PARAM NUMBER
011754  010321                  MOV     R3,(R1)+        ;INSERT NAME ADD POINTER
011756  012721  000001                MOV     #1,(R1)+        ;SET IS VALUE
011762  005021                  CLR     (R1)+           ;SET SB VALUE
011764  010411                  MOV     R4,(R1)         ;INSERT ADD OF CONDITION POINTER
011766  004737  024032                JSR     PC,RPTRES       ;REPORT RESULTS
011772  004737  024240                JSR     PC,RPTREM       ;REPORT REMAINDER
011776  012601                  MOV     (SP)+,R1        ;RESTORE R1
012000  004737  014634                JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
012004                          ENDMSG
(3) 012004                      L10003:
(3) 012004  104023                      EMT     C$MSG
                                
012006                          BGNMSG  ERR5
012006  005277  170646                INC     @ERRPOINT       ;BUMP ERROR COUNT
012012  010146                  MOV     R1,-(SP)        ;STORE R1
012014  004737  023244                JSR     PC,RPTOP        ;REPORT OPERATION
012020  012721  000004                MOV     #4,(R1)+        ;SET PARAM NUMBER
012024  010321                  MOV     R3,(R1)+        ;INSERT NAME ADD POINTER
012026  005021                  CLR     (R1)+           ;SET IS VALUE
012030  012721  000001                MOV     #1,(R1)+        ;SET SB VALUE
012034  010411                  MOV     R4,(R1)         ;INSERT ADD OF CONDITION POINTER
012036  004737  024032                JSR     PC,RPTRES       ;REPORT RESULTS
012042  004737  024240                JSR     PC,RPTREM       ;REPORT REMAINDER
012046  012601                  MOV     (SP)+,R1        ;RESTORE R1
012050  004737  014634                JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
012054                          ENDMSG
```

```
(3)  012054                          L10004:
(3)  012054  104023                  EMT     C$MSG
842
843  012056                          BGNMSG  ERR6
844  012056  105737  003067          TSTB    NOERCT          ;TEST IF ERROR COUNTING INHIBITED
845  012062  001002                  BNE     17$             ;YES - SKIP
846  012064  005277  170570          INC     @ERRPOINT       ;ELSE BUMP ERROR COUNT
847  012070  010146          17$:    MOV     R1,-(SP)        ;STORE R1
848  012072  010346                  MOV     R3,-(SP)        ;STORE R3
849  012074  010446                  MOV     R4,-(SP)        ;STORE R4
850  012076  010546                  MOV     R5,-(SP)        ;STORE R5
851  012100  004737  023244          JSR     PC,RPTOP        ;REPORT OPERATION
852  012104  012721  000003          MOV     #3,(R1)+        ;SET PARAM NUMBER
853  012110  012761  000001  000002  MOV     #1,2(R1)        ;INSERT IS VALUE
854  012116  005037  002546          CLR     TEMP3           ;CLEAR FOR STATUS STORAGE
855  012122  013703  002466          MOV     T.CS,R3         ;GET T.CS
856  012126  042703  177761          BIC     #177761,R3      ;AND CLEAR ALL BUT FUNCTION
857  012132  000004                  CMP     #4,R3           ;CHECK IF IT WAS GET STATUS
858  012136  001437                  BEQ     1$              ;YES - STATUS IS IN T.MP, SKIP
859  012140  012762  000003  000004  MOV     #GETSTAT,RLDA(R2) ;ELSE DO GET STATUS
860  012146  012703  000004          MOV     #4,R3
861  012152  053703  002454          BIS     RLDRV,R3
862  012156  010362  000000          MOV     R3,RLCS(R2)
863  012162                          WAITUS  #10.            ;WAIT FOR CONTROLLER READY
(3)  012162  012700  000012          MOV     #10.,R0
(3)  012166  104027                  EMT     CSWTO
864  012170  032762  000200  000000  BIT     #CRDYMSK,RLCS(R2)       ;TEST IF READY
865  012176  001003                  BNE     10$             ;YES - SKIP
866  012200  012703  001000  9$:     MOV     #BIT9,R3        ;ELSE SET NO DRIVE STATUS BIT
867  012204  000413                  BR      2$              ;IN MESSAGE WORD AND SKIP
868  012206  016203  000006  10$:    MOV     RLMP(R2),R3     ;STORE STATUS FOR REPORT
869  012212  010337  002546          MOV     R3,TEMP3
870  012216  113703  002547          MOVB    TEMP3+1,R3      ;GET ERROR BITS IN PROPER POSITION
871  012222  000402                  BR      13$
872  012224  113703  002475  1$:     MOVB    T.MP+1,R3       ;GET ERROR BITS FROM MP REG
873  012230  042703  177442  13$:    BIC     #177442,R3      ;CLEAR UNUSED BITS
874  012234  013704  002466  2$:     MOV     T.CS,R4         ;GET ERROR BITS FROM CS REG
875  012240  042704  001777          BIC     #1777,R4        ;CLEAR UNUSED BITS
876  012244  050403                  BIS     R4,R3           ;MAKE ONE WORD OF POSSIBLE ERRORS
877  012246  032703  002000          BIT     #OPIERR,R3      ;TEST IF OPI SET
878  012252  001442                  BEQ     11$             ;NO - SKIP
879  012254  032703  010000          BIT     #HNFERR,R3      ;TEST IF HDR NOT FOUND ERROR
880  012260  001026                  BNE     10$             ;YES - SKIP
881  012262  032703  004000          BIT     #HCRCERR,R3     ;TEST IF HDR CRC ERR
882  012266  001020                  BNE     10$             ;YES - SKIP
883  012270  012704  010032          MOV     #MOPERR,R4      ;SET OPI ALONE MESSAGE
884  012274          100$:   PRINTB  #FMT28,#MRSLT,R4,#MERRS ;REPORT ERROR
(10) 012274  012746  010370          MOV     #MERRS,-(SP)
(9)  012300  010446                  MOV     R4,-(SP)
(8)  012302  012746  005157          MOV     #MRSLT,-(SP)
(7)  012306  012746  011543          MOV     #FMT28,-(SP)
(6)  012312  012746  000004          MOV     #4,-(SP)
(3)  012316  010600                  MOV     SP,R0
(4)  012320  104014                  EMT     C$PNTB
(4)  012322  062706  000012          ADD     #12,SP
885  012326  000430                  BR      120$            ;SKIP
```



```
886  012330  012704  007566          105$:   MOV     #MHCRC,R4       ;HDR CRC MESSAGE
887  012334  000757                  BR      100$
888  012336  032703  004000          107$:   BIT     #HCRCERR,R3     ;TEST IF HCRC WITH HDR NOT FND
889  012342  001003                  BNE     109$            ;YES - SKIP
890  012344  012704  007607          MOV     #MHNF,R4        ;MESSAGE HEADER NOT FOUND
891  012350  000751                  BR      100$
892  012352  012704  007635          109$:   MOV     #MHFCRC,R4      ;HNF AND HCRC MESSAGE
893  012356  000746                  BR      100$            ;SKIP
894  012360  032703  004000          115$:   BIT     #DCKERR,R3      ;TEST IF DATA CHECK SET, NOT OPI
895  012364  001403                  BEQ     118$            ;NO - SKIP
896  012366  012704  007576          MOV     #MDCRC,R4       ;SET MESSAGE DATA CHECK
897  012372  000740                  BR      100$            ;SKIP
898  012374  032703  010000          118$:   BIT     #DLTERR,R3      ;TEST IF DATA LATE ERROR
899  012400  001403                  BEQ     120$            ;NO - SKIP
900  012402  012704  007623          MOV     #MDLT,R4        ;SET MESSAGE DATA LATE
901  012406  000732                  BR      100$            ;SKIP
902  012410  012705  100000          120$:   MOV     #BIT15,R5       ;SET BIT POINTER FOR TEST
903  012414  005004                  CLR     R4              ;CLEAR R4 FOR TABLE COUNT
904  012416  030503                  3$:     BIT     R5,R3           ;TEST IF BIT IS SET
905  012420  001005                  BNE     6$              ;YES - SKIP TO REPORT
906  012422  005724                  4$:     TST     (R4)+           ;ELSE BUMP TABLE POINTER
907  012424  000241                  CLC                     ;CLEAR CARRY
908  012426  006005                  ROR     R5              ;SHIFT BIT POINTER TO NEXT BIT
909  012430  001372                  BNE     3$              ;LOOP IF NOT 0
910  012432  000405                  BR      7$              ;ELSE REPORT REMAINDER
911  012434  016411  002174          6$:     MOV     RESTBL(R4),(R1) ;INSERT NAME ADDRESS
912  012440  004737  024032          JSR     PC,RPTRES       ;REPORT RESULTS
913  012444  000766                  BR      4$              ;GET NEXT BIT
914  012446  004737  024240          7$:     JSR     PC,RPTREM       ;REPORT REMAINDER
915  012452  005737  002546          TST     TEMP3           ;TEST IF ANY NEW STATUS
916  012456  001414                  BEQ     15$             ;NO - SKIP
917  012460                          PRINTB  #FMT17,#STAMES,TEMP3
(9)  012460  013746  002546          MOV     TEMP3,-(SP)
(8)  012464  012746  007431          MOV     #STAMES,-(SP)
(7)  012470  012746  011227          MOV     #FMT17,-(SP)
(6)  012474  012746  000003          MOV     #3,-(SP)
(3)  012500  010600                  MOV     SP,R0
(4)  012502  104014                  EMT     C$PNTB
(4)  012504  062706  000010          ADD     #10,SP
918  012510  032737  004000  002466  15$:    BIT     #DCKERR,T.CS    ;TEST IF DATA CHECK ERROR
919  012516  001453                  BEQ     25$             ;NO - SKIP
920  012520  032737  002000  002466  BIT     #OPIERR,T.CS    ;TEST IF OPI SET
921  012526  001047                  BNE     25$             ;YES - SKIP
922  012530  005037  002436          CLR     MORECE          ;CLEAR COMPARE ERROR COUNT
923  012534  012701  000200          MOV     #128.,R1        ;SET COMPARE LENGTH
924  012540  012703  000001          MOV     #1,R3           ;SET WORD COUNT
925  012544  012705  004066          MOV     #OBUFF,R5       ;SET GOOD WORD POINTER
926  012550  012704  003466          MOV     #IBUFF,R4       ;SET TEST WORD POINTER
927  012554  021514          18$:    CMP     (R5),(R4)       ;CHECK WORD
928  012556  001427                  BEQ     19$             ;GOOD - SKIP
929  012560  023727  002436  000012  CMP     MORECE,#10.     ;TEST IF COMPARE LIMIT REACHED
930  012566  003021                  BGT     20$             ;YES - SKIP
931  012570                          PRINTB  #FMT15,#MWORD,R3,#RESE3,(R4),#RESE4,(R5)
(13) 012570  011546                  MOV     (R5),-(SP)
(12) 012572  012746  010407          MOV     #RESE4,-(SP)
(11) 012576  011446                  MOV     (R4),-(SP)
```

```
(10) 012600  012746  010403                    MOV     #RESE3,-(SP)
 (9) 012604  010346                            MOV     R3,-(SP)
 (8) 012606  012746  006005                    MOV     #MWORD,-(SP)
 (7) 012612  012746  011162                    MOV     #FMT15,-(SP)
 (6) 012616  012746  000007                    MOV     #7,-(SP)
 (3) 012622  010600                            MOV     SP,R0
 (4) 012624  104014                            EMT     C$PNTB
 (4) 012626  062706  000020                    ADD     #20,SP
 932 012632  005237  002436          20$:      INC     MORECE          ;BUMP ERROR COUNTER
 933 012636  022524                  19$:      CMP     (R5)+,(R4)+     ;BUMP POINTERS
 934 012640  005203                            INC     R3              ;BUMP COUNTER
 935 012642  005301                            DEC     R1              ;DEC LENGTH COUNT
 936 012644  001343                            BNE     18$             ;LOOP IF NOT DONE
 937 012646  005737  002436          25$:      TST     MORECE          ;TEST IF ANY COMPARE ERRORS
 938 012652  001421                            BEQ     27$             ;NO - SKIP
 939 012654  012701  000200                    MOV     #128.,R1        ;SET COMPARE LENGTH
 940 012660                                    PRINTB  #FMT27,#TCERR,MORECE,#RESE6,R1
(11) 012660  010146                            MOV     R1,-(SP)
(10) 012662  012746  010421                    MOV     #RESE6,-(SP)
 (9) 012666  013746  002436                    MOV     MORECE,-(SP)
 (8) 012672  012746  007520                    MOV     #TCERR,-(SP)
 (7) 012676  012746  011524                    MOV     #FMT27,-(SP)
 (6) 012702  012746  000005                    MOV     #5,-(SP)
 (3) 012706  010600                            MOV     SP,R0
 (4) 012710  104014                            EMT     C$PNTB
 (4) 012712  062706  000014                    ADD     #14,SP
 941 012716  012605                  27$:      MOV     (SP)+,R5        ;RESTORE R5, 4, 3, 1
 942 012720  012604                            MOV     (SP)+,R4
 943 012722  012603                            MOV     (SP)+,R3
 944 012724  012601                            MOV     (SP)+,R1
 945 012726  004737  014634                    JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
 946 012732                          ENDMSG
 (3) 012732                          L10005:
 (3) 012732  104023                            EMT     C$MSG
 947
 948 012734                          BGNMSG    ERR7
 949 012734  005277  167720                    INC     @ERRPOINT       ;BUMP ERROR COUNT
 950 012740  010146                            MOV     R1,-(SP)        ;STORE R1
 951 012742  004737  023244                    JSR     PC,RPTOP        ;REPORT OPERATION
 952 012746  012721  000003                    MOV     #3,(R1)+        ;SET PARAM NUMBER
 953 012752  012721  007713                    MOV     #MDRVST,(R1)+   ;INSERT NAME ADD POINTER
 954 012756  013721  002502                    MOV     I.STAT(R1)+     ;INSERT IS VALUE
 955 012762  010311                            MOV     R3,(R1)+ ;INSERT SB VALUE
 956 012764  004737  024032                    JSR     PC,RPTRES       ;REPORT RESULTS
 957 012770  004737  024240                    JSR     PC,RPTREM       ;REPORT REMAINDER
 958 012774  012601                            MOV     (SP)+,R1        ;RESTORE R1
 959 012776  004737  014634                    JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
 960 013002                          ENDMSG
 (3) 013002                          L10006:
 (3) 013002  104023                            EMT     C$MSG
 961
 962 013004                          BGNMSG    ERR8
 963 013004  005277  167650                    INC     @ERRPOINT       ;BUMP ERROR COUNT
 964 013010  010146                            MOV     R1,-(SP)        ;STORE R1
 965 013012  010346                            MOV     R3,-(SP)        ;STORE R3
 966 013014  004737  023244                    JSR     PC,RPTOP        ;REPORT OPERATION
```



```
 967 013020  012721  000003                    MOV     #3,(R1)+        ;SET PARAM NUMBER
 968 013024  012721  010134                    MOV     #MCYLOC,(R1)+   ;INSERT NAME ADD POINTER
 969 013030  013711  002474                    MOV     HDWRD1,(R1)     ;GET HEADER WORD
 970 013034  012703  000007                    MOV     #7,R3           ;SET SHIFT COUNT
 971 013040  000241                  3$:       CLC
 972 013042  006011                            ROR     (R1)            ;ALIGN CHAR FOR PRINTING
 973 013044  005303                            DEC     R3              ;  AS IS VALUE
 974 013046  001374                            BNE     3$
 975 013050  005721                            TST     (R1)+           ;BUMP PARAM POINTER
 976 013052  013711  002524                    MOV     NEWCYL,(R1)     ;INSERT SB VALUE
 977 013056  004737  024032                    JSR     PC,RPTRES       ;REPORT RESULTS
 978 013062  004737  024240                    JSR     PC,RPTREM       ;REPORT REMAINDER
 979 013066  012603                            MOV     (SP)+,R3        ;RESTORE R3
 980 013070  012601                            MOV     (SP)+,R1        ;RESTORE R1
 981 013072  004737  014634                    JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
 982 013076                          ENDMSG
 (3) 013076                          L10007:
 (3) 013076  104023                            EMT     C$MSG
 983
 984 013100                          BGNMSG    ERR9
 985 013100  005277  167554                    INC     @ERRPOINT       ;BUMP ERROR COUNT
 986 013104  010146                            MOV     R1,-(SP)        ;STORE R1
 987 013106  004737  023244                    JSR     PC,RPTOP        ;REPORT OPERATION
 988 013112  012721  000003                    MOV     #3,(R1)+        ;SET PARAM NUMBER
 989 013116  010321                            MOV     R3,(R1)+        ;INSERT NAME ADD POINTER
 990 013120  010421                            MOV     R4,(R1)+        ;SET IS VALUE
 991 013122  010521                            MOV     R5,(R1)+        ;SET SB VALUE
 992 013124  004737  024032                    JSR     PC,RPTRES       ;REPORT RESULTS
 993 013130  004737  024240                    JSR     PC,RPTREM       ;REPORT REMAINDER
 994 013134  012601                            MOV     (SP)+,R1        ;RESTORE R1
 995 013136  004737  014634                    JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
 996 013142                          ENDMSG
 (3) 013142                          L10010:
 (3) 013142  104023                            EMT     C$MSG
 997 013144                          BGNMSG    ERR10
 998 013144  010146                            MOV     R1,-(SP)        ;STORE R1
 999 013146  005737  002436                    TST     MORECE          ;TEST IF 2ND BAD LINE
1000 013152  001051                            BNE     3$              ;YES - SKIP
1001 013154  005277  167500                    INC     @ERRPOINT       ;BUMP ERROR COUNT
1002 013160  004737  023244                    JSR     PC,RPTOP        ;REPORT OPERATION
1003 013164                                    PRINTB  #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1> ;REPORT ID
 (11) 013164  005046                            CLR     -(SP)
 (11) 013166  153716  002455                    BISB    RLDRV+1,(SP)
 (10) 013172  012746  005633                    MOV     #DRVNAM,-(SP)
  (9) 013176  013746  002450                    MOV     RLBAS,-(SP)
  (8) 013202  012746  005622                    MOV     #BASADD,-(SP)
  (7) 013206  012746  010657                    MOV     #FMT5,-(SP)
  (6) 013212  012746  000005                    MOV     #5,-(SP)
  (3) 013216  010600                            MOV     SP,R0
  (4) 013220  104014                            EMT     C$PNTB
  (4) 013222  062706  000014                    ADD     #14,SP
1004 013226                                    PRINTB  #FMT14,#MRSLT,#MWORD,R3,#RESE3,(R4),#RESE4,(R5)
 (14) 013226  011546                            MOV     (R5),-(SP)
 (13) 013230  012746  010407                    MOV     #RESE4,-(SP)
 (12) 013234  011446                            MOV     (R4),-(SP)
 (11) 013236  012746  010403                    MOV     #RESE3,-(SP)
```

```
  (10) 013242  010346                    MOV     R3,-(SP)
   (9) 013244  012746  006005            MOV     #MWORD,-(SP)
   (8) 013250  012746  005157            MOV     #MRSLT,-(SP)
   (7) 013254  012746  011130            MOV     #FMT14,-(SP)
   (6) 013260  012746  000010            MOV     #10.,-(SP)
   (3) 013264  010600                    MOV     SP,R0
   (4) 013266  104014                    EMT     C$PNTB
   (4) 013270  062706  000022            ADD     #22,SP
  1005 013274  000421                    BR      4$
  1006 013276                    3$:     PRINTB  #FMT15,#MWORD,R3,#RESE3,(R4),#RESE4,(R5) ;REPORT DATA
  (13) 013276  011546                    MOV     (R5),-(SP)
  (12) 013300  012746  010407            MOV     #RESE4,-(SP)
  (11) 013304  011446                    MOV     (R4),-(SP)
  (10) 013306  012746  010403            MOV     #RESE3,-(SP)
   (9) 013312  010346                    MOV     R3,-(SP)
   (8) 013314  012746  006005            MOV     #MWORD,-(SP)
   (7) 013320  012746  011162            MOV     #FMT15,-(SP)
   (6) 013324  012746  000007            MOV     #7,-(SP)
   (3) 013330  010600                    MOV     SP,R0
   (4) 013332  104014                    EMT     C$PNTB
   (4) 013334  062706  000020            ADD     #20,SP
  1007 013340  005237  002436    4$:     INC     MORECE          ;INC COMPARE ERROR COUNT
  1008 013344  012601                    MOV     (SP)+,R1        ;RESTORE R1
  1009 013346  004737  014634            JSR     PC,CKERCM       ;GO CHECK IF ERROR COUNT EXCEEDED
  1010 013352                    ENDMSG
   (3) 013352                    L10011:
   (3) 013352  104023                    EMT     C$MSG
  1011 013354                    ENDMOD
  1012                                   .EVEN
  1013
  1014 013354                    BGNMOD  HPTCODE
  1015 013354                    BGNHW
   (3) 013354  000005                    .WORD   L10012-L$HW/2
  1016 013356  174400                    .WORD   174400          ;CSR BASE ADDRESS DEFAULT
  1017 013360  000160                    .WORD   160             ;VECTOR DEFAULT
  1018 013362  000240                    .WORD   240             ;PRIORITY DEFAULT
  1019 013364  000000                    .WORD   0               ;DRIVE NUMBER DEFAULT
  1020 013366  000001                    .WORD   1               ;RL11 CONTROLLER
  1021 013370                    ENDHW
   (3) 013370                    L10012:
  1022 013370                    ENDMOD
  1023
  1024 013370                    BGNMOD  SPTCODE
  1025 013370                    BGNSW
   (3) 013370  000006                    .WORD   L10013-L$SW/2
  1026 013372  000000            MISWIW: .WORD   0               ;BIT 0 = USE ALL CYLINDERS
  1027                                                           ;BIT 1 = USE ALL SECTORS
  1028                                                           ;BIT 2 = EXECUTE DRIVE SELECT TEST
  1029                                                           ;BIT 3 = EXECUTE HEAD ALIGNMENT
  1030                                                           ;BIT 4 = DROP DRIVE IF NO RESPONSE
  1031                                                           ;BIT 12 = HEAD SELECT SUPPLIED FLAG
  1032                                                           ;BIT 13 = HILIMIT SPECIFIED FLAG
  1033                                                           ;BIT 14 = LO LIMIT SPECIFIED FLAG
  1034                                                           ;BIT 15 = DO MANUAL INTERVENTION
  1035 013374  000000            LOLIMW: .WORD   0
  1036 013376  000377            HILIMW: .WORD   255.
```



```
  1037 013400  000000            HEADW:  .WORD   0
  1038 013402  000024            ERLIMW: .WORD   20.             ;ERROR LIMIT
  1039 013404  000012            DCLIMW: .WORD   10.             ;COMPARE ERROR LIMIT
  1040 013406                    END$W
   (3) 013406                    L10013:
  1041 013406                    ENDMOD
  1042
  1043 013406                    BGNMOD  DSPCODE
  1048 013406                    DISPATCH
   (4) 013406  000023                    .WORD   19
   (6) 013410  024524                    .WORD   T1
   (6) 013412  025046                    .WORD   T2
   (6) 013414  025336                    .WORD   T3
   (6) 013416  025550                    .WORD   T4
   (6) 013420  025760                    .WORD   T5
   (6) 013422  026170                    .WORD   T6
   (6) 013424  026414                    .WORD   T7
   (6) 013426  026614                    .WORD   T8
   (6) 013430  027042                    .WORD   T9
   (6) 013432  027336                    .WORD   T10
   (6) 013434  027634                    .WORD   T11
   (6) 013436  030132                    .WORD   T12
   (6) 013440  031700                    .WORD   T13
   (6) 013442  032410                    .WORD   T14
   (6) 013444  032624                    .WORD   T15
   (6) 013446  033350                    .WORD   T16
   (6) 013450  034372                    .WORD   T17
   (6) 013452  035364                    .WORD   T18
   (6) 013454  036474                    .WORD   T19
  1050 013456                    ENDMOD
  1051
  1052 013456                    BGNMOD  INITCODE
  1053 013456                    BGNINIT
  1054 013456                            SETPRI  #340
   (3) 013456  012700  000340            MOV     #340,R0
   (3) 013462  104041                    EMT     C$SPRI
  1055 013464                            MANUAL                  ;CHECK IF MANUAL INTERVENTION ALLOWED
   (3) 013464  104051                    EMT     C$MANI
  1056 013466                            BCOMPLETE       1$      ;YES - SKIP
   (2) 013466  103403                    BCS     1$
  1057 013470  042737  100014  013372    BIC     #MITEST!DRSELT!HDALIGN,MISWIW ;CLEAR ALL MANUAL
  1058                                                           ; INTERVENTION FLAGS
  1059 013476  005037  002424    1$:     CLR     SSINDX          ;CLEAR SUBROUTINE STACK INDEX
  1060 013502                            READEF  #EF.PWR         ;POWER FAILURE
   (3) 013502  012700  000034            MOV     #EF.PWR,R0
   (3) 013506  104050                    EMT     C$REFG
  1061 013510                            BNCOMPLETE  4$          ;NO, GO CHECK NEW PASS
   (2) 013510  103004                    BCC     4$
  1062 013512  013737  002012  003072    MOV     L$UNIT,PWRFLG   ;SET POWER FAIL FLAG
  1063 013520  000531                    BR      PWCON           ;GO SERVICE POWER FAIL
  1064 013522                    4$:     READEF  #EF.START       ;CHECK IF START
   (3) 013522  012700  000040            MOV     #EF.START,R0
   (3) 013526  104050                    EMT     C$REFG
  1065 013530                            BNCOMPLETE      RESTART ;NO - SKIP
   (2) 013530  103043                    BCC     RESTART
  1066                            ;       ON START INITIALIZE TO START AT FIRST DRIVE, CLEAR INTERNAL
```

```
                                ;       PASS COUNT, AND ERROR COUNT.
013532  013737  002012  002516          MOV     L$UNIT,DRVCNT   ;SET UP UNIT COUNT
013540  005037  003062          RSTRT:  CLR     PASNUM          ;CLEAR PASS NUMBER
013544  012700  002662                  MOV     #ERRCNT,R0
013550  012701  000100                  MOV     #64.,R1         ;GET A COUNT
013554  005020                  1$:     CLR     (R0)+           ;CLEAR A ERROR COUNTER STORAGE AREA
013556  005301                          DEC     R1
013560  001375                          BNE     1$              ;LOOP TILL ALL CLEARED
013562  012737  002660  002660          MOV     #ERRCNT-2,ERRPOINT ;INIT ERROR POINTER
013570  012737  177777  003064          MOV     #-1,PSETNM      ;SET PARAM SELECT TO INITIAL VALUE
013576  012737  177777  002432          MOV     #-1,HADONE      ;PRESET HEAD ALIGN DONE FLAG
013604  032737  020000  013372          BIT     #HICYL,MISWIW   ;TEST IF HI LIMIT SET
013612  001003                          BNE     3$              ;YES - SKIP
013614  012737  000377  013376          MOV     #377,HILIMW     ;ELSE INIT HILIMIT
013622  032737  040000  013372  3$:     BIT     #LOCYL,MISWIW   ;TEST IF LO LIMIT SET
013630  001002                          BNE     5$              ;YES - SKIP
013632  005037  013374                  CLR     LOLIMW          ;ELSE CLEAR LO LIMIT
013636  000432                  5$:     BR      SETDON
013640                          RESTART:
013640                                  READEF  #EF.RESTART     ;CHECK IF RESTART
013640  012700  000037                  MOV     #EF.RESTART,R0
013644  104050                          EMT     CSREFG
013646                                  BCOMPLETE       RSTRT   ;NO - SKIP
013646  103734                          BCS     RSTRT
013650                          CONTINUE:
013650                                  READEF  #EF.CONTINUE    ;TEST IF CONTINUE
013650  012700  000036                  MOV     #EF.CONTINUE,R0
013654  104050                          EMT     CSREFG
013656                                  BCOMPLETE       PWCON
013656  103452                          BCS     PWCON
                                ;       ON CONTINUE PICK UP UNIT LAST UNDER TEST
013660                                  READEF  #EF.NEW         ;CHECK IF STARTING NEW PASS
013660  012700  000035                  MOV     #EF.NEW,R0
013664  104050                          EMT     CSREFG
013666                                  BCOMPLETE       PASNEW
013666  103403                          BCS     PASNEW
013670                          NXTPAS:
013670  005737  002516                  TST     DRVCNT          ;TEST IF ALL UNITS CHECKED
013674  001013                          BNE     SETDON          ;NO - SKIP
013676  005237  003062          PASNEW: INC     PASNUM          ;ELSE BUMP PASS COUNT
013702  012737  002660  002660          MOV     #ERRCNT-2,ERRPOINT ;INIT THE ERROR POINTER
013710  013737  002012  002516          MOV     L$UNIT,DRVCNT   ;GET ALL DRIVES
013716  012737  177777  003064          MOV     #-1,PSETNM      ;SET PARAM SELECT TO INITIAL
013724  005237  003064          SETDON: INC     PSETNM          ;NEXT SET OF PARAMETERS
013730  005337  002516                  DEC     DRVCNT          ;DOWN COUNT DRIVE TOTAL
013734  062737  000002  002660          ADD     #2,ERRPOINT     ;UPDATE THE ERROR POINTER
013742  013700  003064                  MOV     PSETNM,R0       ;SET UP TO GET PARAMETERS
013746  012702  002450                  MOV     #RLBAS,R2
013752                                  GPHARD  R0,R1
013752  104042                          EMT     CSGPHRD
013754  010001                          MOV     R0,R1
013756                                  BCOMPLETE       7$      ;SKIP IF GOOD PARAM
013756  103406                          BCS     7$
013760  005737  003072                  TST     PWRFLG          ;RECENT POWER FAILURE
013764  001741                          BEQ     NXTPAS  ;NO
013766  005337  003072                  DEC     PWRFLG          ;ACCOUNT FOR DRIVE
```



```
000736                                  BR      NXTPAS
012122                          7$:     MOV     (R1)+,(R2)+     ;STORE PARAMETERS CSR
012122                                  MOV     (R1)+,(R2)+     ;               VECTOR
005721                                  TST     (R1)+           ;BUMP PAST PRIORITY
012122                                  MOV     (R1)+,(R2)+     ;                   DRIVE

014004                          PWCON:  SETVEC  RLVEC,#INTHLR,#340      ;SET UP VECTOR
014004  012746  000340                  MOV     #340,-(SP)
014010  012746  014576                  MOV     #INTHLR,-(SP)
014014  013746  002452                  MOV     RLVEC,-(SP)
014020  012746  000003                  MOV     #3,-(SP)
014024  104037                          EMT     CSSVEC
014026  062706  000010                  ADD     #10,SP
014032                          SETPRI  #0              ;SET PRIORITY
014032  012700  000000                  MOV     #0,R0
014036  104041                          EMT     CSSPRI
014040  013702  002450                  MOV     RLBAS,R2        ;SET RL BASE ADDRESS POINTER


                                ;       CHECK IF DOING AUTO SIZE AND DROP DRIVE IF NOT READY AND
                                ;       ERROR SETS ON GET STATUS.
014044  005737  003062                  TST     PASNUM          ;TEST IF PASS 0
014050  001135                          BNE     22$             ;NO - SKIP
014052  032737  000020  013372          BIT     #AUTOSZ,MISWIW  ;TEST IF DOING AUTO SIZE
014060  001531                          BEQ     22$             ;NO - SKIP
                                ;CHECK IF UNIBUS ADDRESS IS THERE BEFORE WE CHECK DRIVE READY
014062  005037  003070                  CLR     TRPFLG          ;TRAP OCCURANCE
014066                                  SETVEC  ERRVEC,#TRPHAN,#340     ;SET TRAP VECTOR
014066  012746  000340                  MOV     #340,-(SP)
014072  012746  014570                  MOV     #TRPHAN,-(SP)
014076  013746  002652                  MOV     ERRVEC,-(SP)
014102  012746  000003                  MOV     #3,-(SP)
014106  104037                          EMT     CSSVEC
014110  062706  000010                  ADD     #10,SP
014114  005762  000000                  TST     RLCS(R2)        ;ACCESS BUS
014120  005737  003070                  TST     TRPFLG          ;TRAP OCCUR??
014124  001032                          BNE     5$              ;YES, DON'T INVESTIGATE FURTHER
014126  013705  002454                  MOV     RLDRV,R5        ;GET DRIVE NUMBER
014132  052705  000200                  BIS     #CRDYMSK,R5     ;INSERT CONT READY
014136  010562  000000                  MOV     R5,RLCS(R2)     ;LOAD IN DRIVE NUMBER
014142  032762  000001  000000          BIT     #DRDYMSK,RLCS(R2) ;CHECK IF DRIVE IS READY
014150  001072                          BNE     20$             ;YES - GO DO TEST
014152  012762  000003  000004          MOV     #GETSTAT,RLDA(R2) ;ELSE INSERT GET STATUS
014160  052705  000004                  BIS     #4,R5           ;LOAD R5 WITH GET STATUS FUNCTION
014164  042705  000200                  BIC     #CRDYMSK,R5     ;CLEAR CONTROLLER READY
014170  010562  000000                  MOV     R5,RLCS(R2)     ;LOAD CS REG
014174                                  WAITMS  #4              ;WAIT 4 MS
014174  012700  000004                  MOV     #4,R0
014200  104026                          EMT     CSWTM
014202  032762  002000  000000          BIT     #OPIERR,RLCS(R2);TEST IF OPI SET
014210  001452                          BEQ     20$             ;NO - SKIP
014212                          5$:     CLRVEC  ERRVEC
014212  013700  002652                  MOV     ERRVEC,R0
014216  104036                          EMT     CSCVEC
014220                                  PRINTF  #FMT24,#DRVNAV
```

```
   (8)  014220  012746  005640                    MOV     #DRVNAV,-(SP)
   (7)  014224  012746  011463                    MOV     #FMT24,-(SP)
   (6)  014230  012746  000002                    MOV     #2,-(SP)
   (3)  014234  010600                            MOV     SP,R0
   (4)  014236  104017                            EMT     C$PNTF
   (4)  014240  062706  000006                    ADD     #6,SP
1149    014244                          10$:      PRINTF  #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
  (11)  014244  005046                            CLR     -(SP)
  (11)  014246  153716  002455                    BISB    RLDRV+1,(SP)
  (10)  014252  012746  005633                    MOV     #DRVNAM,-(SP)
   (9)  014256  013746  002450                    MOV     RLBAS,-(SP)
   (8)  014262  012746  005622                    MOV     #BASADD,-(SP)
   (7)  014266  012746  010657                    MOV     #FMT5,-(SP)
   (6)  014272  012746  000005                    MOV     #5,-(SP)
   (3)  014276  010600                            MOV     SP,R0
   (4)  014300  104017                            EMT     C$PNTF
   (4)  014302  062706  000014                    ADD     #14,SP
1150    014306                                    PRINTF  #FMT3
   (7)  014306  012746  010643                    MOV     #FMT3,-(SP)
   (6)  014312  012746  000001                    MOV     #1,-(SP)
   (3)  014316  010600                            MOV     SP,R0
   (4)  014320  104017                            EMT     C$PNTF
   (4)  014322  062706  000004                    ADD     #4,SP
1151    014326                                    DODU    PSETNM          ;DROP DRIVE
   (3)  014326  013700  003064                    MOV     PSETNM,R0
   (3)  014332  104053                            EMT     C$DODU
1152    014334                                    DOCLN
   (3)  014334  104044                            EMT     C$DCLN
1153    014336                          20$:      CLRVEC  ERRVEC
   (3)  014336  013700  002652                    MOV     ERRVEC,R0
   (3)  014342  104036                            EMT     C$CVEC
1154    014344                          22$:
1165                                    ;CHECK IF POWER FAILURE WAIT IS NEEDED
1166
1167    014344  005737  003072          4$:       TST     PWRFLG          ;NEEDED???
1168    014350  001434                            BEQ     8$              ;NO, SKIP
1169
1170    014352  013705  002454                    MOV     RLDRV,R5        ;DRIVE SELECT
1171    014356  052705  000200                    BIS     #CRDYMSK,R5     ;SET CRDY
1172    014362  010562  000000                    MOV     R5,RLCS(R2)     ;SELECT DRIVE
1173    014366  012701  000074                    MOV     #60.,R1         ;SIXTY SECOND TIMER
1174    014372  032762  000001  000000  9$:       BIT     #DRDYMSK,RLCS(R2)       ;DRIVE UP YET
1175    014400  001020                            BNE     8$              ;YES START TEST
1176
1177    014402                                    WAITMS  #10.            ;WAIT A SECOND
   (3)  014402  012700  000012                    MOV     #10.,R0
   (3)  014406  104026                            EMT     C$WTM
1178    014410  005301                            DEC     R1              ;SIXTY GONE BY
1179    014412  001367                            BNE     9$              ;NO
1180    014414                                    PRINTF  #FMT24,#NOPWR
   (8)  014414  012746  005673                    MOV     #NOPWR,-(SP)
   (7)  014420  012746  011463                    MOV     #FMT24,-(SP)
   (6)  014424  012746  000002                    MOV     #2,-(SP)
   (3)  014430  010600                            MOV     SP,R0
   (4)  014432  104017                            EMT     C$PNTF
   (4)  014434  062706  000006                    ADD     #6,SP
```



```
1181    014440  000701                            BR      10$
1182
1183    014442                          8$:
1184
1185    014442                                    ENDINIT
   (3)  014442                                    L10014:
   (3)  014442  104011                            EMT     C$INIT
1186    014444                                    ENDMOD
1187
1188    014444                                    BGNMOD  CLNCODE
1189    014444                                    BGNCLN
1190
1191    014444                                    SETVEC  ERRVEC,#TRPHAN,#340
   (7)  014444  012746  000340                    MOV     #340,-(SP)
   (6)  014450  012746  014570                    MOV     #TRPHAN,-(SP)
   (5)  014454  013746  002652                    MOV     ERRVEC,-(SP)
   (4)  014460  012746  000003                    MOV     #3,-(SP)
   (3)  014464  104037                            EMT     C$SVEC
   (2)  014466  062706  000010                    ADD     #10,SP
1192
1193    014472                                    SETPRI  #7              ;SET PRORITY TO 7
   (3)  014472  012700  000007                    MOV     #7,R0
   (3)  014476  104041                            EMT     C$SPRI
1194    014500  032762  000200  000000  2$:       BIT     #CRDYMSK,RLCS(R2)       ;TEST IF CONTROLLER READY
1195    014506  001407                            BEQ     3$              ;NO LOOP UNTIL READY
1196    014510  053762  002454  000000            BIS     RLDRV,RLCS(R2)  ;SET DRIVE NUMBER
1197    014516  032762  000001  000000            BIT     #DRDYMSK,RLCS(R2)       ;TEST IF DRIVE BUSY
1198    014524  001003                            BNE     5$              ;NO - SKIP
1199    014526                          3$:       WAITMS  #3              ;WAIT 300 MS
   (3)  014526  012700  000003                    MOV     #3,R0
   (3)  014532  104026                            EMT     C$WTM
1200    014534                          5$:       CLRVEC  RLVEC           ;RELEASE VEC
   (3)  014534  013700  002452                    MOV     RLVEC,R0
   (3)  014540  104036                            EMT     C$CVEC
1201    014542  005737  003072                    TST     PWRFLG  ;PWR FAIL SET
1202    014546  001402                            BEQ     7$              ;NO
1203    014550  005337  003072                    DEC     PWRFLG
1204    014554                          7$:       CLRVEC  ERRVEC
   (3)  014554  013700  002652                    MOV     ERRVEC,R0
   (3)  014560  104036                            EMT     C$CVEC
1205    014562                                    ENDCLN
   (3)  014562                                    L10015:
   (3)  014562  104012                            EMT     C$CLEAN
1206
1207    014564                                    BGNDU
1208    014564  000240                            NOP
1209    014566                                    ENDDU
   (3)  014566                                    L10016:
   (3)  014566  104055                            EMT     C$DU
1210
1211    014570                                    ENDMOD
1212    014570                                    BGNMOD  GLBSUB
1213
1214    014570  005237  003070          TRPHAN:   INC     TRPFLG
1215    014574  000002                            RTI
1216
```

```
                                BGNSRV  INTHLR
                                ;       INTERRUPT HANDLER. ABORTS WAIT TIMER AND STORES ALL RL11 REGS
                                        ABORTWAIT
(3)   014576  104021                     EMT     C$ABRT
      014600  012237  002466              MOV     (R2)+,T.CS      ;STORE RL REGISTERS
      014604  012237  002470              MOV     (R2)+,T.BA
      014610  012237  002472              MOV     (R2)+,T.DA
      014614  012237  002474              MOV     (R2),T.MP
      014620  012737  177777  002430      MOV     #-1,DONE        ;SET DONE FLAG
      014626  013702  002450              MOV     RLBAS,R2        ;RESTORE R2
                                ENDSRV
(3)   014632                            L10017:
(2)   014632  000002                     RTI
                                ;       ERROR LIMIT CHECKING ROUTINE
                                ;       DROPS DRIVE IF ERROR LIMIT EXCEEDED
      014634  027737  166020  013402  CKERLM: CMP     @ERRPOINT,ERLIMW        ;TEST IF ERROR LIMIT EXCEEDED
      014642  002453                     BLT     1$              ;NO - SKIP
                                        INLOOP                  ;CHECK IF IN ERROR LOOP
(3)   014644  104020                     EMT     C$INLP
                                        BCOMPLETE       1$      ;YES - SKIP
(2)   014646  103451                     BCS     1$
                                        PRINTF  #FMT25,ERLIMW,#MEXERS
(9)   014650  012746  010325             MOV     #MEXERS,-(SP)
(8)   014654  013746  013402             MOV     ERLIMW,-(SP)
(7)   014660  012746  011470             MOV     #FMT25,-(SP)
(6)   014664  012746  000003             MOV     #3,-(SP)
(3)   014670  010600                     MOV     SP,R0
(4)   014672  104017                     EMT     C$PNTF
(4)   014674  062706  000010             ADD     #10,SP
                                        PRINTF  #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
(11)  014700  005046                     CLR     -(SP)
(11)  014702  153716  002455             BISB    RLDRV+1,(SP)
(10)  014706  012746  005633             MOV     #DRVNAM,-(SP)
(9)   014712  013746  002450             MOV     RLBAS,-(SP)
(8)   014716  012746  005622             MOV     #BASADD,-(SP)
(7)   014722  012746  010657             MOV     #FMT5,-(SP)
(6)   014726  012746  000005             MOV     #5,-(SP)
(3)   014732  010600                     MOV     SP,R0
(4)   014734  104017                     EMT     C$PNTF
(4)   014736  062706  000014             ADD     #14,SP
                                        PRINTF  #FMT3
(7)   014742  012746  010643             MOV     #FMT3,-(SP)
(6)   014746  012746  000001             MOV     #1,-(SP)
(3)   014752  010600                     MOV     SP,R0
(4)   014754  104017                     EMT     C$PNTF
(4)   014756  062706  000004             ADD     #4,SP
      014762                            DODU    PSETNM          ;DROP DRIVE
(3)   014762  013700  003064             MOV     PSETNM,R0
(3)   014766  104053                     EMT     C$DODU
                                        DOCLN                   ;GO TO CLEAN UP
(3)   014770  104044                     EMT     C$DCLN
      014772  000207                     RTS     PC
                                ;       READ AND STORE ALL RL11 REGISTERS
      014774  016237  000000  002466  READRL: MOV     RLCSR(R2),T.CS  ;GET CS REG
```



```
      015002  016237  000002  002470      MOV     RLBA(R2),T.BA   ;GET BUS ADDRESS REG
      015010  016237  000004  002472      MOV     RLDA(R2),T.DA   ;GET DISK ADDRESS
      015016  016237  000006  002474      MOV     RLMP(R2),T.MP   ;GET MULTI-PURPOSE REG
      015024  000207                     RTS     PC              ;RETURN
                                ;       WAIT FOR CONTROLLER TIMEOUT TO FORCE INTERRUPT ROUTINE
      015026  011646                  WAITIN: MOV     (SP),-(SP)      ;MAKE ROOM FOR ERROR POINTER
      015030  005066  000002             CLR     2(SP)           ;CLEAR FOR POINTER
      015034  032762  000200  000000     BIT     #CRDYMSK,RLCSR(R2)      ;TEST IF CONTROLLER READY
      015042  001420                     BEQ     4$              ;NO - SKIP TO WAIT
      015044  004737  014774             JSR     PC,READRL       ;READ ALL RL REGS
      015050  005737  002430             TST     DONE            ;TEST IF INTERRUPT OCCURRED
      015054  001433                     BEQ     5$              ;NO - GO SET NO INTERRUPT ERR FLAG
      015056  012766  006013  000002  1$: MOV     #MTOSLOW,2(SP)  ;ELSE SET TO SLOW ERROR POINTER
      015064  032737  002000  002466     BIT     #OPIERR,T.CS    ;TEST IF OPI SET
      015072  001403                     BEQ     2$              ;NO - SKIP
      015074  012766  006032  000002     MOV     #MDRRES,2(SP)   ;SET MESSAGE FOR NO DRIVE RESPONSE
      015102  000207                  2$: RTS     PC              ;RETURN
      015104                           4$: WAITMS  #3              ;WAIT 300 MS FOR TIMEOUT
(3)   015104  012700  000003             MOV     #3,R0
(3)   015110  104026                     EMT     C$WTM
      015112  032762  000200  000000     BIT     #CRDYMSK,RLCS(R2)       ;TEST IF READY NOW SET
      015120  001006                     BNE     3$              ;YES - SKIP
      015122  004737  014774             JSR     PC,READRL       ;READ RL REGS
      015126  012766  006115  000002     MOV     #MCONHNG,2(SP)  ;SET MESSAGE FOR CONTROLLER HUNG
      015134  000762                     BR      2$              ;SKIP
      015136  005737  002430          3$: TST     DONE            ;ELSE CHECK IF INTERRUPT OCCURRED
      015142  001345                     BNE     1$              ;YES - SKIP TO SET TO SLOW
      015144  004737  014774          5$: JSR     PC,READRL       ;READ RL REGS
      015150  012766  006062  000002     MOV     #MNOINT,2(SP)   ;ELSE SET NO INTERRUPT FLAG
      015156  000751                     BR      2$              ;GO TO RETURN
                                ;       OPERATION AND TEST INITIALIZE ROUTINE
      015160  005037  002426  TSTINT: CLR     OPFLAG          ;CLEAR OPERATION FLAGS
      015164  105037  003067             CLRB    NORRCT          ;RESET INHIBIT ERROR COUNTING
      015170  005037  002436             CLR     MORECE          ;RESET MORE COMPARE ERRORS
      015174  000207                     RTS     PC
                                ;       GET STATUS AND GET STATUS WITH RESET ROUTINE
      015176  013746  002550  GSTATR: MOV     TEMP4,-(SP)     ;STORE TEMP4
      015202  012737  000013  002550     MOV     #GETSTATIDRSET,TEMP4    ;SET FOR RESET
      015210  000412                     BR      GSTATG
      015212  013746  002550  GSTATC: MOV     TEMP4,-(SP)     ;STORE TEMP4
      015216  012737  000003  002550     MOV     #GETSTAT,TEMP4  ;SET FOR NO RESET
      015224  000404                     BR      GSTATG
      015226  013746  002550  GSTAT:  MOV     TEMP4,-(SP)     ;STORE TEMP4
      015232  005037  002550             CLR     TEMP4           ;SET FOR SAVE L. AND T. REGS
      015236  010346  GSTATG:  MOV     R3,-(SP)        ;STORE R3
      015240  013703  002424             MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
      015244  005723                     TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
      015246  016663  000004  002260     MOV     4(SP),SUBSTK(R3) ;INSERT THIS CALL
      015254  162763  000004  002260     SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
      015262  010337  002424             MOV     R3,SSINDX       ;STORE IT BACK
      015266  010046                     MOV     R0,-(SP)        ;STORE R0
      015270  010146                     MOV     R1,-(SP)        ;STORE R1
      015272  012737  000002  002440     MOV     #2,ERRSWI       ;SET FOR NO ERROR RETURN
```

```
 1297  015300  032737  000010  002550          BIT     #DRSET,TEMP4    ;TEST IF DRIVE RESET
 1298  015306  001453                          BEQ     11$             ;NO - SKIP
 1299  015310  032762  040000  000000          BIT     #DRVERR,RLCS(R2) ;TEST IF DRIVE ERROR SET
 1300  015316  001403                          BEQ     49$             ;NO - SKIP
 1301  015320                                  WAITMS  #3              ;WAIT FOR 300 MS FOR DRIVE TO SETTLE
   (3) 015320  012700  000003                  MOV     #3,R0
   (3) 015324  104026                          EMT     C$WTM
 1302  015326  012701  000062          49$:    MOV     #50.,R1         ;SET WAIT FOR 5 SEC
 1303  015332  004737  015226          50$:    JSR     PC,GSTAT        ;GET DRIVE STATUS
 1304  015336  015772                          3$
 1305  015340  032737  000001  002466          BIT     #DRDYMSK,T.CS   ;TEST IF DRIVE READY
 1306  015346  001051                          BNE     5$              ;YES - GO DO CLEAR
 1307  015350  032737  000020  002474          BIT     #HOSTAT,T.MP    ;ELSE TEST IF HEADS OUT
 1308  015356  001010                          BNE     51$             ;YES - BYPASS RELOAD WAIT FLAG SETTING
 1309  015360  032737  144000  002474          BIT     #SPDSTAT!HCESTAT!WDESTAT,T.MP ;TEST IF DRIVE HAS ERROR
 1310                                                                  ;THAT CAUSED HEADS TO
 1311                                                                  ;UNLOAD
 1312  015366  001441                          BEQ     5$              ;NO - SKIP
 1313  015370  052737  040000  002426          BIS     #RELDWT,OPFLAG  ;ELSE SET WAIT FLAG
 1314  015376  000435                          BR      5$              ;SKIP TO CLEAR
 1315  015400  032737  040000  002466  51$:    BIT     #DRVERR,T.CS    ;TEST IF DRIVE ERROR NOW
 1316  015406  001031                          BNE     5$              ;YES - SKIP TO CLEAR
 1317  015410                                  WAITMS  #1              ;WAIT FOR DRIVE TO GET ERROR, RDY, OR HO
   (3) 015410  012700  000001                  MOV     #1,R0
   (3) 015414  104026                          EMT     C$WTM
 1318  015416  005301                          DEC     R1              ;DEC WAIT COUNTER
 1319  015420  001344                          BNE     50$             ;IF NOT DONE, LOOP
 1320  015422  012703  010201                  MOV     #MUNDEF,R3      ;MESSAGE FOR UNDEFINED STATE
 1321  015426                                  ERRHRD  10001,,,ERR1
   (3) 015426  104443                          TRAP    TSERCODE
   (5) 015430  023421                          .WORD   10001
   (5) 015432  011554                          .WORD   ERR1
 1322  015434  000554                          BR      14$             ;EXIT
 1323  015436  005737  002550          11$:    TST     TEMP4           ;TEST IF SAVE REGISTERS
 1324  015442  001013                          BNE     5$              ;NO SKIP
 1325  015444  012701  000004                  MOV     #4,R1           ;SET SAVE COUNT
 1326  015450  012703  002466                  MOV     #L.MP+2,R3      ;SET ADDRESS OF FIRST SAVE
 1327  015454  014346                  8$:     MOV     -(R3),-(SP)     ;PUT REG ON STACK
 1328  015456  005301                          DEC     R1              ;DEC COUNT
 1329  015460  001375                          BNE     8$              ;LOOP UNTIL ALL SAVED
 1330  015462  012737  000003  002462          MOV     #GETSTAT,L.DA   ;SET FOR GET STATUS
 1331  015470  000403                          BR      6$              ;SKIP
 1332  015472  013737  002550  002462  5$:     MOV     TEMP4,L.DA      ;INSERT PRESET FOR STATUS
 1333  015500                          6$:
 1334  015500  005037  002430                  CLR     DONE            ;CLEAR INTERRUPT FLAG
 1335  015504  013737  002454  002456          MOV     RLDRV,L.CS      ;SET UP TO GET STATUS
 1336  015512  042737  002000  002456          BIC     #BIT10,L.CS     ;CLEAR FOR DRIVE 4 - 7 SPEC'D
 1337  015520  052737  000104  002456          BIS     #GTSTAT,L.CS
 1338  015526  013762  002462  000004          MOV     L.DA,RLDA(R2)   ;LOAD RL REGS
 1339  015534  013762  002456  000000          MOV     L.CS,RLCSR(R2)  ;LOAD CS REG
 1340  015542                                  WAITUS  #1              ;WAIT 100 US FOR INTERRUPT
   (3) 015542  012700  000001                  MOV     #1,R0
   (3) 015546  104027                          EMT     C$WTU
 1341  015550  005737  002430                  TST     DONE            ;CHECK IF INTERRUPT OCCURRED
 1342  015554  001476                          BEQ     1$              ;NO - SKIP
 1343  015556  013737  002474  002502  4$:     MOV     T.MP,T.STAT     ;STORE MP REGISTER
```



```
 1344  015564  042737  177770  002502          BIC     #^C<STAMSK>,T.STAT ;CLEAR ALL BUT STATE
 1345  015572  032737  000010  002462          BIT     #DRSET,L.DA     ;TEST IF RESET WAS SPECIFIED
 1346  015600  001474                          BEQ     3$              ;NO - SKIP TO EXIT
 1347  015602  032737  040000  002426          BIT     #RELDWT,OPFLAG  ;TEST IF RELOAD WAIT FLAG SET
 1348  015610  001424                          BEQ     12$             ;NO - SKIP
 1349  015612  012701  001130                  MOV     #600.,R1        ;SET WAIT COUNT FOR 60 SECONDS
 1350  015616  032762  000001  000000  13$:    BIT     #DRDYMSK,RLCS(R2) ;TEST IF DRIVE NOW READY
 1351  015624  001016                          BNE     12$             ;YES - SKIP
 1352  015626                                  WAITMS  #1              ;CALL WAIT
   (3) 015626  012700  000001                  MOV     #1,R0
   (3) 015632  104026                          EMT     C$WTM
 1353  015634  005301                          DEC     R1              ;DEC COUNT
 1354  015636  001367                          BNE     13$             ;LOOP IF NOT 0
 1355  015640  004737  015226                  JSR     PC,GSTAT        ;GET DRIVE STATUS
 1356  015644  015772                          3$                      ;ERROR RETURN
 1357  015646  012703  010246                  MOV     #MRLFAL,R3      ;SET RESULT MESSAGE POINTER
 1358  015652                                  ERRHRD  10003,,,ERR1
   (3) 015652  104443                          TRAP    TSERCODE
   (5) 015654  023423                          .WORD   10003
   (5) 015656  011554                          .WORD   ERR1
 1359  015660  000442                          BR      14$             ;GO TO EXIT
 1360  015662                          12$:    WAITUS  #10.            ;WAIT FOR 1MS
   (3) 015662  012700  000012                  MOV     #10.,R0
   (3) 015666  104027                          EMT     C$WTU
 1361  015670  004737  015226                  JSR     PC,GSTAT        ;GET DRIVE STATUS
 1362  015674  015772                          3$
 1363  015676  032737  100000  002466          BIT     #ANYERR,T.CS    ;TEST IF ANY ERROR
 1364  015704  001432                          BEQ     3$              ;NO - SKIP
 1365  015706  032737  001000  002474          BIT     #VCSTAT,T.MP    ;CHECK IF VOLUME CHECK RESET
 1366  015714  001403                          BEQ     7$              ;YES SKIP
 1367  015716  012703  006161                  MOV     #VCNRST,R3      ;SET REASON POINTER
 1368  015722  000416                          BR      2$              ;EXIT
 1369  015724  032737  040000  002466  7$:     BIT     #DRVERR,T.CS    ;CHECK IF DRIVE ERROR
 1370  015732  001404                          BEQ     9$              ;NO - SKIP
 1371  015734                                  ERRHRD  10004,,,ERR6
   (3) 015734  104443                          TRAP    TSERCODE
   (5) 015736  023424                          .WORD   10004
   (5) 015740  012056                          .WORD   ERR6
 1372  015742  000411                          BR      14$             ;EXIT
 1373  015744  012703  006202          9$:     MOV     #UNXERR,R3      ;SET REASON POINTER
 1374  015750  000403                          BR      2$              ;EXIT
 1375  015752  004737  015026          1$:     JSR     PC,WAITIN       ;WAIT FOR INTERRUPT
 1376  015756  012603                          MOV     (SP)+,R3        ;STORE REASON POINTER FOR RETURN
 1377  015760                          2$:     ERRHRD  10002,,,ERR1
   (3) 015760  104443                          TRAP    TSERCODE
   (5) 015762  023422                          .WORD   10002
   (5) 015764  011554                          .WORD   ERR1
 1378  015766  005037  002440          14$:    CLR     ERRSWI          ;CLEAR FOR ERROR RETURN
 1379  015772  005737  002550          3$:     TST     TEMP4           ;TEST IF REGISTERS WERE SAVED
 1380  015776  001007                          BNE     22$             ;NO - SKIP
 1381  016000  012703  002456                  MOV     #L.CS,R3        ;SET POINTER TO RESTORE
 1382  016004  012701  000004                  MOV     #4,R1           ;SET REGISTER COUNT
 1383  016010  012623                  20$:    MOV     (SP)+,(R3)+     ;RESTORE REG
 1384  016012  005301                          DEC     R1              ;DEC COUNT
 1385  016014  001375                          BNE     20$             ;LOOP UNTIL ALL ARE RESTORED
 1386  016016  162737  000002  002424  22$:    SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
```

```
1387  016024  012601                          MOV     (SP)+,R1        ;RESTORE R1
1388  016026  012600                          MOV     (SP)+,R0
1389  016030  012603                          MOV     (SP)+,R3        ;RESTORE R3
1390  016032  012637  002550                  MOV     (SP)+,TEMP4     ;RESTORE TEMP4
1391  016036  005737  002440                  TST     ERRSWI          ;TEST IF ERROR RETURN
1392  016042  001403                          BEQ     99$             ;YES - SKIP
1393  016044  063716  002440                  ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
1394  016050  000207                          RTS     PC
1395  016052  017626  000000          99$:    MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
1396  016056  000207                          RTS     PC
1397
1398
1400                                  ;       SEEK ROUTINE
1401  016060  012737  177777  002542  XSEEKT: MOV     #-1,TEMP1       ;SET SPECIAL TIMING SEEK FLAG
1402  016066  000402                          BR      XSEEK1
1403  016070  005037  002542          XSEEK:  CLR     TEMP1           ;CLEAR SPECIAL SEEK FOR TIMING FLAG
1404  016074  010346                  XSEEK1: MOV     R3,-(SP)        ;STORE R3
1405  016076  013703  002424                  MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
1406  016102  005723                          TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
1407  016104  016663  000002  002260          MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
1408  016112  162763  000004  002260          SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
1409  016120  010337  002424                  MOV     R3,SSINDX       ;STORE IT BACK
1410  016124  010046                          MOV     R0,-(SP)
1411  016126  010146                          MOV     R1,-(SP)
1412  016130  010546                          MOV     R5,-(SP)        ;STORE REG
1413  016132  012737  000002  002440          MOV     #2,@#ERRSWI     ;SET FOR NO ERROR RETURN
1414  016140  005037  002520                  CLR     DIFAUG          ;CLEAR DIFFERENCE AUGMENT (FOR SEEKING
                                                                  ;       PAST GUARD BAND)
1416  016144  004737  021116                  JSR     PC,GETPOS       ;GET PRESENT POSITION
1417  016150  016530                          65$
1418  016152  013737  002526  002522          MOV     CURCYL,OLDCYL   ;MOVE CURRENT TO OLD CYLINDER
1419  016160  023727  002524  000377          CMP     NEWCYL,#255.    ;TEST IF NEW IS GREATER THAN 255
1420  016166  003412                          BLE     3$              ;NO - SKIP
1421  016170  162737  000377  002524          SUB     #255.,NEWCYL    ;ELSE SUBTRACT 255.
1422  016176  013737  002524  002520          MOV     NEWCYL,DIFAUG   ;STORE DIFFERENCE AS AUGMENT
1423  016204  012737  000377  002524          MOV     #255.,NEWCYL    ;SET NEWCYL AS 255.
1424  016212  000412                          BR      6$              ;SKIP
1425  016214  005737  002524          3$:     TST     NEWCYL          ;TEST IF NEWCYL HAS NEGATIVE VALUE
1426  016220  100007                          BPL     6$              ;NO - SKIP
1427  016222  005437  002524                  NEG     NEWCYL          ;ELSE MAKE IT POSITIVE
1428  016226  013737  002524  002520          MOV     NEWCYL,DIFAUG   ;AND STORE IT AS AUGMENT
1429  016234  005037  002524                  CLR     NEWCYL          ;AND SET NEWCYL TO 0
1430  016240  013705  002526          6$:     MOV     CURCYL,R5       ;COMPUTE DIFFERENCE AND NEW CYLINDER
1431  016244  163705  002524                  SUB     NEWCYL,R5       ;SUB NEWCYL FROM CURCYL
1432  016250  100005                          BPL     13$             ;IF DIFF IS POSITIVE - SKIP(REV SEEK)
1433  016252  012737  000001  002532          MOV     #1,DESSGN       ;ELSE SET SIGN FOR FORWARD
1434  016260  005405                          NEG     R5              ;MAKE DIFFERENCE POSITIVE
1435  016262  000402                          BR      14$             ;SKIP
1436  016264  005037  002532          13$:    CLR     DESSGN          ;SET SIGN FOR REVERSE
1437  016270  010537  002530          14$:    MOV     R5,DESDIF       ;STORE DIFFERENCE
1438  016274  005737  002520                  TST     DIFAUG          ;IS THERE A DIFFERENCE AUGMENT
1439  016300  001412                          BEQ     18$             ;NO - SKIP
1440  016302  023727  002524  000377          CMP     NEWCYL,#255.    ;CHECK IF NEW CYL IS 255.
1441  016310  001003                          BNE     17$             ;NO - SKIP
1442  016312  012737  000001  002532          MOV     #1,DESSGN       ;ELSE FORCE SIGN FOR FORWARD
1443                                                                  ;(INNER GUARD BAND)
```



```
1444  016320  063737  002520  002530  17$:    ADD     DIFAUG,DESDIF   ;ADD ANY AUGMENT TO DIFFERENCE
1445  016326                          18$:
1446  016326  012705  002456                  MOV     #L.CS,R5        ;GET L REG ADDRESS
1447  016332  012715  000106                  MOV     #SEEK,(R5)      ;SET FOR SEEK
1448  016336  053715  002454                  BIS     RLDRV,(R5)      ;INSERT DRIVE NUMBER
1449  016342  042725  002000                  BIC     #BIT10,(R5)+    ;CLEAR IF DRIVE 4 - 7 SPEC'D
1450  016346  005025                          CLR     (R5)+           ;CLEAR BUS ADDRESS
1451  016350  013715  002530                  MOV     DESDIF,(R5)     ;LOAD DIFFERENCE
1452  016354  012700  000007                  MOV     #7,R0           ;SET TO SHIFT DIFFERENCE
1453  016360  006315                  21$:    ASL     (R5)
1454  016362  005300                          DEC     R0
1455  016364  001375                          BNE     21$             ;LOOP UNTIL ALIGNED
1456  016366  005737  002532                  TST     DESSGN          ;TEST SIGN
1457  016372  001402                          BEQ     23$             ;SKIP IF 0
1458  016374  052715  000004                  BIS     #DIRBIT,(R5)    ;ELSE INSERT SIGN
1459  016400  005737  002534          23$:    TST     DESHD           ;TEST IF HEAD 0
1460  016404  001402                          BEQ     25$             ;YES - SKIP
1461  016406  052715  000020                  BIS     #HDSEL,(R5)     ;ELSE SET HEAD BIT
1462  016412  052725  000001          25$:    BIS     #MBSET0,(R5)+   ;INSERT MARKER BIT
1463  016416  004737  017130                  JSR     PC,RDYCHK       ;CHECK IF DRIVE READY
1464  016422  016530                          65$
1465  016424  005037  002430                  CLR     DONE            ;CLEAR INTERRUPT FLAG
1466  016430  005737  002542                  TST     TEMP1           ;CHECK IF SPECIAL SEEK FLAG SET
1467  016434  001035                          BNE     65$             ;YES - SKIP, DO NOT START SEEK
1468  016436  014562  000004                  MOV     -(R5),RLDA(R2)  ;LOAD RL REGISTERS
1469  016442  014562  000002                  MOV     -(R5),RLBA(R2)
1470  016446  014562  000000                  MOV     -(R5),RLCS(R2)
1471  016452                          30$:    WAITUS  #10.
  (3) 016452  012700  000012                  MOV     #10.,R0
  (3) 016456  104027                          EMT     CSWTU
1472  016460  005737  002430                  TST     DONE            ;TEST IF INTERRUPT DONE
1473  016464  001011                          BNE     32$             ;YES - SKIP
1474  016466  004737  015026                  JSR     PC,WAITIN       ;GO WAIT FOR INTERRUPT
1475  016472  012603                          MOV     (SP)+,R3        ;GET RESULT MESSAGE POINTER
1476  016474                                  ERRHRD  10005,,ERR1
  (3) 016474  104443                          TRAP    TSERCODE
  (5) 016476  023425                          .WORD   10005
  (5) 016500  011554                          .WORD   ERR1
1477  016502  005037  002440                  CLR     ERRSWI          ;CLEAR FOR ERROR ERROR RETURN
1478  016506  000410                          BR      65$
1479  016510  005737  002466          32$:    TST     T.CS            ;TEST IF ANY ERROR
1480  016514  100005                          BPL     65$             ;NO - SKIP
1481  016516                                  ERRHRD  10006,,ERR6
  (3) 016516  104443                          TRAP    TSERCODE
  (5) 016520  023426                          .WORD   10006
  (5) 016522  012056                          .WORD   ERR6
1482  016524  005037  002440                  CLR     ERRSWI          ;CLEAR FOR ERROR ERROR RETURN
1483  016530  162737  000002  002424  65$:    SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
1484  016536  012605                          MOV     (SP)+,R5        ;RESTORE REGISTER
1485  016540  012601                          MOV     (SP)+,R1
1486  016542  012600                          MOV     (SP)+,R0
1487  016544  012603                          MOV     (SP)+,R3        ;RESTORE R3
1488  016546  005737  002440                  TST     ERRSWI          ;TEST IF ERROR RETURN
1489  016552  001403                          BEQ     99$             ;YES - SKIP
1490  016554  063716  002440                  ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
1491  016560  000207                          RTS     PC
```

```
 1492  016562  017616  000000          99$:    MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
 1493  016566  000207                          RTS     PC
 1494
 1552
 1554                                  ;       POSITION HEADS ROUTINE. POSITIONS HEADS USING 1 CYLINDER SEEKS
 1555                                  ;       TO CYLINDER SPECIFIED IN R5 BY THE CALLING ROUTINE
 1556  016570  010346                  POSHDS: MOV     R3,-(SP)        ;SAVE REGS
 1557  016572  013703  002424                  MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
 1558  016576  005723                          TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
 1559  016600  016663  000002  002260          MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
 1560  016606  162763  000004  002260          SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
 1561  016614  010337  002424                  MOV     R3,SSINDX       ;STORE IT BACK
 1562  016620  010346                          MOV     R3,-(SP)
 1563  016622  010446                          MOV     R4,-(SP)
 1564  016624  012737  000002  002440          MOV     #2,ERRSWI       ;SET FOR NO ERROR RETURN
 1565  016632  004737  021116                  JSR     PC,GETPOS       ;GET CURRENT POSITION
 1566  016636  017072                          PH65$
 1567  016640  012704  000012                  MOV     #10.,R4         ;SET RETRY COUNT
 1568  016644                          BGNSEG
   (3) 016644  104004                          EMT     C$BSEG
 1569  016646                          1$:     INLOOP                  ;CHECK IF IN ERROR LOOP
   (3) 016646  104020                          EMT     C$INLP
 1570  016650                                  BNCOMPLETE      5$      ;NO - SKIP
   (2) 016650  103012                          BCC     5$
 1571  016652  004737  021116                  JSR     PC,GETPOS       ;ELSE GET POSITION
 1572  016656  017070                          60$
 1573  016660  023737  002526  002524          CMP     CURCYL,NEWCYL   ;CHECK IF AT INTENDED POSITION
 1574  016666  001017                          BNE     8$              ;NO - SKIP
 1575  016670  004737  017454                  JSR     PC,ONSWAP       ;SWAP OLDCYL AND NEWCYL
 1576  016674  000414                          BR      8$              ;SKIP
 1577  016676  013737  002526  002522  5$:     MOV     CURCYL,OLDCYL   ;IN NOT LOOPING, STORE CURCYL AS OLDCYL
 1578  016704  023705  002526                  CMP     CURCYL,R5       ;CHECK IF HDS AT FINAL POSITION
 1579  016710  001467                          BEQ     60$             ;YES - GO TO EXIT
 1580  016712  003003                          BGT     7$              ;IF CURCYL > FINAL POSITION - SKIP
 1581  016714  005237  002524                  INC     NEWCYL          ;ELSE BUMP NEWCYL (MOVE HDS IN)
 1582  016720  000402                          BR      8$              ;SKIP
 1583  016722  005337  002524          7$:     DEC     NEWCYL          ;DEC NEWCYL (MOVE HDS OUT)
 1584  016726  004737  016070          8$:     JSR     PC,XSEEK        ;DO SEEK
 1585  016732  017070                          60$
 1586  016734  012701  005670                  MOV     #3000.,R1       ;SET WAIT COUNT 300 MS
 1587  016740  004737  020650                  JSR     PC,RDYWAIT      ;WAIT FOR DRIVE READY
 1588  016744  017070                          60$
 1589  016746  005737  002466                  TST     T.CS            ;TEST IF ANY ERROR
 1590  016752  100006                          BPL     10$             ;NO - SKIP
 1591  016754                                  ERRHRD  10008.,6,ERR6
   (3) 016754  104443                          TRAP    T$ERCODE
   (5) 016756  023430                          .WORD   10008
   (5) 016760  012056                          .WORD   ERR6
 1592  016762  005037  002440                  CLR     ERRSWI          ;CLEAR FOR ERROR ERROR RETURN
 1593  016766  000440                          BR      60$
 1594  016770  004737  021116          10$:    JSR     PC,GETPOS       ;GET POSITION
 1595  016774  017070                          60$
 1596  016776  023737  002526  002524          CMP     CURCYL,NEWCYL   ;CHECK IF ARRIVED AT DESIRED PLACE
 1597  017004  001003                          BNE     15$             ;NO - SKIP
 1598  017006  012704  000012          14$:    MOV     #10.,R4         ;ELSE INIT RETRY COUNT
 1599  017012  000715                          BR      1$              ;GO DO NEXT SEEK
```



```
 1600  017014  005737  002532          15$:    TST     DESSGN          ;TEST IF GOING IN
 1601  017020  001016                          BNE     17$             ;YES - SKIP
 1602  017022  023737  002526  002524          CMP     CURCYL,NEWCYL   ;CHECK IF HEADS DID NOT MOVE IN
 1603  017030  003366                          BGT     14$             ;YES - SKIP
 1604  017032  005304                  16$:    DEC     R4              ;DEC RETRY COUNT
 1605  017034  001334                          BNE     8$              ;DO ANOTHER SEEK IF NOT 0
 1606  017036  012703  007172                  MOV     #HDMOVF,R3      ;ELSE SET RESULT MESSAGE POINTER
 1607  017042                                  ERRHRD  10009.,6,ERR1
   (3) 017042  104443                          TRAP    T$ERCODE
   (5) 017044  023431                          .WORD   10009
   (5) 017046  011554                          .WORD   ERR1
 1608  017050  005037  002440                  CLR     ERRSWI          ;CLEAR FOR ERROR ERROR RETURN
 1609  017054  000405                          BR      60$
 1610  017056  023737  002526  002524  17$:    CMP     CURCYL,NEWCYL   ;HDS SHOULD MOVE OUT, CHK THEY DID
 1611  017064  002750                          BLT     14$             ;YES - SKIP
 1612  017066  000761                          BR      16$             ;ELSE GO DEC AND RETRY
 1613  017070                          20$:
 1614  017070                          60$:
 1615  017070                          ENDSEG
 1616  017070                          10000$:
   (3) 017070  104005                          EMT     C$ESEG
 1616  017072  162737  000002  002424  PH65$:  SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
 1617  017100  012604                          MOV     (SP)+,R4        ;RESTORE REGISTERS
 1618  017102  012600                          MOV     (SP)+,R0
 1619  017104  012603                          MOV     (SP)+,R3
 1620  017106  005737  002440                  TST     ERRSWI          ;TEST IF ERROR RETURN
 1621  017112  001403                          BEQ     99$             ;YES - SKIP
 1622  017114  063716  002440                  ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
 1623  017120  000207                          RTS     PC
 1624  017122  017616  000000          99$:    MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
 1625  017126  000207                          RTS     PC
 1626
 1628                                  ;       DRIVE READY TEST ROUTINE. CHECKS DEIVE IS READY. IF NOT, WAIT
 1629                                  ;       500MS FOR READY TO SET.
 1630  017130  010346                  RDYCHK: MOV     R3,-(SP)        ;STORE REGS
 1631  017132  013703  002424                  MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
 1632  017136  005723                          TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
 1633  017140  016663  000002  002260          MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
 1634  017146  162763  000004  002260          SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
 1635  017154  010337  002424                  MOV     R3,SSINDX       ;STORE IT BACK
 1636  017160  010046                          MOV     R0,-(SP)
 1637  017162  010146                          MOV     R1,-(SP)
 1638  017164  010446                          MOV     R4,-(SP)
 1639  017166  012737  000002  002440          MOV     #2,ERRSWI       ;SET FOR NO ERROR RETURN
 1640  017174  012701  011610                  MOV     #5000.,R1       ;SET WAIT COUNT
 1641  017200  004737  015226          1$:     JSR     PC,GSTAT        ;GET DRIVE STATUS
 1642  017204  017324                          4$
 1643  017206  032737  000001  002466          BIT     #DRDYMSK,T.CS   ;TEST IF DRIVE READY
 1644  017214  001045                          BNE     5$              ;YES - EXIT
 1645  017216                                  WAITUS  #1
   (3) 017216  012700  000001                  MOV     #1,R0
   (3) 017222  104027                          EMT     C$WTU
 1646  017224  005301                          DEC     R1              ;DEC WAIT COUNT
 1647  017226  001364                          BNE     1$              ;LOOP IF NOT 0
 1648  017230  012703  007543                  MOV     #MDRDY,R3       ;SET RESULT MESSAGE POINTER
 1649  017234  012704  010471                  MOV     #C500MS,R4      ;SET CONDITION MESSAGE POINTER
```

```
1650  017240                                ERRHRD  10010,,ERR5
 (3)  017240  104443                        TRAP    TSERCODE
 (5)  017242  023432                        .WORD   10010
 (5)  017244  012006                        .WORD   ERR5
1651  017246  012701  000062                MOV     #50.,R1         ;SET WAIT COUNT FOR 5 SECONDS
1652  017252  004737  015226          2$:   JSR     PC,GSTAT        ;GET DRIVE STATUS
1653  017256  017324                        4$
1654  017260  032737  000001  002466        BIT     #DRDYMSK,T.CS   ;TEST IF DRIVE READY
1655  017266  001005                        BNE     3$              ;YES - SKIP
1656  017270                                WAITMS  #1              ;WAIT FOR 100MS
 (3)  017270  012700  000001                MOV     #1,R0
 (3)  017274  104026                        EMT     CSWTM
1657  017276  005301                        DEC     R1              ;DEC WAIT COUNTER
1658  017300  001364                        BNE     2$              ;LOOP UNTIL TIME DONE
1659  017302  032737  100000  002466  3$:   BIT     #ANYERR,T.CS    ;TEST IF ANYERR SET
1660  017310  001405                        BEQ     4$              ;NO - SKIP
1661  017312                                ERRHRD  10011,,ERR6     ;REPORT ALL ERRORS
 (3)  017312  104443                        TRAP    TSERCODE
 (5)  017314  023433                        .WORD   10011
 (5)  017316  012056                        .WORD   ERR6
1662  017320  005337  002662                DEC     ERRCNT          ;REDUCE ERROR COUNT FOR DUAL ERRORS
1663  017324  005037  002440          4$:   CLR     ERRSWI          ;CLEAR FOR ERROR RETURN
1664  017330  162737  000002  002424  5$:   SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
1665  017336  012604                        MOV     (SP)+,R4        ;RESTORE REGS
1666  017340  012601                        MOV     (SP)+,R1
1667  017342  012600                        MOV     (SP)+,R0
1668  017344  012603                        MOV     (SP)+,R3
1669  017346  005737  002440                TST     ERRSWI          ;TEST IF ERROR RETURN
1670  017352  001403                        BEQ     99$             ;YES - SKIP
1671  017354  063716  002440                ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
1672  017360  000207                        RTS     PC
1673  017362  017616  000000          99$:  MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
1674  017366  000207                        RTS     PC
1675
1677                                ;       CHOSE HEAD ROUTINE. PICKS HEAD 0 UNLESS SPECIFIC HEAD IS
1678                                ;       SELECTED BY SOFTWARE PARAMETER.
1679  017370  005037  002534        CHOSHD: CLR     DESHD           ;CLEAR TO HEAD 0
1680  017374  032737  010000  013372        BIT     #HEADLM,MISWIW  ;TEST IF HEAD SPECIFIED
1681  017402  001403                        BEQ     1$              ;NO - SKIP
1682  017404  013737  013400  002534        MOV     HEADW,DESHD     ;INSERT SPECIFIED HEAD
1683  017412  000207                  1$:   RTS     PC
1684
1685                                ;       SWAP HEAD ROUTINE. CHANGES SELECTED HEAD TO HEAD 1
1686                                ;       UNLESS HEAD 0 SPECIFICALLY SELECTED BY SOFTWARE PARAMETER.
1687  017414  032737  010000  013372  SWAPHD: BIT     #HEADLM,MISWIW  ;TEST IF HEAD SPECIFIED
1688  017422  001011                        BNE     2$              ;YES - TAKE ABORT EXIT
1689  017424  005737  002534                TST     DESHD           ;TEST IF HEAD ONE USED
1690  017430  001006                        BNE     2$              ;YES - TAKE ABORT EXIT
1691  017432  012737  000001  002534        MOV     #1,DESHD        ;ELSE SET FOR HEAD ONE
1692  017440  062716  000002                ADD     #2,(SP)         ;BUMP PAST ABORT RETURN
1693  017444  000207                        RTS     PC              ;RETURN
1694  017446  017616  000000          2$:   MOV     @(SP),(SP)      ;GET ABORT DESTINATION
1695  017452  000207                  3$:   RTS     PC
1696
1697                                ;       SWAP OLD CYLINDER AND NEW CYLINDER ROUTINE.
1698  017454  010046                  ONSWAP: MOV   R0,-(SP)        ;STORE R0
```



```
1699  017456  013700  002522                MOV     OLDCYL,R0       ;MOVE OLD TO R0
1700  017462  013737  002524  002522        MOV     NEWCYL,OLDCYL   ;MOVE NEW TO OLD
1701  017470  010037  002524                MOV     R0,NEWCYL       ;PUT OLD IN NEW
1702  017474  012600                        MOV     (SP)+,R0        ;RESTORE R0
1703  017476  000207                        RTS     PC
1704
1705                                ;       BAD SECTOR FILES VALID CHECK ROUTINE. CHECKS IF BAD SECTOR
1706                                ;       FILES HAVE BEEN READ AND STORED. IF NOT, REPORT AND FORCE
1707                                ;       FORCE FILES TO LOOK LIKE ALL SECTORS OK.
1708  017500  005737  003074        CKBSVD: TST     BSFVAL          ;TEST IF BAD SECTORS STORED
1709  017504  001051                        BNE     5$              ;YES - EXIT
1710  017506                                PRINTF  #FMT9,#BSNSTR   ;REPORT
 (8)  017506  012746  007442                MOV     #BSNSTR,-(SP)
 (7)  017512  012746  011043                MOV     #FMT9,-(SP)
 (6)  017516  012746  000002                MOV     #2,-(SP)
 (3)  017522  010600                        MOV     SP,R0
 (4)  017524  104017                        EMT     CSPNTF
 (4)  017526  062706  000006                ADD     #6,SP
1711  017532                                PRINTF  #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
(11)  017532  005046                        CLR     -(SP)
(11)  017534  153716  002455                BISB    RLDRV+1,(SP)
(10)  017540  012746  005633                MOV     #DRVNAM,-(SP)
 (9)  017544  013746  002450                MOV     RLBAS,-(SP)
 (8)  017550  012746  005622                MOV     #BASADD,-(SP)
 (7)  017554  012746  010657                MOV     #FMT5,-(SP)
 (6)  017560  012746  000005                MOV     #5,-(SP)
 (3)  017564  010600                        MOV     SP,R0
 (4)  017566  104017                        EMT     CSPNTF
 (4)  017570  062706  000014                ADD     #14,SP
1712  017574                                PRINTF  #FMT3
 (7)  017574  012746  010643                MOV     #FMT3,-(SP)
 (6)  017600  012746  000001                MOV     #1,-(SP)
 (3)  017604  010600                        MOV     SP,R0
 (4)  017606  104017                        EMT     CSPNTF
 (4)  017610  062706  000004                ADD     #4,SP
1713  017614  012737  177777  003076        MOV     #-1,SBSFIL      ;FORCE FILES TO NO ENTRIES
1714  017622  012737  177777  003272        MOV     #-1,FBSFIL
1715  017630  000207                  5$:   RTS     PC
1716
1718                                ;       READ HEADERS ROUTINE.
1719  017632  012737  000001  002550  XRDHDC: MOV     #1,TEMP4        ;SET FLAG TO BYPASS REG STORAGE
1720  017640  000402                        BR      XRDHDG          ;GO DO IT
1721  017642  005037  002550          XRDHD:  CLR     TEMP4           ;SET FLAG TO SAVE T. AMD L. REGS
1722  017646  010346                  XRDHDG: MOV     R3,-(SP)        ;STORE REGISTERS
1723  017650  013703  002424                MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
1724  017654  005723                        TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
1725  017656  016663  000002  002260        MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
1726  017664  162763  000004  002260        SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
1727  017672  010337  002424                MOV     R3,SSINDX       ;STORE IT BACK
1728  017676  010046                        MOV     R0,-(SP)
1729  017700  010146                        MOV     R1,-(SP)
1730  017702  010446                        MOV     R4,-(SP)
1731  017704  012737  000002  002440        MOV     #2,ERRSWI       ;SET FOR NO ERROR RETURN
1732  017712  005737  002550                TST     TEMP4           ;TEST IF REGISTERS TO BE SAVED
1733  017716  001007                        BNE     2$              ;NO - SKIP
1734  017720  012703  002466                MOV     #L.MP+2,R3      ;SET POINTER FOR REGS
```

```
 1735  017724  012701  000004            MOV     #4,R1           ;SET COUNT
 1736  017730  014346            1$:     MOV     -(R3),-(SP)     ;SAVE REGISTER
 1737  017732  005301                    DEC     R1              ;DEC COUNT
 1738  017734  001375                    BNE     1$              ;LOOP UNTIL ALL ARE SAVED
 1739  017736  004737  017130            2$:     JSR     PC,RDYCHK       ;CHECK DRIVE READY
 1740  017742  020176                    65$
 1741  017744  005037  002430            CLR     DONE            ;CLEAR INTERRUPT FLAG
 1742  017750  012701  002456            MOV     #L.CS,R1        ;GET ADDRESS OF LOAD REGS
 1743  017754  013711  002454            MOV     RLDRV,(R1)      ;LOAD DRIVE NUMBER
 1744  017760  042711  002000            BIC     #BIT10,(R1)     ;CLEAR FOR DRIVE 4 - 7 SPEC'D
 1745  017764  052721  000110            BIS     #RDHEAD,(R1)+   ;INSERT COMMAND
 1746  017770  005021                    CLR     (R1)+           ;CLEAR BA
 1747  017772  005021                    CLR     (R1)+           ;CLEAR DA
 1748  017774  014162  000004            MOV     -(R1),RLDA(R2)  ;LOAD RL11 REGS
 1749  020000  014162  000002            MOV     -(R1),RLBA(R2)
 1750  020004  014162  000000            MOV     -(R1),RLCSR(R2)
 1751  020010                    3$:     WAITUS  #10.            ;WAIT 1MS FOR INTERRUPT
   (3) 020010  012700  000012            MOV     #10.,R0
   (5) 020014  104027                    EMT     C$WT10
 1752  020016  005737  002430            TST     DONE            ;TEST IN INTERRUPT FLAG SET
 1753  020022  001455                    BEQ     14$             ;NO - SKIP
 1754  020024  032737  000001  002466    5$:     BIT     #DRDYMSK,T.CS   ;TEST IF DRIVE READY
 1755  020032  001033                    BNE     10$             ;YES - SKIP
 1756  020034  012703  007543            MOV     #MDRDY,R3       ;SET NO READY MESSAGE
 1757  020040  012704  010510            MOV     #CAFDT,R4       ;CONDITION OF AFTER DATA XFER
 1758  020044                            ERRHRD  10017,,ERR5
   (3) 020044  104443                    TRAP    T$ERCODE
   (5) 020046  023441                    .WORD   10017
   (5) 020050  012006                    .WORD   ERR5
 1759  020052  012701  000062            MOV     #50.,R1         ;SET WAIT COUNT FOR 5 SECONDS
 1760  020056  004737  015226            4$:     JSR     PC,GSTAT        ;GET STATUS
 1761  020062  020172                    60$
 1762  020064  032737  000001  002466    BIT     #DRDYMSK,T.CS   ;TEST IF DRIVE HAS COME READY
 1763  020072  001401                    BEQ     11$             ;NO - SKIP
 1764  020074  005037  002440            CLR     ERRSWI          ;CLEAR ERROR SWITCH
 1765  020100  000410                    BR      10$             ;SKIP
 1766  020102  005301            11$:    DEC     R1              ;DEC WAIT COUNT
 1767  020104  001364                    BNE     4$              ;LOOP UNTIL TIME DONE
 1768  020106  012704  010522            MOV     #C5SEC,R4       ;SET CONDITION AFTER 5 SECONDS
 1769  020112                            ERRHRD  10014,,ERR5
   (3) 020112  104443                    TRAP    T$ERCODE
   (5) 020114  023436                    .WORD   10014
   (5) 020116  012006                    .WORD   ERR5
 1770  020120  000424                    BR      60$             ;EXIT
 1771  020122  005737  002466    10$:    TST     T.CS            ;CHECK FOR ANY ERRORS
 1772  020126  100004                    BPL     12$             ;NO - SKIP
 1773  020130                            ERRHRD  10016,,ERR6     ;REPORT ALL ERRORS
   (3) 020130  104443                    TRAP    T$ERCODE
   (5) 020132  023440                    .WORD   10016
   (5) 020134  012056                    .WORD   ERR6
 1774  020136  000415                    BR      60$
 1775  020140  012701  002476    12$:    MOV     #HDWRD2,R1      ;GET POINTER
 1776  020144  016221  000006            MOV     RLMP(R2),(R1)+  ;STORE LAST TWO HEADER WORDS
 1777  020150  016221  000006            MOV     RLMP(R2),(R1)+
 1778  020154  000410                    BR      65$             ;EXIT
 1779  020156  004737  015026    14$:    JSR     PC,WAITIN       ;WAIT FOR INTERRUPT
```



```
 1780  020162  012603                    MOV     (SP)+,R3        ;GET RESULTS
 1781  020164                            ERRHRD  10015,,ERR1     ;REPORT
   (3) 020164  104443                    TRAP    T$ERCODE
   (5) 020166  023437                    .WORD   10015
   (5) 020170  011554                    .WORD   ERR1
 1782  020172  005037  002440    60$:    CLR     ERRSWI          ;CLEAR FOR ERROR ERROR RETURN
 1783  020176  005737  002550    65$:    TST     TEMP4           ;TEST IF REGISTERS WERE SAVED
 1784  020202  001007                    BNE     22$             ;NO - SKIP
 1785  020204  012703  002456            MOV     #L.CS,R3        ;SET POINTER TO RESTORE REGS
 1786  020210  012701  000004            MOV     #4,R1           ;SET COUNT
 1787  020214  012623            20$:    MOV     (SP)+,(R3)+     ;RESTORE REGISTER
 1788  020216  005301                    DEC     R1              ;DEC COUNT
 1789  020220  001375                    BNE     20$             ;LOOP UNTIL ALL ARE RESTORED
 1790  020222  162737  000002  002424    22$:    SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
 1791  020230  012604                    MOV     (SP)+,R4        ;RESTORE REGS
 1792  020232  012601                    MOV     (SP)+,R1
 1793  020234  012600                    MOV     (SP)+,R0
 1794  020236  012603                    MOV     (SP)+,R3
 1795  020240  005737  002440            TST     ERRSWI          ;TEST IF ERROR RETURN
 1796  020244  001403                    BEQ     99$             ;YES - SKIP
 1797  020246  063716  002440            ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
 1798  020252  000207                    RTS     PC
 1799  020254  017616  000000    99$:    MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
 1800  020260  000207                    RTS     PC
 1801
 1803                            ;       VERIFY HEADERS ROUTINE. COMPARES 40 HEADERS FOR CONTENT AND
 1804                            ;       SEQUENCE.
 1805  020262  010346            VERHDR: MOV     R3,-(SP)        ;STORE REGS
 1806  020264  013703  002424            MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
 1807  020270  005723                    TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
 1808  020272  016663  000002  002260    MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
 1809  020300  162763  000004  002260    SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
 1810  020306  010337  002424            MOV     R3,SSINDX       ;STORE IT BACK
 1811  020312  010046                    MOV     R0,-(SP)
 1812  020314  010146                    MOV     R1,-(SP)
 1813  020316  010446                    MOV     R4,-(SP)
 1814  020320  010546                    MOV     R5,-(SP)
 1815  020322  012737  000002  002440    MOV     #2,ERRSWI       ;SET FOR NO ERROR RETURN
 1816  020330  052737  000002  002426    BIS     #HDRCMP,OPFLAG  ;SET HEADER COMPARE FLAG
 1817  020336  005037  002436            CLR     MORECE          ;CLEAR MORE ERRORS FLAG
 1818  020342  012704  003466            MOV     #IBUFF,R4       ;SET POINTER TO HEADERS
 1819  020346  012705  002540            MOV     #TEMP0,R5       ;SET POINTER TO WORK AREA
 1820  020352  005003                    CLR     R3              ;CLEAR FOR WORD COUNTER
 1821  020354  011415                    MOV     (R4),(R5)       ;MOVE HDR WORD TO WORK AREA
 1822  020356  011401                    MOV     (R4),R1         ;PUT WORD IN REG 1
 1823  020360  042701  100177            BIC     #^CHDCYL,R1     ;CLEAR ALL BUT CYLINDER
 1824  020364  012700  000007            MOV     #7,R0           ;SET SHIFT COUNT
 1825  020370  006201            3$:     ASR     R1              ;SHIFT
 1826  020372  005300                    DEC     R0              ;DEC
 1827  020374  001375                    BNE     3$              ;LOOP
 1828  020376  020137  002524            CMP     R1,NEWCYL       ;CHECK IF CYLINDER PART GOOD
 1829  020402  001406                    BEQ     4$              ;YES - SKIP
 1830  020404                            ERRHRD  10018,,ERR10    ;REPORT ERROR
   (3) 020404  104443                    TRAP    T$ERCODE
   (5) 020406  023442                    .WORD   10018
   (5) 020410  013144                    .WORD   ERR10
```

```
 1831  020412  005037  002440              CLR     ERRSWI          ;CLEAR FOR ERROR ERROR RETURN
 1832  020416  000454                      BR      65$
 1833  020420  012701  000050      4$:     MOV     #40.,R1         ;SET HEADER COUNT
 1834  020424  042715  100100              BIC     #BIT15!HDHSEL,(R5)      ;CLEAR HEAD SELECT AND 0 BIT
 1835  020430  005737  002534              TST     DESHD           ;ARE WE USING HD 0?
 1836  020434  001402                      BEQ     5$              ;YES - SKIP
 1837  020436  052715  000100              BIS     #HDHSEL,(R5)    ;INSERT HEAD BIT
 1838  020442  005065  000002      5$:     CLR     2(R5)           ;CLEAR 2ND WORD OF WORK AREA
 1839  020446  021524                      CMP     (R5),(R4)+      ;TEST FIRST WORD OK
 1840  020450  001407                      BEQ     8$              ;YES - SKIP
 1841  020452  005744                      TST     -(R4)           ;ELSE SET POINTER FOR ERROR
 1842  020454                              ERRHRD  10018.,,ERR10   ;REPORT
  (3)  020454  104443                      TRAP    T$ERCODE
  (5)  020456  023442                      .WORD   10018
  (5)  020460  013144                      .WORD   ERR10
 1843  020462  005037  002440              CLR     ERRSWI          ;CLEAR FOR ERROR ERROR RETURN
 1844  020466  005724                      TST     (R4)+           ;RESET POINTER
 1845  020470  005203              8$:     INC     R3              ;BUMP WORD COUNTER
 1846  020472  005724                      TST     (R4)+           ;TEST 2ND WORD IS 0
 1847  020474  001407                      BEQ     12$             ;YES - SKIP
 1848  020476  022544                      CMP     (R5)+,-(R4)     ;ADJUST POINTERS FOR REPORT
 1849  020500                              ERRHRD  10018.,,ERR10   ;REPORT
  (3)  020500  104443                      TRAP    T$ERCODE
  (5)  020502  023442                      .WORD   10018
  (5)  020504  013144                      .WORD   ERR10
 1850  020506  005037  002440              CLR     ERRSWI          ;CLEAR FOR ERROR ERROR RETURN
 1851  020512  024524                      CMP     -(R5),(R4)+     ;RESET POINTERS
 1852  020514  005724              12$:    TST     (R4)+           ;BUMP PAST ECC WORD
 1853  020516  005203                      INC     R3              ;BUMP WORD COUNTER
 1854  020520  005215                      INC     (R5)            ;BUMP SECTOR OF EXPECTED HEADER
 1855  020522  011500                      MOV     (R5),R0         ;MOVE EXPECTED HDR TO R0
 1856  020524  042700  177700              BIC     #^CHDSEC,R0     ;CLEAR ALL BUT SECTOR
 1857  020530  022700  000050              CMP     #40.,R0         ;TEST IF AT SECTOR 40
 1858  020534  001002                      BNE     15$             ;NO - SKIP
 1859  020536  042715  000077              BIC     #HDSEC,(R5)     ;CLEAR SECTOR TO 0
 1860  020542  005203              15$:    INC     R3              ;BUMP HDR WORD COUNTER
 1861  020544  005301                      DEC     R1              ;DEC HEADER COUNT
 1862  020546  001337                      BNE     6$              ;LOOP IF NOT YET DONE
 1863  020550  162737  000002  002424  65$:     SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
 1864  020556  012605                      MOV     (SP)+,R5        ;RESTORE REGISTERS
 1865  020560  012604                      MOV     (SP)+,R4
 1866  020562  012601                      MOV     (SP)+,R1
 1867  020564  012600                      MOV     (SP)+,R0
 1868  020566  012603                      MOV     (SP)+,R3
 1869  020570  005737  002440              TST     ERRSWI          ;TEST IF ERROR RETURN
 1870  020574  001403                      BEQ     99$             ;YES - SKIP
 1871  020576  063716  002440              ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
 1872  020602  000207                      RTS     PC
 1873  020604  017616  000000      99$:    MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
 1874  020610  000207                      RTS     PC
 1875
 1877                              ;       POSITION HEAD BIT FROM HEADER OR MULTIPURPOSE REGISTER TO LSB.
 1878  020612  013705  002474      POSHW1: MOV     HDWRD1,R5       ;START FOR POSITION HD BIT IN WD 1
 1879  020616  000402                      BR      POSHD0          ;SKIP
 1880  020620  013705  002474      POSHSB: MOV     T.MP,R5 ;START FOR POSITION HD BIT IN MP
 1881  020624  010146              POSHD0: MOV     R1,-(SP)        ;STORE R1
```



```
 1882  020626  042705  177677              BIC     #^CHSSTAT,R5    ;CLEAR ALL BUT HEAD SEL BIT
 1883  020632  012701  000006              MOV     #6,R1           ;SET SHIFT COUNT
 1884  020636  006205              1$:     ASR     R5              ;SHIFT FOR RIGHT JUSTIFY
 1885  020640  005301                      DEC     R1
 1886  020642  001375                      BNE     1$
 1887  020644  012601                      MOV     (SP)+,R1        ;RESTORE R1
 1888  020646  000207                      RTS     PC              ;RETURN
 1889
 1890                              ;       WAIT FOR READY ROUTINE. DURATION OF WAIT PASSED TO THE ROUTINE
 1891                              ;       FROM THE CALLING ROUTINE IN R1.
 1892  020650  010346              RDYWAIT:        MOV     R3,-(SP)        ;STORE R3
 1893  020652  013703  002424                      MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
 1894  020656  005723                      TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
 1895  020660  016663  000002  002260      MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
 1896  020666  162763  000004  002260      SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
 1897  020674  010337  002424              MOV     R3,SSINDX       ;STORE IT BACK
 1898  020700  010046                      MOV     R0,-(SP)
 1899  020702  010146                      MOV     R1,-(SP)
 1900  020704  010446                      MOV     R4,-(SP)
 1901  020706  012737  000002  002440      MOV     #2,ERRSWI       ;SET FOR NO ERROR RETURN
 1902  020714  004737  015226              JSR     PC,GSTAT        ;GET DRIVE STATUS
 1903  020720  021052                      10$
 1904  020722  032737  000001  002466      BIT     #DRDYMSK,T.CS   ;CHECK IF READY
 1905  020730  001052                      BNE     5$              ;YES - SKIP
 1906  020732  005301                      DEC     R1              ;DEC WAIT COUNT
 1907  020734  001404                      BEQ     7$              ;SKIP IF 0
 1908  020736                              WAITUS  #1
  (3)  020736  012700  000001                  MOV     #1,R0
  (3)  020742  104027                      EMT     CSWTU
 1909  020744  000763                      BR      5$
 1910  020746  012703  007543      7$:     MOV     #MDRDY,R3       ;SET NAME MESSAGE PTR
 1911  020752                              ERRHRD  10020.,,ERR3    ;REPORT READY ERROR
  (3)  020752  104443                      TRAP    T$ERCODE
  (5)  020754  023444                      .WORD   10020
  (5)  020756  011670                      .WORD   ERR3
 1912  020760  012701  000062              MOV     #50.,R1         ;SET WAIT COUNT FOR 5 SECONDS
 1913  020764  004737  015226      6$:     JSR     PC,GSTAT        ;GET DRIVE STATUS
 1914  020770  021052                      10$
 1915  020772  032737  000001  002466      BIT     #DRDYMSK,T.CS   ;TEST IF DRIVE READY
 1916  021000  001013                      BNE     8$              ;YES - SKIP
 1917  021002                              WAITMS  #1              ;WAIT 100 MS
  (3)  021002  012700  000001                  MOV     #1,R0
  (3)  021006  104026                      EMT     CSWTM
 1918  021010  005301                      DEC     R1              ;DEC WAIT COUNT
 1919  021012  001364                      BNE     6$              ;LOOP UNTIL TIME DONE
 1920  021014  012704  010522              MOV     #C5SEC,R4       ;SET CONDITION AFTER 5 SECDS
 1921  021020                              ERRHRD  10021.,,ERR5
  (3)  021020  104443                      TRAP    T$ERCODE
  (5)  021022  023445                      .WORD   10021
  (5)  021024  012006                      .WORD   ERR5
 1922  021026  000407                      BR      11$             ;EXIT
 1923  021030  032737  100000  002466  8$:     BIT     #ANYERR,T.CS    ;TEST IF ANY ERROR SET
 1924  021036  001405                      BEQ     10$             ;NO - SKIP
 1925  021040                              ERRHRD  10022.,,ERR6    ;REPORT ALL ERRORS
  (3)  021040  104443                      TRAP    T$ERCODE
  (5)  021042  023446                      .WORD   10022
```

```
   (5)  021044  012056                          .WORD   ERR6
 1926   021046  005337  002662          11$:    DEC     ERRCNT          ;DEC FOR DOUBLE ERROR REPORT
 1927   021052  005037  002440          10$:    CLR     ERRSWI          ;CLEAR FOR ERROR ERROR RETURN
 1928   021056  162737  000002  002424  9$:     SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
 1929   021064  012604                          MOV     (SP)+,R4        ;RESTORE REGISTERS
 1930   021066  012601                          MOV     (SP)+,R1
 1931   021070  012600                          MOV     (SP)+,R0
 1932   021072  012603                          MOV     (SP)+,R3        ;RESTORE R3
 1933   021074  005737  002440                  TST     ERRSWI          ;TEST IF ERROR RETURN
 1934   021100  001403                          BEQ     99$             ;YES - SKIP
 1935   021102  063716  002440                  ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
 1936   021106  000207                          RTS     PC
 1937   021110  017616  000000          99$:    MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
 1938   021114  000207                          RTS     PC
 1939
 1940                                  ;       GET POSITION ROUTINE. READS A HEADER FROM CURRENT CYLINDER
 1941                                  ;       (WHERE IT IS PRESENTLY POSITIONED) AND STORES CYLINDER
 1942                                  ;       NUMBER IN CURCYL.
 1943   021116  010346          GETPOS: MOV     R3,-(SP)        ;STORE REGISTERS
 1944   021120  013703  002424                  MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
 1945   021124  005723                          TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
 1946   021126  016663  000002  002260          MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
 1947   021134  162763  000004  002260          SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
 1948   021142  010337  002424                  MOV     R3,SSINDX       ;STORE IT BACK
 1949   021146  010046                          MOV     R0,-(SP)
 1950   021150  010546                          MOV     R5,-(SP)
 1951   021152  004737  017642                  JSR     PC,XRDHD        ;DO READ HEADER
 1952   021156  021206                          65$
 1953   021160  013703  002474                  MOV     HDWRD1,R3       ;GET HEADER WORD
 1954   021164  042703  100177                  BIC     #^CHDCYL,R3     ;CLEAR ALL BUT CYLINDER
 1955   021170  012705  000007                  MOV     #7,R5           ;SET SHIFT COUNT
 1956   021174  006203                  4$:     ASR     R3              ;SHIFT TO RIGHT JUSTIFY
 1957   021176  005305                          DEC     R5
 1958   021200  001375                          BNE     4$
 1959   021202  010337  002526                  MOV     R3,CURCYL       ;STORE AS CURRENT CYLINDER
 1960   021206  162737  000002  002424  65$:    SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
 1961   021214  012605                          MOV     (SP)+,R5        ;RESTORE REGISTERS
 1962   021216  012600                          MOV     (SP)+,R0
 1963   021220  012603                          MOV     (SP)+,R3
 1964   021224  005737  002440                  TST     ERRSWI          ;TEST IF ERROR RETURN
 1965   021232  001403                          BEQ     99$             ;YES - SKIP
 1966   021230  063716  002440                  ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
 1967   021234  000207                          RTS     PC
 1968   021236  017616  000000          99$:    MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
 1969   021242  000207                          RTS     PC
 1970
 1972                                  ;       VERIFY POSITION ROUTINE. READS A HEADER (USING GETPOS) AND
 1973                                  ;       CHECKS HEADS ARE POSITIONED AT NEW CYLINDER (CURCYL = NEWCYL).
 1974   021244  010346          VERPOS: MOV     R3,-(SP)        ;STORE R3
 1975   021246  013703  002424                  MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
 1976   021252  005723                          TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
 1977   021254  016663  000002  002260          MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
 1978   021262  162763  000004  002260          SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
 1979   021270  010337  002424                  MOV     R3,SSINDX       ;STORE IT BACK
 1980
 1981   021274  012737  000002  002440          MOV     #2,ERRSWI       ;SET FOR NO ERROR RETURN
```



```
 1982   021302  004737  021116                  JSR     PC,GETPOS       ;GET POSITION
 1983   021306  021332                          65$
 1984   021310  023737  002524  002526          CMP     NEWCYL,CURCYL   ;CHECK IF CURRENT CYL IS NEW CYL
 1985   021316  001405                          BEQ     1$              ;YES - SKIP
 1986   021320                                  ERRHRD  10022.,ERR8
   (3)  021320  104443                          TRAP    T$ERCODE
   (5)  021322  023446                          .WORD   10022
   (5)  021324  013004                          .WORD   ERR8
 1987   021326  005037  002440                  CLR     ERRSWI          ;CLEAR FOR ERROR ERROR RETURN
 1988   021332                          1$:
 1989   021332  162737  000002  002424  65$:    SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
 1990   021340  012603                          MOV     (SP)+,R3        ;RESTORE R3
 1991   021342  005737  002440                  TST     ERRSWI          ;TEST IF ERROR RETURN
 1992   021346  001403                          BEQ     99$             ;YES - SKIP
 1993   021350  063716  002440                  ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
 1994   021354  000207                          RTS     PC
 1995   021356  017616  000000          99$:    MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
 1996   021362  000207                          RTS     PC
 1997
 1999                                  ;       READ ALL HEADERS ROUTINE. 40 HEADERS ARE READ AND STORED
 2000                                  ;       IN IBUFF.
 2001   021364  010346          RDALHD: MOV     R3,-(SP)        ;STORE REGISTERS
 2002   021366  013703  002424                  MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
 2003   021372  005723                          TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
 2004   021374  016663  000002  002260          MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
 2005   021402  162763  000004  002260          SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
 2006   021410  010337  002424                  MOV     R3,SSINDX       ;STORE IT BACK
 2007   021414  010046                          MOV     R0,-(SP)
 2008   021416  010146                          MOV     R1,-(SP)
 2009   021420  010446                          MOV     R4,-(SP)
 2010   021422  012737  000002  002440          MOV     #2,ERRSWI       ;SET FOR NO ERROR RETURN
 2011   021430  012701  000050                  MOV     #40.,R1         ;SET HEADER COUNT
 2012   021434  052737  100000  002426          BIS     #HDR40,OPFLAG   ;SET 40 HDR OP FLAG
 2013   021442  012703  003466                  MOV     #IBUFF,R3       ;SET POINTER TO STORE HDRS
 2014   021446  013704  002450                  MOV     RLBAS,R4        ;GET BASE ADDRESS
 2015   021452  062704  000006                  ADD     #RLMP,R4        ;MAKE IT POINT TO MP REG
 2016   021456  012737  000010  002456          MOV     #10,L.CS        ;LOAD FOR READ HEADER, NO INTERRUPT
 2017   021464  053737  002454  002456          BIS     RLDRV,L.CS      ;INSERT DRIVE NUMBER
 2018   021472  042737  002000  002456          BIC     #BIT10,L.CS     ;CLEAR FOR DRIVE 4 - 7 SPEC'D
 2019   021500  005037  002460                  CLR     L.BA            ;CLEAR BA
 2020   021504  005037  002462                  CLR     L.DA            ;CLEAR DA
 2021   021510  005737  002534                  TST     DESHD           ;TEST IF HEAD 0
 2022   021514  001403                          BEQ     3$              ;YES - SKIP
 2023   021516  052737  000020  002462          BIS     #HDSEL,L.DA     ;ELSE INSERT HEAD 0
 2024   021524  013762  002462  000004  3$:     MOV     L.DA,RLDA(R2)   ;LOAD RLDA REG
 2025   021532  013762  002460  000002          MOV     L.BA,RLBA(R2)   ;LOAD RLBA
 2026   021540  032762  000200  000000          BIT     #CRDYMSK,RLCS(R2) ;TEST IF CONTROLLER READY
 2027   021546  001003                          BNE     6$              ;YES - SKIP
 2028   021550  004737  017130                  JSR     PC,RDYCHK       ;ELSE CHECK READY
 2029   021554  021666                          65$
 2030   021556  013762  002456  000000  6$:     MOV     L.CS,RLCS(R2)   ;LOAD RLCS REG
 2031   021564  012700  077777                  MOV     #77777,R0       ;SET COUNT FOR WAIT
 2032   021570  032762  000200  000000  7$:     BIT     #CRDYMSK,RLCS(R2) ;CHECK THAT OPERATION COMPLETED
 2033   021576  001015                          BNE     8$              ;YES - SKIP
 2034   021600  005300                          DEC     R0              ;DEC COUNT
 2035   021602  001372                          BNE     7$              ;SKIP IF NOT YET 0
```

```
 2036  021604  004737  014774                  JSR     PC,READRL       ;ELSE GET ALL REGISTERS
 2037  021610  004737  015026                  JSR     PC,WAITIN       ;ELSE WAIT FOR TIMEOUT
 2038  021614  012603                          MOV     (SP)+,R3        ;GET RESULT MESSAGE POINTER
 2039  021616                                  ERRHRD  10025.,,ERR1
   (3) 021616  104443                          TRAP    TSERCODE
   (5) 021620  023451                          .WORD   10025
   (5) 021622  011554                          .WORD   ERR1
 2040  021624  005037  002440                  CLR     ERRSWI          ;CLEAR FOR ERROR RETURN
 2041  021630  000416                          BR      65$
 2042  021632  005737  002466          8$:     TST     T.CS            ;TEST FOR ANY ERRORS
 2043  021636  100006                          BPL     12$             ;NO - SKIP
 2044  021640                                  ERRHRD  10026.,,ERR6
   (3) 021640  104443                          TRAP    TSERCODE
   (5) 021642  023452                          .WORD   10026
   (5) 021644  012056                          .WORD   ERR6
 2045  021646  005037  002440                  CLR     ERRSWI          ;CLEAR FOR ERROR RETURN
 2046  021652  000405                          BR      65$
 2047  021654  011423          12$:    MOV     (R4),(R3)+      ;STORE HEADER WORDS
 2048  021656  011423                  MOV     (R4),(R3)+
 2049  021660  011423                  MOV     (R4),(R3)+
 2050  021662  005301                  DEC     R1              ;DEC HEADER COUNT
 2051  021664  001334                  BNE     6$
 2052  021666  162737  000002  002424  65$:    SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
 2053  021674  012604                  MOV     (SP)+,R4        ;RESTORE REGISTERS
 2054  021676  012601                  MOV     (SP)+,R1
 2055  021700  012600                  MOV     (SP)+,R0
 2056  021702  012603                  MOV     (SP)+,R3
 2057  021704  005737  002440                  TST     ERRSWI          ;TEST IF ERROR RETURN
 2058  021710  001403                  BEQ     99$             ;YES - SKIP
 2059  021712  063716  002440                  ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
 2060  021716  000207                  RTS     PC
 2061  021720  017616  000000          99$:    MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
 2062  021724  000207                  RTS     PC
 2063
 2064
 2066                                  ;       GENERATE DATA ROUTINE. PATTERN TO BE GENERATED IS GIVEN
 2067                                  ;       IN THE WORD FOLLOWING THE CALL. 128 WORDS ARE GENERATED
 2068                                  ;       IN OBUFF.
 2069  021726  010146          DATGEN: MOV     R1,-(SP)        ;STORE REGISTERS
 2070  021730  010346                  MOV     R3,-(SP)
 2071  021732  010446                  MOV     R4,-(SP)
 2072  021734  012701  004066                  MOV     #OBUFF,R1       ;SET POINTER TO OBUFF
 2073  021740  012504                  MOV     (R5)+,R4        ;GET DATA PATTERN SELECTOR
 2074  021742  006304                  ASL     R4              ;ADJUST IT FOR INDEXING
 2075  021744  016403  002234                  MOV     PATTBL(R4),R3   ;GET ADDRESS OF PATTERN
 2076  021750  011321                  MOV     (R3),(R1)+      ;MOVE FIRST PATTERN WORD
 2077  021752  001427                  BEQ     5$              ;SKIP IF PATTERN IS 0
 2078  021754  021327  177777                  CMP     (R3),#-1        ;CHECK IF PATTERN IS ALL 1'S
 2079  021760  001416                  BEQ     5$              ;YES - SKIP
 2080  021762  020427  000010                  CMP     R4,#8.          ;TEST IF PATTERN 5
 2081  021766  001403                  BEQ     3$              ;YES - SKIP
 2082  021770  020427  000020                  CMP     R4,#16.         ;CHECK IF PATTERN 9 OR 10
 2083  021774  002413                  BLT     6$              ;NO - SKIP
 2084  021776  005723          3$:     TST     (R3)+           ;BUMP SOURCE POINTER
 2085  022000  012321                  MOV     (R3)+,(R1)+     ;MOVE TWO MORE WORDS FORM SOURCE
 2086  022002  012321                  MOV     (R3)+,(R1)+
```



```
 2087  022004  012704  000015                  MOV     #13.,R4         ;SET COUNT
 2088  022010  012703  004066                  MOV     #OBUFF,R3       ;RESET POINTER
 2089  022014  000406                          BR      8$
 2090  022016  012703  004066          5$:     MOV     #OBUFF,R3       ;ELSE SET OBUFF AS PATTERN SOURCE
 2091  022022  000401                          BR      7$              ;GO TO FILL
 2092  022024  005723                  6$:     TST     (R3)+           ;BUMP SOURCE POINTER
 2093  022026  012704  000017          7$:     MOV     #15.,R4         ;SET MOVE COUNT
 2094  022032  012321                  8$:     MOV     (R3)+,(R1)+     ;MOVE 15 WORDS INTO BUFFER
 2095  022034  005304                  DEC     R4
 2096  022036  001375                  BNE     8$
 2097  022040  012703  004066                  MOV     #OBUFF,R3       ;SET SOURCE TO TOP OF OBUFF
 2098  022044  012704  000160                  MOV     #112.,R4        ;SET COUNT FOR REST OF BUFFER
 2099  022050  012321          10$:    MOV     (R3)+,(R1)+     ;REPEAT PATTERN IN BUFFER
 2100  022052  005304                  DEC     R4
 2101  022054  001375                  BNE     10$
 2102  022056  012604                  MOV     (SP)+,R4        ;RESTORE REGISTERS
 2103  022060  012603                  MOV     (SP)+,R3
 2104  022062  012601                  MOV     (SP)+,R1
 2105  022064  000205                  RTS     R5              ;RETURN
 2106
 2107                                  ;       DATA COMPARE ROUTINE. COMPARES THE CONTENTS OF IBUFF AND OBUFF.
 2108                                  ;       ERROR REPORTING IS LIMITED BY SOFTWARE PARAMETER.
 2109  022066  010346          DATCOM: MOV     R3,-(SP)        ;STORE R3
 2110  022070  013703  002424                  MOV     SSINDX,R3       ;GET SUBROUTINE STACK INDEX
 2111  022074  005723                  TST     (R3)+           ;BUMP INDEX TO NEXT ENTRY
 2112  022076  016663  000002  002260          MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
 2113  022104  162763  000004  002260          SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
 2114  022112  010337  002424                  MOV     R3,SSINDX       ;STORE IT BACK
 2115  022116  010146                          MOV     R1,-(SP)        ;STORE OTHER REGISTERS
 2116  022120  010446                          MOV     R4,-(SP)
 2117  022122  010546                          MOV     R5,-(SP)
 2118  022124  052737  000001  002426          BIS     #DATACMP,OPFLAG ;SET DATA COMPARE FLAG
 2119  022132  005037  002436                  CLR     MORECE          ;CLEAR MORE ERROR FLAG
 2120  022136  012705  004066                  MOV     #OBUFF,R5       ;SET POINTERS TO DATA FOR COMPARE
 2121  022142  012704  003466                  MOV     #IBUFF,R4
 2122  022146  012703  000001                  MOV     #1,R3           ;SET WORD COUNTER
 2123  022152  012701  000200                  MOV     #128.,R1        ;SET COMPARE COUNT
 2124  022156  022425          5$:     CMP     (R4)+,(R5)+     ;COMPARE DATA
 2125  022160  001052                  BNE     10$             ;ERROR - SKIP TO REPORT
 2126  022162  005203          7$:     INC     R3              ;BUMP WORD COUNT
 2127  022164  005301                  DEC     R1              ;DEC COMPARE COUNT
 2128  022166  001373                  BNE     5$              ;LOOP IF NOT 0
 2129  022170  042737  000001  002426  9$:     BIC     #DATACMP,OPFLAG ;CLEAR DATA COMPARE FLAG
 2130  022176  005737  002440                  TST     ERRSWI          ;TEST IF ANY COMPARE ERRORS
 2131  022202  001021                          BNE     15$             ;NO - SKIP
 2132  022204  012701  000200                  MOV     #128.,R1        ;SET REPORT VALUE
 2133  022210                                  PRINTB  #FMT27,#TCERR,MORECE,#RESE6,R1
  (11) 022210  010146                          MOV     R1,-(SP)
  (10) 022212  012746  010421                  MOV     #RESE6,-(SP)
   (9) 022216  013746  002436                  MOV     MORECE,-(SP)
   (8) 022222  012746  007520                  MOV     #TCERR,-(SP)
   (7) 022226  012746  011524                  MOV     #FMT27,-(SP)
   (6) 022232  012746  000005                  MOV     #5,-(SP)
   (3) 022236  010600                          MOV     SP,R0
   (4) 022240  104014                          EMT     CSPNTB
   (4) 022242  062706  000014                  ADD     #14,SP
```

```
2134  022246  162737  000002  002424  15$:    SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
2135  022254  017605                          MOV     (SP)+,R5        ;RESTORE REGS
2136  022256  012604                          MOV     (SP)+,R4
2137  022260  012601                          MOV     (SP)+,R1
2138  022262  012603                          MOV     (SP)+,R3
2139  022264  005737  002440                  TST     ERRSWI          ;TEST IF ERROR RETURN
2140  022270  001403                          BEQ     99$             ;YES - SKIP
2141  022272  063716  002440                  ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
2142  022276  000207                          RTS     PC
2143  022300  017616  000000          99$:    MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
2144  022304  000207                          RTS     PC
2145  022306  023737  002436  013404  10$:    CMP     MORECE,DCLIMW   ;TEST IF COMPARE ERRORS LIMIT EXCEEDED
2146  022314  002010                          BGE     13$             ;YES - SKIP
2147  022316  024445                          CMP     -(R4),-(R5)     ;SET PTRS BACK TO ERROR WORDS
2148  022320                                  ERRHRD  10035,,,ERR10   ;REPORT ERROR
  (3) 022320  104443                          TRAP    TSERCODE
  (5) 022322  023463                          .WORD   10035
  (5) 022324  012414                          .WORD   ERR10
2149  022326  005737  002440                  CLR     ERRSWI          ;CLEAR ERROR SWITCH
2150  022332  022425                          CMP     (R4)+,(R5)+     ;BUMP PTRS PAST ERROR WORDS
2151  022334  000712                          BR      9$              ;DO NEXT COMPARE
2152  022336  005237  002436          13$:    INC     MORECE          ;BUMP ERROR COUNTER
2153  022342  000707                          BR      7$              ;DO NEXT COMPARE
2154
2155                                          ;
                                              WRITE AND READ DATA ROUTINE.
2156  022344  012737  177777  002542  XWRITT: MOV     #-1,TEMP1       ;SET SPECIAL WRITE FOR TIMING FLAG
2157  022352  000402                          BR      XWRIT1
2158  022354  005037  002542          XWRITE: CLR     TEMP1           ;CLEAR SPECIAL WRITE FLAG
2159  022360  012737  000112  002556  XWRIT1: MOV     #WTDATA,TEMP7   ;SET FOR WRITE
2160  022366  027737  000377  002526          CMP     #255.,CURCYL    ;TEST IF CYLINDER 255 (BAD SEC)
2161  022374  001006                          BNE     1$              ;NO - SKIP
2162  022376  005737  002534                  TST     DESHD           ;TEST IF HEAD 1 (BAD SECTOR FILES)
2163  022402  001403                          BEQ     1$              ;NO - SKIP
2164  022404  052737  004000  002426          BIS     #BADADD,OPFLAG  ;SET BAD ADDRESS FLAG
2165  022412  000403                          BR      XREADG          ;SKIP TO EXECUTE
2166  022414  012737  000114  002556  XREAD:  MOV     #RDDATA,TEMP7   ;SET FOR READ
2167  022422  010346                  XREADG: MOV     R3,-(SP)        ;STORE R3
2168  022424  013703  002424                  MOV     SSINDX,R3       ;SET SUBROUTINE INDEX
2169  022430  005723                          TST     (R3)+           ;BUMP TO NEXT STACK ENTRY
2170  022432  016663  000002  002260          MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
2171  022440  162763  000004  002260          SUB     #4,SUBSTK(R3)   ;ADJUST TO POINT TO CALL
2172  022446  010337  002424                  MOV     R3,SSINDX       ;STORE IT BACK
2173  022452  010046                          MOV     R0,-(SP)
2174  022454  010146                          MOV     R1,-(SP)        ;STORE OTHER REGISTERS
2175  022456  010446                          MOV     R4,-(SP)
2176  022460  004737  017130                  JSR     PC,RDYCHK       ;CHECK IF DRIVE READY
2177  022464  023034                          65$
2178  022466  012703  002456                  MOV     #L.CS,R3        ;GET ADDRESS OF LOAD REGS
2179  022472  013713  002556                  MOV     TEMP7,(R3)      ;SET COMMAND
2180  022476  053713  002454                  BIS     RLDRV,(R3)      ;INSERT DRIVE NUMBER
2181  022502  042713  002000                  BIC     #BIT10,(R3)     ;CLEAR FOR DRIVE 4 - 7 SPEC'D
2182  022506  032723  000004                  BIT     #BIT2,(R3)+     ;TEST IF WRITE DATA
2183  022512  001403                          BEQ     3$              ;YES - SKIP
2184  022514  012723  003466                  MOV     #IBUFF,(R3)+    ;ELSE SET BA FOR READ
2185  022520  000402                          BR      4$
2186  022522  012723  004066          3$:     MOV     #OBUFF,(R3)+    ;SET BA FOR WRITE
```



```
2187  022526  013713  002526          4$:     MOV     CURCYL,(R3)     ;GET CURRENT CYLINDER
2188  022532  012704  000007                  MOV     #7,R4           ;ALIGN IT IN DA
2189  022536  006313                  5$:     ASL     (R3)
2190  022540  005304                          DEC     R4
2191  022542  001375                          BNE     5$
2192  022544  005737  002534                  TST     DESHD           ;TEST IF HEAD 0
2193  022550  001402                          BEQ     7$              ;YES - SKIP
2194  022552  052713  000100                  BIS     #HSMSK,(R3)     ;SET FOR HEAD 1
2195  022556  053723  002536          7$:     BIS     DESSEC,(R3)+    ;INSERT DESIRED SECTOR
2196  022562  012713  177600                  MOV     #177600,(R3)    ;INSERT WORD COUNT
2197  022566  005737  002542                  TST     TEMP1           ;CHECK IF SPECIAL WRITE FOR TIMING
2198  022572  001402                          BEQ     8$              ;NO - SKIP
2199  022574  012713  177777                  MOV     #177777,(R3)    ;ELSE SET FOR 1 WORD TRANSFER
2200  022600  032737  004000  002426  8$:     BIT     #BADADD,OPFLAG  ;TEST IF BAD ADDRESS FLAG SET
2201  022606  001413                          BEQ     2$              ;NO - SKIP
2202  022610  042737  173777  002426          BIC     #^CBADADD,OPFLAG  ;CLEAR ALL BUT THIS FLAG
2203  022616  012703  010307                  MOV     #MWRTAB,R3      ;SET RESULT MESSAGE POINTER
2204  022622                                  ERRHRD  10032,,,ERR1
  (3) 022622  104443                          TRAP    TSERCODE
  (5) 022624  023460                          .WORD   10032
  (5) 022626  011554                          .WORD   ERR1
2205  022630  005037  002426                  CLR     OPFLAG          ;CLEAR ALL FLAGS
2206  022634  000475                          BR      64$
2207  022636  005037  002430          2$:     CLR     DONE            ;CLEAR INTERRUPT FLAG
2208  022642  005737  002542                  TST     TEMP1           ;CHECK IF SPECIAL WRITE FLAG SET
2209  022646  001072                          BNE     65$             ;YES - DO NOT START WRITE
2210  022650  011362  000006                  MOV     (R3),RLMP(R2)   ;LOAD RL REGS
2211  022654  014362  000004                  MOV     -(R3),RLDA(R2)
2212  022660  014362  000002                  MOV     -(R3),RLBA(R2)
2213  022664  014362  000000                  MOV     -(R3),RLCS(R2)
2214  022670                          10$:    WAITUS  #3000.          ;WAIT 300MS FOR INTERRUPT
  (3) 022670  012700  005670                  MOV     #3000.,R0
  (3) 022674  104027                          EMT     CSWTU
2215  022676  005737  002430                  TST     DONE            ;CHECK IF INTERRUPT
2216  022702  001007                          BNE     14$             ;YES - SKIP
2217  022704  004737  015026                  JSR     PC,WAITIN       ;WAIT FOR INTERRUPT
2218  022710  012603                          MOV     (SP)+,R3        ;GET RESULT MESSAGE
2219  022712                                  ERRHRD  10030,,,ERR1
  (3) 022712  104443                          TRAP    TSERCODE
  (5) 022714  023456                          .WORD   10030
  (5) 022716  011554                          .WORD   ERR1
2220  022720  000443                          BR      64$
2221  022722  032737  000001  002466  14$:    BIT     #DRDYMSK,T.CS   ;TEST IF DRIVE READY
2222  022730  001031                          BNE     20$             ;YES - SKIP
2223  022732  012703  007543                  MOV     #MDRDY,R3       ;SET RESULT MESSAGE
2224  022736  012704  010510                  MOV     #CAFDT,R4       ;CONDITION AFTER DATA XFER
2225  022742                                  ERRHRD  10032,,,ERR5
  (3) 022742  104443                          TRAP    TSERCODE
  (5) 022744  023460                          .WORD   10032
  (5) 022746  012006                          .WORD   ERR5
2226  022750  012701  000062                  MOV     #50.,R1         ;SET WAIT COUNT FOR 5 SECDS
2227  022754  004737  015226          17$:    JSR     PC,GSTAT        ;GET DRIVE STATUS
2228  022760  023030                          64$
2229  022762  032737  000001  002466          BIT     #DRDYMSK,T.CS   ;TEST IF DRIVE READY NOW
2230  022770  001011                          BNE     20$             ;YES - SKIP
2231  022772  005301                          DEC     R1              ;DEC WAIT COUNT
```

```
 2232  022774  001367                        BNE     17$             ;LOOP IF NOT TIME DONE
 2233  022776  012704  010522                MOV     #CSSEC,R4       ;SET CONDITION 5 SECONDS
 2234  023002                                ERRHRD  10033.,6,ERR5
   (3) 023002  104443                        TRAP    TSERCODE
   (5) 023004  023461                        .WORD   10033
   (5) 023006  012006                        .WORD   ERR5
 2235  023010  005037  002440                CLR     ERRSWI          ;CLEAR ERROR SWITCH
 2236  023014  005737  002466          20$:  TST     T.CS            ;CHECK IF ANY ERROR
 2237  023020  100005                        BPL     65$             ;NO - SKIP
 2238  023022                                ERRHRD  10031.,6,ERR6
   (3) 023022  104443                        TRAP    TSERCODE
   (5) 023024  023457                        .WORD   10031
   (5) 023026  012056                        .WORD   ERR6
 2239  023030  005037  002440          64$:  CLR     ERRSWI          ;CLEAR ERROR SWITCH
 2240  023034  162737  000002  002424  65$:  SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
 2241  023042  012604                        MOV     (SP)+,R4        ;RESTORE REGISTERS
 2242  023044  012601                        MOV     (SP)+,R1
 2243  023046  012600                        MOV     (SP)+,R0
 2244  023050  012603                        MOV     (SP)+,R3
 2245  023052  005737  002440                TST     ERRSWI          ;TEST IF ERROR RETURN
 2246  023056  001403                        BEQ     99$             ;YES - SKIP
 2247  023060  062716  002440                ADD     ERRSWI,(SP)     ;ELSE ADD IN ERROR RETURN
 2248  023064  000207                        RTS     PC
 2249  023066  017616  000000          99$:  MOV     @(SP),(SP)      ;ADJUST FOR ERROR RETURN
 2250  023072  000207                        RTS     PC
 2251
 2252                                  ;       BAD SECTOR CHECK ROUTINE. CHECKS IF SECTOR SPECIFIED IN CURCYL,
 2253                                  ;       DESHD, AND DESSEC IS LISTED AS BAD IN THE BAD SECTOR FILES.
 2254  023074  010046                  BSCHK:  MOV     R0,-(SP)        ;STORE REGISTERS
 2255  023076  010146                        MOV     R1,-(SP)
 2256  023100  010346                        MOV     R3,-(SP)
 2257  023102  005037  002442                CLR     BSFLAG          ;CLEAR FLAG
 2258  023106  012703  003277                MOV     #FBSFIL,R3      ;GET POINTER TO FACTORY FILE
 2259  023112  022713  177777          1$:   CMP     #-1,(R3)        ;CHECK IF ALL ONES
 2260  023116  001005                        BNE     4$              ;NO SKIP TO TEST
 2261  023120  012703  003076                MOV     #SBSFIL,R3      ;ELSE SET POINTER TO SOFTWARE FILE
 2262  023124  022713  177777                CMP     #-1,(R3)        ;CHECK IF ALL ONES
 2263  023130  001431                        BEQ     20$             ;YES - EXIT
 2264  023132  013700  002524          4$:   MOV     NEWCYL,R0       ;BUILD HEADER OF ADDRESS IN QUESTION
 2265  023136  012701  000007                MOV     #7,R1           ;POSITION CYLINDER
 2266  023142  006300                  5$:   ASL     R0
 2267  023144  005301                        DEC     R1
 2268  023146  001375                        BNE     5$
 2269  023150  005737  002534                TST     DESHD           ;CHECK IF HEAD 0
 2270  023154  001402                        BEQ     7$              ;YES - SKIP
 2271  023156  052700  000100                BIS     #BIT6,R0        ;INSERT HEAD 1
 2272  023162  053700  002536          7$:   BIS     DESSEC,R0       ;INSERT SECTOR
 2273  023166  022300                  8$:   CMP     (R3)+,R0        ;CHECK THIS WORD IN FILE
 2274  023170  001402                        BEQ     12$             ;YES - EXIT,ERROR
 2275  023172  101005                        BHI     15$             ;EXIT- NO ERROR
 2276  023174  000774                        BR      8$
 2277  023176  012737  000001  002442  12$:  MOV     #1,BSFLAG       ;SET ERROR FLAG
 2278  023204  000403                        BR      20$             ;GO TO EXIT
 2279  023206  020327  003272          15$:  CMP     R3,#FBSFIL      ;DONE BOTH FILES?
 2280  023212  003342                        BGT     2$              ;NO GO DO SOFTWARE FILE
 2281  023214  012603                  20$:  MOV     (SP)+,R3        ;ELSE RESTORE REGISTERS
```



```
 2282  023216  012601                        MOV     (SP)+,R1
 2283  023220  012600                        MOV     (SP)+,R0
 2284  023222  005737  002442                TST     BSFLAG          ;CHECK IF ERROR
 2285  023226  001003                        BNE     99$             ;YES - SKIP
 2286  023230  062716  000002                ADD     #2,(SP)         ;ELSE BUMP ERROR RETURN
 2287  023234  000207                        RTS     PC
 2288  023236  017616  000000          99$:  MOV     @(SP),(SP)      ;SET FOR ERROR RETURN
 2289  023242  000207                        RTS     PC
 2290
 2292                                  ;       REPORT OPERATION ROUTINE. PRINTS SUBROUTINE TRACE SEQUENCE AND
 2293                                  ;       OPERATION BEING PERFORMED PORTION OF ALL
 2294                                  ;       ERROR MESSAGES.
 2295  023244  010446                  RPTOP:  MOV     R4,-(SP)
 2296  023246  005737  002424                TST     SSINDX          ;TEST SUBROUTINE INDEX 0
 2297  023252  001433                        BEQ     1$              ;SKIP IF 0
 2298  023254  012704  000002                MOV     #2,R4           ;SET INDEXER TO FIRST ENTRY
 2299  023260                                PRINTB  #FMT9,#SEQMES   ;PRINT "SUBROUTINE CALL SEQ"
   (8) 023260  012746  007366                MOV     #SEQMES,-(SP)
   (7) 023264  012746  011043                MOV     #FMT9,-(SP)
   (6) 023270  012746  000002                MOV     #2,-(SP)
   (3) 023274  010600                        MOV     SP,R0
   (4) 023276  104014                        EMT     CSPNTB
   (4) 023300  062706  000006                ADD     #6,SP
 2300  023304                          3$:   PRINTB  #FMT16,SUBSTK(R4)       ;PRINT CALLING LOCATION
   (8) 023304  016446  002260                MOV     SUBSTK(R4),-(SP)
   (7) 023310  012746  011216                MOV     #FMT16,-(SP)
   (6) 023314  012746  000002                MOV     #2,-(SP)
   (3) 023320  010600                        MOV     SP,R0
   (4) 023322  104014                        EMT     CSPNTB
   (4) 023324  062706  000006                ADD     #6,SP
 2301  023330  062704  000002                ADD     #2,R4           ;BUMP INDEX
 2302  023334  020437  002424                CMP     R4,SSINDX       ;CHECK IF ALL PRINTED
 2303  023340  003761                        BLE     3$              ;LOOP IF NOT ALL PRINTED YET
 2304  023342                          1$:   PRINTB  #FMT4,ERHEAD,#TSTLAB    ;PRINT ERROR HEADER
   (9) 023342  012746  006217                MOV     #TSTLAB,-(SP)
   (8) 023346  013746  002434                MOV     ERHEAD,-(SP)
   (7) 023352  012746  010646                MOV     #FMT4,-(SP)
   (6) 023356  012746  000003                MOV     #3,-(SP)
   (3) 023362  010600                        MOV     SP,R0
   (4) 023364  104014                        EMT     CSPNTB
   (4) 023366  062706  000010                ADD     #10,SP
 2305  023372  042737  030000  002426        BIC     #SEEKOP!RORWOP,OPFLAG   ;CLEAR SK & RD OR WRT FLAG
 2306  023400  013701  002456                MOV     L.CS,R1         ;GET COMMAND EXECUTED
 2307  023404  042701  177741                BIC     #177741,R1      ;STRIP ALL BUT FUNCTION CODE
 2308  023410  022701  000006                CMP     #6,R1   ;TEST IF SEEK OPERATION
 2309  023414  001003                        BNE     2$              ;NO - SKIP
 2310  023416  052737  010000  002426        BIS     #SEEKOP,OPFLAG  ;ELSE SET SEEK FLAG
 2311  023424  022701  000012          2$:   CMP     #12,R1          ;TEST IF WRITE
 2312  023430  001003                        BNE     20$             ;NO - SKIP
 2313  023432  052737  020000  002426        BIS     #RORWOP,OPFLAG  ;SET RD OR WRT FLAG
 2314  023440  022701  000014          20$:  CMP     #14,R1          ;TEST IF READ
 2315  023444  001003                        BNE     22$             ;NO - SKIP
 2316  023446  052737  020000  002426        BIS     #RORWOP,OPFLAG  ;SET RD OR WRT FLAG
 2317  023454                          22$:  PRINTB  #FMT1,#MOPER,OPMSGS(R1) ;PRINT OPERATION
   (9) 023454  016146  002122                MOV     OPMSGS(R1),-(SP)
   (8) 023460  012746  005143                MOV     #MOPER,-(SP)
```

```
                023464  012746  010624                  MOV     #FMT1,-(SP)
  (6)           023470  012746  000003                  MOV     #3,-(SP)
  (3)           023474  010600                          MOV     SP,R0
  (4)           023476  104014                          EMT     C$PNTB
  (4)           023500  062706  000010                  ADD     #10,SP
2318            023504  020127  000004                  CMP     R1,#4           ;CHECK IF GET STATUS
2319            023510  001007                          BNE     4$              ;NO - SKIP
2320            023512  032737  000010  002462          BIT     #DRSET,L.DA     ;TEST IF RESET INCLUDED
2321            023520  001403                          BEQ     4$              ;NO - SKIP
2322            023522  012701  000016                  MOV     #16,R1          ;SET TO PRINT WITH RESET
2323            023526  000436                          BR      9$
2324            023530  032737  007777  002426  4$:     BIT     #COMPOP,OPFLAG  ;TEST IF ANY OTHER OPERATION
2325            023536  001424                          BEQ     8$              ;NO - SKIP
2326            023540  013704  002426                  MOV     OPFLAG,R4       ;SET UP TO DETERMINE WHICH ONE
2327            023544  012701  000020                  MOV     #20,R1          ;PRESET THE POINTER
2328            023550  032704  000001          5$:     BIT     #BIT00,R4       ;CHECK THE BIT
2329            023554  001003                          BNE     6$              ;IF SET - SKIP
2330            023556  005721                          TST     (R1)+           ;BUMP POINTER
2331            023560  006204                          ASR     R4
2332            023562  000772                          BR      5$
2333            023564                          6$:     PRINTB  #FMT2,OPMSGS(R1)
  (8)           023564  016146  002122                  MOV     OPMSGS(R1),-(SP)
  (7)           023570  012746  010640                  MOV     #FMT2,-(SP)
  (6)           023574  012746  000002                  MOV     #2,-(SP)
  (3)           023600  010600                          MOV     SP,R0
  (4)           023602  104014                          EMT     C$PNTB
  (4)           023604  062706  000006                  ADD     #6,SP
2334            023610  032737  100000  002426  8$:     BIT     #HDR40,OPFLAG   ;TEST IF 40 HEADER OPERATION
2335            023616  001415                          BEQ     10$             ;NO - SKIP
2336            023620  012701  000050                  MOV     #50,R1          ;ELSE PRINT IT
2337            023624                          9$:     PRINTB  #FMT2,OPMSGS(R1)
  (8)           023624  016146  002122                  MOV     OPMSGS(R1),-(SP)
  (7)           023630  012746  010640                  MOV     #FMT2,-(SP)
  (6)           023634  012746  000002                  MOV     #2,-(SP)
  (3)           023640  010600                          MOV     SP,R0
  (4)           023642  104014                          EMT     C$PNTB
  (4)           023644  062706  000006                  ADD     #6,SP
2338            023650  000434                          BR      15$             ;SKIP
2339            023652  032737  010000  002426  10$:    BIT     #SEEKOP,OPFLAG  ;TEST IF SEEK
2340            023660  001430                          BEQ     15$             ;NO - SKIP
2341            023662                                  PRINTB  #FMT13,#FRMWD,OLDCYL,#DIFWD,DESDIF,#SGNWD,DESSGN,#HDWD,DESHD
  (15)          023662  013746  002534                  MOV     DESHD,-(SP)
  (14)          023666  012746  007327                  MOV     #HDWD,-(SP)
  (13)          023672  013746  002532                  MOV     DESSGN,-(SP)
  (12)          023676  012746  007322                  MOV     #SGNWD,-(SP)
  (11)          023702  013746  002530                  MOV     DESDIF,-(SP)
  (10)          023706  012746  007314                  MOV     #DIFWD,-(SP)
  (9)           023712  013746  002527                  MOV     OLDCYL,-(SP)
  (8)           023716  012746  007345                  MOV     #FRMWD,-(SP)
  (7)           023722  012746  010064                  MOV     #FMT13,-(SP)
  (6)           023726  012746  000011                  MOV     #11,-(SP)
  (3)           023732  010600                          MOV     SP,R0
  (4)           023734  104014                          EMT     C$PNTB
  (4)           023736  062706  000024                  ADD     #24,SP
2342            023742  032737  020000  002426  15$:    BIT     #RDRWOP,OPFLAG  ;TEST IF READ OR WRITE SET
2343            023750  001424                          BEQ     17$             ;NO - SKIP
```



```
2344            023752                                  PRINTB  #FMT22,#CYLWD,CURCYL,#HDWD,DESHD,#SECWD,DESSEC
  (13)          023752  013746  002536                  MOV     DESSEC,-(SP)
  (12)          023756  012746  007333                  MOV     #SECWD,-(SP)
  (11)          023762  013746  002534                  MOV     DESHD,-(SP)
  (10)          023766  012746  007327                  MOV     #HDWD,-(SP)
  (9)           023772  013746  002526                  MOV     CURCYL,-(SP)
  (8)           023776  012746  007340                  MOV     #CYLWD,-(SP)
  (7)           024002  012746  011413                  MOV     #FMT22,-(SP)
  (6)           024006  012746  000007                  MOV     #7,-(SP)
  (3)           024012  010600                          MOV     SP,R0
  (4)           024014  104014                          EMT     C$PNTB
  (4)           024016  062706  000020                  ADD     #20,SP
2345            024022  004737  024474          17$:    JSR     PC,CLRPARM      ;CLEAR PARAM TABLE
2346            024026  012604                          MOV     (SP)+,R4        ;RESTORE R4
2347            024030  000207                          RTS     PC
2348
2349                                            ;       REPORT REASON ROUTINE
2350                                            ;       PRINTS REASON PORTION FOR ALL ERROR REPORTS.
2351            024032  010146                  RPTRES: MOV     R1,-(SP)        ;STORE R1
2352            024034  010346                          MOV     R3,-(SP)        ;STORE R3
2353            024036  010446                          MOV     R4,-(SP)        ;STORE R4
2354            024040  012701  002504                  MOV     #RESPARM,R1     ;GET START OF PARAM
2355            024044  012103                          MOV     (R1)+,R3        ;GET NUMBER OF PARAM
2356            024046                                  PRINTB  #FMT1.1,#MRSLT,(R1)     ;PRINT NAME
  (9)           024046  011146                          MOV     (R1),-(SP)
  (8)           024050  012746  005157                  MOV     #MRSLT,-(SP)
  (7)           024054  012746  010631                  MOV     #FMT1.1,-(SP)
  (6)           024060  012746  000003                  MOV     #3,-(SP)
  (3)           024064  010600                          MOV     SP,R0
  (4)           024066  104014                          EMT     C$PNTB
  (4)           024070  062706  000010                  ADD     #10,SP
2357            024074  021127  010141                  CMP     (R1),#MNDRST    ;TEST IF MESSAGE IS NO DRV STATUS
2358            024100  001453                          BEQ     6$              ;YES - SKIP REST OF REPORT
2359            024102  012704  011050                  MOV     #FMT11,R4       ;PRISET FOR FORMAT 11
2360            024106  021127  010134                  CMP     (R1)+,#MCYLOC   ;CHECK IF REPORTING CYLINDER LOC
2361            024112  001002                          BNE     3$              ;NO - SKIP
2362            024114  012704  011056                  MOV     #FMT12,R4       ;ELSE CHANGE TO FORMAT 12
2363            024120  005303          3$:     DEC     R3              ;DEC PARAM COUNT
2364            024122  001442                          BEQ     6$              ;IF 0 - EXIT
2365            024124                                  PRINTB  R4,#RESE3,(R1)+  ;REPORT IS VALUE
  (9)           024124  012146                          MOV     (R1)+,-(SP)
  (8)           024126  012746  010403                  MOV     #RESE3,-(SP)
  (7)           024132  010446                          MOV     R4,-(SP)
  (6)           024134  012746  000003                  MOV     #3,-(SP)
  (3)           024140  010600                          MOV     SP,R0
  (4)           024142  104014                          EMT     C$PNTB
  (4)           024144  062706  000010                  ADD     #10,SP
2366            024150                                  PRINTB  R4,#RESE4,(R1)+  ;REPORT SB VALUE
  (9)           024150  012146                          MOV     (R1)+,-(SP)
  (8)           024152  012746  010407                  MOV     #RESE4,-(SP)
  (7)           024156  010446                          MOV     R4,-(SP)
  (6)           024160  012746  000003                  MOV     #3,-(SP)
  (3)           024164  010600                          MOV     SP,R0
  (4)           024166  104014                          EMT     C$PNTB
  (4)           024170  062706  000010                  ADD     #10,SP
2367            024174  162703  000002                  SUB     #2,R3           ;DEC PARAM COUNT
```

```
 2368  024200  001413                          BEQ     6$                      ;IF 0 - EXIT
 2369  024202                                  PRINTB  #FMT1,#RESE5,(R1)+  ;REPORT CONDITION
   (9) 024202  012146                          MOV     (R1)+,-(SP)
   (8) 024204  012746  010414                  MOV     #RESE5,-(SP)
   (7) 024210  012746  010624                  MOV     #FMT1,-(SP)
   (6) 024214  012746  000003                  MOV     #3,-(SP)
   (3) 024220  010600                          MOV     SP,R0
   (4) 024222  104014                          EMT     CSPNTB
   (4) 024224  062706  000010                  ADD     #10,SP
 2370  024230  012604                  6$:     MOV     (SP)+,R4        ;RESTORE REGS
 2371  024232  012603                          MOV     (SP)+,R3
 2372  024234  012601                          MOV     (SP)+,R1
 2373  024236  000207                          RTS     PC              ;RETURN
 2374
 2375                                  ;       REPORT PHYSICAL ADDRESS OF DEVICE UNDER TEST
 2376                                  ;       AND ALL REGISTER CONTENTS.
 2377  024240                          RPTREM: PRINTB  #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
   (11) 024240  005046                          CLR     -(SP)
   (11) 024242  153716  002455                  BISB    RLDRV+1,(SP)
   (10) 024246  012746  005633                  MOV     #DRVNAM,-(SP)
   (9)  024252  013746  002450                  MOV     RLBAS,-(SP)
   (8)  024256  012746  005622                  MOV     #BASADD,-(SP)
   (7)  024262  012746  010657                  MOV     #FMT5,-(SP)
   (6)  024266  012746  000005                  MOV     #5,-(SP)
   (3)  024272  010600                          MOV     SP,R0
   (4)  024274  104014                          EMT     CSPNTB
   (4)  024276  062706  000014                  ADD     #14,SP
 2378                                  ;       REPORT RL11 REGISTERS
 2379  024302                                  PRINTB  #FMT6,#CSNAM,#DANAM,#BANAM,#MPNAM,#CYLWD,#HDWD
   (13) 024302  012746  007327                  MOV     #HDWD,-(SP)
   (12) 024306  012746  007340                  MOV     #CYLWD,-(SP)
   (11) 024312  012746  005752                  MOV     #MPNAM,-(SP)
   (10) 024316  012746  005740                  MOV     #BANAM,-(SP)
   (9)  024322  012746  005745                  MOV     #DANAM,-(SP)
   (8)  024326  012746  005733                  MOV     #CSNAM,-(SP)
   (7)  024332  012746  010677                  MOV     #FMT6,-(SP)
   (6)  024336  012746  000007                  MOV     #7,-(SP)
   (3)  024342  010600                          MOV     SP,R0
   (4)  024344  104014                          EMT     CSPNTB
   (4)  024346  062706  000020                  ADD     #20,SP
 2380  024352                                  PRINTB  #FMT8,#LAB1,L.CS,L.DA,L.BA,L.MP
   (12) 024352  013746  002464                  MOV     L.MP,-(SP)
   (11) 024356  013746  002460                  MOV     L.BA,-(SP)
   (10) 024362  013746  002462                  MOV     L.DA,-(SP)
   (9)  024366  013746  002456                  MOV     L.CS,-(SP)
   (8)  024372  012746  005757                  MOV     #LAB1,-(SP)
   (7)  024376  012746  011011                  MOV     #FMT8,-(SP)
   (6)  024402  012746  000006                  MOV     #6,-(SP)
   (3)  024406  010600                          MOV     SP,R0
   (4)  024410  104014                          EMT     CSPNTB
   (4)  024412  062706  000016                  ADD     #16,SP
 2381  024416                                  PRINTB  #FMT7,#LAB2,T.CS,T.DA,T.BA,T.MP,CURCYL,DESHD
   (14) 024416  013746  002534                  MOV     DESHD,-(SP)
   (13) 024422  013746  002526                  MOV     CURCYL,-(SP)
   (12) 024426  013746  002474                  MOV     T.MP,-(SP)
   (11) 024432  013746  002470                  MOV     T.BA,-(SP)
```



```
   (10) 024436  013746  002472                  MOV     T.DA,-(SP)
   (9)  024442  013746  002466                  MOV     T.CS,-(SP)
   (8)  024446  012746  005772                  MOV     #LAB2,-(SP)
   (7)  024452  012746  010741                  MOV     #FMT7,-(SP)
   (6)  024456  012746  000010                  MOV     #10.,-(SP)
   (3)  024462  010600                          MOV     SP,R0
   (4)  024464  104014                          EMT     CSPNTB
   (4)  024466  062706  000022                  ADD     #22,SP
 2382  024472  000207                          RTS     PC
 2383
 2384                                  ;       CLEAR PARAMETER BLOCK FOR REPORTING
 2385  024474  010546                  CLRPARM:        MOV     R5,-(SP)        ;STORE R5
 2386  024476  012701  002504                  MOV     #RESPARM,R1     ;GET ADDRESS OF BLOCK
 2387  024502  012705  000005                  MOV     #5,R5           ;SET COUNT
 2388  024506  005021                  2$:     CLR     (R1)+           ;CLEAR WORD
 2389  024510  005305                          DEC     R5              ;DEC COUNT
 2390  024512  001375                          BNE     2$              ;LOOP UNTIL 0
 2391  024514  012701  002504                  MOV     #RESPARM,R1     ;RESET POINTER
 2392  024520  012605                          MOV     (SP)+,R5        ;RESTORE R5
 2393  024522  000207                          RTS     PC
 2394
 2395  024524                          ENDMOD
 2396
```

```
                        BGMMOD  HRDWTST
                        .SBTTL  *TEST 1          **DIFFERENCE OF 1 SEEK (PART 1)
                        BGMTST                   ;TEST 1
 (3) 024524                                                                  T1::
2401 024524 012737 006225 002434        MOV     #P2T01E,ERHEAD  ;SET ERROR HEADER
2402 024532 012737 000004 002540        MOV     #4,TEMP0        ;SET PASS COUNT
2403 024540 004737 015160               JSR     PC,TSTINT       ;INITIALIZE TEST
2404 024544 004737 015176               JSR     P6,GSTATR       ;GET STATUS
2405 024550 025044                      T1765$
2406 024552 012737 177777 002544        MOV     #-1,TEMP2       ;SET -1 INTO DIFF AUGMENT FOR -1 SEEK
2407 024560 012704 002526               MOV     #CURCYL,R4      ;SET POINTERS
2408 024564 012705 002524               MOV     #NEWCYL,R5
2409 024570 004737 017370               JSR     PC,CHOSHD       ;GO CHOSE HEAD
2410 024574                     T172$:
2411 024574                     BGMSUB
 (3) 024574                                                                  T1.1:
 (3) 024574 104002                      EMT     C$BSUB
2412 024576 004737 021116               JSR     PC,GETPOS       ;GET POSITION
2413 024602 025002                      60$
2414 024604                             INLOOP                  ;CHECK IF IN ERROR LOOP
 (3) 024604 104020                      EMT     C$INLP
2415 024606                             BNCOMPLETE      3$      ;NO - SKIP
 (2) 024606 103005                      BCC     3$
2416 024610 021415                      CMP     (R4),(R5)       ;CHECK IF CURRENT = NEW
2417 024612 001005                      BNE     4$              ;NO - SKIP
2418 024614 004737 017454               JSR     PC,ONSWAP       ;ELSE SWAP OLD AND NEW
2419 024620 000421                      BR      9$              ;SKIP TO SEEK
2420 024622 005437 002544       3$:     NEG     TEMP2           ;CHANGE DIFF AUGMENT FOR OPPOSITE DIR
2421 024626 011415              4$:     MOV     (R4),(R5)       ;MOV    CURRENT INTO OLD
2422 024630 021414                      CMP     #255.,(R4)      ;CHECK IF CURRENT AT 255
2423 024634 000377
2424 024636 001004                      BNE     7$              ;NO - SKIP
2425 024640 012737 177777 002544        MOV     #-1,TEMP2       ;AT MAX CYL, MAKE NEXT SEEK REV
2426 024644 000405                      BR      8$              ;SKIP
2427 024646 005714              7$:     TST     (R4)            ;TEST IF CURRENT AT 0
2428 024650 001003                      BNE     8$              ;NO - SKIP
2429 024652 012737 000001 002544        MOV     #1,TEMP2        ;AT CYL 0, MAKE NEXT SEEK FWRD
2430 024660 063715 002544       8$:     ADD     TEMP2,(R5)      ;ADD DIFF TO NEW CYL (+1 OR -1)
2431 024664 004737 016070       9$:     JSR     PC,XSEEK        ;DO SEEK
2432 024670 025002                      60$
2433 024672 004737 015226               JSR     PC,GSTAT        ;GET STATUS
2434 024676 025002                      60$
2435
2435 024700 012703 000004               MOV     #4,R3           ;SET EXPECTED STATE
2436 024704 020337 002502               CMP     R3,T.STAT       ;CHECK IF STATE COUNT
2437 024710 001404                      BEQ     10$             ;YES-SKIP
2438 024712                             ERRHRD  101,,ERR7       ;REPORT STATE ERROR
 (3) 024712 104443                      TRAP    T$ERCODE
 (5) 024714 000145                      .WORD   101
 (5) 024716 012734                      .WORD   ERR7
2439 024720 000423                      BR      16$             ;EXIT TEST
2440 024722 012703 000005       10$:    MOV     #5,R3           ;SET EXPECTED STATE
2441 024726 012701 000062               MOV     #50.,R1         ;SET WAIT COUNT FOR 5 MS
2442 024732 004737 015226       12$:    JSR     PC,GSTAT        ;GET STATUS
2443 024736 025002                      60$
2444 024740 020337 002502               CMP     R3,T.STAT       ;IS STATE 5?
2445 024744 001411                      BEQ     16$             ;YES-SKIP
```



```
2446 024746 005301                      DEC     R1              ;DEC WAIT COUNT
2447 024750 001404                      BEQ     14$             ;SKIP IF 0
2448 024752                             WAITUS  #1
 (3) 024752 012700 000001               MOV     #1,R0
 (3) 024756 104027                      EMT     C$WTU
2449 024760 000764                      BR      12$
2450 024762                     14$:    ERRHRD  102,,ERR7       ;REPORT STATE ERROR
 (3) 024762 104443                      TRAP    T$ERCODE
 (5) 024764 000146                      .WORD   102
 (5) 024766 012734                      .WORD   ERR7
2451 024770 012701 000062       16$:    MOV     #50.,R1         ;SET WAIT COUNT FOR 5 MS
2452 024774 004737 020650               JSR     PC,RDYWAIT      ;GO WAIT FOR DRIVE READY
2453 025000 025002                      60$
2454 025002 012737 000002 002440 60$:   MOV     #2,ERRSWI       ;INIT ERROR SWITCH
2455 025010                     ENDSUB
 (3) 025010                     L10021:
 (3) 025010 104003                      EMT     C$ESUB
2456 025012                             ESCAPE  TST             ;EXIT TEST IF ERROR
 (3) 025012 104010                      EMT     C$ESCAPE
 (3) 025014 000030                      .WORD   L10020-.
2457 025016 005337 002540               DEC     TEMP0           ;DEC PASS COUNT
2458 025022 001410                      BEQ     24$             ;SKIP IF 0-DONE
2459
2460 025024 032737 000001 002540        BIT     #BIT0,TEMP0     ;TEST IF PASS=2
2461 025032 001003                      BNE     23$             ;NO-SKIP
2462 025034 004737 017414               JSR     PC,SWAPHD       ;GO SWAP TO HEAD 1 OR END TEST
2463 025040 025044                      24$                     ;ABORT RETURN
2464 025042 000654              23$:    BR      T172$
2465 025044                     24$:
2466 025044                     T1765$:
2467 025044                     ENDTST
 (3) 025044                     L10020:
 (3) 025044 104001                      EMT     C$ETST
```

```
 2469
 2470
 2471                                           .SBTTL  *TEST 2          **DIFFERENCE OF 1 SEEK (PART 2)
 2472   025046                                  BGNTST                   ;TEST 2
  (3)   025046                                                                                    T2::
 2473   025046  012737  006225  002434                  MOV     #P2T02E,ERHEAD  ;SET ERROR HEADER
 2474   025054  012737  000004  002540                  MOV     #4,TEMP0        ;SET PASS COUNT
 2475   025062  004737  015160                          JSR     PC,TSTINT       ;INITIALIZE TEST
 2476   025066  004737  015176                          JSR     PC,GSTATR       ;GET STATUS, CLEAR DRIVE
 2477   025072  025334                                  T1865$
 2478   025074  004737  017370                          JSR     PC,CHOSHD       ;GO CHOSE HEAD
 2479   025100  012737  177777  002544                  MOV     #-1,TEMP2       ;SET DIFF AUGMENT TO -1 (REVERSE)
 2480   025106  012703  002524                          MOV     #NEWCYL,R3      ;GET ADDRESSES
 2481   025112  012704  002526                          MOV     #CURCYL,R4
 2482   025116  012705  002522                          MOV     #OLDCYL,R5
 2483   025122                                  T187$:
 2484   025122                                  BGNSUB
  (3)   025122                                                                                    T2.1:
  (3)   025122  104002                                  EMT     C$BSUB
 2485   025124  004737  021116                          JSR     PC,GETPOS       ;GET CURRENT POSITION
 2486   025130  025272                                  60$
 2487   025132                                          INLOOP                  ;CHECK IF IN ERROR LOOP
  (3)   025132  104020                                  EMT     C$INLP
 2488   025134                                          BNCOMPLETE      3$      ;NO - SKIP
  (2)   025134  103005                                  BCC     3$
 2489   025136  021413                                  CMP     (R4),(R3)       ;CHECK IF CURRENT = NEW
 2490   025140  001005                                  BNE     4$              ;NO - SKIP
 2491   025142  004737  017454                          JSR     PC,ONSWAP       ;ELSE SWAP OLD AND NEW
 2492   025146  000421                                  BR      9$              ;SKIP TO SEEK
 2493   025150  005437  002544          3$:             NEG     TEMP2           ;CHANGE DIFF AUGMENT FOR OPPOSITE DIR
 2494   025154  011413                  4$:             MOV     (R4),(R3)       ;MOV CURRENT INTO NEW
 2495   025156  022714  000377                          CMP     #255.,(R4)      ;CHECK IF CURRENT AT 255
 2496   025162  001004                                  BNE     7$              ;NO - SKIP
 2497   025164  012737  177777  002544                  MOV     #-1,TEMP2       ;AT MAX CYL, ,AKE NEXT SEEK REV
 2498   025172  000405                                  BR      8$              ;SKIP
 2499   025174  005714                  7$:             TST     (R4)            ;TEST IF CURRENT AT 0
 2500   025176  001003                                  BNE     8$              ;NO - SKIP
 2501   025200  012737  000001  002544                  MOV     #1,TEMP2        ;AT CYL 0, MAKE NEXT SEEK FWRD
 2502   025206  063713  002544          8$:             ADD     TEMP2,(R3)      ;ADD DIFF TO NEW CYL (+1 OR -1)
 2503   025212  004737  016070          9$:             JSR     PC,XSEEK        ;DO SEEK
 2504   025216  025401                                  60$
 2505   025220  012701  000226                          MOV     #150.,R1        ;SET WAIT COUNT FOR 15 MS
 2506   025224  004737  020650                          JSR     PC,RDYWAIT      ;WAIT FOR READY
 2507   025230  025272                                  60$
 2508   025232  004737  021116                          JSR     PC,GETPOS       ;STORE POSITION
 2509   025236  025272                                  60$
 2510   025240  011501                                  MOV     (R5),R1         ;GET OLD POSITION
 2511   025242  161401                                  SUB     (R4),R1         ;SUBTRACT FROM NEW POINTER (FORWARD)
 2512   025244  005737  002532                          TST     DESSGN          ;CHECK IF SIGN FORWARD
 2513   025250  001402                                  BEQ     10$             ;YES-SKIP, ELSE SUB FOR SEEK REVERSE
 2514   025252  011401                                  MOV     (R4),R1         ;GET NEW CYLINDER
 2515   025254  161501                                  SUB     (R5),R1         ;SUBTRACT FROM OLD CYL
 2516   025256  022701  000001          10$:            CMP     #1,R1           ;CHECK IF RESULT IS DIFFERENCE OF 1
 2517   025262  001403                                  BEQ     12$             ;YES-SKIP
 2518   025264                                          ERRHRD  201,,ERR8       ;ELSE REPORT ERROR
  (3)   025264  104443                                  TRAP    T$ERCODE
```



```
  (5)   025266  000311                                  .WORD   201
  (5)   025270  013004                                  .WORD   ERR8
 2519   025272                                  12$:
 2520   025272  012737  000002  002440          60$:    MOV     #2,ERRSWI       ;INIT ERROR SWITCH
 2521   025300                                  ENDSUB
  (3)   025300                                  L10023:
  (3)   025300  104003                                  EMT     C$ESUB
 2522   025302                                          ESCAPE  TST             ;EXIT TEST IF ERROR
  (3)   025302  104010                                  EMT     C$ESCAPE
  (3)   025304  000030                                  .WORD   L10022-.
 2523   025306  005337  002540                          DEC     TEMP0           ;DEC PASS COUNT
 2524   025312  001410                                  BEQ     30$             ;EXIT IF DONE
 2525
 2526   025314  032737  000001  002540                  BIT     #BIT0,TEMP0     ;TEST IF PASS 1 OR 3
 2527   025322  001003                                  BNE     20$             ;YES-SKIP
 2528   025324  004737  017414                          JSR     PC,SWAPHD       ;GO SWAP TO HEAD 1 OR END TEST
 2529   025330  025334                                  30$                     ;ABORT RETURN
 2530   025332  000673                          20$:    BR      T187$           ;LOOP
 2531   025334                                  30$:
 2532   025334                                  T1865$:
 2533   025334                                  ENDTST
  (3)   025334                                  L10022:
  (3)   025334  104001                                  EMT     C$ETST
```

```
                                               .SBTTL  *TEST 3         **OUTER GUARD BAND DETECTION
        025336                                 BGNTST                  ;TEST 3
 (3)    025336                                                                                  T3::
        025336  012737  006244  002434                 MOV     #P2T03E,ERHEAD  ;SET ERROR HEADER
        025344  004737  015160                         JSR     PC,TSTINT       ;INITIALIZE TEST
        025350  004737  015176                         JSR     PC,GSTATR       ;CLEAR DRIVE
        025354  025546                                 T1965$
        025356  004737  017370                         JSR     PC,CHOSHD       ;GO CHOSE HEAD
        025362  005005                         T197$:  CLR     R5              ;CLEAR FOR POSITION TO 0
        025364  004737  016570                         JSR     PC,POSHDS       ;POSITION HEADS
        025370  025546                                 T1965$
        025372                                 BGNSUB
 (3)    025372                                                                                  T3.1:
 (3)    025372  104002                                 EMT     C$BSUB
        025374  012737  177777  002524                 MOV     #-1,NEWCYL      ;SET FOR GUARD BAND SEEK
        025402  004737  016070                         JSR     PC,XSEEK        ;DO SEEK
        025406  025522                                 60$
        025410  012701  000003                         MOV     #3.,R1          ;SET WAIT COUNT FOR 3MS
        025414  032762  000001  000000         8$:     BIT     #DRDYMSK,RLCS(R2) ;TEST IF DRIVE READY
        025422  001413                                 BEQ     9$              ;NO-SKIP
        025424  004737  015226                         JSR     PC,GSTAT        ;GET DRIVE STATUS
        025430  025522                                 60$
        025432  012703  007543                         MOV     #MDRDY,R3       ;SET NAME MESSAGE PTR
        025436  012704  010456                         MOV     #C10MS,R4       ;SET CONDITION MESSAGE PTR
        025442                                         ERRHRD  301.,,ERR4      ;REPORT READY ERROR
 (3)    025442  104443                                 TRAP    T$ERCODE
 (5)    025444  000455                                 .WORD   301
 (5)    025446  011736                                 .WORD   ERR4
        025450  000424                                 BR      60$             ;EXIT TEST
        025452  005301                         9$:     DEC     R1              ;DEC WAIT COUNT
        025454  001404                                 BEQ     12$             ;SKIP IF 0
        025456                                         WAITUS  #10.
 (3)    025456  012700  000012                         MOV     #10.,R0
 (3)    025462  104027                                 EMT     C$WTU
        025464  000753                                 BR      8$              ;LOOP
        025466  012701  000226                 12$:    MOV     #150.,R1        ;SET WAIT COUNT FOR 15 MS
        025472  004737  020650                         JSR     PC,RDYWAIT      ;WAIT FOR READY & REPORT IF NOT READY
        025476  025522                                 60$

        025500  004737  021116                         JSR     PC,GETPOS       ;GET POSITION
        025504  025522                                 60$
        025506  005737  002526                         TST     CURCYL          ;CHECK IF HEADS STILL AT 0
        025512  001403                                 BEQ     15$             ;YES-SKIP
        025514                                         ERRHRD  302.,,ERR8      ;ELSE REPORT CYLINDER ERROR
 (3)    025514  104443                                 TRAP    T$ERCODE
 (5)    025516  000456                                 .WORD   302
 (5)    025520  013004                                 .WORD   ERR8
        025522                                 15$:
        025522  012737  000002  002440         60$:    MOV     #2,ERRSWI       ;INIT ERROR SWITCH
        025530                                 ENDSUB
 (3)    025530                                 L10025:
 (3)    025530  104003                                 EMT     C$ESUB
        025532                                         ESCAPE  TST             ;EXIT TEST IF ERROR
 (3)    025532  104010                                 EMT     C$ESCAPE
```



```
 (3)    025534  000012                                 .WORD   L10024-.
        025536  004737  017414                         JSR     PC,SWAPHD       ;GO SWAP TO HEAD 1 OR END TEST
        025542  025546                                 17$                     ;ABORT RETURN
        025544  000706                                 BR      T197$           ;REDO TEST
        025546                                 17$:
        025546                                 T1965$:
        025546                                 ENDTST
 (3)    025546                                 L10024:
 (3)    025546  104001                                 EMT     C$ETST
```

```
2584
2585
2586                                    .SBTTL  *TEST 4          **INCREMENTAL FORWARD SEEK HEAD 0
2587  025550                            BGNTST                   ;TEST 4
 (3)  025550                                                                                T4::
2588  025550  012737  006270  002434            MOV     #P2T04E,ERRHEAD  ;SET ERROR HEADER
2589  025556  004737  015160                    JSR     PC,TSTINT        ;INITIALIZE TEST
2590  025562  004737  015176                    JSR     PC,GSTATR        ;CLEAR DRIVE
2591  025566  025756                            T2065$
2592  025570  004737  017370                    JSR     PC,CHOSHD        ;GO CHOSE HEAD
2593  025574  005737  002534                    TST     DESHD            ;TEST IF THIS IS HEAD 0
2594  025600  001402                            BEQ     2$               ;YES - SKIP
2595  025602                                    EXIT    #ST              ;ELSE EXIT TEST
 (3)  025602  104032                            EMT     C$EXIT
 (3)  025604  000152                            .WORD   L10026-.
2596  025606  013705  013374          2$:       MOV     LOLIMW,R5        ;CLEAR TO POSITION HEADS TO LOLIMIT
2597  025612  004737  016570                    JSR     PC,POSHDS        ;POSITION HEADS
2598  025616  025756                            T2065$
2599  025620                            BGNSUB
 (3)  025620                                                                                T4.1:
 (3)  025620  104002                            EMT     C$BSUB
2600  025622  004737  021116          T206$:    JSR     PC,GETPOS        ;GET POSITION
2601  025626  025746                            60$
2602  025630                                    INLOOP                   ;CHECK IF IN ERROR LOOP
 (3)  025630  104020                            EMT     C$INLP
2603  025632                                    BNCOMPLETE      5$       ;NO - SKIP
 (2)  025632  103007                            BCC     5$
2604  025634  023737  002526  002524            CMP     CURCYL,NEWCYL    ;CHECK IF POSITIONED AT DESIRED LOC
2605  025642  001003                            BNE     5$               ;NO - SKIP
2606  025644  004737  017454                    JSR     PC,ONSWAP        ;ELSE SWAP NEW AND OLD CYLINDERS
2607  025650  000405                            BR      7$               ;SKIP
2608  025652  013737  002526  002524  5$:       MOV     CURCYL,NEWCYL    ;PLACE CURRENT INTO NEW
2609  025660  005237  002524                    INC     NEWCYL           ;BUMP FOR ONE CYLINDER SEEK
2610  025664                            7$:
2611  025664  004737  016070                    JSR     PC,XSEEK         ;DO SEEK
2612  025670  025746                            60$
2613  025672  012701  000226                    MOV     #150.,R1         ;SET WAIT TIME 15 MS
2614  025676  004737  020650                    JSR     PC,RDYWAIT       ;WAIT FOR READY
2615  025702  025746                            60$
2616
2617  025704  004737  021244                    JSR     PC,VERPOS        ;GO VERIFY POSITON
2618  025710  025746                            60$
2619
2620  025712  032737  000002  013372            BIT     #ALLSEC,MISWIW   ;TEST IF CHECK ALL SECTORS
2621  025720  001406                            BEQ     11$              ;NO-SKIP
2622  025722  004737  021364                    JSR     PC,RDALHD        ;GO READ ALL HEADERS
2623  025726  025746                            60$
2624  025730  004737  020262                    JSR     PC,VERHDR        ;GO VERIFY HEADER
2625  025734  025746                            60$
2626  025736                            11$:
2627  025736  023737  013376  002524            CMP     HILIMW,NEWCYL    ;CHECK IF HILIMIT REACHED
2628  025744  103726                            BLO     T206$            ;NO-LOOP
2629  025746  012737  000002  002440  60$:       MOV     #2,ERRSWI        ;INIT ERROR SWITCH
2630  025754                            ENDSUB
 (3)  025754                            L10027:
 (3)  025754  104003                            EMT     C$ESUB
```



```
2631  025756                            T2065$:
2632  025756                            ENDTST
 (3)  025756                            L10026:
 (3)  025756  104001                            EMT     C$ETST
```

```
                                        .SBTTL  *TEST 5          **INCREMENTAL REVERSE SEEK HEAD 0
                                        BGNTST                   ;TEST 5
025760                                                                                  T5::
025760  012737  006312  002434                MOV     #P2T05E,ERHEAD  ;SET ERROR HEADER
025766  004737  015160                        JSR     PC,TSTINT       ;INITIALIZE TEST
025772  004737  015176                        JSR     PC,CSTATR       ;CLEAR DRIVE
025776  026166                                T2165$
026000  004737  017370                        JSR     PC,CHOSHD       ;GO CHOSE HEAD
026004  005737  002534                        TST     DESHD           ;TEST IF HEAD 0 SELECTED
026010  001402                                BEQ     2$              ;YES - SKIP
026012                                        EXIT    TST             ;ELSE EXIT TEST
026012  104032                                EMT     C$EXIT
026014  000152                                .WORD   L10030-.
026016  013705  013376          2$:           MOV     HILIMW,R5       ;SET TO POSITION HDS TO HILIMIT
026022  004737  016570                        JSR     PC,POSHDS       ;POSITION HEADS
026026  026166                                T2165$
026030                                  BGNSUB
026030                                                                                  T5.1:
026030  104002                                EMT     C$BSUB
026032  004737  021116          T216$:        JSR     PC,GETPOS       ;GET POSITION
026036  026156                                60$
026040                                        INLOOP                  ;CHECK IF IN ERROR LOOP
026040  104020                                EMT     C$INLP
026042                                        BNCOMPLETE      5$      ;NO - SKIP
026042  103007                                BCC     5$
026044  023737  002526  002524                CMP     CURCYL,NEWCYL   ;CHECK IF POSITIONED AT DES LOC
026052  001003                                BNE     5$              ;NO - SKIP
026054  004737  017454                        JSR     PC,ONSWAP       ;ELSE SWAP OLD AND NEW CYLINDERS
026060  000405                                BR      7$              ;SKIP
026062  013737  002526  002524  5$:           MOV     CURCYL,NEWCYL   ;PUT CURRENT INTO NEW
026070  005337  002524                        DEC     NEWCYL          ;DEC FOR ONE CYLINDER REVERSE SEEK
026074  004737  016070          7$:           JSR     PC,XSEEK        ;SEEK TO IT
026100  026156                                60$
026102  012701  000226                        MOV     #150.,R1        ;SET WAIT FOR 15 MS
026106  004737  020650                        JSR     PC,RDYWAIT      ;WAIT FOR READY
026112  026156                                60$

026114  004737  021244                        JSR     PC,VERPOS       ;VERIFY POSITION
026120  026156                                60$

026122  032737  000002  002426                BIT     #ALLSEC,OPFLAG  ;TEST IF USE ALL SECTORS
026130  001406                                BEQ     11$             ;NO-SKIP
026132  004737  021364                        JSR     PC,RDALHD       ;ELSE READ ALL THE HDRS
026136  026156                                60$
026140  004737  020262                        JSR     PC,VERHDR       ;VERIFY THE HEADERS
026144  026156                                60$
026146                          11$:
026146  023737  013374  002524                CMP     LOLIMW,NEWCYL   ;CHECK IF REACHED LOLIMIT
026154  103726                                BLO     T216$           ;NO - LOOP
026156  012737  000002  002440  60$:           MOV     #2,ERRSWI       ;INIT ERROR SWITCH
026164                                  ENDSUB
026164                                  L10031:
026164  104003                                EMT     C$ESUB
026166                          T2165$:
```



```
026166                                  ENDTST
026166                                  L10030:
026166  104001                                EMT     C$ETST
```

```
                                        .SBTTL  *TEST 6          **INCREMENTAL FORWARD SEEK HEAD 1
                                        BGNTST                   ;TEST 6
(3)  026170                                                                                  T6::
2687 026170  012737  006334  002434              MOV     #P2T06E,ERHEAD  ;SET ERROR HEADER
2688 026176  004737  015160                      JSR     PC,TSTINT       ;INITIALIZE TEST
2689 026202  004737  015176                      JSR     PC,GSTATR       ;CLEAR DRIVE
2690 026206  026412                              T2265$
2691 026210  005037  002534                      CLR     DESHD           ;SET HEAD TO 0
2692 026214  013705  013374                      MOV     LOLIMW,R5       ;CLEAR FOR POSITION HDS TO LOLIMIT
2693 026220  004737  016570                      JSR     PC,POSHDS       ;POSITION HDS
2694 026224  026412                              T2265$
2695 026226  012737  000001  002534              MOV     #1,DESHD        ;SET TO HEAD 1
2696 026234  032737  010000  013372              BIT     #HEADLM,MISWIW  ;TEST IF HEAD SPECIFIED
2697 026242  001405                              BEQ     2$              ;NO - SKIP
2698 026244  005737  013400                      TST     HEADW           ;TEST IF IT IS HEAD 0
2699 026250  001002                              BNE     2$              ;NO - SKIP
2700 026252                                      EXIT    #ST             ;ELSE EXIT TEST
(3)  026252  104032                              EMT     C$EXIT
(3)  026254  000136                              .WORD   L10032-.
2701 026256                              2$:
2702 026256                              BGNSUB                                              T6.1:
(3)  026256
(3)  026256  104002                              EMT     C$BSUB
2703 026260  004737  021116              T227$:  JSR     PC,GETPOS       ;GET CURRENT POSITION
2704 026264                                      INLOOP                  ;CHECK IF IN ERROR LOOP
(3)  026264  104020                              EMT     C$INLP
2705 026266                                      BNCOMPLETE 5$           ;NO - SKIP
(2)  026266  103007                              BCC     5$
2706 026270  023737  002526  002524              CMP     CURCYL,NEWCYL   ;CHECK IF AT DESIRED LOCATION
2707 026276  001003                              BNE     3$              ;NO - SKIP
2708 026300  004737  017454                      JSR     PC,OWSWAP       ;SWAP OLD AND NEW CYLINDER
2709 026304  000405                              BR      7$              ;SKIP
2710 026306  013737  002526  002524  5$:         MOV     CURCYL,NEWCYL   ;MOVE CURRENT INTO NEW
2711 026314  005237  002524                      INC     NEWCYL          ;BUMP NEWCYL FOR ONE CYL FWRD SEEK
2712 026320                              7$:
2713 026320  004737  016070                      JSR     PC,XSEEK        ;DO SEEK
2714 026324  026402                              60$
2715 026326  012701  000226                      MOV     #150.,R1        ;SET WAIT COUNT 15 MS
2716 026332  004737  020650                      JSR     PC,RDYWAIT      ;WAIT FOR READY
2717 026336  026402                              60$
2718 026340  004737  021244                      JSR     PC,VERPOS       ;VERIFY POSITION IS CORRECT
2719 026344  026402                              60$
2720
2721 026346  032737  000002  013372              BIT     #ALLSEC,MISWIW  ;CHECK IF USE ALL SECTORS
2722 026354  001406                              BEQ     9$              ;NO-SKIP
2723 026356  004737  021364                      JSR     PC,RDALHD       ;ELSE READ ALL HEADERS
2724 026362  026402                              60$
2725 026364  004737  020262                      JSR     PC,VERHDR       ;VERIFY HEADERS
2726 026370  026402                              60$
2727 026372                              9$:
2728 026372  023737  013376  002524              CMP     HILIMW,NEWCYL   ;CHECK IF DONE
2729 026400  101327                              BHI     T227$           ;NO - LOOP
2730 026402  012737  000002  00244Q  60$:         MOV     #2,ERRSWI       ;INIT ERROR SWITCH
2731 026410                              ENDSUB
```



```
(3)  026410                              L10033:
(3)  026410  104003                              EMT     C$ESUB
2732 026412                              T2265$:
2733 026412                              ENDTST
(3)  026412                              L10032:
(3)  026412  104001                              EMT     C$ETST
```

```
                                .SBTTL  *TEST 7          **INNER GUARD BAND DETECTION
                                BGMTST                   ;TEST 7
026414                                                                          T7::
026414  012737  006356  002434          MOV     #P2T07E,ERHEAD  ;SET ERROR HEADER
026422  004737  015160                  JSR     PC,TSTINT       ;INITIALIZE TEST
026426  004737  015176                  JSR     PC,GSTATR       ;CLEAR DRIVE
026432  026612                          T2365$
026434  004737  017370                  JSR     PC,CHOSHD       ;GO CHOSE HEAD
026440  012705  000377          T233$:  MOV     #255.,R5        ;SET FOR POSITION TO 255.
026444  004737  016570                  JSR     PC,POSHDS       ;POSITION HEADS
026450  026612                          T2365$
026452                          BGMSUB
                                                                                T7.1:
026452  104002                          EMT     C$BSUB
026454  012737  000400  002524          MOV     #256.,NEWCYL    ;SET FOR INNER GUARD BAND SEEK
026462  004737  016070                  JSR     PC,XSEEK        ;DO IT
026466  026566                          60$
026470  012701  000003                  MOV     #3.,R1          ;SET WAIT COUNT 3 MS
                                7$:
026474  032762  000001  000000          BIT     #DRDYMSK,RLCS(R2) ;CHECK IF READY
026502  001413                          BEQ     9$              ;NO-SKIP
026504  004737  015226                  JSR     PC,GSTAT        ;GET DRIVE STATUS
026510  026566                          60$
026512  012703  007543                  MOV     #MDRDY,R3       ;SET NAME MESSAGE PTR
026516  012704  010456                  MOV     #C1OMS,R4       ;SET CONDITION MESSAGE PTR
026522                                  ERRMRD  701.,ERR4       ;REPORT READY ERROR
026522  104443                          TRAP    T$ERCODE
026524  001275                          .WORD   701
026526  011736                          .WORD   ERR4
026530  000416                          BR      60$             ;EXIT TEST
026532  005301                  9$:     DEC     R1              ;DEC WAIT COUNT
026534  001404                          BEQ     11$             ;SKIP IF 0
026536                                  WAITUS  #10.            ;WAIT 100 US
026536  012700  000012                  MOV     #10.,R0
026542  104027                          EMT     C$WTU
026544  000753                          BR      7$              ;LOOP
026546  012701  000226          11$:    MOV     #150.,R1                ;SET WAIT COUNT 15 MS
026552  004737  020650                  JSR     PC,RDYWAIT      ;GO WAIT FOR READY
026556  026566                          60$

026560  004737  021244                  JSR     PC,VERPOS       ;GO VERIFY POSITION IS 255
026564  026566                          60$
026566  012737  000002  002440  60$:    MOV     #2,ERRSWI       ;INIT ERROR SWITCH
026574                          ENDSUB
026574                          L10035:
026574  104003                          EMT     C$ESUB
026576                                  ESCAPE  TST             ;EXIT TEST IF ERROR
026576  104010                          EMT     C$ESCAPE
026600  000012                          .WORD   L10034-.
026602  004737  017414                  JSR     PC,SWAPHD       ;GO SWAP TO HEAD 1 OR END TEST
026606  026612                          15$                     ;ABORT RETURN
026610  000713                          BR      T233$           ;REPEAT THE TESTS
026612                          15$:
026612                          T2365$:
```



```
026612                          ENDTST
026612                          L10034:
026612  104001                          EMT     C$ETST
```

```
2781
2782
2783                                    .SBTTL  *TEST 8          **INCREMENTAL REVERSE SEEK HEAD 1
2784   026614                           BGNTST                   ;TEST 8
  (3)  026614                                                                                T8::
2785   026614  012737  006402  002434            MOV     #P2T08E,ERHEAD  ;SET ERROR HEADER
2786   026622  004737  015160                    JSR     PC,$STINT       ;INITIALIZE TEST
2787   026626  004737  015176                    JSR     PC,GSTATR       ;GET STATUS & CLEAR
2788   026632  027040                           T2465$
2789   026634  005037  002534                    CLR     DESHD           ;SET TO HEAD 0
2790   026640  013705  013376                    MOV     HILIMW,R5       ;SET TO POSITION HDS AT HILIMIT
2791   026644  004737  016570                    JSR     PC,POSHDS       ;POSITION HDS
2792   026650  027040                           T2465$
2793   026652  012737  000001  002534            MOV     #1,DESHD        ;SET TO SELECT HD 1
2794   026660  032737  010000  013372            BIT     #HEADLM,MISWIW  ;TEST IF HEAD SPECIFIED
2795   026666  001405                           BEQ     2$              ;NO - SKIP
2796   026670  005737  013400                    TST     HEADW           ;TEST IF HEAD SPECIFIED IS 0
2797   026674  001002                           BNE     2$              ;NO - SKIP
2798   026676                                   EXIT    TST             ;ESLE EXIT TEST
  (3)  026676  104032                           EMT     C$EXIT
  (3)  026700  000140                           .WORD   L10036-.
2799   026702                           2$:
2800   026702                           BGNSUB
  (3)  026702                                                                                T8.1:
  (3)  026702  104002                           EMT     C$BSUB
2801   026704  004737  021116          T247$:   JSR     PC,GETPOS       ;GET CURRENT POSITION
2802   026710  027030                           60$
2803   026712                                   INLOOP                  ;CHECK IF IN ERROR LOOP
  (3)  026712  104020                           EMT     C$INLP
2804   026714                                   BNCOMPLETE      5$      ;NO - SKIP
  (2)  026714  103007                           BCC     5$
2805   026716  023737  002526  002524            CMP     CURCYL,NEWCYL   ;CHECK IF POSITIONED AT DESIRED LOC
2806   026724  001003                           BNE     5$              ;NO - SKIP
2807   026726  004737  017454                    JSR     PC,ONSWAP       ;ELSE SWAP OLD AND NEW CYLINDER
2808   026732  000405                           BR      7$              ;SKIP
2809   026734  013737  002526  002524  5$:      MOV     CURCYL,NEWCYL   ;MOV CUR TO NEW
2810   026742  005337  002524                    DEC     NEWCYL          ;DEC NEWCYL FOR 1 CYL REV SEEK
2811   026746  004737  016070          7$:      JSR     PC,XSEEK        ;DO SEEK
2812   026752  027030                           60$
2813   026754  012701  000226                    MOV     #150.,R1                ;SET WAIT FOR 15 MS
2814   026760  004737  020650                    JSR     PC,RDYWAIT      ;WAIT FOR READY
2815   026764  027030                           60$
2816   026766  004737  021244                    JSR     PC,VERPOS       ;VERIFY POSITION
2817   026772  027030                           60$
2818   026774  032737  000002  013372            BIT     #ALLSEC,MISWIW  ;TEST IF ALL SECTORS
2819   027002  001406                           BEQ     9$              ;NO-EXIT
2820   027004  004737  021364                    JSR     PC,RDALHD       ;READ ALL HEADERS
2821   027010  027030                           60$
2822   027012  004737  020262                    JSR     PC,VERHDR       ;VERIFY HEADER
2823   027016  027030                           60$
2824   027020                           9$:
2825   027020  023737  013374  002524            CMP     LOLIMW,NEWCYL   ;CHECK IF AT LOLIMIT
2826   027026  103726                           BLO     T247$           ;NO - LOOP
2827   027030  012737  000002  002440  60$:      MOV     #2,ERRSWI       ;INIT ERROR SWITCH
2828   027036                           ENDSUB
  (3)  027036                           L10037:
```



```
  (3)  027036  104003                           EMT     C$ESUB
2829   027040                           T2465$:
2830   027040                           ENDTST
  (3)  027040                           L10036:
  (3)  027040  104001                           EMT     C$ETST
```

```
 2832                                          .SBTTL  *TEST 9        **SEEK TESTS
 2833  027042                                  BGNTST                 ;TEST 9
  (3)  027042                                                                                  T9::
 2834  027042  012737  006424  002434          MOV     #P2T09E,ERHEAD  ;SET ERROR HEADER
 2835  027050  004737  015160                  JSR     PC,TSTINT       ;INITIALIZE TEST
 2836  027054  004737  015176                  JSR     PC,GSTATR       ;CLEAR DRIVE
 2837  027060  027334                          T2565$
 2838  027062  004737  017370                  JSR     PC,CHOSHD       ;GO CHOSE HEAD
 2839  027066  013705  013374                  MOV     LOLIMW,R5       ;SET TO POSTION HEADS TO LOLIMIT
 2840  027072  004737  016570                  JSR     PC,POSHDS       ;POSITION HDS TO LOWLIMIT
 2841  027076  027334                          T2565$
 2842  027100  004737  021116         T256$:   JSR     PC,GETPOS       ;GET CURRENT POSITION
 2843  027104  027334                          T2565$
 2844  027106  013737  002526  002524          MOV     CURCYL,NEWCYL   ;PUT CURRENT INTO NEW
 2845  027114  012704  002304                  MOV     #T25TBL,R4      ;SET POINTER TO TABLE OF SEEK DIFF
 2846  027120  012405                 T258$:   MOV     (R4)+,R5        ;PUT FIRST IN R5
 2847  027122  013701  013376                  MOV     HILIMW,R1       ;GET HILIMIT
 2848  027126  163701  013374                  SUB     LOLIMW,R1       ;SUBTRACT LOLIMIT
 2849  027132  021401                          CMP     (R4),R1         ;CHECK IF NEW DIFFERENCE IS IN BOUNDS
 2850  027134  101073                          BHI     T2517$$         ;NO - SKIP TEST
 2851  027136  060537  002524         T257$:   ADD     R5,NEWCYL       ;ADD TO PRESENT POSITION
 2852  027142  023737  002524  013374          CMP     NEWCYL,LOLIMW   ;CHECK IF AT OR PAST LOLIMIT
 2853  027150  002004                          BGE     9$              ;NO - SKIP
 2854  027152  013737  013374  002524          MOV     LOLIMW,NEWCYL   ;ELSE SET TO LOLIMIT
 2855  027160  000407                          BR      11$
 2856  027162  023737  002524  013376 9$:      CMP     NEWCYL,HILIMW   ;CHECK IF AT HILIMIT OR GREATER
 2857  027170  003403                          BLE     11$             ;NO - SKIP
 2858  027172  013737  013376  002524          MOV     HILIMW,NEWCYL   ;ELSE SET FOR HILIMIT
 2859  027200                         11$:
 2860  027200                         BGNSUB
  (3)  027200                                                                                  T9.1:
  (3)  027200  104002                          EMT     C$BSUB
 2861  027202                                  INLOOP                  ;CHECK IF IN ERROR LOOP
  (3)  027202  104020                          EMT     C$INLP
 2862  027204                                  BNCOMPLETE      13$     ;NO - SKIP
  (2)  027204  103011                          BCC     13$
 2863  027206  004737  021116                  JSR     PC,GETPOS       ;GET CURRENT POSITION
 2864  027212  027256                          60$
 2865  027214  023737  002526  002524          CMP     CURCYL,NEWCYL   ;CHECK IF HEADS AT DESIRED POSITION
 2866  027222  001002                          BNE     13$             ;NO - SKIP
 2867  027224  004737  017454                  JSR     PC,ONSWAP       ;ELSE SWAP CURRENT AND NEW CYLINDERS
 2868  027230  004737  016070         13$:     JSR     PC,XSEEK        ;DO SEEK
 2869  027234  027256                          60$
 2870  027236  012701  005670                  MOV     #3000.,R1       ;SET WAIT COUNT
 2871  027242  004737  020650                  JSR     PC,RDYWAIT      ;WAIT FOR READY
 2872  027246  027256                          60$
 2873  027250  004737  021244                  JSR     PC,VERPOS       ;VERIFY POSITION
 2874  027254  027256                          60$
 2875  027256  012737  000002  002440 60$:     MOV     #2,ERRSWI       ;INITIALIZE ERROR SWITCH
 2876  027264                         ENDSUB
  (3)  027264                         L10041:
  (3)  027264  104003                          EMT     C$ESUB
 2877  027266                                  ESCAPE  TST             ;EXIT TEST IF ERROR
  (3)  027266  104010                          EMT     C$ESCAPE
  (3)  027270  000044                          .WORD   L10040-.
 2878  027272  023737  013376  002524          CMP     HILIMW,NEWCYL   ;CHECK IF SEEK WAS TO HILIMIT
```



```
 2879  027300  001002                          BNE     15$             ;NO - SKIP
 2880  027302  005405                          NEG     R5              ;ELSE SET R5 TO REPEAT DIFF IN REVERSE
 2881  027304  000714                          BR      T257$
 2882  027306  023737  013374  002524 15$:     CMP     LOLIMW,NEWCYL   ;TEST IF LAST SEEK WAS TO LOLIMIT
 2883  027314  001310                          BNE     T257$           ;NO - GO DO SEEK TEST
 2884  027316  021427  000377                  CMP     (R4),#255.      ;CHECK IF ALL TABLE DIFF USED
 2885  027322  001276                          BNE     T258$           ;NO - SKIP
 2886  027324  004737  017414         T2517$:  JSR     PC,SWAPHD       ;GO SWAP TO HEAD 1 OR END TEST
 2887  027330  027334                          T2565$                  ;ABORT RETURN
 2888  027332  000662                          BR      T256$           ;REPEAT TEST HEAD 1
 2889  027334                         T2565$:
 2890  027334                         ENDTST
  (3)  027334                         L10040:
  (3)  027334  104001                          EMT     C$ETST
```

```
                                .SBTTL  *TEST 10         **FORWARD OSCILLATING SEEK
                                BGNTST                   ;TEST 10
  (3)  027336                                                                    T10::
                                MOV     #P2T10E,ERRHEAD  ;SET ERROR HEADER
                                JSR     PC,INITST        ;INITIALIZE TEST
                                JSR     PC,CSTATR        ;CLEAR DRIVE
                                T2665$
                                JSR     PS,CHOSHD        ;GO CHOSE HEAD
                                MOV     #1,R5            ;LOAD R5 FOR FIRST SEEK
                                BIT     #HICYL,MISWIW    ;TEST IF AT CYLINDER SPECED
                                BEQ     2$               ;NO - SKIP
                                MOV     #YLIMW,R5        ;ELSE SET UPPER LIMIT
                        2$:     CLR     NEWCYL           ;SET TO SEEK TO CYL 0
                                BIT     #LOCYL,MISWIW    ;CHECK IF LO CYL SPEC'D
                                BEQ     5$               ;NO - SKIP
                                MOV     LOLIMW,NEWCYL    ;ELSE SET LOWER LIMIT
                        5$:     JSR     PC,XSEEK         ;DO SEEK
                                T2665$
                                MOV     #3000.,R1        ;SET WAIT COUNT FOR 120 MS
                                JSR     PC,RDYWAIT       ;WAIT FOR READY
                                T2665$
                        T267$:  JSR     PC,GETPOS        ;GET HEAD POSITION
                                T2665$
                                MOV     R5,NEWCYL        ;LOAD NEW CYLINDER INTO NEWCYL
                                BGWSUB
  (3)  027456                                                                    T10.1:
                                EMT     C$BSUB
                        INLOOP                           ;CHECK IF IN ERROR LOOP
                                EMT     C$INLP
                        BWCOMPLETE      18$              ;NO - SKIP
                                BCC     18$
                                JSR     PC,GETPOS        ;GET POSITION
                                60$
                                CMP     CURCYL,NEWCYL    ;CHECK IF HEADS AT DESIRED LOC
                                BNE     18$              ;NO - SKIP
                                JSR     PC,ONSWAP        ;SWAP OLD AND NEW
                                JSR     PC,XSEEK         ;DO SEEK
                        18$:    60$
                                MOV     #3000.,R1        ;SET WAIT COUNT 120 MS
                                JSR     PC,RDYWAIT       ;WAIT FOR READY
                                60$
                                JSR     PC,VERPOS        ;VERIFY HEAD POSITION
                                60$
                                TST     DESSGW           ;TEST IF JUST SEEK REV
                                BEQ     60$              ;YES - SKIP
                                CLR     NEWCYL           ;ELSE SET TO SEEK TO 0
                                BIT     #LOCYL,MISWIW    ;CHECK IF LO LIMIT SPEC'D
                                BEQ     18$              ;NO - SKIP
                                MOV     LOLIMW,NEWCYL    ;ELSE SET LOW LIMIT FOR SEEK
                                BR      18$
                        60$:    MOV     #2,ERRSWI        ;INIT ERROR SWITCH
                        ENDSUB
                        L10043:
  (3)  027574                                                                    T10.2:
                                EMT     C$ESUB
                        ESCAPE  TST                      ;EXIT TEST IF ERROR
  (3)  027576                   EMT     C$ESCAPE
```



```
  (3)  027600                   .WORD   L10042-.
                                BIT     #HICYL,MISWIW    ;TEST IF UPPER LIMIT SPEC'D
                                BNE     20$              ;YES - SKIP
                                INC     R5               ;BUMP R5
                                CMP     R5,#256.         ;ALL CYLINDERS DONE
                                BNE     T267$            ;NO - GO DO NEXT CYLINDER
                        20$:    JSR     PC,SWAPHD        ;GO SWAP TO HEAD 1 OR END TEST
                                T2665$                   ;ABORT RETURN
                                BR      T266$            ;GO DO TESTS
                        T2665$:
                        ENDTST
                        L10042:
  (3)  027632                                                                    T10.3:
                                EMT     C$ETST
```

```
                                        .SBTTL  *TEST 11        **REVERSE OSCILLATING SEEK
                                        BGNTST                  ;TEST 11
(3)  027634                                                                                 T11::
2953 027634  012737  006446  002434              MOV     #P2T11E,ERHEAD  ;SET ERROR HEADER
2954 027642  004737  015160                      JSR     PC,TSTINT       ;INITIALIZE TEST
2955 027646  004737  015176                      JSR     PC,GSTATR       ;CLEAR DRIVE
2956 027652  030130                              T2765$
2957 027654  004737  017370                      JSR     PC,CHOSHD       ;GO CHOSE HEAD
2958 027660  012737  000377  002524      T275$:  MOV     #255.,NEWCYL    ;SEEK OUT TO 255.
2959 027666  032737  020000  013372              BIT     #HICYL,MISWIW   ;TEST IF UPPER LIMIT SPEC'D
2960 027674  001403                              BEQ     2$              ;NO - SKIP
2961 027676  013737  013376  002524              MOV     HILIMW,NEWCYL   ;ELSE SET UPPER LIMIT
2962 027704  012705  000376                  2$: MOV     #254.,R5        ;SET R5 FOR FIRST SEEKS
2963 027710  032737  040000  013372              BIT     #LOCYL,MISWIW   ;CHECK IF LO LIMIT SPEC'D
2964 027716  001402                              BEQ     5$              ;NO - SKIP
2965 027720  013705  013374                      MOV     LOLIMW,R5       ;SET LOWER LIMIT
2966 027724  004737  016070              5$:     JSR     PC,XSEEK        ;DO SEEK
2967 027730  030130                              T2765$
2968 027732  012704  005670                      MOV     #3000.,R1       ;SET WAIT FOR 120 MS
2969 027736  004737  020650                      JSR     PC,RDYWAIT      ;WAIT FOR DRIVE READY
2970 027742  030130                              T2765$
2971 027744  004737  021116              T276$:  JSR     PC,GETPOS       ;GET POSITION
2972 027750  030130                              T2765$
2973 027752  010537  002524                      MOV     R5,NEWCYL       ;SET FOR NEXT SEEK
2974 027756                                     BGNSUB
(3)  027756                                                                                 T11.1:
(3)  027756  104002                              EMT     C$BSUB
2975 027760                                     INLOOP                  ;CHECK IF IN ERROR LOOP
(3)  027760  104020                              EMT     C$INLP
2976 027762                                     BNCOMPLETE      18$     ;NO - SKIP
(2)  027762  103011                              BCC     18$
2977 027764  004737  021116                      JSR     PC,GETPOS       ;ELSE GET POSITION
2978 027770  030070                              60$
2979 027772  023737  002526  002524              CMP     CURCYL,NEWCYL   ;CHECK IF AT DESIRED CYL
2980 030000  001002                              BNE     18$             ;NO - SKIP
2981 030002  004737  017454                      JSR     PC,ONSWAP       ;ELSE SWAP OLD AND NEW CYL
2982 030006  004737  016070              18$:    JSR     PC,XSEEK        ;DO SEEK
2983 030012  030070                              60$
2984 030014  012701  005670                      MOV     #3000.,R1       ;SET WAIT FOR 120 MS
2985 030020  004737  020650                      JSR     PC,RDYWAIT      ;WAIT FOR READY
2986 030024  030070                              60$
2987 030026  004737  021244                      JSR     PC,VERPOS       ;VERIFY POSITION
2988 030032  030070                              60$
2989 030034  005737  002532                      TST     DESSGN          ;CHECK IF JUST SEEK FWD
2990 030040  001013                              BNE     60$             ;YES - SKIP
2991 030042  012737  000377  002524              MOV     #255.,NEWCYL    ;ELSE SEEK TO TO 255
2992 030050  032737  020000  013372              BIT     #HICYL,MISWIW   ;TEST IF HILIMIT SPEC'D
2993 030056  001753                              BEQ     18$             ;NO - SKIP
2994 030060  013737  013376  002524              MOV     HILIMW,NEWCYL   ;SET TO UPPER LIMIT
2995 030066  000747                              BR      18$
2996 030070  012737  000002  002440      60$:    MOV     #2,ERRSWI       ;INIT ERROR SWITCH
2997 030076                                     ENDSUB
(3)  030076                                     L10045:
(3)  030076  104003                              EMT     C$ESUB
2998 030100                                     ESCAPE  TST             ;EXIT TEST IF ERROR
(3)  030100  104010                              EMT     C$ESCAPE
```



```
(3)  030102  000026                              .WORD   L10044-.
2999 030104  032737  040000  013372              BIT     #LOCYL,MISWIW   ;TEST IF LOLIMIT SPEC'D
3000 030112  001002                              BNE     20$             ;YES - SKIP
3001 030114  005305                              DEC     R5              ;DEC CYLINDER COUNT
3002 030116  100312                              BPL     T276$           ;IF STILL POSITIVE, DO SEEKS AGAIN
3003 030120  004737  017414              20$:    JSR     PC,SWAPHD       ;GO SWAP TO HEAD 1 OR END TEST
3004 030124  030130                              T2765$                  ;ABORT RETURN
3005 030126  000654                              BR      T275$           ;LOOP AGAIN
3006 030130                                     T2765$:
3007 030130                                     ENDTST
(3)  030130                                     L10044:
(3)  030130  104001                              EMT     C$ETST
```

```
3009                                    .SBTTL  *TEST 12         **SEEK TIMING
3010  030132                            BGNTST                   ;TEST 12
  (3) 030132                                                                    T12::
3011  030132  012737  006463  002434            MOV     #P2T12E,ERHEAD  ;SET ERROR HEADER
3012  030140  005737  003062                    TST     PASNUM          ;TEST IF PASS 0
3013  030144  001003                            BNE     2$              ;NO - SKIP
3014  030146  005737  013372                    TST     MISWIW          ;TEST IF MANUAL TESTS WERE RUN
3015  030152  100402                            BMI     1$              ;YES - SKIP
3016  030154  000137  031676          2$:       JMP     65$             ;ELSE EXIT TEST
3017  030160  004737  015160          1$:       JSR     PC,TSTINT       ;INITIALIZE TEST
3018  030164  004737  015176                    JSR     PC,GSTATR       ;CLEAR DRIVE
3019  030170  031676                            65$
3020  030172  012700  002562                    MOV     #OFIN,R0        ;GET ADDRESS OF 1ST TIME VALUE
3021  030176  012701  000030                    MOV     #24.,R1         ;SET COUNT FOR CLEAR
3022  030202  005020                    4$:     CLR     (R0)+           ;CLEAR TIMER STORAGE
3023  030204  005301                            DEC     R1
3024  030206  001375                            BNE     4$
3025  030210  005037  002654                    CLR     PASCNT          ;CLEAR PASS COUNTER
3026  030214  005037  002524                    CLR     NEWCYL          ;POSITION HEADS AT 0
3027  030220  004737  016070                    JSR     PC,XSEEK        ;DO SEEK
3028  030224  031676                            65$
3029  030226  012701  005670                    MOV     #3000.,R1       ;SET WAIT FOR 300 MS
3030  030232  004737  020650                    JSR     PC,RDYWAIT      ;WAIT FOR READY
3031  030236  031676                            65$
3032  030240  004737  021244                    JSR     PC,VERPOS       ;VERIFY POSITION
3033  030244  031676                            65$
3034  030246  004737  017370                    JSR     PC,CHOSHD       ;GO CHOSE HEAD
3035  030252  012700  002572                    MOV     #OFOUT,R0       ;SET PTRS FOR 1 CYL FWD OUTER TIMER
3036  030256  012701  002574                    MOV     #OFOUTU,R1
3037  030262  012703  002606                    MOV     #OROUT,R3
3038  030266  012704  002610                    MOV     #OROUTU,R4
3039  030272  012737  000001  002524            MOV     #1,NEWCYL       ;SET NEWCYL TO CYL 1
3040  030300  012737  000200  002656  8$:       MOV     #128.,COUNT     ;SET COUNTER FOR SEEK LOOP
3041  030306  012737  000110  002560            MOV     #RDHEAD,TEMP8   ;BUILD READ HEADER COMMAND
3042  030314  053737  002454  002560            BIS     RLDRV,TEMP8
3043  030322  042737  002000  002560            BIC     #BIT10,TEMP8
3044  030330  004737  016060          9$:       JSR     PC,XSEEKT       ;DO SEEK BUILD BUT DO NOT START
3045  030334  031676                            65$
3046  030336  013762  002462  000004            MOV     L.DA,RLDA(R2)   ;LOAD RL REGISTERS
3047  030344  013762  002456  000000            MOV     L.CS,RLCS(R2)
3048  030352  010046                            MOV     R0,-(SP)        ;STORE R0
3049  030354                            WAITUS  #10.            ;WAIT FOR INTERRUPT
  (3) 030354  012700  000012                    MOV     #10.,R0
  (3) 030360  104027                            EMT     C$WTU
3050  030362  005737  002430                    TST     DONE            ;TEST IF INTERRUPT
3051  030366  001010                            BNE     17$             ;YES - SKIP
3052  030370  004737  015026                    JSR     PC,WAITIN       ;WAIT FOR INTERRUPT
3053  030374  012603                            MOV     (SP)+,R3        ;GET MESSAGE POINTER
3054  030376                            ERRHRD  1201.,,ERR1
  (3) 030376  104443                            TRAP    T$ERCODE
  (5) 030400  002261                            .WORD   1201
  (5) 030402  011554                            .WORD   ERR1
3055  030404  000137  031676                    JMP     65$
3056  030410  005737  002466          17$:      TST     T.CS            ;CHECK IF ANY ERRORS
3057  030414  100005                            BPL     14$             ;NO - SKIP
3058  030416                            ERRHRD  1202.,,ERR6
```



```
  (3) 030416  104443                            TRAP    T$ERCODE
  (5) 030420  002262                            .WORD   1202
  (5) 030422  012056                            .WORD   ERR6
3059  030424  000137  031676                    JMP     65$
3060  030430  005037  002430          14$:      CLR     DONE            ;CLEAR INTERRUPT FLAG
3061  030434  013762  002560  000000            MOV     TEMP8,RLCS(R2)  ;LOAD RL REGISTER
3062  030442                            WAITUS  #2000.          ;WAIT FOR INTERRUPT
  (3) 030442  012700  003720                    MOV     #2000.,R0
  (3) 030446  104027                            EMT     C$WTU
3063  030450                            GETTIM  R5              ;GET TIME USED
  (3) 030450  104052                            EMT     C$GTIM
  (3) 030452  012005                            MOV     R0,R5
3064  030454  012600                            MOV     (SP)+,R0        ;RESTORE R0
3065  030456  013737  002560  002456            MOV     TEMP8,L.CS      ;SET IF ERROR TO REPORT
3066  030464  004737  021244                    JSR     PC,VERPOS       ;VERIFY POSITION
3067  030470  031676                            65$
3068  030472  005737  002532                    TST     DESSGN          ;CHECK WHICH SEEK DIRECTION
3069  030476  001403                            BEQ     15$             ;REVERSE - SKIP
3070  030500  060510                            ADD     R5,(R0)         ;ADD TO FORWARD TOTAL
3071  030502  005511                            ADC     (R1)            ;ADD IN OVERFLOW
3072  030504  000402                            BR      16$             ;SKIP
3073  030506  060513                    15$:    ADD     R5,(R3)         ;ADD TO REVERSE TOTAL
3074  030510  005514                            ADC     (R4)            ;ADD IN OVERFLOW
3075  030512  005337  002656          16$:      DEC     COUNT           ;DEC SEEK COUNT
3076  030516  001403                            BEQ     18$             ;SKIP IF 0
3077  030520  004737  017454                    JSR     PC,ONSWAP       ;ELSE SWAP OLD AND NEW CYL
3078  030524  000701                            BR      9$              ;REDO SEEK LOOP
3079  030526  162710  000470          18$:      SUB     #312.,(R0)      ;SUB CONSTANT FOR READ HEADER TIME
3080  030532  162713  000470                    SUB     #312.,(R3)
3081  030536  012705  000006                    MOV     #6,R5           ;SET SHIFT COUNT TO DIVIDE BY 64
3082  030542  000241                    10$:    CLC                     ;DIVIDE BOTH TOTALS BY 64
3083  030544  006011                            ROR     (R1)
3084  030546  006010                            ROR     (R0)
3085  030550  000241                            CLC
3086  030552  006014                            ROR     (R4)
3087  030554  006013                            ROR     (R3)
3088  030556  005305                            DEC     R5
3089  030560  001370                            BNE     10$
3090  030562  005237  002654                    INC     PASCNT          ;BUMP PASS COUNT
3091  030566  022737  000001  002654            CMP     #1,PASCNT       ;TEST IF PASS 1
3092  030574  001033                            BNE     24$             ;NO - SKIP
3093  030576  012737  000177  002524            MOV     #127.,NEWCYL    ;ELSE SET TO POSITION HDS TO 127
3094  030604  004737  016070                    JSR     PC,XSEEK        ;DO SEEK
3095  030610  031676                            65$
3096  030612  012701  005670                    MOV     #3000.,R1       ;SET WAIT COUNT FOR 300 MS
3097  030616  004737  020650                    JSR     PC,RDYWAIT      ;WAIT FOR READY
3098  030622  031676                            65$
3099  030624  004737  021244                    JSR     PC,VERPOS       ;VERIFY POSITION
3100  030630  031676                            65$
3101  030632  012700  002566                    MOV     #OFMID,R0       ;SET PTRS FOR TIMING 1 CYL SK AT 127
3102  030636  012701  002570                    MOV     #OFMIDU,R1
3103  030642  012703  002602                    MOV     #ORMID,R3
3104  030646  012704  002604                    MOV     #ORMIDU,R4
3105  030652  012737  000200  002524            MOV     #128.,NEWCYL    ;SET NEWCYL TO 128
3106  030660  000137  030300                    JMP     8$              ;DO SEEK LOOP
3107  030664  022737  000002  002654  24$:      CMP     #2,PASCNT       ;TEST IF PASS 2
```

```
3108  030672  001033                        BNE     28$             ;NO - SKIP
3109  030674  012737  000376  002524        MOV     #254.,NEWCYL    ;SET UP TO TIME 1 CYL SEEK AT INNER
3110  030702  004737  016070                JSR     PC,XSEEK        ; LIMIT
3111  030706  031676                        65$
3112  030710  012701  005670                MOV     #3000.,R1       ;SET WAIT COUNT FOR 300 MS
3113  030714  004737  020650                JSR     PC,RDYWAIT      ;WAIT FOR READY
3114  030720  031676                        65$
3115  030722  004737  021244                JSR     PC,VERPOS       ;VERIFY POSITION
3116  030726  031676                        65$
3117  030730  012700  002562                MOV     #OFIN,R0        ;SET POINTERS
3118  030734  012701  002564                MOV     #OFINO,R1
3119  030740  012703  002576                MOV     #ORIN,R3
3120  030744  012704  002600                MOV     #ORINO,R4
3121  030750  012737  000377  002524        MOV     #255.,NEWCYL    ;LOAD NEW CYLINDER
3122  030756  000137  030300                JMP     8$              ;DO SEEK LOOP
3123  030762  027737  000003  002654  28$:  CMP     #3,PASCNT       ;TEST IF PASS 3
3124  030770  001031                        BNE     32$             ;NO - SKIP
3125  030772  005037  002524                CLR     NEWCYL          ;ELSE SET UP TO TIME 128 CYL SEEK
3126  030776  004737  016070                JSR     PC,XSEEK        ;  AT OUTER LIMIT
3127  031002  031676                        65$
3128  031004  012701  005670                MOV     #3000.,R1       ;SET WAIT COUNT FOR 300 MS
3129  031010  004737  020650                JSR     PC,RDYWAIT      ;WAIT FOR DRIVE READY
3130  031014  031676                        65$
3131  031016  004737  021244                JSR     PC,VERPOS       ;VERIFY POSITION
3132  031022  031676                        65$
3133  031024  012700  002616                MOV     #HFOUT,R0       ;SET POINTERS
3134  031030  012701  002620                MOV     #HFOUTO,R1
3135  031034  012703  002626                MOV     #HROUT,R3
3136  031040  012704  002620                MOV     #HFOUTO,R4
3137  031044  012737  000177  002524        MOV     #127.,NEWCYL    ;LOAD NEWCYL FOR 128 CYL SEEK
3138  031052  000472                        BR      39$
3139  031054  027737  000004  002654  32$:  CMP     #4,PASCNT       ;TEST IF PASS 4
3140  031062  001032                        BNE     36$             ;NO - SKIP
3141  031064  012737  000200  002524        MOV     #128.,NEWCYL    ;ELSE SET UP TO TIME 128 CYL SEEK
3142  031072  004737  016070                JSR     PC,XSEEK        ; AT INNER LIMIT
3143  031076  031676                        65$
3144  031100  012701  005670                MOV     #3000.,R1       ;SET WAIT COUNT FOR 300 MS
3145  031104  004737  020650                JSR     PC,RDYWAIT      ;WAIT FOR READY
3146  031110  031676                        65$
3147  031112  004737  021244                JSR     PC,VERPOS       ;VERIFY POSITION
3148  031116  031676                        65$
3149  031120  012700  002612                MOV     #HFIN,R0        ;SET POINTERS
3150  031124  012701  002614                MOV     #HFINO,R1
3151  031130  012703  002622                MOV     #HRIN,R3
3152  031134  012704  002624                MOV     #HRINO,R4
3153  031140  012737  000377  002524        MOV     #255.,NEWCYL    ;SET NEWCYL TO 255 FOR 128 CYL SEEK
3154  031146  000434                        BR      39$             ;DO TIMING LOOP
3155  031150  027737  000005  002654  36$:  CMP     #5,PASCNT       ;TEST IF PASS 5
3156  031156  001032                        BNE     40$             ;NO - SKIP
3157  031160  005037  002524                CLR     NEWCYL          ;ELSE SET UP TO TIME 256 CYL SEEK
3158  031164  004737  016070                JSR     PC,XSEEK        ;  OVER ALL SURFACE
3159  031170  031676                        65$
3160  031172  012701  005670                MOV     #3000.,R1       ;SET WAIT COUNT FOR 300 MS
3161  031176  004737  020650                JSR     PC,RDYWAIT      ;WAIT FOR DRIVE READY
3162  031202  031676                        65$
3163  031204  004737  021244                JSR     PC,VERPOS       ;VERIFY POSITION
```



```
3164  031210  031676                        65$
3165  031212  012700  002632                MOV     #AFMID,R0       ;SET POINTERS
3166  031216  012701  002634                MOV     #AFMIDO,R1
3167  031222  012703  002636                MOV     #ARMID,R3
3168  031226  012704  002640                MOV     #ARMIDO,R4
3169  031232  012737  000377  002524        MOV     #255.,NEWCYL    ;SET NEWCYL
3170  031240  000137  030300          39$:  JMP     8$
3171  031244                          40$:  PRINTF  #FMT1.1,#SKTMES,#VALDES
 (9)  031244  012746  006757                MOV     #VALDES,-(SP)
 (8)  031250  012746  006714                MOV     #SKTMES,-(SP)
 (7)  031254  012746  010631                MOV     #FMT1.1,-(SP)
 (6)  031260  012746  000003                MOV     #3,-(SP)
 (3)  031264  010600                        MOV     SP,R0
 (4)  031266  104017                        EMT     C$PNTF
 (4)  031270  062706  000010                ADD     #10,SP
3172  031274                                PRINTF  #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
(11)  031274  005046                        CLR     -(SP)
(11)  031276  153716  002455                BISB    RLDRV+1,(SP)
(10)  031302  012746  005633                MOV     #DRVNAM,-(SP)
 (9)  031306  013746  002450                MOV     RLBAS,-(SP)
 (8)  031312  012746  005624                MOV     #BASADD,-(SP)
 (7)  031316  012746  010657                MOV     #FMT5,-(SP)
 (6)  031322  012746  000005                MOV     #5,-(SP)
 (3)  031326  010600                        MOV     SP,R0
 (4)  031330  104017                        EMT     C$PNTF
 (4)  031332  062706  000014                ADD     #14,SP
3173  031336                                PRINTF  #FMT18,#LABIN,#LABMID,#LABOUT,#LABEXP
(11)  031336  012746  007051                MOV     #LABEXP,-(SP)
(10)  031342  012746  007043                MOV     #LABOUT,-(SP)
 (9)  031346  012746  007034                MOV     #LABMID,-(SP)
 (8)  031352  012746  007026                MOV     #LABIN,-(SP)
 (7)  031356  012746  011251                MOV     #FMT18,-(SP)
 (6)  031362  012746  000005                MOV     #5,-(SP)
 (3)  031366  010600                        MOV     SP,R0
 (4)  031370  104017                        EMT     C$PNTF
 (4)  031372  062706  000014                ADD     #14,SP
3174  031376                                PRINTF  #FMT19,#LABOCF,OFIN,OFMID,OFOUT,EXOCYL
(12)  031376  013746  002642                MOV     EXOCYL,-(SP)
(11)  031402  013746  002572                MOV     OFOUT,-(SP)
(10)  031406  013746  002566                MOV     OFMID,-(SP)
 (9)  031412  013746  002562                MOV     OFIN,-(SP)
 (8)  031416  012746  007062                MOV     #LABOCF,-(SP)
 (7)  031422  012746  011303                MOV     #FMT19,-(SP)
 (6)  031426  012746  000006                MOV     #6,-(SP)
 (3)  031432  010600                        MOV     SP,R0
 (4)  031434  104017                        EMT     C$PNTF
 (4)  031436  062706  000016                ADD     #16,SP
3175  031442                                PRINTF  #FMT19,#LABOCR,ORIN,ORMID,OROUT,EXOCYL
(12)  031442  013746  002642                MOV     EXOCYL,-(SP)
(11)  031446  013746  002606                MOV     OROUT,-(SP)
(10)  031452  013746  002602                MOV     ORMID,-(SP)
 (9)  031456  013746  002576                MOV     ORIN,-(SP)
 (8)  031462  012746  007076                MOV     #LABOCR,-(SP)
 (7)  031466  012746  011303                MOV     #FMT19,-(SP)
 (6)  031472  012746  000006                MOV     #6,-(SP)
 (3)  031476  010600                        MOV     SP,R0
```

```
   (4)  031500  104017                          EMT     C$PNTF
   (4)  031502  062706  000016                  ADD     #16,SP
 3176   031506                                  PRINTF  #FMT20,#LABHCF,HFIN,HFOUT,EXHCYL
  (11)  031506  013746  002644                  MOV     EXHCYL,-(SP)
  (10)  031512  013746  002616                  MOV     HFOUT,-(SP)
   (9)  031516  013746  002612                  MOV     HFIN,-(SP)
   (8)  031522  012746  007117                  MOV     #LABHCF,-(SP)
   (7)  031526  012746  011340                  MOV     #FMT20,-(SP)
   (6)  031532  012746  000005                  MOV     #5,-(SP)
   (3)  031536  010600                          MOV     SP,R0
   (4)  031540  104017                          EMT     C$PNTF
   (4)  031542  062706  000014                  ADD     #14,SP
 3177   031546                                  PRINTF  #FMT20,#LABHCR,HRIN,HROUT,EXHCYL
  (11)  031546  013746  002644                  MOV     EXHCYL,-(SP)
  (10)  031552  013746  002626                  MOV     HROUT,-(SP)
   (9)  031556  013746  002622                  MOV     HRIN,-(SP)
   (8)  031562  012746  007126                  MOV     #LABHCR,-(SP)
   (7)  031566  012746  011340                  MOV     #FMT20,-(SP)
   (6)  031572  012746  000005                  MOV     #5,-(SP)
   (3)  031576  010600                          MOV     SP,R0
   (4)  031600  104017                          EMT     C$PNTF
   (4)  031602  062706  000014                  ADD     #14,SP
 3178   031606                                  PRINTF  #FMT21,#LABACF,AFMID,EXACYL
  (10)  031606  013746  002646                  MOV     EXACYL,-(SP)
   (9)  031612  013746  002632                  MOV     AFMID,-(SP)
   (8)  031616  012746  007142                  MOV     #LABACF,-(SP)
   (7)  031622  012746  011370                  MOV     #FMT21,-(SP)
   (6)  031626  012746  000004                  MOV     #4,-(SP)
   (3)  031632  010600                          MOV     SP,R0
   (4)  031634  104017                          EMT     C$PNTF
   (4)  031636  062706  000012                  ADD     #12,SP
 3179   031642                                  PRINTF  #FMT21,#LABACR,ARMID,EXACYL
  (10)  031642  013746  002646                  MOV     EXACYL,-(SP)
   (9)  031646  013746  002636                  MOV     ARMID,-(SP)
   (8)  031652  012746  007156                  MOV     #LABACR,-(SP)
   (7)  031656  012746  011370                  MOV     #FMT21,-(SP)
   (6)  031662  012746  000004                  MOV     #4,-(SP)
   (3)  031666  010600                          MOV     SP,R0
   (4)  031670  104017                          EMT     C$PNTF
   (4)  031672  062706  000012                  ADD     #12,SP
 3180   031676                        65$:
 3181   031676                        ENDTST
   (3)  031676                        L10046:
   (3)  031676  104001                          EMT     C$ETST
```



```
 3183                                    .SBTTL  *TEST 13        **BASIC READ DATA (BAD SECTOR FILE)
 3184   031700                           BGNTST  ;TEST 13
   (3)  031700                                                                      T13::
 3185   031700  012737  006477  002434           MOV     #P2T13E,ERHEAD  ;SET ERROR HEADER
 3186   031706  004737  015160                   JSR     PC,TSTINT       ;INITIALIZE TEST
 3187   031712  004737  015176                   JSR     PC,GSTATR       ;CLEAR DRIVE
 3188   031716  032360                           65$
 3189   031720  012737  000001  002534           MOV     #1,DESHD        ;SET TO HEAD 1
 3190   031726  032737  010000  013372           BIT     #HEADLM,MISWIW  ;TEST IF HEAD SPEC'D
 3191   031734  001405                           BEQ     2$              ;NO - SKIP
 3192   031736  005737  013400                   TST     HEADW           ;TEST IF HEAD 0
 3193   031742  001002                           BNE     2$              ;NO - SKIP
 3194   031744                                   EXIT    TST             ;ELSE EXIT TEST
   (3)  031744  104032                           EMT     C$EXIT
   (3)  031746  000440                           .WORD   L10047-.
 3195   031750  012737  000377  002524  2$:      MOV     #255.,NEWCYL    ;POSITION HEADS AT 255
 3196   031756  004737  016070                   JSR     PC,XSEEK        ;DO SEEK
 3197   031762  032360                           65$
 3198   031764  012701  005670                   MOV     #3000.,R1       ;SET WAIT COUNT FOR 300 MS
 3199   031770  004737  020650                   JSR     PC,RDYWAIT      ;WAIT FOR INTERRUPT
 3200   031774  032360                           65$
 3201   031776  004737  021244                   JSR     PC,VERPOS       ;VERIFY POSITION
 3202   032002  032360                           65$
 3203   032004  005037  002536                   CLR     DESSEC          ;SET FOR SECTOR 0
 3204   032010  012737  003272  002552           MOV     #FBSFIL,TEMP5   ;SET TEMP STORAGE FOR FACTORY BS FILE
 3205   032016  012737  000020  002554           MOV     #16.,TEMP6      ;SET MAX SECTOR COUNT
 3206   032024  112737  000001  003067           MOVB    #1,NOERCT       ;SET FOR NO ERROR COUNTING
 3207   032032  105037  003066                   CLRB    LOCERR          ;CLEAR LOCAL ERROR COUNTER
 3208   032036  005037  002546           4$:     CLR     TEMP3           ;CLEAR ONES DETECTED FLAG
 3209   032042  013701  002552                   MOV     TEMP5,R1        ;INIT POINTERS
 3210   032046  013700  002554                   MOV     TEMP6,R0
 3211   032052  012703  003466                   MOV     #IBUFF,R3
 3212   032056  012737  000002  002440           MOV     #2,ERRSWI       ;INIT ERROR SWITCH
 3213   032064  004737  022414                   JSR     PC,XREAD        ;DO READ
 3214   032070  032242                           39$
 3215   032072  005723                           TST     (R3)+           ;TEST IF WORD 0 NOT NEG
 3216   032074  100515                           BMI     45$             ;YES, BAD FMT ERROR
 3217   032076  005723                           TST     (R3)+           ;ELSE TEST WORD 1 NOT NEG
 3218   032100  100513                           BMI     45$             ;YES - BAD FMT ERROR REPORT
 3219   032102  005723                   7$:     TST     (R3)+           ;TEST WORD 2 IS 0
 3220   032104  001111                           BNE     45$             ;NO - SKIP TO FMT ERROR RPT
 3221   032106  005723                           TST     (R3)+           ;TEST WORD 3 IS 0
 3222   032110  001107                           BNE     45$             ;NO - SKIP TO FMT ERROR RPT
 3223   032112  021327  177777           8$:     CMP     (R3),#-1        ;TEST IF NEXT WORD IS ALL 1'S
 3224   032116  001004                           BNE     10$             ;NO  - SKIP
 3225   032120  012737  000001  002546           MOV     #1,TEMP3        ;ELSE SET 1'S DETECTED FLAG
 3226   032126  000403                           BR      11$             ;SKIP
 3227   032130  005737  002546           10$:    TST     TEMP3           ;TEST IF ONES HAVE BEEN DETECTED
 3228   032134  001075                           BNE     45$             ;YES - SKIP TO FMT ERROR RPT
 3229   032136  012311           11$:    MOV     (R3)+,(R1)      ;STORE CYLINDER WORD
 3230   032140  012705  000007                   MOV     #7,R5           ;ALIGN IT TO LOOK LIKE HEADER
 3231   032144  006311           12$:    ASL     (R1)
 3232   032146  005305                   DEC     R5
 3233   032150  001375                   BNE     12$
 3234   032152  032713  000400                   BIT     #BIT8,(R3)      ;TEST IF HEAD 1
 3235   032156  001402                           BEQ     15$             ;NO - SKIP
```

```
  3236  032160  052711  000100            BIS     #BIT6,(R1)          ;INSERT HEAD BIT
  3237  032164  042713  177400            BIC     #177400,(R3)        ;CLEAR ALL BUT SECTOR
  3238  032170  052123                    BIS     (R3)+,(R1)+         ;INSERT SECTOR NUMBER
  3239  032172  020327  004066            CMP     R3,#IBUFF+256.      ;CHECK IF IBUFF EMPTY
  3240  032176  001345                    BNE     8$                  ;NO GET NEXT CYLINDER
  3241  032200  005737  002546            TST     TEMP3               ;ELSE TEST IF 1'S DETECTED
  3242  032204  001457                    BEQ     40$                 ;TO MANY ERRORS - REPORT
  3243  032206  022737  000044  002554    CMP     #36.,TEMP6          ;CHECK IF SOFTWARE BAD READ
  3244  032214  001461                    BEQ     65$                 ;YES - SKIP
  3245  032216  012737  003076  002552  37$:  MOV   #SBSFIL,TEMP5     ;ELSE CHANGE POINTERS
  3246  032224  012737  000044  002554        MOV   #36.,TEMP6        ;       MAX SECTOR NUMBER
  3247  032232  012737  000024  002536        MOV   #20.,DESSEC       ;       SECTOR NUMBER START
  3248  032240  000676                    BR      4$                  ;DO READ
  3249  032242  005237  003066  39$:     INC     LOCERR              ;BUMP LOCAL ERROR COUNTER
  3250  032246  012777  177777  150276  40$:     MOV     #-1,@TEMP5      ;MOV  1'S INTO FILE STORAGE
  3251  032254                            INLOOP                      ;CHECK IF IN ERROR LOOP
    (3) 032254  104020                      EMT     C$INLP
  3252  032256                            BCOMPLETE       4$                  ;YES - GO DO READ
    (4) 032256  103667                      BCS     4$
  3253  032260  023737  002536  002554  41$:     CMP     DESSEC,TEMP6    ;CHECK IF ALL SECTORS READ
  3254  032266  001014                    BNE     43$                 ;NO - SKIP
  3255  032270  012703  005503            MOV     #MBADSF,R3          ;SET RESULT MESSAGE POINTER
  3256  032274  005237  003066            INC     LOCERR              ;BUMP LOCAL ERROR COUNTER
  3257  032300                            ERRHRD  1301.,,ERR1
    (3) 032300  104443                      TRAP    T$ERCODE
    (5) 032302  002425                      .WORD   1301
    (5) 032304  011554                      .WORD   ERR1
  3258  032306  022737  003076  002552      CMP     #SBSFIL,TEMP5   ;TEST IF SOFTWARE FILES CHECKED
  3259  032314  001421                      BEQ     65$             ;YES - EXIT
  3260  032316  000737                      BR      37$             ;ELSE GO CHECK SOFTWARE FILES
  3261  032320  062737  000004  002536  43$:     ADD     #4,DESSEC       ;BUMP TO NEXT SECTOR
  3262  032326  000643                      BR      4$              ;GO DO READ
  3263  032330  012703  005533            45$:    MOV     #MFMTER,R3      ;SET RESULT MESSAGE POINTER
  3264  032334                            ERRHRD  1302.,,ERR1
    (3) 032334  104443                      TRAP    T$ERCODE
    (5) 032336  002426                      .WORD   1302
    (5) 032340  011554                      .WORD   ERR1
  3265  032342  000737                      BR      39$             ;GO CHECK FOR LOOP
  3266  032344  012703  005560            48$:    MOV     #MTMBS,R3       ;SET RESULT MESSAGE PTR
  3267  032350                            ERRHRD  1303.,,ERR1
    (3) 032350  104443                      TRAP    T$ERCODE
    (5) 032352  002427                      .WORD   1303
    (5) 032354  011554                      .WORD   ERR1
  3268  032356  000733                      BR      40$             ;GO CHECK FOR LOOP
  3269  032360  012737  000002  002440  65$:     MOV     #2,ERRSWI       ;INIT ERROR SWITCH
  3270  032366  012737  000001  003074          MOV     #1,BSFVAL       ;SET BAD SECTOR FILES VALID FLAG
  3271  032374  105737  003066                  TSTB    LOCERR          ;TEST IF LOCAL ERRORS
  3272  032400  001402                          BEQ     66$             ;NO - SKIP
  3273  032402  005237  002662                  INC     ERRCNT          ;ELSE BUMP ERROR COUNT
  3274  032406                            66$:
  3275  032406                            ENDTST
    (3) 032406                            L10047:
    (3) 032406  104001                      EMT     C$ETST
```



```
  3277                              .SBTTL  *TEST 14        **WRITE/READ DATA (PART 1)
  3278  032410                      BGNTST          ;TEST 14
    (3) 032410                                                                  T14::
  3279  032410  012737  006517  002434          MOV     #P2T14E,ERHEAD  ;SET ERROR HEADER
  3280  032416  004737  017500                  JSR     PC,CKBSVD       ;GO CHECK IF BAD SECTOR FILES VALID
  3281  032422  004737  015160                  JSR     PC,TSTINT       ;INITIALIZE TEST
  3282  032426  004737  015176                  JSR     PC,GSTATR       ;CLEAR DRIVE
  3283  032432  032622                          T3065$
  3284  032434  004737  017370                  JSR     PC,CHOSHD       ;GO CHOSE HEAD
  3285  032440  005037  002536                  CLR     DESSEC          ;       SECTOR 0
  3286  032444  005037  002524                  CLR     NEWCYL          ;       CYLINDER 0
  3287  032450  005037  032514                  CLR     T310$           ;CLEAR PATTERN SELECT
  3288  032454  004737  016070          T306$:  JSR     PC,XSEEK        ;POSITION HEADS
  3289  032460  032622                          T3065$
  3290  032462  012701  005670                  MOV     #3000.,R1       ;SET WAIT COUNT FOR 300 MS
  3291  032466  004737  020650                  JSR     PC,RDYWAIT      ;WAIT FOR READY
  3292  032472  032622                          T3065$
  3293  032474  004737  021244                  JSR     PC,VERPOS       ;VERIFY POSITION
  3294  032500  032622                          T3065$
  3295  032502  005037  032514                  CLR     T310$           ;CLEAR PATTERN SELECTOR
  3296  032506                            T307$:
  3297  032506                            BGNSUB                                  T14.1:
    (3) 032506
    (3) 032506  104002                      EMT     C$BSUB
  3298  032510  004537  021726                  JSR     R5,DATGEN       ;GENERATE DATA
  3299  032514  000000          T310$:  .WORD   0               ;PATTERN SELECT WORD
  3300  032516  004737  022354                  JSR     PC,XWRITE       ;DO WRITE DATA
  3301  032522  032540                          60$
  3302  032524  004737  022414                  JSR     PC,XREAD        ;DO READ DATA
  3303  032530  032540                          60$
  3304  032532  004737  022066                  JSR     PC,DATCOM       ;COMPARE DATA
  3305  032536  032540                          60$
  3306  032540  012737  000002  002440  60$:     MOV     #2,ERRSWI       ;INIT ERROR SWITCH
  3307  032546                            ENDSUB
    (3) 032546                            L10051:
    (3) 032546  104003                      EMT     C$ESUB
  3308  032550                            ESCAPE  TST             ;EXIT TEST IF ERROR
    (3) 032550  104010                      EMT     C$ESCAPE
    (3) 032552  000050                      .WORD   L10050-.
  3309  032554  022737  000010  032514      CMP     #8.,T310$       ;WAS DATA PAT 8 USED?
  3310  032562  001403                      BEQ     10$             ;YES - SKIP
  3311  032564  005237  032514              INC     T310$           ;ELSE BUMP TO NEXT PATTERN
  3312  032570  000746                      BR      T307$           ;DO TEST WITH NEW PATTERN
  3313  032572  004737  017414          10$:    JSR     PC,SWAPHD       ;GO SWAP TO HEAD 1 OR END TEST
  3314  032576  032622                          T3065$          ;ABORT RETURN
  3315  032600  005037  032514                  CLR     T310$           ;SET PATTERN SELECT TO 0
  3316  032604  004737  023074          11$:    JSR     PC,BSCHK        ;CHECK IF SECTOR BAD
  3317  032610  032614                          13$             ;YES RETURN - SKIP TO 13$
  3318  032612  000720                      BR      T306$           ;NO RETURN - DO TEST THIS SECTOR
  3319  032614  005237  002524          13$:    INC     NEWCYL          ;BUMP TO NEXT CYLINDER
  3320  032620  000771                      BR      11$             ;CHECK IF THIS ONE BAD
  3321  032622                            T3065$:
  3322  032622                            ENDTST
    (3) 032622                            L10050:
    (3) 032622  104001                      EMT     C$ETST
  3323
```

```
                                        .SBTTL  *TEST 15        **SPINDLE TIMING TEST
     032624                             BGNTST          ;TEST 15
(3)  032624                                                                      T15::
     032624  012737  006542  002434.            MOV     #P2T15E,ERHEAD  ;SET ERROR HEADER
     032632  005737  003062                     TST     PASNUM          ;TEST IF PASS 0
     032636  001003                             BNE     2$              ;NO - SKIP
     032640  005737  013372                     TST     MISWIW          ;TEST IF MANUAL TESTS WERE RUN
     032644  100402                             BMI     1$              ;YES - SKIP
     032646                             2$:     EXIT    TST             ;ELSE SKIP TEST
(3)  032646  104032                             EMT     C$EXIT
(3)  032650  000476                             .WORD   L10052-.
     032652  005003                     1$:     CLR     R3              ;CLEAR FOR TIMING STORAGE
     032654  005004                             CLR     R4
     032656  004737  015160                     JSR     PC,TSTINT       ;INITIALIZE TEST
     032662  004737  015176                     JSR     PC,GSTATR       ;CLEAR DRIVE
     032666  033340                             60$
     032670  004537  021726                     JSR     R5,DATGEN       ;GENERATE DATA
     032674  000000                             0                       ;PATTERN 0
     032676  005037  002536                     CLR     DESSEC          ;CLEAR TO SECTOR 0
     032702  004737  017370                     JSR     PC,CHOSHD       ;GO SELECT HEAD
     032706  013737  013374  002524             MOV     LOLIMW,NEWCYL   ;SET FOR CYLINDER
     032714  004737  016070                     JSR     PC,XSEEK        ;DO SEEK
     032720  033340                             60$
     032722  012701  005670                     MOV     #3000.,R1       ;SET WAIT FOR 300 MS
     032726  004737  020650                     JSR     PC,RDYWAIT      ;WAIT FOR READY
     032732  033340                             60$
     032734  004737  021244                     JSR     PC,VERPOS       ;VERIFY POSITION
     032740  033340                             60$
     032742  012701  000100                     MOV     #64.,R1 ;SET LOOP COUNTER
     032746  012705  002464             5$:     MOV     #L.MP,R5        ;SET A POINTER
     032752  004737  022344                     JSR     PC,XWRITT       ;DO FIRST WRITE
     032756  033340                             60$
     032760  011562  000006                     MOV     (R5),RLMP(R2)   ;LOAD RL REGISTERS
     032764  014562  000004                     MOV     -(R5),RLDA(R2)
     032770  014562  000002                     MOV     -(R5),RLBA(R2)
     032774  014562  000000                     MOV     -(R5),RLCS(R2)
     033000                                     WAITUS  #3000.
(3)  033000  012700  005670                     MOV     #3000.,R0
(3)  033004  104027                             EMT     C$WTU
     033006  005737  002430                     TST     DONE            ;TEST IF INTERRUPT
     033012  001010                             BNE     6$              ;YES - SKIP
     033014  004737  015026                     JSR     PC,WAITIN       ;ELSE WAIT FOR TIMEOUT
     033020  012603                             MOV     (SP)+,R3        ;GET MESSAGE POINTER
     033022                                     ERRHRD  1501.,,ERR1
(3)  033022  104443                             TRAP    T$ERCODE
(5)  033024  002735                             .WORD   1501
(5)  033026  011554                             .WORD   ERR1
     033030  000137  033340                     JMP     60$
     033034  005737  002466             6$:     TST     T.CS            ;TEST IF ANY ERRORS
     033040  100005                             BPL     4$              ;NO - SKIP
     033042                                     ERRHRD  1502.,,ERR6
(3)  033042  104443                             TRAP    T$ERCODE
(5)  033044  002736                             .WORD   1502
(5)  033046  012056                             .WORD   ERR6
     033050  000137  033340                     JMP     60$
     033054  012705  002464             4$:     MOV     #L.MP,R5        ;SET POINTER TO RL LOAD REGS
```



```
     033060  005037  002430                     CLR     DONE            ;CLEAR INTERRUPT INDICATOR
     033064  011562  000006                     MOV     (R5),RLMP(R2)   ;LOAD RL REGISTERS FOR 2ND WRITE
     033070  014562  000004                     MOV     -(R5),RLDA(R2)
     033074  014562  000002                     MOV     -(R5),RLBA(R2)
     033100  014562  000000                     MOV     -(R5),RLCS(R2)
     033104                                     WAITUS  #3000.          ;WAIT FOR INTERRUPT
(3)  033104  012700  005670                     MOV     #3000.,R0
(3)  033110  104027                             EMT     C$WTU
     033112                                     GETTIM  R0              ;GET TIME WAITED
(3)  033112  104052                             EMT     C$GTIM
     033114  005737  002430                     TST     DONE            ;TEST IN INTERRUPT OCCURRED
     033120  001007                             BNE     7$              ;YES - SKIP
     033122  004737  015026                     JSR     PC,WAITIN       ;GO WAIT FOR INTERRUPT
     033126  012603                             MOV     (SP)+,R3        ;GET MESSAGE POINTER
     033130                                     ERRHRD  1503.,,ERR1     ;REPORT
(3)  033130  104443                             TRAP    T$ERCODE
(5)  033132  002737                             .WORD   1503
(5)  033134  011554                             .WORD   ERR1
     033136  000500                             BR      60$
     033140  005737  002466             7$:     TST     T.CS            ;TEST IN ANY ERROR
     033144  100004                             BPL     8$              ;NO - SKIP
     033146                                     ERRHRD  1504.,,ERR6     ;REPORT ERRORS
(3)  033146  104443                             TRAP    T$ERCODE
(5)  033150  002740                             .WORD   1504
(5)  033152  012056                             .WORD   ERR6
     033154  000471                             BR      60$
     033156  060003                     8$:     ADD     R0,R3           ;ADD IN TIME USED
     033160  005504                             ADC     R4              ;DOUBLE PRECISSION
     033162  005301                             DEC     R1              ;DEC LOOP COUNTER
     033164  001270                             BNE     5$              ;LOOP UNTIL 0
     033166  012701  000006                     MOV     #6,R1           ;SET DIVIDE COUNT
     033172  000241                     10$:    CLC                     ;CLEAR CARRY FOR DIVIDE
     033174  006004                             ROR     R4              ;DIVIDE SUM BY 100(8)
     033176  006003                             ROR     R3
     033200  005301                             DEC     R1              ;DEC DIVIDE COUNT
     033202  001373                             BNE     10$             ;LOOP UNTIL DONE
     033204                                     PRINTF  #FMT1.1,#SRTMES,#VALDES
(9)  033204  012746  006757                     MOV     #VALDES,-(SP)
(8)  033210  012746  006730                     MOV     #SRTMES,-(SP)
(7)  033214  012746  010631                     MOV     #FMT1.1,-(SP)
(6)  033220  012746  000003                     MOV     #3,-(SP)
(3)  033224  010600                             MOV     SP,R0
(4)  033226  104017                             EMT     C$PNTF
(4)  033230  062706  000010                     ADD     #10,SP
     033234                                     PRINTF  #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
(11) 033234  005046                             CLR     -(SP)
(11) 033236  153716  002455                     BISB    RLDRV+1,(SP)
(10) 033242  012746  005633                     MOV     #DRVNAM,-(SP)
(9)  033246  013746  002450                     MOV     RLBAS,-(SP)
(8)  033252  012746  005622                     MOV     #BASADD,-(SP)
(7)  033256  012746  010657                     MOV     #FMT5,-(SP)
(6)  033262  012746  000005                     MOV     #5,-(SP)
(3)  033266  010600                             MOV     SP,R0
(4)  033270  104017                             EMT     C$PNTF
(4)  033272  062706  000014                     ADD     #14,SP
     033276                                     PRINTF  #FMT26,#RESE3,R3,#RESE4,#MAPROX,EXROT
```

```
  (12) 033276 013746 002650                MOV     EXROT,-(SP)
  (11) 033302 012746 007016                MOV     #MAPROX,-(SP)
  (10) 033306 012746 010407                MOV     #RESE4,-(SP)
   (9) 033312 010346                       MOV     R3,-(SP)
   (8) 033314 012746 010403                MOV     #RESE3,-(SP)
   (7) 033320 012746 011500                MOV     #FMT26,-(SP)
   (6) 033324 012746 000006                MOV     #6,-(SP)
   (3) 033330 010600                       MOV     SP,R0
   (4) 033332 104017                       EMT     C$PWTF
   (4) 033334 062706 000016                ADD     #16,SP
3400   033340 012737 000002 002440 60$:    MOV     #2,ERRSWI       ;INITIALIZE ERROR SWITCH
3401   033346                      ENDTST
   (3) 033346                      L10052:
   (3) 033346 104001                       EMT     C$ETST
3402
```



```
3404                                     .SBTTL  *TEST 16         **WRITE/READ DATA (PART 2)
3405   033350                            BGNTST                   ;TEST 16
   (3) 033350                                                                              T16::
3406   033350 012737 006572 002434               MOV     #P2T16E,ERRHEAD  ;SET ERROR HEADER
3407   033356 004737 017500                      JSR     PC,CKRSVD        ;GO CHECK IF BAD SECTOR FILES VALID
3408   033362 004737 015160                      JSR     PC,ISTINT        ;INITIALIZE TEST
3409   033366 004737 015176                      JSR     PC,GSTATR        ;CLEAR DRIVE
3410   033372 034340                             T3165$
3411   033374 005037 002654                      CLR     PASCNT           ;CLEAR PASS TO 0
3412   033400 012705 177777                      MOV     #-1,R5           ;SET R5
3413   033404 005737 003062                      TST     PASNUM           ;TEST IF FIRST PASS (QUICK VERIFY)
3414   033410 001006                             BNE     1$               ;NO - SKIP
3415   033412 032737 000001 013372               BIT     #ALLCYL,MISWIW   ;TEST IF USE ALL CYLINDERS
3416   033420 001002                             BNE     1$               ;YES - SKIP
3417   033422 012705 177770                      MOV     #-8.,R5          ;ELSE SET R5 TO NEG 8
3418   033426                            1$:
3419   033426 012701 002332                      MOV     #T33TBL,R1       ;GET ADDRESS OF WORK TABLE
3420   033432 012703 000010                      MOV     #10,R3           ;SET CLEAR COUNT
3421   033436 013721 013374              2$:     MOV     LOLIMW,(R1)+     ;CLEAR LOCATIONS TO LO LIMIT
3422   033442 005303                             DEC     R3               ;DEC COUNT
3423   033444 001374                             BNE     2$               ;LOOP UNTIL 0
3424   033446 113737 013376 002336               MOVB    HILIMW,T33TBL+4  ;INSERT HILIMIT
3425   033454 113737 013376 002340               MOVB    HILIMW,T33TBL+6  ;INTO APPROPRIATE LOCATIONS
3426   033462 113737 013376 002342               MOVB    HILIMW,T33TBL+10
3427   033470 005205                     T3100$: INC     R5               ;BUMP R5
3428   033472 032737 000001 013372               BIT     #ALLCYL,MISWIW   ;TEST IF USE ALL CYLINDERS
3429   033500 001017                             BNE     5$               ;YES - SKIP
3430   033502 005737 003062                      TST     PASNUM           ;TEST IF FIRST PASS (QUICK VERIFY)
3431   033506 001002                             BNE     3$               ;NO - SKIP
3432   033510 062705 000007                      ADD     #7,R5            ;ELSE BUMP CYLINDER POINTER BY 7
3433   033514 020527 000051              3$:     CMP     R5,#41.          ;TEST IF PAST TABLE
3434   033520 103005                             BHIS    4$               ;YES - GO TO EXIT
3435
3436   033522 116503 002352                      MOVB    CYLTBL(R5),R3    ;GET NEXT TABLE ENTRY
3437   033526 042703 177400                      BIC     #177400,R3       ;CLEAR UPPER BYTE
3438   033532 001006                             BNE     8$               ;SKIP IF NOT 0
3439   033534 000137 034370              4$:     JMP     T3165$           ;EXIT TEST
3440   033540 022705 000377              5$:     CMP     #255.,R5         ;TEST IF ALL CYLINDERS USED
3441   033544 001773                             BEQ     4$               ;YES - EXIT TEST
3442   033546 010503                             MOV     R5,R3            ;USE R5 AS NEXT CYLINDER
3443   033550 020337 013374              8$:     CMP     R3,LOLIMW        ;CHECK IF LOWER THAN LOLIMIT
3444   033554 103745                             BLO     T3100$           ;YES - SKIP
3445   033556 020337 013376                      CMP     R3,HILIMW        ;CHECK IF HIGHER THAN HILIMIT
3446   033562 101342                             BHI     T3100$           ;YES - SKIP
3447   033564 012704 002332                      MOV     #T33TBL,R4       ;GET ADDRESS OF SEEK TABLE
3448   033570 110364 000001                      MOVB    R3,1(R4)         ;INSERT CC IN APPROPRIATE TABLE
3449   033574 110364 000003                      MOVB    R3,3(R4)         ; LOCATIONS FOR TEST SEEK SEQUENCE
3450   033600 110364 000005                      MOVB    R3,5(R4)
3451   033604 110364 000007                      MOVB    R3,7(R4)
3452   033610 110364 000011                      MOVB    R3,11(R4)
3453   033614 110364 000013                      MOVB    R3,13(R4)
3454   033620 010437 002446                      MOV     R4,TBLSTR        ;STORE TABLE ADDRESS
3455   033624 004737 017370                      JSR     PC,CHOSHD        ;GO CHOSE HEAD
3456   033630                            T3101$:
3457   033630                            BGNSUB
   (3) 033630                                                                              T16.1:
```

```
  (3)  033630  104002                          EMT     C$BSUB
3458   033632  042737  003760  002426          BIC     #MQUALS,OPFLAG  ;CLEAR ALL MESSAGE QUALIFIERS
3459   033640  005737  002654                  TST     PASCNT          ;TEST IF PASS 0
3460   033644  001414                          BEQ     11$             ;YES - SKIP
3461   033646  023727  002654  000003          CMP     PASCNT,#3       ;TEST IF PASS 3
3462   033654  001404                          BEQ     10$             ;YES - SKIP
3463   033656  002407                          BLT     11$             ;CHECK IF LESS THAN 3, IF YES CLEAR TO 0
3464   033660  012737  000003  002654          MOV     #3,PASCNT       ;ELSE SET TO 3
3465   033666  052737  000020  002426  10$:    BIS     #INOUTS,OPFLAG  ;SET MESSAGE QUAL
3466   033674  000405                          BR      12$             ;SKIP
3467   033676  005037  002654          11$:    CLR     PASCNT          ;SET PASS COUNT TO 0
3468   033702  052737  000040  002426          BIS     #OUTINS,OPFLAG  ;SET MESSAGE QUAL
3469   033710  012737  000003  002444  12$:    MOV     #3,WRTSWI       ;SET READ AND WRITE SWITCH
3470   033716  013704  002446                  MOV     TBLSTR,R4       ;GET STORED TABLE ADDRESS
3471   033722  005037  002536          15$:    CLR     DESSEC          ;CLEAR TO SECTOR 0
3472   033726  112437  002524                  MOVB    (R4)+,NEWCYL    ;GET NEXT TABLE ENTRY
3473   033732  004737  016070                  JSR     PC,XSEEK        ;DO SEEK
3474   033736  034302                          60$
3475   033740  012701  005670                  MOV     #3000.,R1       ;SET WAIT COUNT FOR 300 MS
3476   033744  004737  020650                  JSR     PC,RDYWAIT      ;WAIT FOR READY
3477   033750  034302                          60$
3478   033752  112437  002524                  MOVB    (R4)+,NEWCYL    ;GET NEXT TABLE ENTRY
3479   033756  004737  016070                  JSR     PC,XSEEK        ;DO SEEK
3480   033762  034302                          60$
3481   033764  012701  005670                  MOV     #3000.,R1       ;SET WAIT COUNT FOR 300 MS
3482   033770  004737  020650                  JSR     PC,RDYWAIT      ;WAIT FOR READY
3483   033774  034302                          60$
3484   033776  004737  021244                  JSR     PC,VERPOS       ;VERIFY POSITION
3485   034002  034302                          60$
3486   034004  004737  023074          16$:    JSR     PC,BSCHK        ;CHECK FOR BAD SECTOR
3487   034010  034142                          32$                     ;"YES" RETURN
3488   034012  013737  002536  034032          MOV     DESSEC,25$      ;SET DATA PATTERN = TO SECTOR NUMBER
3489   034020  042737  177770  034032          BIC     #177770,25$     ;CLEAR ALL BUT LSD
3490   034026  004537  021726                  JSR     R5,DATGEN       ;GO GENERATE DATA
3491   034032  000000          25$:    .WORD   0
3492   034034  032737  000001  002444          BIT     #BIT0,WRTSWI    ;TEST IF WRITE THIS PASS
3493   034042  001425                          BEQ     29$             ;NO - SKIP
3494   034044  004737  022354                  JSR     PC,XWRITE       ;DO WRITE
3495   034050  034302                          60$
3496   034052  005237  002536                  INC     DESSEC          ;INC SECTOR
3497   034056  022737  000050  002536          CMP     #40.,DESSEC     ;TEST IF ALL SECTORS USED
3498   034064  001347                          BNE     16$             ;NO - SKIP
3499   034066  042737  000060  002426          BIC     #INOUTS!OUTINS,OPFLAG ;CLEAR QUALIFIERS
3500   034074  042737  000001  002444          BIC     #BIT0,WRTSWI    ;CLEAR WRITE REQUIRED SWITCH
3501   034102  052737  000100  002426          BIS     #FOLWRT,OPFLAG  ;SET FOLLOWING WRITE QUALIFIER
3502   034110  005037  002536                  CLR     DESSEC          ;CLEAR TO SECTOR 0
3503   034114  000733                          BR      16$             ;SKIP
3504   034116  032737  000002  002444  29$:    BIT     #BIT1,WRTSWI    ;TEST IF READ THIS PASS
3505   034124  001414                          BEQ     33$             ;NO - SKIP
3506   034126  004737  022414          31$:    JSR     PC,XREAD        ;ELSE DO READ
3507   034132  034302                          60$
3508   034134  004737  022066                  JSR     PC,DATCOM       ;COMPARE DATA
3509   034140  034302                          60$
3510   034142  005237  002536          32$:    INC     DESSEC          ;BUMP SECTOR
3511   034146  022737  000050  002536          CMP     #40.,DESSEC     ;TEST IF ALL SECTORS USED
3512   034154  001313                          BNE     16$             ;NO - LOOP
```



```
3513   034156  005037  002536          33$:    CLR     DESSEC          ;CLEAR DESIRED SECTOR
3514   034162  005037  002444                  CLR     WRTSWI          ;CLEAR WRITE/READ SWITCH
3515   034166  005237  002654                  INC     PASCNT          ;BUMP PASS COUNT
3516   034172  042737  003760  002426          BIC     #MQUALS,OPFLAG  ;CLEAR ALL QUALIFIERS
3517   034200  023727  002654  000003          CMP     PASCNT,#3       ;TEST IS PASS 3
3518   034206  001435                          BEQ     60$             ;YES - SKIP
3519   034210  023727  002654  000006          CMP     PASCNT,#6       ;TEST IF PASS 6
3520   034216  001431                          BEQ     60$             ;YES - SKIP
3521   034220  012737  000002  002444          MOV     #BIT1,WRTSWI    ;SET READ REQUIRED BIT
3522   034226  023727  002654  000001          CMP     PASCNT,#1       ;TEST IF PASS 1
3523   034234  001415                          BEQ     40$             ;YES - SKIP
3524   034236  023727  002654  000005          CMP     PASCNT,#5       ;TEST IF PASS 4
3525   034244  001411                          BEQ     40$             ;YES - SKIP
3526   034246  000404                          BR      39$             ;SKIP
3527   034250  052737  002000  002426  37$:    BIS     #FWDSKO,OPFLAG  ;SET FWD QUALIFIER
3528   034256  000407                          BR      36$             ;GO DO NEXT PASS
3529   034260  052737  000020  002426  39$:    BIS     #INOUTS,OPFLAG  ;SET QUALIFIER
3530   034266  000403                          BR      36$             ;SKIP
3531   034270  052737  000040  002426  40$:    BIS     #OUTINS,OPFLAG  ;SET MESSAGE QUALIFIER
3532   034276  000137  033722          36$:    JMP     15$             ;GO DO NEXT PASS
3533   034302  012737  000002  002440  60$:    MOV     #2,ERRSWI       ;INIT ERROR SWITCH
3534   034310                          ENDSUB
  (3)  034310                          L10054:
  (3)  034310  104003                          EMT     C$ESUB
3535   034312                          ESCAPE  TST                     ;EXIT TEST IF ERROR
  (3)  034312  104010                          EMT     C$ESCAPE
  (3)  034314  000054                          .WORD   L10053-.
3536   034316  012737  000003  002444          MOV     #3,WRTSWI       ;SET FOR READ AND WRITE REQ.
3537   034324  023727  002654  000003          CMP     PASCNT,#3       ;TEST IF PASS 3
3538   034332  001004                          BNE     45$             ;NO - SKIP
3539   034334  012737  002340  002446          MOV     #T33TBL+6,TBLSTR ;STORE MID POINT IN TABLE
3540   034342  000410                          BR      48$             ;GO START PASS 4
3541   034344  005037  002654          45$:    CLR     PASCNT          ;CLEAR TO PASS 0
3542   034350  004737  017414                  JSR     PC,SWAPHD       ;GO SWAP TO HEAD 1 OR END TEST
3543   034354  033470                          T3100$                  ;ABORT RETURN
3544   034356  012737  002332  002446          MOV     #T33TBL,TBLSTR  ;STORE START OF TABLE
3545   034364  000137  033630          48$:    JMP     T3101$          ;GO DO HEAD 1
3546   034370                          T3165$:
3547   034370                          ENDTST
  (3)  034370                          L10053:
  (3)  034370  104001                          EMT     C$ETST
```

```
                                        .SBTTL  *TEST 17         **WRITE LOCK ERROR AND DATA PROTECTION
                                        BGNTST                   ;TEST 17
                                                                                          T17::
                                        TST     PASNUM           ;TEST IF FIRST PASS
                                        BNE     2$               ;NO - SKIP
                                        TST     MISWIW           ;TEST IF RUN MANUAL INTERVENTION
                                        BMI     3$               ;YES - SKIP
                                        JMP     T3265$           ;EXIT TST
                                2$:
                                3$:
                                        BGNSUB
                                                                                          T17.1:
                                        EMT     C$BSUB
                                        MOV     #P2T17E,ERHEAD   ;SET ERROR HEADER
                                        JSR     PC,*TSTINT       ;INITIALIZE TEST
                                        JSR     PC,GSTATR        ;CLEAR DRIVE
                                        60$
                                        CLR     DESHD            ;SET TO HEAD 0
                                        CLR     DESSEC           ;SET TO SECTOR 0
                                        CLR     NEWCYL           ;CLEAR TO CYLINDER 0
                                        JSR     PC,XSEEK         ;DO SEEK
                                        60$
                                        MOV     #3000.,R1        ;SET WAIT FOR 300 MS
                                        JSR     PC,RDYWAIT       ;WAIT FOR READY
                                        60$
                                        JSR     PC,VERPOS        ;VERIFY POSITION
                                        60$
                                        BIT     #WLSTAT,T.MP     ;TEST IF WRITE LOCK SET
                                        BNE     7$               ;YES - SKIP
                                        JSR     R5,DATGEN        ;GENERATE DATA
                                        7                        ;PATTERN 7
                                        JSR     PC,XWRITE        ;WRITE DATA
                                        60$
                                        JSR     PC,XREAD         ;READ DATA
                                        60$
                                        JSR     PC,DATCOM        ;CHECK DATA
                                        60$
                                        PRINTF  #FMTOP1,#OPR004,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1> ;REQUEST SET WR
                                        -(SP)
                                        BISB    RLDRV+1,(SP)
                                        MOV     #DRVNAM,-(SP)
                                        MOV     RLBAS,-(SP)
                                        MOV     #BASADD,-(SP)
                                        MOV     #OPR1A,-(SP)
                                        MOV     #OPR004,-(SP)
                                        MOV     #FMTOP1,-(SP)
                                        MOV     #7,-(SP)
                                        MOV     SP,R0
                                        EMT     C$PNTF
                                        ADD     #20,SP
                                        MOV     #6.,R1           ;SET WAIT COUNT FOR 30 SECONDS
                                5$:     WAITMS  #50.             ;CALL WAIT
                                        MOV     #50.,R0
                                        EMT     C$WTM
                                        JSR     PC,GSTATR        ;GET STATUS
                                        60$
                                        BIT     #WLSTAT,T.MP     ;CHECK IF WRITE LOCK SET
```



```
                                        BNE     7$               ;YES - SKIP
                                        PRINTF  #FMT2,#BELL      ;RING BELL
                                        MOV     #BELL,-(SP)
                                        MOV     #FMT2,-(SP)
                                        MOV     #2,-(SP)
                                        MOV     SP,R0
                                        EMT     C$PNTF
                                        ADD     #6,SP
                                        DEC     R1               ;DEC COUNT
                                        BNE     5$               ;SKIP IF NOT 0
                                        PRINTF  #FMT23,#P2T17E,#BYPSNM,#OPR1A,<B,RLDRV+1> ;RPT BYPASSED
                                        CLR     -(SP)
                                        BISB    RLDRV+1,(SP)
                                        MOV     #OPR1A,-(SP)
                                        MOV     #BYPSNM,-(SP)
                                        MOV     #P2T17E,-(SP)
                                        MOV     #FMT23,-(SP)
                                        MOV     #5,-(SP)
                                        MOV     SP,R0
                                        EMT     C$PNTF
                                        ADD     #14,SP
                                        EXIT    T$T
                                        EMT     C$EXIT
                                        .WORD   L10055-.
                                7$:     JSR     R5,DATGEN        ;GENERATE DATA
                                        1                        ;PATTERN 1
                                        MOV     #L.CS,R5         ;GET ADDRESS OF L REGS
                                        MOV     #WTDATA,(R5)     ;LOAD WRITE COMMAND
                                        BIS     RLDRV,(R5)       ;INSERT DRIVE NUMBER
                                        BIC     #BIT10,(R5)+     ;CLEAR FOR DRIVE 4 - 7 SPEC'D
                                        MOV     #OBUFF,(R5)+     ;LOAD BUS ADDRESS
                                        CLR     (R5)+            ;CYL 0, HD 0, SECTOR 0
                                        MOV     #177600,(R5)+    ;128 WORDS
                                        MOV     #300.,R1         ;SET WAIT COUNT FOR 30 MS
                                        CLR     DONE             ;CLEAR INTERRUPT FLAG
                                        MOV     -(R5),RLMP(R2)   ;LOAD RL REGS
                                        MOV     -(R5),RLDA(R2)
                                        MOV     -(R5),RLBA(R2)
                                        MOV     -(R5),RLCS(R2)
                                10$:    WAITUS  #1
                                        MOV     #1,R0
                                        EMT     C$WTU
                                        TST     DONE             ;CHECK IF INTERRUPT
                                        BNE     14$              ;YES - SKIP
                                        DEC     R1               ;DEC WAIT COUNT
                                        BNE     10$              ;LOOP IF NOT 0
                                        JSR     PC,WAITIN        ;WAIT FOR INTERRUPT
                                        MOV     (SP)+,R3         ;GET RESULT MESSAGE
                                        ERRHRD  1701.,ERR1
                                        TRAP    T$ERCODE
                                        .WORD   1701
                                        .WORD   ERR1
                                        EXIT    SUB
                                        EMT     C$EXIT
                                        .WORD   L10056-.
                                14$:    JSR     PC,GSTAT         ;GET STATUS
```

```
                                        60$
035072  035234                          BIT     #DRVERR,T.CS    ;TEST IF ANY ERROR SET
035074  032737  040000  002466          BNE     15$             ;YES - SKIP
035102  001005                          MOV     #NDRERR,R3      ;SET RESULT MESSAGE POINTER
035104  012703  007665                  ERRHRD  1702.,,ERR3     ;REPORT ERROR NOT SET
035110                                  TRAP    T$ERCODE
035110  104443                          .WORD   1702
035112  003246                          .WORD   ERR3
035114  011670
035116  032737  002000  002474  15$:    BIT     #WGESTAT,T.MP   ;TEST IF WGE SET
035124  001005                          BNE     18$             ;YES - SKIP
035126  012703  007744                  MOV     #WGERR,R3       ;SET MESSAGE FOR WGE NOT SET
035132                                  ERRHRD  1704.,,ERR3
035132  104443                          TRAP    T$ERCODE
035134  003250                          .WORD   1704
035136  011670                          .WORD   ERR3
035140  042737  040000  002466  18$:    BIC     #DRVERR,T.CS    ;CLEAR DRIVE ERROR BIT
035146  042737  002000  002474          BIC     #WGESTAT,T.MP   ;CLEAR WGE BIT
035154  032737  157400  002474          BIT     #157400,T.MP    ;TEST IF ANY OTHER ERRORS
035162  001004                          BNE     16$             ;YES - GO REPORT
035164  032737  036000  002466          BIT     #36000,T.CS     ;TEST ANY ERRORS IN CS REG
035172  001404                          BEQ     17$             ;NO - SKIP
035174                          16$:    ERRHRD  1703.,,ERR6     ;REPORT ERRORS
035174  104443                          TRAP    T$ERCODE
035176  003247                          .WORD   1703
035200  012056                          .WORD   ERR6
035202  000414                          BR      60$             ;EXIT TEST
035204  004737  015176          17$:    JSR     PC,GSTATR       ;GET STATUS AND RESET ERROR
035210  035234                          60$
035212  004537  021726                  JSR     R5,DATGEN       ;GO GENERATE DATA
035216  000007                          7                       ;PATTERN 7
035220  004737  022414                  JSR     PC,XREAD        ;READ DATA
035224  035234                          60$
035226  004737  022066                  JSR     PC,DATCOM       ;COMPARE DATA
035232  035234                          60$
035234  012737  000002  002440  60$:    MOV     #2,ERRSWI       ;INIT ERROR SWITCH
035242                                  ENDSUB
035242                                  L10056:
035242  104003                          EMT     C$ESUB
035244  012737  000002  002440  T3204$: MOV     #2,ERRSWI       ;INIT ERROR SWITCH
035252                                  PRINTF  #FMTOP1,#OPR12,#OPR1A,#BASADD,RLBAS,#DRVWAM,<B,RLDRV+1> ;REQ RESET WRT L
035252  005046                          -(SP)
035254  153716  002455                  CLR     RLDRV+1,(SP)
035260  012746  005633                  BISB    #DRVWAM,-(SP)
035264  013746  002450                  MOV     RLBAS,-(SP)
035270  012746  005622                  MOV     #BASADD,-(SP)
035274  012746  007250                  MOV     #OPR1A,-(SP)
035300  012746  007231                  MOV     #OPR12,-(SP)
035304  012746  010537                  MOV     #FMTOP1,-(SP)
035310  012746  000007                  MOV     #7,-(SP)
035314  010600                          MOV     SP,R0
035316  104017                          EMT     C$PNTF
035320  062706  000020                  ADD     #20,SP
035324  012701  000454                  MOV     #300.,R1        ;SET WAIT FOR 30 SEC
035330                          16$:    WAITMS  #1
035330  012700  000001                  MOV     #1,R0
035334  104026                          EMT     C$WTM
```



```
035336  004737  015176                  JSR     PC,GSTATR       ;GET STATUS
035342  035244                          T3204$
035344  032737  020000  002474          BIT     #WLSTAT,T.MP    ;CHECK IF WRITE LOCK RESET
035352  001403                          BEQ     T3265$
035354  005301                          DEC     R1              ;DEC WAIT COUNT
035356  001364                          BNE     16$             ;LOOP IF NOT 0
035360  000731                          BR      T3204$          ;ELSE REPEAT MESSAGE
035362                          T3265$:
035362                                  ENDTST
035362                                  L10055:
035362  104001                          EMT     C$ETST
```

```
                                        .SBTTL  *TEST 18        **ADJACENT CYLINDER INTERFERENCE
035364                                  BGNTST                  ;TEST 18
035364                                                                                  T18::
035364  012737  006655  002434          MOV     #P2T18E,ERHEAD  ;SET ERROR HEADER
035372  004737  017500                  JSR     PC,CKBSVD       ;GO CHECK IF BAD SECTOR FILES VALID
035376  004737  015160                  JSR     PC,TSTINT       ;INITIALIZE TEST
035402  004737  015176                  JSR     PC,GSTATR       ;CLEAR DRIVE
035406  036472                          T3365$
035410  005037  002654                  CLR     PASCNT          ;CLEAR PASS TO 0
035414  012705  177777                  MOV     #-1,R5          ;SET R5
035420  005737  003062                  TST     PASNUM          ;TEST IF FIRST PASS (QUICK VERIFY)
035424  001007                          BNE     1$              ;NO - SKIP
035426  032737  000001  013372          BIT     #ALLCYL,MISWIW  ;TEST IF USE ALL CYLINDERS
035434  001003                          BNE     1$              ;YES - SKIP
035436  012705  177754                  MOV     #-20.,R5                ;ELSE SET R5 TO NEG 20
035442  000402                          BR      9$              ;SKIP
035444  012705  177774          1$:     MOV     #-4,R5          ;ELSE SET FOR NEG 4
035450  012701  002332          9$:     MOV     #T33TBL,R1      ;GET ADDRESS OF WORK TABLE
035454  012703  000010                  MOV     #10,R3          ;SET CLEAR COUNT
035460  013721  013374          2$:     MOV     LOLIMW,(R1)+    ;CLEAR LOCATIONS TO LOLIMIT
035464  005303                          DEC     R3              ;DEC COUNT
035466  001374                          BNE     2$              ;LOOP UNTIL 0
035470  004537  021726                  JSR     R5,DATGEN       ;GO GENERATE DATA
035474  000011                          9.                      ;PATTERN 9
035476  113737  013376  002334          MOVB    HILIMW,T33TBL+2 ;INSERT HILIMIT
035504  113737  013376  002336          MOVB    HILIMW,T33TBL+4 ;INTO APPROPRIATE LOCATIONS
035512  113737  013376  002342          MOVB    HILIMW,T33TBL+10
035520  113737  013376  002350          MOVB    HILIMW,T33TBL+16
035526  005205                  T3300$: INC     R5              ;BUMP R5
035530  032737  000001  013372          BIT     #ALLCYL,MISWIW  ;TEST IF USE ALL CYLINDERS
035536  001022                          BNE     5$              ;YES - SKIP
035540  005737  003062                  TST     PASNUM          ;TEST IF FIRST PASS (QUICK VERIFY)
035544  001403                          BEQ     3$              ;NO - SKIP
035546  062705  000003                  ADD     #3,R5           ;ELSE BUMP CYLINDER POINTER BY 3
035552  000402                          BR      6$              ;SKIP
035554  062705  000023          3$:     ADD     #19.,R5         ;BUMP TO NEXT ENTRY
035560  020527  000051          6$:     CMP     R5,#41.         ;CHECK IF PAST TABLE
035564  103005                          BHIS    4$              ;YES - SKIP TO EXIT
035566  116503  002352                  MOVB    CYLTBL(R5),R3   ;GET NEXT TABLE ENTRY
035572  042703  177400                  BIC     #177400,R3      ;CLEAR UPPER BYTE
035576  001011                          BNE     8$              ;SKIP IF NOT 0
035600  000137  036472          4$:     JMP     T3365$          ;EXIT TEST
035604  005705                  5$:     TST     R5              ;TEST IF R5 0
035606  001001                          BNE     7$              ;NO - SKIP
035610  005205                          INC     R5              ;ELSE BUMP R5 AGAIN
035612  022705  000377          7$:     CMP     #255.,R5        ;TEST IF ALL CYLINDERS USED
035616  001770                          BEQ     4$              ;YES - EXIT TEST
035620  010503                          MOV     R5,R3           ;USE R5 AS NEXT CYLINDER
035622  020337  013374          8$:     CMP     R3,LOLIMW       ;CHECK IF LOWER THAN LOLIMIT
035626  103737                          BLO     T3300$          ;YES - SKIP
035630  020337  013376                  CMP     R3,HILIMW       ;CHECK IF HIGHER THAN HILIMIT
035634  101334                          BHI     T3300$          ;YES - SKIP
035636  012704  002332                  MOV     #T33TBL,R4      ;GET ADDRESS OF SEEK TABLE
035642  110364  000001                  MOVB    R3,1(R4)        ;INSERT CC IN APPROPRIATE TABLE
035646  110364  000007                  MOVB    R3,7(R4)
035652  110364  000011                  MOVB    R3,11(R4)
```



```
035656  110364  000017                  MOVB    R3,17(R4)
035662  005203                          INC     R3              ;BUMP R3 TO CC+1
035664  110364  000005                  MOVB    R3,5(R4)        ;INSERT AS NEEDED
035670  110364  000015                  MOVB    R3,15(R4)
035674  162703  000002                  SUB     #2,R3           ;SET R3 TO CC-1
035700  110364  000003                  MOVB    R3,3(R4)        ;INSERT AS NEEDED
035704  110364  000013                  MOVB    R3,13(R4)
035710  010437  002446                  MOV     R4,TBLSTR       ;STORE TABLE ADDRESS
035714  004737  017370                  JSR     PC,CHOSHD       ;GO CHOSE HEAD
035720                          T3301$:
035720                          BGNSUB
035720                                                                                  T18.1:
035720  104002                          EMT     C$BSUB
035722  042737  003760  002426          BIC     #MQUALS,OPFLAG  ;CLEAR ALL MESSAGE QUALIFIERS
035730  005737  002654                  TST     PASCNT          ;TEST IF PASS 0
035734  001414                          BEQ     11$             ;YES - SKIP
035736  023727  002654  000004          CMP     PASCNT,#4       ;TEST IF PASS 4
035744  001404                          BEQ     10$             ;YES - SKIP
035746  002407                          BLT     11$             ;CHECK IF LESS THAN 4, IF YES CLEAR TO 0
035750  012737  000004  002654          MOV     #4,PASCNT       ;ELSE SET TO 4
035756  052737  000020  002426  10$:     BIS     #INOUTS,OPFLAG  ;SET MESSAGE QUAL
035764  000405                          BR      12$             ;SKIP
035766  005037  002654          11$:    CLR     PASCNT          ;SET PASS COUNT TO 0
035772  052737  000040  002426          BIS     #OUTINS,OPFLAG  ;SET MESSAGE QUAL
036000  012737  000003  002444  12$:     MOV     #3,WRTSWI       ;SET READ AND WRITE SWITCH
036006  013704  002446                  MOV     TBLSTR,R4       ;GET STORED TABLE ADDRESS
036012  005037  002536          15$:    CLR     DESSEC          ;CLEAR TO SECTOR 0
036016  112437  002524                  MOVB    (R4)+,NEWCYL    ;GET NEXT TABLE ENTRY
036022  004737  016070                  JSR     PC,XSEEK        ;DO SEEK
036026  036404                          60$
036030  012701  005670                  MOV     #3000.,R1       ;SET WAIT COUNT FOR 300 MS
036034  004737  020650                  JSR     PC,RDYWAIT      ;WAIT FOR READY
036040  036404                          60$
036042  112437  002524                  MOVB    (R4)+,NEWCYL    ;GET NEXT TABLE ENTRY
036046  004737  016070                  JSR     PC,XSEEK        ;DO SEEK
036052  036404                          60$
036054  012701  005670                  MOV     #3000.,R1       ;SET WAIT COUNT FOR 300 MS
036060  004737  020650                  JSR     PC,RDYWAIT      ;WAIT FOR READY
036064  036404                          60$
036066  004737  021244                  JSR     PC,VERPOS       ;VERIFY POSITION
036072  036404                          60$
036074  004737  023074          16$:    JSR     PC,BSCHK        ;CHECK FOR BAD SECTOR
036100  036210                          32$                     ;"YES" RETURN
036102  032737  000001  002444          BIT     #BIT0,WRTSWI    ;TEST IF WRITE THIS PASS
036110  001425                          BEQ     29$             ;NO - SKIP
036112  004737  022354                  JSR     PC,XWRITE       ;DO WRITE
036116  036404                          60$
036120  005237  002536                  INC     DESSEC          ;INC SECTOR
036124  022737  000050  002536          CMP     #40.,DESSEC     ;TEST IF ALL SECTORS USED
036132  001360                          BNE     16$             ;NO - SKIP
036134  042737  000060  002426          BIC     #INOUTS!OUTINS,OPFLAG ;CLEAR QUALIFIERS
036142  042737  000001  002444          BIC     #BIT0,WRTSWI    ;CLEAR WRITE REQUIRED SWITCH
036150  052737  000100  002426          BIS     #FOLWRT,OPFLAG  ;SET FOLLOWING WRITE QUALIFIER
036156  005037  002536                  CLR     DESSEC          ;CLEAR TO SECTOR 0
036162  000744                          BR      16$             ;SKIP
036164  032737  000002  002444  29$:     BIT     #BIT1,WRTSWI    ;TEST IF READ THIS PASS
```

```
 3770  036172  001414                        BEQ     33$             ;NO - SKIP
 3771  036174  004737  022414          31$:  JSR     PC,XREAD        ;ELSE DO READ
 3772  036200  036404                        60$
 3773  036202  004737  022066                JSR     PC,DATCOM       ;COMPARE DATA
 3774  036206  036404                        60$
 3775  036210  005237  002536          32$:  INC     DESSEC          ;BUMP SECTOR
 3776  036214  022737  000050  002536        CMP     #40.,DESSEC     ;TEST IF ALL SECTORS USED
 3777  036222  001324                        BNE     16$             ;NO - LOOP
 3778  036224  005037  002536          33$:  CLR     DESSEC          ;CLEAR DESIRED SECTOR
 3779  036230  005037  002444                CLR     WRTSWI          ;CLEAR WRITE/READ SWITCH
 3780  036234  005237  002654                INC     PASCNT          ;BUMP PASS COUNT
 3781  036240  042737  003760  002426        BIC     #MQUALS,OPFLAG  ;CLEAR ALL QUALIFIERS
 3782  036246  023727  002654  000004        CMP     PASCNT,#4       ;TEST IS PASS 4
 3783  036254  001453                        BEQ     60$             ;YES - SKIP
 3784  036256  023727  002654  000010        CMP     PASCNT,#8.      ;TEST IF PASS 8.
 3785  036264  001447                        BEQ     60$             ;YES - SKIP
 3786  036266  023727  002654  000003        CMP     PASCNT,#3       ;TEST IF PASS 3
 3787  036274  001430                        BEQ     39$             ;YES - SKIP
 3788  036276  023727  002654  000007        CMP     PASCNT,#7       ;TEST IF PASS 7
 3789  036304  001431                        BEQ     40$             ;YES - SKIP
 3790  036306  012737  000001  002444        MOV     #BIT0,WRTSWI    ;SET WRITE REQUIRED
 3791  036314  023727  002654  000001        CMP     PASCNT,#1       ;TEST IF PASS 1
 3792  036322  001411                        BEQ     37$             ;YES - SKIP
 3793  036324  023727  002654  000002        CMP     PASCNT,#2       ;TEST IF PASS 2
 3794  036332  001405                        BEQ     37$             ;YES - SKIP
 3795  036334  052737  000040  002426        BIS     #OUTINS,OPFLAG  ;SET MESSAGE QUALIFIER
 3796  036342  000137  036012                JMP     15$             ;GO DO NEXT PASS
 3797  036346  052737  000020  002426  37$:  BIS     #INOUTS,OPFLAG  ;SET MESSAGE QUALIFIER
 3798  036354  000772                        BR      36$
 3799  036356  052737  000200  002426  39$:  BIS     #REVSKS,OPFLAG  ;SET MESSAGE QUALIFIER
 3800  036364  000403                        BR      41$
 3801  036366  052737  000400  002426  40$:  BIS     #FWDSKS,OPFLAG  ;SET MESSAGE QUALIFIER
 3802  036374  012737  000002  002444  41$:  MOV     #BIT1,WRTSWI    ;SET READ REQUIRED
 3803  036402  000757                        BR      36$
 3804  036404  012737  000002  002440  60$:  MOV     #2,ERRSWI       ;INIT ERROR SWITCH
 3805  036412                        ENDSUB
                                      L10060:
   (3) 036412
   (3) 036412  104003                        EMT     C$ESUB
 3806  036414                                ESCAPE  T$T             ;EXIT TEST IF ERROR
   (3) 036414  104010                        EMT     C$ESCAPE
   (3) 036416  000054                        .WORD   L10057-.
 3807  036420  012737  000003  002444        MOV     #3,WRTSWI       ;SET FOR READ AND WRITE REQ.
 3808  036426  023727  002654  000004        CMP     PASCNT,#4       ;TEST IF PASS 4
 3809  036434  001004                        BNE     45$             ;NO - SKIP
 3810  036436  012737  002342  002446        MOV     #T33TBL+10,TBLSTR ;STORE MID POINT IN TABLE
 3811  036444  000410                        BR      48$             ;GO START PASS 4
 3812  036446  005037  002654          45$:  CLR     PASCNT          ;CLEAR TO PASS 0
 3813  036452  004737  017414                JSR     PC,SWAPHD       ;GO SWAP TO HEAD 1 OR END TEST
 3814  036456  035526                        T3300$                  ;ABORT RETURN
 3815  036460  012737  002332  002446        MOV     #T33TBL,TBLSTR  ;STORE START OF TABLE
 3816  036466  000137  035720          48$:  JMP     T3301$          ;GO DO HEAD 1
 3817  036472                        T3365$:
 3818  036472                        ENDTST
                                      L10057:
   (3) 036472
   (3) 036472  104001                        EMT     C$ETST
```



```
 3820                           .SBTTL  *TEST 19         **OVERWRITE
 3821  036474                   BGNTST                   ;TEST 19
   (3) 036474                                                              T19::
 3822  036474  012737  006702  002434        MOV     #P2T19E,ERHEAD  ;SET ERROR HEADER
 3823  036502  004737  017500                JSR     PC,CKBSVD       ;GO CHECK IF BAD SECTOR FILES VALID
 3824  036506  004737  015160                JSR     PC,TSTINT       ;INITIALIZE TEST
 3825  036512  004737  015176                JSR     PC,GSTATR       ;CLEAR DRIVE
 3826  036516  037600                        T3465$
 3827  036520  005037  002654                CLR     PASCNT          ;CLEAR PASS TO 0
 3828  036524  012705  177777                MOV     #-1,R5          ;SET R5
 3829  036530  005737  003062                TST     PASNUM          ;TEST IF FIRST PASS (QUICK VERIFY)
 3830  036534  001007                        BNE     1$              ;NO - SKIP
 3831  036536  032737  000001  013372        BIT     #ALLCYL,MISWIW  ;TEST IF USE ALL CYLINDERS
 3832  036544  001003                        BNE     1$              ;YES - SKIP
 3833  036546  012705  177754                MOV     #-20.,R5        ;ELSE SET R5 TO NEG 20
 3834  036552  000402                        BR      9$              ;SKIP
 3835  036554  012705  177774          1$:   MOV     #-4,R5          ;SET FOR NEXT ENTRY
 3836  036560  012701  002332          9$:   MOV     #T33TBL,R1      ;GET ADDRESS OF WORK TABLE
 3837  036564  012703  000010                MOV     #10,R3          ;SET CLEAR COUNT
 3838  036570  013721  013374          2$:   MOV     LOLIMW,(R1)+    ;CLEAR LOCATIONS TO LOLIMIT
 3839  036574  005303                        DEC     R3              ;DEC COUNT
 3840  036576  001374                        BNE     2$              ;LOOP UNTIL 0
 3841  036600  113737  013376  002334        MOVB    HILIMW,T33TBL+2 ;INSERT HILIMIT
 3842  036606  113737  013376  002340        MOVB    HILIMW,T33TBL+6 ;INTO APPROPRIATE LOCATIONS
 3843  036614  113737  013376  002344        MOVB    HILIMW,T33TBL+12
 3844  036622  005205                T3400$: INC     R5              ;BUMP R5
 3845  036624  032737  000001  013372        BIT     #ALLCYL,MISWIW  ;TEST IF USE ALL CYLINDERS
 3846  036632  001022                        BNE     5$              ;YES - SKIP
 3847  036634  005737  003062                TST     PASNUM          ;TEST IF FIRST PASS (QUICK VERIFY)
 3848  036640  001003                        BNE     3$              ;NO - SKIP
 3849  036642  062705  000023                ADD     #19.,R5         ;ELSE BUMP CYLINDER POINTER BY 19
 3850  036646  000402                        BR      6$              ;SKIP
 3851  036650  062705  000003          3$:   ADD     #3,R5           ;BUMP CYLINDER POINTER BY 3
 3852  036654  020527  000051          6$:   CMP     R5,#41.         ;TEST IF PAST VALID TABLE
 3853  036660  103005                        BHIS    4$              ;YES - SKIP
 3854  036662  116503  002352                MOVB    CYLTBL(R5),R3   ;GET NEXT TABLE ENTRY
 3855  036666  042703  177400                BIC     #177400,R3      ;CLEAR UPPER BYTE
 3856  036672  001011                        BNE     8$              ;SKIP IF NOT 0
 3857  036674  000137  037600          4$:   JMP     T3465$          ;EXIT TEST
 3858  036700  005705                5$:   TST     R5              ;TEST IF R5 0
 3859  036702  001001                        BNE     7$              ;NO - SKIP
 3860  036704  005205                        INC     R5              ;ELSE BUMP R5 AGAIN
 3861  036706  022705  000377          7$:   CMP     #255.,R5        ;TEST IF ALL CYLINDERS USED
 3862  036712  001770                        BEQ     4$              ;YES - EXIT TEST
 3863  036714  010503                        MOV     R5,R3           ;USE R5 AS NEXT CYLINDER
 3864  036716  020337  013374          8$:   CMP     R3,LOLIMW       ;TEST IF PAST LO LIMIT
 3865  036722  103737                        BLO     T3400$          ;YES - SKIP
 3866  036724  020337  013376                CMP     R3,HILIMW       ;TEST IF PAST HILIMIT
 3867  036730  101334                        BHI     T3400$          ;YES - SKIP
 3868  036732  012704  002332                MOV     #T33TBL,R4      ;GET ADDRESS OF SEEK TABLE
 3869  036736  110364  000001                MOVB    R3,1(R4)        ;INSERT CC IN APPROPRIATE TABLE
 3870  036742  110364  000003                MOVB    R3,3(R4)        ; LOCATIONS FOR TEST SEEK SEQUENCE
 3871  036746  110364  000005                MOVB    R3,5(R4)
 3872  036752  110364  000007                MOVB    R3,7(R4)
 3873  036756  110364  000011                MOVB    R3,11(R4)
 3874  036762  110364  000013                MOVB    R3,13(R4)
```

```
     036766  010437  002446                    MOV     R4,TBLSTR          ;STORE TABLE ADDRESS
     036772  004737  017370                    JSR     PC,CHOSHD          ;GO CHOSE HEAD
     036776                           T3401$:
     036776                           BGNSUB
 (3) 036776                                                                         T19.1:
 (3) 036776  104002                            EMT     C$BSUB
     037000  042737  003760  002426            BIC     #MQUALS,OPFLAG     ;CLEAR ALL MESSAGE QUALIFIERS
     037006  005737  002654                    TST     PASCNT             ;TEST IF PASS 0
     037012  001414                            BEQ     11$                ;YES - SKIP
     037014  023727  002654  000003            CMP     PASCNT,#3          ;TEST IF PASS 3
     037022  001404                            BEQ     10$                ;YES - SKIP
     037024  002407                            BLT     11$                ;CHECK IF LESS THAN 3, IF YES CLEAR TO 0
     037026  012737  000003  002654            MOV     #3,PASCNT          ;ELSE SET TO 3
     037034  052737  000020  002426   10$:     BIS     #INOUTS,OPFLAG     ;SET MESSAGE QUAL
     037042  000405                            BR      12$                ;SKIP
     037044  005037  002654           11$:     CLR     PASCNT             ;SET PASS COUNT TO 0
     037050  052737  000040  002426            BIS     #OUTINS,OPFLAG     ;SET MESSAGE QUAL
     037056  012737  000003  002444   12$:     MOV     #3,WRTSWI          ;SET READ AND WRITE SWITCH
     037064  013704  002446                    MOV     TBLSTR,R4          ;GET STORED TABLE ADDRESS
     037070  005037  002536           15$:     CLR     DESSEC             ;CLEAR TO SECTOR 0
     037074  112437  002524                    MOVB    (R4)+,NEWCYL       ;GET NEXT TABLE ENTRY
     037100  004737  016070                    JSR     PC,XSEEK           ;DO SEEK
     037104  037512                                    60$
     037106  012701  005670                    MOV     #3000.,R1          ;SET WAIT COUNT FOR 300 MS
     037112  004737  020650                    JSR     PC,RDYWAIT         ;WAIT FOR READY
     037116  037512                                    60$
     037120  112437  002524                    MOVB    (R4)+,NEWCYL       ;GET NEXT TABLE ENTRY
     037124  004737  016070                    JSR     PC,XSEEK           ;DO SEEK
     037130  037512                                    60$
     037132  012701  005670                    MOV     #3000.,R1          ;SET WAIT COUNT FOR 300 MS
     037136  004737  020650                    JSR     PC,RDYWAIT         ;WAIT FOR READY
     037142  037512                                    60$
     037144  004737  021244                    JSR     PC,VERPOS          ;VERIFY POSITION
     037150  037512                                    60$
     037152  004737  023074           16$:     JSR     PC,BSCHK           ;CHECK FOR BAD SECTOR
     037156  037326                                    32$                ;"YES" RETURN
     037160  005737  002654                    TST     PASCNT             ;TEST IF PASS 0
     037164  001407                            BEQ     17$                ;YES - SKIP
     037166  022737  000003  002654            CMP     #3,PASCNT          ;TEST IF PASS 3
     037174  001403                            BEQ     17$                ;YES - SKIP
     037176  005037  037216                    CLR     25$                ;ELSE CLEAR DATA PATTERN SELECTOR
     037202  000403                            BR      18$
     037204  012737  000010  037216   17$:     MOV     #8.,25$            ;SET DATA PATTERN SELECTOR TO 8
     037212  004537  021726           18$:     JSR     R5,DATGEN          ;GO GENERATE DATA
     037216  000000                   25$:     .WORD   0
     037220  032737  000001  002444            BIT     #BIT0,WRTSWI       ;TEST IF WRITE THIS PASS
     037226  001425                            BEQ     29$                ;NO - SKIP
     037230  004737  022354                    JSR     PC,XWRITE          ;DO WRITE
     037234  037512                                    60$
     037236  005237  002536                    INC     DESSEC             ;INC SECTOR
     037242  022737  000050  002536            CMP     #40.,DESSEC        ;TEST IF ALL SECTORS USED
     037250  001340                            BNE     16$                ;NO - SKIP
     037252  042737  000060  002426            BIC     #INOUTS!OUTINS,OPFLAG ;CLEAR QUALIFIERS
     037260  042737  000001  002444            BIC     #BIT0,WRTSWI       ;CLEAR WRITE REQUIRED SWITCH
     037266  052737  000100  002426            BIS     #FOLWRT,OPFLAG     ;SET FOLLOWING WRITE QUALIFIER
     037274  005037  002536                    CLR     DESSEC             ;CLEAR TO SECTOR 0
```



```
     037300  000724                            BR      16$                ;SKIP
     037302  032737  000002  002444   29$:     BIT     #BIT1,WRTSWI       ;TEST IF READ THIS PASS
     037310  001414                            BEQ     33$                ;NO - SKIP
     037312  004737  022414           31$:     JSR     PC,XREAD           ;ELSE DO READ
     037316  037512                                    60$
     037320  004737  022066                    JSR     PC,DATCOM          ;COMPARE DATA
     037324  037512                                    60$
     037326  005237  002536           32$:     INC     DESSEC             ;BUMP SECTOR
     037332  022737  000050  002536            CMP     #40.,DESSEC        ;TEST IF ALL SECTORS USED
     037340  001304                            BNE     16$                ;NO - LOOP
     037342  005037  002536           33$:     CLR     DESSEC             ;CLEAR DESIRED SECTOR
     037346  005037  002444                    CLR     WRTSWI             ;CLEAR WRITE/READ SWITCH
     037352  005237  002654                    INC     PASCNT             ;BUMP PASS COUNT
     037356  042737  003760  002426            BIC     #MQUALS,OPFLAG     ;CLEAR ALL QUALIFIERS
     037364  023727  002654  000003            CMP     PASCNT,#3          ;TEST IS PASS 3
     037372  001447                            BEQ     60$                ;YES - SKIP
     037374  023727  002654  000006            CMP     PASCNT,#6          ;TEST IF PASS 6
     037402  001443                            BEQ     60$                ;YES - SKIP
     037404  023727  002654  000001            CMP     PASCNT,#1          ;TEST IF PASS 1
     037412  001424                            BEQ     39$                ;YES - SKIP
     037414  023727  002654  000004            CMP     PASCNT,#4          ;TEST IF PASS 4
     037422  001424                            BEQ     40$                ;YES - SKIP
     037424  012737  000002  002444            MOV     #BIT1,WRTSWI       ;SET WRITE REQUIRED BIT
     037432  023727  002654  000002            CMP     PASCNT,#2          ;TEST IF PASS 2
     037440  001405                            BEQ     37$                ;YES - SKIP
     037442  052737  001000  002426            BIS     #REVSKO,OPFLAG     ;SET REVERSE QUALIFIER
     037450  000137  037070           36$:     JMP     15$                ;GO DO NEXT PASS
     037454  052737  002000  002426   37$:     BIS     #FWDSKO,OPFLAG     ;SET FWD QUALIFIER
     037462  000772                            BR      36$                ;GO DO NEXT PASS
     037464  052737  000020  002426   39$:     BIS     #INOUTS,OPFLAG     ;SET QUALIFIER
     037472  000403                            BR      41$                ;SKIP
     037474  052737  000040  002426   40$:     BIS     #OUTINS,OPFLAG     ;SET MESSAGE QUALIFIER
     037502  012737  000001  002444   41$:     MOV     #BIT0,WRTSWI       ;SET WRITE REQUIRED BIT
     037510  000757                            BR      36$                ;GO DO NEXT PASS
     037512  012737  000002  002440   60$:     MOV     #2,ERRSWI          ;INIT ERROR SWITCH
     037520                           ENDSUB
 (3) 037520                           L10062:
 (3) 037520  104003                            EMT     C$ESUB
     037522                                    ESCAPE  TST                ;EXIT TEST IF ERROR
 (3) 037522  104010                            EMT     C$ESCAPE
 (3) 037524  000054                            .WORD   L10061-.
     037526  012737  000003  002444            MOV     #3,WRTSWI          ;SET FOR READ AND WRITE REQ.
     037534  023727  002654  000003            CMP     PASCNT,#3          ;TEST IF PASS 3
     037542  001004                            BNE     45$                ;NO - SKIP
     037544  012737  002340  002446            MOV     #T33TBL+6,TBLSTR ;STORE MID POINT IN TABLE
     037552  000410                            BR      48$                ;GO START PASS 4
     037554  005037  002654           45$:     CLR     PASCNT             ;CLEAR TO PASS 0
     037560  004737  017414                    JSR     PC,SWAPHD          ;GO SWAP TO HEAD ONE OR ABORT TEST
     037564  036622                                    T3400$             ;ABORT RETURN
     037566  012737  002332  002446            MOV     #T33TBL,TBLSTR     ;STORE START OF TABLE
     037574  000137  036776           48$:     JMP     T3401$             ;GO DO HEAD 1
     037600                           T3465$:
     037600                           ENDTST
 (3) 037600                           L10061:
 (3) 037600  104001                            EMT     C$ETST
     037602                           ENDMOD
```

```
 3980  037602                              BGNMOD  HRDPRM
 3981  037602                              BGNHRD
  (3)  037602  000025                              .WORD L10063-L$HARD/2
 3982  037604                              GPRML   CNTYPE,CNT,1,YES
  (4)  037604  004130                              .WORD   T$CODE
  (4)  037606  037720                              .WORD   CNTYPE
  (4)  037610  000001                              .WORD   1
 3983  037612                              GPRMA   CSRMSG,CSR,0,160000,177776,YES
  (4)  037612  000031                              .WORD   T$CODE
  (4)  037614  037656                              .WORD   CSRMSG
  (4)  037616  160000                              .WORD   T$LOLIM
  (4)  037620  177776                              .WORD   T$HILIM
 3984  037622                              GPRMA   VECMSG,VECT,0,0,776,YES
  (4)  037622  001031                              .WORD   T$CODE
  (4)  037624  037672                              .WORD   VECMSG
  (4)  037626  000000                              .WORD   T$LOLIM
  (4)  037630  000776                              .WORD   T$HILIM
 3985  037632                              GPRMD   BRMSG,PRIOR,0,340,0,7,YES
  (4)  037632  002032                              .WORD   T$CODE
  (4)  037634  037701                              .WORD   BRMSG
  (4)  037636  000340                              .WORD   340
  (4)  037640  000000                              .WORD   T$LOLIM
  (4)  037642  000007                              .WORD   T$HILIM
 3986  037644                              GPRMD   DRMSG,DRSD,0,3400,0,7,YES
  (4)  037644  003032                              .WORD   T$CODE
  (4)  037646  037712                              .WORD   DRMSG
  (4)  037650  003400                              .WORD   3400
  (4)  037652  000000                              .WORD   T$LOLIM
  (4)  037654  000007                              .WORD   T$HILIM
 3987
 3988  037656                              ENDHRD
  (2)                                              .EVEN
  (3)  037656                              L10063:
 3989
 3990  037656  052502  020123  042101      CSRMSG: .ASCIZ  /BUS ADDRESS/
       037664  051104  051505  000123
 3991  037672  042526  052103  051117      VECMSG: .ASCIZ  /VECTOR/
       037700     000
 3992  037701     102  020122  042514      BRMSG:  .ASCIZ  /BR LEVEL/
       037706  042526  000114
 3993  037712  051104  053111  000105      DRMSG:  .ASCIZ  /DRIVE/
 3994  037720  046122  030461     000      CNTYPE: .ASCIZ  /RL11/
 3995  037725                              ENDMOD
 3996          037726                              .EVEN
 3997
 3998  037726                              BGNMOD  SFTPRM
 3999  037726                              BGNSFT
  (3)  037726  000061                              .WORD L10064-L$SOFT/2
 4000
 4002  037730                              GPRML   CYLQ,MISWI,1,YES
  (4)  037730  000130                              .WORD   T$CODE
  (4)  037732  040072                              .WORD   CYLQ
  (4)  037734  000001                              .WORD   1
 4003  037736                              GPRML   SECQ,MISWI,2,YES
  (4)  037736  000130                              .WORD   T$CODE
  (4)  037740  040114                              .WORD   SECQ
```



```
  (4)  037742  000002                              .WORD   2
 4009  037744                              GPRML   MANQ,MISWI,100000,YES
  (4)  037744  000130                              .WORD   T$CODE
  (4)  037746  040134                              .WORD   MANQ
  (4)  037750  100000                              .WORD   100000
 4010
 4012  037752                              GPRML   LOLIMQ,MISWI,40000,YES
  (4)  037752  000130                              .WORD   T$CODE
  (4)  037754  040176                              .WORD   LOLIMQ
  (4)  037756  040000                              .WORD   40000
 4013  037760                              XFERF   1$
  (5)  037760  006044                              .WORD   T$CODE
 4014  037762                              GPRMD   LIMVAL,LOLIM,D,255.,0,253.,YES
  (4)  037762  001052                              .WORD   T$CODE
  (4)  037764  040217                              .WORD   LIMVAL
  (4)  037766  000377                              .WORD   255.
  (4)  037770  000000                              .WORD   T$LOLIM
  (4)  037772  000375                              .WORD   T$HILIM
 4015  037774                      1$:     GPRML   HILIMQ,MISWI,20000,YES
  (4)  037774  000130                              .WORD   T$CODE
  (4)  037776  040233                              .WORD   HILIMQ
  (4)  040000  020000                              .WORD   20000
 4016  040002                              XFERF   2$
  (5)  040002  006044                              .WORD   T$CODE
 4017  040004                              GPRMD   LIMVAL,HILIM,D,255.,0,255.,YES
  (4)  040004  002052                              .WORD   T$CODE
  (4)  040006  040217                              .WORD   LIMVAL
  (4)  040010  000377                              .WORD   255.
  (4)  040012  000000                              .WORD   T$LOLIM
  (4)  040014  000377                              .WORD   T$HILIM
 4018  040016                      2$:     GPRML   HEADQ,MISWI,10000,YES
  (4)  040016  000130                              .WORD   T$CODE
  (4)  040020  040254                              .WORD   HEADQ
  (4)  040022  010000                              .WORD   10000
 4019  040024                              XFERF   3$
  (5)  040024  006044                              .WORD   T$CODE
 4020  040026                              GPRMD   HEADV,HEAD,D,17,0,1,YES
  (4)  040026  003052                              .WORD   T$CODE
  (4)  040030  040301                              .WORD   HEADV
  (4)  040032  000017                              .WORD   17
  (4)  040034  000000                              .WORD   T$LOLIM
  (4)  040036  000001                              .WORD   T$HILIM
 4022  040040                      3$:     GPRMD   ERLIMQ,ERLIM,D,377,0,377,YES
  (4)  040040  004052                              .WORD   T$CODE
  (4)  040042  040332                              .WORD   ERLIMQ
  (4)  040044  000377                              .WORD   377
  (4)  040046  000000                              .WORD   T$LOLIM
  (4)  040050  000377                              .WORD   T$HILIM
 4024  040052                              GPRMD   DCLIMQ,DCLIM,D,377,1,377,YES
  (4)  040052  005052                              .WORD   T$CODE
  (4)  040054  040410                              .WORD   DCLIMQ
  (4)  040056  000377                              .WORD   377
  (4)  040060  000001                              .WORD   T$LOLIM
  (4)  040062  000377                              .WORD   T$HILIM
 4026  040064                              GPRML   AUTOQ,MISWI,20,YES
  (4)  040064  000130                              .WORD   T$CODE
```

```
(4)  040066  040356                        .WORD   AUTOQ
(4)  040070  000020                        .WORD   20
4027 040072                        ENDSFT  .EVEN
(2)
(3)  040072                        L10064:
4028
4030 040072  051525  020105  046101  CYLQ:   .ASCIZ  /USE ALL CYLINDERS/
     040100  020114  054503  044514
     040106  042440  051105  040123
4031 040114  051525  020105  046101  SECQ:   .ASCIZ  /USE ALL SECTORS/
     040120  020114  042523  052103
     040126  051117  000123
4037 040134  052105  041505  052125  MANQ:   .ASCIZ  /EXECUTE MANUAL INTERVENTION TESTS/
     040142  052105  040516  052516
     040150  046101  044440  052116
     040156  051105  042526  052116
     040164  047511  020116  042524
     040172  052123  000123
4039 040176  047514  042527  020122  LOLIMQ: .ASCIZ  /LOWER SEEK LIMIT/
     040204  042523  045505  046040
     040212  046511  052111     000
4040 040217      165  052116  051105  LIMVAL: .ASCIZ  /ENTER VALUE/
     040224  053440  046101  042525
     040232      000
4041 040233      125  050120  051105  HILIMQ: .ASCIZ  /UPPER SEEK LIMIT/
     040240  051440  042505  020113
     040246  044514  044515  000124
4042 040254  051525  020105  047117  HEADQ:  .ASCIZ  /USE ONLY ONE SURFACE/
     040262  054505  047440  042516
     040270  051440  051125  040506
     040276  042503     060
4043 040301      123  042520  044503  HEADV:  .ASCIZ  /SPECIFY SURFACE (0 OR 1)/
     040306  054506  051440  051125
     040314  040506  042503  020440
     040322  020060  051117  030440
     040330  000051
4045 040332  050123  041505  043111  ERLIMQ: .ASCIZ  /SPECIFY ERROR LIMIT/
     040340  020131  051105  047522
     040346  020122  044514  044515
     040354  000124
4046 040356  051104  050117  042040  AUTOQ:  .ASCIZ  /DROP DRIVE IF NO RESPONSE/
     040364  042522  042526  044440
     040372  020106  047516  051040
     040400  051505  047520  051516
     040406  000105
4048 040410  040504  040524  041440  DCLIMQ: .ASCIZ  /DATA COMPARE ERROR LIMIT/
     040416  046517  040520  042522
     040424  042440  051122  051117
     040432  046040  046511  052111
     040440     000
4050           040442                        .EVEN
4051 040442                        ENDMOD
4052
4063
4065
4066           040514                        .=40514
```



```
4067
4068                               ;AREA RESERVED AS PATCH AREA FOR DIAGNOSTIC
4069                               ;.=40514 WAS SELECTED AS "LASTAD" TO PROVIDE APT TO LSI-11 COMPATIBILITY.
4070                               ;BIT 7 OF "LASTAD" MUST BE CLEARED TO ACHIEVE A VALID MAILBOX ADDRESS
4071                               ;WHEN RUNNING ON THE LSI-11 UNDER APT.
4073
4074 040514                        LASTAD
(2)                                       .EVEN
(3)  040514                        L$LAST::
```

```
                                .SBTTL  DIAGNOSTIC SUPERVISOR -- LOW CORE SET UP
14947           000000          .WORD   0               ;SPACE FOR USER POOL POINTER
14948           000000          .WORD   0               ;SIZE
14949   071414  000000          .WORD   0               ;CHECKSUM (NOT CURRENTLY USED)
14950   071316  000000          .WORD   0               ;SIZE OF H.W. PTAB. ALLOCATION
14951           071322          END.SUPV=.+2
14952           000200          .END 200
```



```
ABOFLA  041040 G        BIT7   = 000200 G       CONHNG= 000004 G       C$KWON= 000034         DRSELT= 000004
ABOPAS  040756 G        BIT8   = 000400 G       CONTCL  067672 G       C$LOOP= 000100         DRSET = 000010
ABO.FM  043320          BIT9   = 001000 G       CONTIN  013650         C$MANI= 000051         DRVCNT  002516
AFMID   002632          BLD.HW  046202          COSTAT= 000040         C$MSG = 000023         DRVERR= 040000
AFMIDU  002634          BLOCK   063614          COUNT   002656         C$PNTB= 000014         DRVNAM  005633
AFSI    040546 G        BRMSG   037701          CRDYMS= 000200         C$PNTF= 000017         DRVNAV  005640
ALLCYL= 000001          BSCHK   023074          CRLF    060172         C$PNTS= 000016         DSESTA= 000400
ALLOC   061460          BSFLAG  002442          CSNAM   005733         C$PNTX= 000015         DSMSK = 001400
ALLSEC= 000002          BSPVAL  003074          CSR    = 000000         C$POIN= 000040         DSPCOD  013406 G
AMYERR= 100000          BSNSTR  007442          CSRMSG  037656         C$QIO = 000377         DUNIT.  040762 G
APT.ER  042450          BYPSNM  007353          CURCYL  002526         C$RDBU= 000007         DVC.FT  054042
ARMID   002636          B$AAB   047604          CURR.S  040522 G       C$REFG= 000050         D$AAG   054746
ARMIDU  002640          B$AAF   047516          CURR.T  040524 G       C$REQT= 000045         D$AAH   054764
ASSEMB= 000010          CAFDT   010510          CYLQ    040072         C$RESE= 000033         D$AAI   057532
AUTOQ   040356          CALLPC= 000022          CYLTBL  002352         C$REVI= 000002         D$AAJ   057536
AUTOSZ= 000020          CALLPS= 000024          CYLUP  = 000004         C$RPT = 000025         D$AAK   057554
A$AAV   045316          CALLSP= 000026          CYLWD   007340         C$SEFG= 000047         D$AAL   057572
A$AAW   045332          CALLTC= 000030          C$AAD   053062         C$SPRI= 000041         D$AAM   057602
A$AAX   045344          CAL.CL  066202          C$AAE   053074         C$SVEC= 000037         EF.CON= 000036 G
A$AAY   045352          CAL.TI  066240 G        C$AAK   054072         C$TPRI= 000013         EF.NEW= 000035 G
A$AAZ   045366          CAMSK  = 077600         C$AAL   054236         C$UNBU= 000031         EF.PWR= 000034 G
A$ABA   045376          CCYLUP  010477          C$ABRT= 000021         C$WTM = 000026         EF.RES= 000037 G
BADADD= 004000          CHKLUP  047620          C$ADR = 000020         C$WTU = 000027         EF.STA= 000040 G
BAMSK  = 000060         CHKSTR  062072          C$AU  = 000054         C10MS   010456         EF01   = 000001 G
BAMAM   005740          CHKTYV  060110          C$BRK = 000022         C5SEC   010522         EF02   = 000002 G
BASADD  005622          CHK.MA  045760          C$BSEG= 000004         C500MS  010471         EF03   = 000003 G
BELL    010377          CHK.PC  053110          C$BSUB= 000002         DANAM   005745         EF04   = 000004 G
BGM.SU= 040514          CHK.SW  042150          C$BUFF= 000030         DATACM= 000001         EF05   = 000005 G
BHSTAT= 000010          CHOSHD  017370          C$CEFG= 000046         DATCOM  022066         EF06   = 000006 G
BINMSG  057770          CHRCNT  061342          C$CLEA= 000012         DATGEN  021726         EF07   = 000007 G
BIT0   = 000001 G       CH.FLA  045466          C$CLP1= 000006         DCKERR= 004000         EF08   = 000010 G
BIT00  = 000001 G       CH.PAS  045504          C$CVEC= 000036         DCLIM  = 000012        EF09   = 000011 G
BIT01  = 000002 G       CKBSYD  017500          C$DCLN= 000044         DCLIMQ  040410         EF10   = 000012 G
BIT02  = 000004 G       CKDATA= 000102          C$DODU= 000053         DCLIMW  013404         EF11   = 000013 G
BIT03  = 000010 G       CKERLM  014634          C$DRPT= 000024         DECMSG  060004         EF12   = 000014 G
BIT04  = 000020 G       CLEAR.  047102          C$DU   = 000055        DESDIF  002530         EF13   = 000015 G
BIT05  = 000040 G       CLKACC  040754 G        C$EDIT= 000002         DESHD   002534         EF14   = 000016 G
BIT06  = 000100 G       CLKBFR  066204          C$ERDF= 000002         DESSEC  002536         EF15   = 000017 G
BIT07  = 000200 G       CLKCNT  040752 G        C$ERHR= 000003         DESSGN  002532         EF16   = 000020 G
BIT08  = 000400 G       CLKJUM  066610 G        C$ERSF= 000001         DEV.CO  040526 G       EMT.TR  041044 G
BIT09  = 001000 G       CLKRES  067612 G        C$ERSO= 000004         DIAGMC= 000000         END.OP  047370
BIT1   = 000002 G       CLKSER  067746 G        C$ESCA= 000010         DIAG.T  041046 G       END.SU= 071322
BIT10  = 002000 G       CLKSOM  041012 G        C$ESEG= 000005         DIPAUG  002520         ENVIRO  040566 G
BIT11  = 004000 G       CLK.SE  045562          C$ESUB= 000003         DIPWD   007314         EOP.CH  067770 G
BIT12  = 010000 G       CLNCOD  014444 G        C$ETST= 000001         DIRBIT= 000004         EOP.FM  043334
BIT13  = 020000 G       CLRPAR  024474          C$EXIT= 000032         DIRMSK= 077600         EOP.IN  045500
BIT14  = 040000 G       CLR.MA  046036          C$GMAN= 000043         DLTERR= 010000         ERHEAD  002434
BIT15  = 100000 G       CNT    = 000010         C$GPHR= 000042         DONE    002430         ERLIM  = 000010
BIT2   = 000004 G       CNTYPE  037720          C$GPRI= 000040         DPDVD   070456 G       ERLIMQ  040332
BIT3   = 000010 G       CNVT    064260          C$GTIM= 000052         DPMUL   070344 G       ERLIMW  013402
BIT4   = 000020 G       COMMAN  040564 G        C$INIT= 000011         DRDYMS= 000001         ERRCNT  002662
BIT5   = 000040 G       COMMTA  064074          C$INLP= 000020         DRMSG   037712         ERRFOR  054314
BIT6   = 000100 G       COMPOP= 007777          C$KWOF= 000035         DRSB   = 000006        ERRHAN  053114
```

ERRPOI	002660	
ERRSWI	002440	
ERRVEC	002652	
ERR.HR	054052	
ERR.MU	040516	G
ERR.SF	054056	
ERR1	011554	G
ERR1FO	054400	
ERR10	013144	G
ERR2	011622	G
ERR3	011670	G
ERR4	011736	G
ERR5	012006	G
ERR6	012056	G
ERR7	012734	G
ERR8	013004	G
ERR9	013100	G
[illegible]C.PC	053106	
[illegible].COU	040520	G
[illegible]ACYL	002646	
[illegible]MCYL	002644	
[illegible]OCYL	002642	
[illegible]DY[illegible]	002650	
[illegible]FIL	003272	
FILL	060640	
FILL.C	000204	G
FLAGS	040560	G
FLAGS1	040562	G
FLAGTA	064012	
FLAG.I	045546	
FLA.SE	063760	
FLG.MA	045506	
FMTOP1	010532	
FMTOP2	010561	
FMTOP3	010603	
FMT1	010624	
FMT1.1	010631	
FMT11	011050	
FMT12	011056	
FMT13	011064	
FMT14	011130	
FMT15	011162	
FMT16	011216	
FMT17	011227	
FMT18	011251	
FMT19	011303	
FMT2	010640	
FMT20	011340	
FMT21	011370	
FMT22	011413	
FMT23	011447	
FMT24	011463	
FMT25	011470	
FMT26	011500	
FMT27	011524	
FMT28	011543	
FMT3	010643	
FMT4	010646	
FMT5	010657	
FMT6	010677	
FMT7	010741	
FMT8	011011	
FMT9	011043	
FOLWRT=	000100	
FORM.T	054410	
FREE	061716	
FRMWD	007345	
FWDSKO=	002000	
FWDSKS=	000400	
F$AU =	000015	
F$BGN =	000040	
F$CLEA=	000007	
F$DU =	000016	
F$END =	000041	
F$HARD=	000004	
F$HW =	000013	
F$INIT=	000006	
F$JMP =	000050	
F$MOD =	000000	
F$MSG =	000011	
F$PWR =	000017	
F$RPT =	000012	
F$SEG =	000003	
F$SOFT=	000005	
F$SRV =	000010	
F$SUB =	000002	
F$SW =	000014	
F$TEST=	000001	
GARBAG	061344	
GETCHR	060050	
GETCMN	063434	
GETPAR	055126	
GETPOS	021116	
GETSTA=	000003	
GETSWI	062430	
GET.TW	062200	
GLBDAT	002122	G
GLBEQA	002122	G
GLBERR	011554	G
GLBSUB	014570	G
GLBTXT	004744	G
GSTAT	015226	
GSTATC	015212	
GSTATG	015236	
GSTATR	015176	
GTSTAT=	000104	
G$EXCP=	000400	
G$HILI=	000002	
G$LOLI=	000001	
G$NO =	000000	
G$OFFS=	000400	
G$OFSI=	000376	
G$PRMA=	000001	
G$PRMD=	000002	
G$PRML=	000000	
G$RADA=	000140	
G$RADB=	000000	
G$RADD=	000040	
G$RADF=	000200	
G$RADL=	000120	
G$RADO=	000020	
G$RADT=	000100	
G$XFER=	000004	
G$YES =	000010	
HADONE	002432	
HCESTA=	040000	
HCORED	045256	
HCOREQ	045166	
HCORET	041002	G
HCRCER=	004000	
HC.ADR	040552	G
HC.DEF	040544	G
HC.DIA	040542	G
HDALIG=	000010	
HDCYL =	077600	
HDHSEL=	000100	
HDMOVF	007172	
HDRCMP=	000002	
HDR40 =	100000	
HDSEC =	000077	
HDSEL =	000020	
HDWD	007327	
HDWRD1	002474	
HDWRD2	002476	
HDWRD3	002500	
HEAD =	000006	
HEADLM=	010000	
HEADQ	040254	
HEADV	040301	
HEADW	013400	
HERTZ.	045126	
HFIN	002612	
HFINU	002614	
HFOUT	002616	
HFOUTU	002620	
HICYL =	020000	
HILIM =	000004	
HILIMQ	040233	
HILIMW	013376	
HNFERR=	010000	
HOLDSP=	000020	
HOSTAT=	000020	
HPTCOD	013354	G
HRDPRM	037602	G
HRDWTS	024524	G
HRIN	002622	
HRINU	002624	
HROUT	002626	
HROUTU	002630	
HSMSK =	000100	
HSSTAT=	000100	
HW.ADR	040550	G
H$AAB	064606	
IBUFF	003466	
ININIT	040772	G
INITCO	013456	G
INITIA	060020	
INIT.M	046104	
INIT.R	040606	G
INOUTS=	000020	
INPUTA	060746	
INTEBL=	000100	
INTFOR	054244	
INTHLR	014576	G
INVAL.	045052	
INVINT	054102	
INV.SW	042104	
IN.SUF	047054	
I$AU =	000041	
I$CLN =	000041	
I$DU =	000041	
I$HRD =	000041	
I$INIT=	000041	
I$MOD =	000041	
I$MSG =	000041	
I$PWR =	000041	
I$RPT =	000041	
I$SEG =	000041	
I$SFT =	000041	
I$SRV =	000041	
I$SUB =	000041	
I$TST =	000041	
J$JMP =	000167	
KBPTR	040624	G
KBUF	040626	G
LABACF	007142	
LABACR	007156	
LABEXP	007051	
LABHCF	007112	
LABHCR	007126	
LABIN	007026	
LABMID	007034	
LABOCF	007062	
LABOCR	007076	
LABOUT	007043	
LAB1	005757	
LAB2	005772	
LIMVAL	040217	
LINE.F	041042	G
LOAD.F	045502	
LOCERR	003066	
LOCYL =	040000	
LOGMSG	060012	
LOLIM =	000002	
LOLIMQ	040176	
LOLIMW	013374	
LPBFR	040622	G
LPCNTR	040620	G
LPT.AD	045144	
LPT.RE	045140	
LSI.RE	045134	
LUP	066106	
LUP.AD	053112	
L$APT	002036	G
L$AUT	002074	G
L$CCP	002106	G
L$CLEA	014444	G
L$CO	002032	G
L$DEPO	002011	G
L$DESC	002102	G
L$DEVP	002064	G
L$DISP	013410	G
L$DR	002112	G
L$DRCT	002070	G
L$DRS	002072	G
L$DRST	002112	G
L$DTP	002040	G
L$DU	014564	G
L$DUT	002076	G
L$DVTY	002114	G
L$EF	002056	G
L$EFLG	002034	G
L$EXP1	002042	G
L$EXP2	002044	G
L$EXP3	002046	G
L$HARD	037604	G
L$HPCP	002016	G
L$HPTP	002022	G
L$HW	013356	G
L$ICP	002104	G
L$INIT	013456	G
L$LADP	002026	G
L$LAST	040514	G
L$MREV	002050	G
L$NAME	002000	G

L$REPP	002066	G
L$REV	002010	G
L$SOFT	037730	G
L$SPC	002062	G
L$SPCP	002020	G
L$SPTP	002024	G
L$STA	002030	G
L$SW	013372	G
L$TIML	002014	G
L$TIMU	002054	G
L$TIM1	002052	G
L$TSTI	002100	G
L$UNIT	002012	G
L.BA	002460	
L.CLK.	045112	
L.CS	002456	
L.DA	002462	
L.MP	002464	
L10000	011620	
L10001	011666	
L10002	011734	
L10003	012004	
L10004	012054	
L10005	012732	
L10006	013002	
L10007	013076	
L10010	013142	
L10011	013352	
L10012	013370	
L10013	013406	
L10014	014442	
L10015	014562	
L10016	014566	
L10017	014632	
L10020	025044	
L10021	025010	
L10022	025334	
L10023	025300	
L10024	025546	
L10025	025530	
L10026	025756	
L10027	025754	
L10030	026166	
L10031	026164	
L10032	026412	
L10033	026410	
L10034	026612	
L10035	026574	
L10036	027040	
L10037	027036	
L10040	027334	
L10041	027264	
L10042	027632	
L10043	027574	
L10044	030130	
L10045	030076	
L10046	031676	
L10047	032406	
L10050	032622	
L10051	032546	
L10052	033346	
L10053	034370	
L10054	034310	
L10055	035362	
L10056	035242	
L10057	036472	
L10060	036412	
L10061	037600	
L10062	037520	
L10063	037656	
L10064	040072	
MAJ.IN	040576	G
MAJ.LO	066206	
MAJ.US	040600	G
MANQ	040134	
MAN.TI	001244	
MAPROX	007016	
MAP16	070714	G
MASK.B	047616	
MASK.W	047614	
MBADAD	005462	
MBADSF	005503	
MBSETO=	000001	
MCERR	007554	
MCONHN	006115	
MCYLOC	010134	
MCYLUP	005204	
MDATCP	005046	
MDCRC	007576	
MDHEDR	002000	G
MDLT	007623	
MDRDY	007543	
MDRERR	007665	
MDRRES	006032	
MDRVST	007713	
MDSERR	007676	
MEM.SI	045154	
MERRS	010370	
MEXERS	010325	
MFLERR	010066	
MFMTER	005533	
MFOLWR	005260	
MFWDSK	005337	
MFWSKO	005372	
MGTSTA	005032	
MHCERR	007777	
MHCRC	007566	
MHDERR	010051	
MHDRCP	005071	
MHFCRC	007635	
MHNF	007607	
MINOUT	005235	
MIN.IN	040572	G
MIN.US	040574	G
MISWI =	000000	
MISWIW	013372	
MITEST=	100000	
MNDRST	010141	
MNEERR	010114	
MNOCLR	006141	
MNOINT	006062	
MODR	070256	G
MOPER	005143	
MOPERR	010032	
MORECE	002436	
MOUTIN	005214	
MPNAM	005752	
MQUALS=	003760	
MREAD	004752	
MREADH	004765	
MRESKO	005426	
MREVSK	005304	
MRLFAL	010246	
MRSLT	005157	
MSEEK	004744	
MSG.AD	040540	G
MSG.TY	040514	G
MSPERR	007726	
MSTEPR	007761	
MTMBS	005560	
MTOSLO	006013	
MUL	070212	G
MULOAD	005173	
MUNDEF	010201	
MWDERR	010015	
MWGERR	007744	
MWORD	006005	
MWRCHK	005002	
MWRITE	005016	
MWRSET	005127	
MWRTAB	010307	
M40HDR	005113	
NEWCYL	002524	
NEWPRI	067736	G
NEXTAR	064176	
NOCLR =	000010	
NOERCT	003067	
NOIRPT=	000002	
NOOP =	000100	
NOPWR	005673	
NO.CLK	045102	
NO.FLA	063772	
NO.LPT	061310	
NO.PTA	045306	
NR =	000000	
NUMBIN	054434	
NUM.LA	054602	
NUM.NO	040556	G
NUM.UN	041164	
NUNITS	047572	
NXMERR=	020000	
NXTFOR	064252	
NXTPAS	013670	
OBUFF	004066	
OCTMSG	057776	
OFIN	002562	
OFINU	002564	
OFMID	002566	
OFMIDU	002570	
OFOUT	002572	
OFOUTU	002574	
OLDCYL	002522	
ONSWAP	017454	
OPFLAG	002426	
OPIERR=	002000	
OPMSGS	002122	
OPR004	007277	
OPR1A	007250	
OPR1B	007254	
OPR12	007231	
ORIN	002576	
ORINU	002600	
ORMID	002602	
ORMIDU	002604	
OROUT	002606	
OROUTU	002610	
OUTINS=	000040	
O$APTS=	000000	
O$AU =	000000	
O$BGNR=	000000	
O$BGNS=	000001	
O$DU =	000001	
O$GNSW=	000001	
O$POIN=	000001	
PARSES	063506	
PART2 =	000000	
PAR.LA	057474	
PASCNT	002654	
PASNEW	013676	
PASNUM	003062	
PASS.C	040530	G
PATTBL	002234	
PAT1	004466	
PAT10	004742	
PAT2	004470	
PAT3	004530	
PAT4	004570	
PAT5	004630	
PAT6	004636	
PAT7	004676	
PAT8	004700	
PAT9	004740	
PH65$	017072	
POSHDS	016570	
POSHDO	020624	
POSHSB	020620	
POSHW1	020612	
PRINTC	061320	
PRINTF	064626	
PRIOR =	000004	
PRI00 =	000000	G
PRI01 =	000040	G
PRI02 =	000100	G
PRI03 =	000140	G
PRI04 =	000200	G
PRI05 =	000240	G
PRI06 =	000300	G
PRI07 =	000340	G
PRNTST	061210	
PRO.CM	045460	
PSETNM	003064	
PTAB.S	041000	G
PUTCHR	060024	
PWCON	014004	
PWRFLG	003072	
PWR.FA	071150	G
PWR.FL	040604	G
PWR.MS	071276	
PWR.SA	071272	
PWR.UP	071274	
P.CLK.	045120	
P2T01E	006225	
P2T02E	006225	
P2T03E	006244	
P2T04E	006270	
P2T05E	006312	
P2T06E	006334	
P2T07E	006356	
P2T08E	006402	
P2T09E	006424	
P2T10E	006431	
P2T11E	006446	
P2T12E	006463	
P2T13E	006477	
P2T14E	006517	