

RL11,RLV11

DRIVE TEST PART 1
CZRLCB0

AH-E045B-MC

COPYRIGHT ' 77-78

FICHE 1 OF 1

JAN 1979

digital

MADE IN USA

This microfiche card contains a grid of frames. The first column contains frames with text, likely a table of contents or index. The remaining columns contain frames with vertical bar patterns, which are likely data or test results. The text in the first column is too small to read accurately but appears to be organized in a list-like structure.

IDENTIFICATION

B 1

SEQ 0001

PRODUCT CODE: AC-E044B-MC
PRODUCT NAME: CZRLCBO RL01 DRIVE TEST PART 1
DATE CREATED: 11-OCT-78
MAINTAINER: DIAGNOSTIC ENGINEERING
AUTHOR: D. DEKNIS

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS MANUAL.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED TO THE PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DIGITAL'S COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DIGITAL.

DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (C) 1977, 1978, DIGITAL EQUIPMENT CORPORATION

TABLE OF CONTENTS

- 1.0 GENERAL INFORMATION
- 1.1 PROGRAM ABSTRACT
- 1.2 SYSTEM REQUIREMENTS
- 1.3 RELATED DOCUMENTS AND STANDARDS
- 1.4 DIAGNOSTIC HIERARCHY PREREQUISITES
- 1.5 ASSUMPTIONS

- 2.0 OPERATING INSTRUCTIONS
- 2.1 HOW TO RUN THIS DIAGNOSTIC
- 2.1.1 THE SIX STEPS OF EXECUTION
- 2.1.2 SAMPLE RUN-THROUGH
- 2.2 HOW TO CREATE A CHAINABLE FILE
- 2.3 DETAILS OF COMMANDS AND SYNTAX
- 2.3.1 TABLE OF COMMAND VALIDITY
- 2.3.2 COMMAND SYNTAX
- 2.4 EXTENDED P-TABLE DIALOGUE
- 2.5 HARDWARE PARAMETERS
- 2.6 SOFTWARE PARAMETERS

- 3.0 ERROR INFORMATION

- 4.0 PERFORMANCE AND PROGRESS REPORTS

- 5.0 DEVICE INFORMATION TABLES

- 6.0 TEST SUMMARIES

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 RLO1 DISK DRIVES RL11/RLV11 CONTROLLERS (4 DRIVES PER CONTROLLER). THE ENTIRE PROGRAM IS RUN ON THE FIRST DRIVE BEFORE STARTING ON THE SECOND. THE PROGRAM STARTS BY TESTING THE SIMPLEST FUNCTIONS FIRST USING THE LOGIC TESTED IN EARLIER TESTS TO TEST MORE COMPLEX FUNCTIONS.

THIS PROGRAM TESTS THE RLO1 INTERFACE AND BASIC DRIVE LOGIC. GET STATUS WITH RESET, GET STATUS, SEEK, AND READ HEADER ARE THE ONLY COMMANDS EXECUTED IN THE PROGRAM. ONLY SEEKS WITH 0 DIFFERENCE ARE USED SO NO HEAD MOVEMENT IS REQUIRED.

A SIGNIFICANT PORTION OF THE PROGRAM REQUIRES MANUAL INTERVENTION. THESE TESTS TEST THE COVER OPEN AND WRITE LOCK STATUS. THE DRIVE MUST BE LOADED AND UNLOADED TO TEST ALL THE CONDITIONS OF HEADS OUT, BRUSH HOME, AND DRIVE STATES. THE PROGRAM CAN BE RUN IN AUTOMATIC MODE IN WHICH CASE ALL TESTS REQUIRING MANUAL INTERVENTION ARE BYPASSED.

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 RLO1 DRIVES
1 - 8 RLO1K CARTRIDGES WITH BAD SECTOR FILE
KW11P, KW11L (OPTIONAL)
LINEPRINTER(OPTIONAL)

1.2.2 SOFTWARE REQUIREMENTS

CZRLCBO RLO1 DRIVE TEST PART 1
(FORMERLY MD-11-DZRLC-A)

1.3 RELATED DOCUMENTS AND STANDARDS

RLO1 USERS MANUAL (EK-RLO1-UG-PRE)
XXDP USERS MANUAL

1.4 DIAGNOSTIC HIERARCY PREREQUISITES

THE RLO1 SUBSYSTEM SHOULD HAVE SUCCESSFULLY RUN THE FOLLOWING PROGRAMS:

CVRLAAO	RLV11 RLO1 DISKLESS TEST (RLV11)
CZRLABO	RL11/RLV11 CONTROLLER TEST PART 1
CZRLCBO	RL11/RLV11 CONTROLLER TEST PART 2

1.5 ASSUMPTIONS

THE HARDWARE OTHER THAN THE RLO1 SUBSYSTEM IS ASSJMED 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 ?	IS THERE A 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 DIAGNOS IC.

* 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 AGAIN:

1. ISSUE ANOTHER 'START' COMMAND (THUS GOING THRU ALL OF STEPS 2, 3, 4, 5, AND 6 AGAIN)
2. ISSUE A 'RESTART' COMMAND (SAME AS START COMMAND EXCEPT THAT THE HARDWARE QUESTIONS ARE NOT ASKED)
3. ISSUE A 'CONTINUE' COMMAND (EXECUTION WILL RESUME AT THE BEGINNING OF THE PARTICULAR HARDWARE TEST (MOST DIAGNOSTICS CONSIST OF A NUMBER OF THESE) THAT IT WAS IN WHEN THE ERROR HALT OCCURED. NO QUESTIONS ASKED.
4. ISSUE A 'PROCEED' COMMAND: EXECUTION WILL RESUME AT THE INSTRUCTION FOLLOWING THE ERROR REPORT (THIS IS A SPECIAL COMMAND AND CAN BE ISSUED ONLY AT A HALT 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/FLAG: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
SOHZ (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. THAT IS WHY IT BEARS THE EXTENSION 'BIN' INSTEAD OF 'BIC'. THERE IS A WAY, HOWEVER, TO CREATE A CHAINABLE PROGRAM FROM WHAT YOU'VE GOT.

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

HERE IS A SAMPLE DIALOGUE TO ACCOMPLISH THIS:

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

```
DS-B>CCI
# UNITS (D) ? 4
-----
-----
```

```
CHANGE SW (L) ? N
PTAB END. 60632
```

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

```
*HCORE 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 'R/JN 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 NUMERICAL ORDER REGARDLESS OF THE ORDER OF SPECIFICATION. THE DEFAULT IS TO EXECUTE ALL TESTS.

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

- HGE 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 TESTS BEING EXECUTED
- BOE BELL ON ERROR
- UAM RUN IN UNATTENDED MODE, BYPASSING MANUAL INTERVENTION TESTS
- ISR INHIBIT STATISTICAL REPORTS
- IDU INHIBIT DROPPING OF UNITS BY DIAGNOSTIC

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

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

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

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

THE QUESTION 'CHANGE SW?' IS ASKED, AND THE ANSWERS IF GIVEN BECOME THE NEW DEFAULTS. THE COMMAND MAY BE ISSUED WHEN COMMAND 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

PROCEED)/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 FORMATS.

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 TRANSLATES TO THE STRING 6,7,8,9,10 (AN INCREMENT OF 1). IF THE VALUES ARE ADDRESSES, THE SAMPLE RANGE TRANSLATES TO THE STRING 6,8,10 (AN INCREMENT OF 2).

NOW LET US SEE HOW WE COULD USE THESE CAPABILITIES TO CONSTRUCT A SET OF P-TABLES. ASSUME THAT WE HAVE 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 SHOULD 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 EXPLICIT 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 RESPONSE.

RL11 (L) Y?

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

BUS ADDRESS (O) 174400?

ANSWER WITH THE BUS ADDRESS OF THE CONTROLLER.

VECTOR (O) 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 FLEXIBILITY IN THE WAY THE PROGRAM BEHAVES. THE SOFTWARE PARAMETERS GIVE THE PROGRAM FLEXIBILITY IN THE WAY IT RUNS. THE PARAMETERS CAN BE MODIFIED ON A START, RESTART, OR CONTINUE BY ANSWERING (Y)ES TO THE FOLLOWING QUESTION:

CHANGE S.W. ?

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

EXECUTE DRIVE SELECT TESTS (N)?

IF 'YES' TESTS 5 AND 6 ARE EXECUTED IN THE FIRST PASS OF THE PROGRAM. THESE TESTS REQUIRE MANUAL INTERVENTION TO CHANGE ADDRESS PLUGS AND REQUIRE A FULL COMPLEMENT OF ADDRESS PLUGS (0 - 3).

EXECUTE HEAD ALIGNMENT SUPPORT (N)?

IF 'YES', TEST 11 IS EXECUTED IN THE FIRST PASS.

EXECUTE MANUAL INTERVENTION TESTS (N)?

IF 'YES', TESTS 1, 2, 3, AND 4 ARE EXECUTED TO TEST BASIC INTERFACE OPERATIONS, HEAD LOADING, HEAD UNLOADING, AND ALL STATE CHANGES.

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 FURTHER TESTING.

DROP DRIVE IF NO RESPONSE (N)?

IF THIS PARAMETER IS SPECIFIED AS YES, THE PROGRAM WILL CHECK EACH DRIVE BEFORE TESTING STARTS TO DETERMINE IF IT IS READY OR IF IT WILL RESPOND TO A GET STATUS. 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 RELEVANT DATA.

3.1 ERROR REPORTING

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 BEING 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/DL	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 CONTENTS 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.

3.1.1 SPECIFIC OPERATION MESSAGES

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 HEADERS - 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 'FOI 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)
CJR 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 LOSING READY BEFORE STARTING A READ HEADER AND THEREFORE NOT COMPLETING THE READ HEADER IN 1MS.

'FAIL TO RELOAD HDS 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 CONTROLLER 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 NOT 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. 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.

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 COMPLEMENT)

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 BASIC INTERFACE TEST (PART 1)

LOAD IN DRIVE NUMBER. DO GET STATUS WITH RESET. IF OPI SETS:
DRIVE INTERFACE IS DEAD
DRIVE COMMAND SHIFT REGISTER NOT LOADING/SHIFTING
MARKER DETECTION FAILED
DRIVE IS NOT SELECTING OR AC LOW IS SET
SYSTEM OR STATUS CLOCKS NOT OPERATIONAL
GET STATUS DETECTION FAILED.

IF INTERRUPT WITH NO OPI, CHECK STATUS RECEIVED. COVER OPEN
AND BRUSH HOME SHOULD BE SET. IF NOT:
BAD STATUS DATA LINE
BAD COVER SWITCH OR LOGIC
DRIVE COMMAND SHIFT REGISTER
BAD BRUSH HOME SWITCH OR LOGIC

CHECK WRITE LOCK STATUS BIT SET. IF NOT:
BAD SWITCH OR WRITE LOCK LOGIC

DRIVE COMMAND SHIFT REGISTER

CHECK STATE FOR 0. IF NOT:
BAD STATE ROM
DRIVE COMMAND SHIFT REGISTER

CHECK VOLUME CHECK RESET. IF NOT:
BAD RESET DETECTION
BAD VOLUME CHECK LOGIC
DRIVE COMMAND SHIFT REGISTER

CHECK DRIVE ERROR RESET. IF NOT:
BAD DRIVE ERROR INTERFACE
SOME OTHER ERROR STUCK ON. REPORT WHICH ERROR.

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

TEST 2 BASIC INTERFACE TEST (PART 2)

REQUEST OPERATOR TO CLOSE COVER AND RESET WRITE LOCK.

DO GET STATUS LOOP CHECKING IF COVER OPEN OR WRITE LOCK RESETS. WAIT 15 SECONDS FOR BOTH TO CHANGE. IF NO CHANGE, ASK OPERATOR TO TYPE CR IF PROCEDURE WAS FOLLOWED.

IF ONE CHANGED BUT NOT THE OTHER, REPORT WHICH FAILURE:

WRITE LOCK SWITCH OR LOGIC
(OR) COVER OPEN SWITCH OR LOGIC
DRIVE COMMAND SHIFT REGISTER

IF NEITHER CHANGED, REPORT BOTH FAILURES.

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

TEST 3 HEAD LOADING TEST

REQUEST OPERATOR TO PRESS LOAD SWITCH.

DO GET STATUS LOOP CHECKING FOR STATE TO GO TO 1. WAIT 30 SECONDS FOR CHANGE. IF NO CHANGE, ASK OPERATOR TO CONFIRM ACTION BY TYPING CR.

IF LOAD WAS PRESSED:

BAD STATE ROM

BAD LOAD SWITCH OR LOGIC

CHECK THAT STATE 1 REMAINS FOR LESS THAN 30 SECONDS. IF NOT:

SPINDLE NOT TURNING OR TOO SLOW (AC SERVO)
SECTOR PULSE DETECTION OR LOGIC BAD
BAD CLOCK SHIFT REGISTER IN SPEED CONTROL
BAD DISK ON SPEED LOGIC
BAD STATE ROM

AND CHECK IF SPINUP TIMEOUT ERROR SET. IF NOT:

BAD STATE ROM
BAD TIMEOUT DETECTION LOGIC

CHECK THAT STATE GOES TO 2. IF NOT:

BAD STATE ROM

CHECK THAT BRUSH HOME IS RESET 5 SECONDS OR LESS AFTER STATE IS 2. IF NOT:

BAD BRUSH HOME SWITCH OR LOGIC
BAD BRUSH MOTOR (AC SERVO)

WAIT 30 SECONDS FOR BRUSH HOME TO SET. IF NOT:

BAD AC SERVO
BAD SWITCH OR LATCH

CHECK THAT STATE HAS CHANGED TO 3. IF NOT:

BAD STATE ROM

AFTER STATE IS 3, CHECK HEADS OUT IS SET. IF NOT:

BAD SWITCH
BAD SEEK CONTROL ROM
BAD VELOCITY ROM
BAD DC SERVO

CHECK VOLUME CHECK IS SET. IF NOT:

BAD VOLUME CHECK LOGIC

CHECK IF DRIVE ERROR IS SET. IF NOT:

BAD DRIVE ERROR LOGIC OR INTERFACE

WAIT 300 MS FOR STATE TO CHANGE TO 4. IF IT DOESN'T CHANGE:

STATE ROM BAD
SEEK ROM
VEL ROM
GUARD BAND DETECTION

WAIT 15 MS FOR STATE TO CHANGE TO 5.

8 MS AFTER STATE GOES TO 5, DRIVE READY SHOULD SET. IF NOT:

INTEGRATOR OR NULL DETECTION FAILURE
READY ONE SHOT BAD
ENABLE TIMEOUT H NOT SETTING OR COUNT LOGIC BAD

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

TEST 4 HEAD UNLOADING TEST

CHECK DRIVE IS READY. IF NOT REPORT AND ASK OPERATOR TO MAKE DRIVE READY.

REQUEST OPERATOR TO UNLOAD DRIVE.

LOOP ON GET STATUS WAITING FOR STATE TO CHANGE TO 6. IF NO CHANGE:

BAD STATE ROM
BAD SWITCH

WAIT 300 MS FOR STATE TO CHANGE TO 7. IF NO CHANGE:

BAD STATE ROM

AFTER STATE IS 7, WAIT 30 SEC FOR STATE TO CHANGE TO STATE 0.
IF NO CHANGE:

NO BRAKING
BAD AC SERVO

REQUEST OPERATOR TO LOAD DRIVE. WAIT UNTIL DRIVE BECOMES READY.

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

TEST 5 DRIVE SELECT TEST

INSTRUCT THE OPERATOR TO REMOVE DRIVE ADDRESS PLUGS FROM ALL DRIVES EXCEPT THE DRIVE UNDER TEST. ASK THAT CARRIAGE RETURN

BE TYPED WHEN DONE.

DO GET STATUS TO ADDRESS OF DRIVE UNDER TEST. CHECK THAT NO ERRORS ARE REPORTED. DO GET STATUS TO ALL OTHER ADDRESSES AND CHECK THAT OP1 SETS FOR ALL OTHER ADDRESSES.

DO GET STATUS TO ADDRESS OF NEXT SEQUENTIAL ADDRESS. CHECK THAT NO ERRORS ARE REPORTED. DO GET STATUS TO ALL OTHER ADDRESSES AND CHECK THAT OP1 SETS.

REPEAT FOR ALL DRIVE ADDRESSES (0,1,2,3 - 0 IS SEQUENTIAL AFTER 3).

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

TEST 6 DRIVE SELECT ERROR TEST

REQUEST OPERATOR INSERT IDENTICAL ADDRESS PLUGS IN TWO DRIVES (MUST BE IDENTICAL TO NUMBER SPECIFIED EARLIER). REQUEST OPERATOR TYPE CARRIAGE RETURN WHEN READY.

PROCEDURE WILL BE TO GET STATUS AND CHECK FOR DRIVE SELECT ERROR. THEN RESET THAT DRIVE AND VERIFY THAT DRIVE SELECT ERROR IS NOT REPORTED AGAIN. WAIT 1 SECOND, THEN CHANGE DRIVE SELECT TO A DIFFERENT NUMBER AND BACK AGAIN. DRIVE SELECT ERROR SHOULD SET AGAIN.

OPERATOR SHOULD SEE THE FAULT LIGHT ON ON BOTH DRIVES. IF INDICATOR IS NOT SEEN ON A DRIVE:

DRIVE SELECT ERROR DETECTION IS BAD IN THAT DRIVE.

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

4.2 STANDARD TESTS

IF THE PROGRAM OPERATION MODE 1 IS SELECTED, THIS WILL BE THE FIRST TEST EXECUTED. THE DRIVE(S) TO BE TESTED MUST BE POWERED UP, HEADS LOADED, AND WRITE LOCK RESET.

TEST 7 INITIAL STATE TEST

DO GET STATUS, WAIT FOR INTERRUPT.

IF OPI OCCURS:

DRIVE INTERFACE IS DEAD

DRIVE COMMAND SHIFT REGISTER NOT LOADING/SHIFTING
DRIVE IS NOT SELECTING OR AC LOW IS SET
SYSTEM OR STATUS CLOCKS NOT OPERATIONAL
GET STATUS DETECTION FAILED.

IF INTERRUPT OCCURS WITHOUT OPI, CHECK DRIVE READY. READY SET INDICATES HEADS ARE LOADED AND ARE TRACKING (POSITION WORKING).

IF MANUAL INTERVENTION TESTS WERE RUN, CHECK THAT HEAD 0 IS SELECTED. IF NOT:

DRIVE CYCLE UP DID NOT SELECT HEAD 0

IF DRIVE READY IS SET, CHECK STATUS MESSAGE RECEIVED. HEADS DU* AND BRUSH HOME MUST BE SET. IF NOT:

DRIVE COMMAND SHIFT REGISTER NOT LOADING/SHIFTING
HEADS OUT OR BRUSH HOME SWITCH OR ASSOCIATED
CIRCUITRY BAD
STATUS DATA BAD

IF MANUAL INTERVENTION TESTS WERE RUN AND THIS IS THE FIRST
PASS CHECK THAT VOLUME CHECK AND DRIVE ERROR ARE SET.

CHECK ALL ERROR BITS ARE 0.

CHECK STATE IS 5. IF NOT:

DRIVE COMMAND SHIFT REGISTER BAD

TEST 8 INITIAL RESET STATE TEST

DO GET STATUS HEAD SELECT 0, WAIT FOR INTERRUPT.

DO GET STATUS WITH RESET, WAIT FOR INTERRUPT. BOTH DRIVE
ERROR AND VOLUME CHECK SHOULD NOW BE RESET. IF NOT:

RESET DETECTION, RESET ERROR, OR VOLUME CHECK FLOP BAD
DRIVE COMMAND SHIFT REGISTER BAD

HEAD SELECTED BIT SHOULD STILL BE ZERO. IF NOT:

DRIVE COMMAND SHIFT REGISTER BAD
HEAD SELECT SHIFT REGISTER NOT LOADING

TEST 9 DRIVE READY TEST

DO SEEK WITH 0 DIFFERENCE, SIGN 0, HEAD 0. WAIT FOR
INTERRUPT. GET STATUS. CHECK STATE IS 5. IF NOT:

DIFFERENCE COUNTER PICKING UP BITS
COUNTER CIRCUITRY IS NOT INDICATING 0 DIFFERENCE

CHECK DRIVE READY IS RESET. IF NOT:

ENABLE TIMEOUT OR READY LATCH/ONE SHOT BAD

WAIT APPROX 8 MS FOR READY TO SET. IF IT TAKES LONGER OR
DOESN'T SET AT ALL:

HEADS MAY HAVE MOVED (INTEGRATOR OR NULL DETECTION)
READY ONE SHOT FAILED

CHECK DRIVE ERROR DID NOT SET. IF IT SET, DO GET STATUS AND
REPORT WHICH ERROR.

VERIFY HEAD SELECT IS ZERO.

TEST 10 SEEK SIGN SWITCH TEST

DO SEEK WITH DIFFERENCE 0, SIGN 1, HEAD 0. WAIT FOR INTERRUPT. GET STATUS AND CHECK STATE IS 5. IF NOT:

COUNT ROM
DIFFERENCE COUNTER PICKING UP BITS
COUNTER CIRCUITRY IS NOT INDICATING 0 DIFFERENCE

VERIFY DRIVE IS NOT READY

WAIT APPROX 8 MS FOR READY TO SET. IF IT TAKES LONGER OR DOESN'T SET AT ALL:

HEADS ARE MOVING (INTEGRATOR OR NULL DETECTION)
READY ONE SHOT FAILED
COUNT ROM

VERIFY DRIVE ERROR DID NOT SET

VERIFY HEAD SELECT IS ZERO.

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

TEST 11 HEAD ALIGNMENT SUPPORT ROUTINE

THIS TEST IS EXECUTED WHEN THE PROGRAM IS STARTED AT ADDRESS 204, HEAD ALIGNMENT SUPPORT IS REQUESTED, AND IN THE FIRST PASS ONLY. IT IS BYPASSED IF THE PROGRAM IS STARTED AT ANY OTHER ADDRESS AND IN THE SECOND AND SUBSEQUENT PASSES.

THIS TEST SELECTS THE DRIVE UNDER TEST AND LOOPS ON A GET

STATUS WITH RESET. THE WRITE LOCK BIT IS MONITORED AND WHEN WRITE LOCK IS RESET HEAD 0 IS SELECTED AND WHEN WRITE LOCK IS SET HEAD 1 IS SELECTED. THIS WILL PERMIT THE HEADS TO BE ALIGNED IN KEEPING WITH THE PRESENT HEAD ALIGNMENT PROCEDURE WITHOUT RETURNING TO THE CONSOLE.

TYPING A CARRIAGE RETURN ON THE CONSOLE WILL TERMINATE THIS TEST ON THE DRIVE UNDER TEST. BEFORE TERMINATING, THE TEST WILL CHECK THAT WRITE LOCK IS RESET. IF NOT, THE OPERATOR WILL BE REQUESTED TO RESET WRITE LOCK.

TEST 12 HEAD SWITCHING TEST

DO SEEK WITH 0 DIFFERENCE, SIGN 0, HEAD 1. WAIT FOR INTERRUPT. GET STATUS AND CHECK STATE IS 5. IF NOT:

DIFFERENCE COUNTER IS PICKING UP BITS
ASSOCIATED CIRCUITRY IS BAD

VERIFY DRIVE READY RESET. IF NOT:

ENABLE TIMEOUT OR READY LATCH/ONE SHOT BAD

WAIT APPROX 8 MS FOR READY TO SET. IF IT TAKES LONGER OR DOESN'T SET AT ALL:

HEADS ARE MOVING (INTEGRATOR OR NULL DETECTION)
READY ONE SHOT FAILED
DRIVE CANNOT TRACK WITH THIS HEAD

VERIFY DRIVE ERROR DID NOT SET.

DO GET STATUS, CHECK HEAD SELECT IS CORRECT. IF NOT:

HEAD SELECT REGISTER BAD
DRIVE COMMAND SHIFT REGISTER BAD

DO SEEK WITH 0 DIFFERENCE, SIGN 0, HEAD 0. REPEAT ABOVE TESTS.

TEST 13 READ HEADER TEST (PART 1)

DO SEEK WITH DIFFERENCE 0, HEAD 0, SIGN 0. WAIT FOR INTERRUPT AND WAIT FOR DRIVE READY.

DO READ HEADER, WAIT FOR INTERRUPT.

CHECK IF HEADER CRC ERROR SET. IF SET:

READ/WRITE BOARD BAD
READ DATA LINE BAD

CHECK IF BIT 6 OF WORD 1 IS SAME AS HEAD SELECT BIT IN STATUS. IF NOT:

HEADS ARE SWITCHED (CABLE)
HEAD SELECT LOGIC

IF MANUAL INTERVENTION TESTS WERE RUN AND HEAD ALIGNMENT TESTS WERE NOT RUN, CHECK THAT HEADER WORD 0 INDICATES HEADS ARE POSITIONED OVER CYLINDER 0. STORE HEADER WORD 1.

REPEAT TESTS USING HEAD 1.

CHECK THAT CYLINDER PORTION OF STORED HEADER WORD 1 IS THE SAME AS HEADER WORD 1 OF THIS HEADER. IF NOT:

HEADS ARE MISALIGNED

TEST 14 READ HEADER TEST (PART 2)

DO SEEK WITH DIFFERENCE 0, SIGN 0, HEAD 0. WAIT FOR INTERRUPT. WAIT FOR READY.

DO 40 CONSECUTIVE READ HEADER, STORE 3 HEADER WORDS AFTER EACH READ.

CHECK ALL HEADERS FOR SEQUENCE AND CONTENT (WORD 2 ALL ZERO, BIT 15 WORD 1 AND 3 IS 0, HS BIT WORD 1 IS 0). IF NOT:

BAD READ/WRITE BOARD
BAD PACK

DO SEEK WITH DIFFERENCE 0, SIGN 0, HEAD 1. REPEAT ABOVE TEST FOR HEAD 1. %

.MAIN. MACY11 30A(1052) 22-NOV-78 16:20
CZRLCB.P11 23-OCT-78 14:39

L 3

TABLE OF CONTENTS

SEQ 0037

2399	*TEST 1	BASIC INTERFACE (PART 1)
2437	*TEST 2	BASIC INTERFACE (PART 2)
2466	*TEST 3	HEAD LOADING
2624	*TEST 4	HEAD UNLOADING
2705	*TEST 5	DRIVE SELECT
2754	*TEST 6	DRIVE SELECT TEST
2847	*TEST 7	INITIAL STATE
2911	*TEST 8	INITIAL RESET STATE
2934	*TEST 9	DRIVE READY
2992	*TEST 10	SEEK SIGN SWITCH
3064	*TEST 11	HEAD ALIGNMENT SUPPORT
3123	*TEST 12	HEAD SWITCHING
3194	*TEST 13	READ HEADER (PART 1)
3241	*TEST 14	READ HEADER (PART 2)
3402	DIAGNOSTIC SUPERVISOR	-- LOW CORE SET UP

1			.NLIST	CND,MD,ME
2			.ENABL	ABS,AMA
3		002000	.=2000	
5				
C	002000		SVC	
7		000001	SVCTST=1	
8		000001	SVCSUB=1	
9		000001	SVCBGL=1	
10		000000	SVCINS=0	
11		000000	SVCTAG=0	
12	002000		POINTER	BGNSW,BGNSFT,BGNDU
13				
14	002000		BGNMOD	MDHEDR
16	002000		HEADER	CZRLC,B,0,1,1,1,RL01
(4)	002000	103	.ASCII	/C/
(4)	002001	132	.ASCII	/Z/
(4)	002002	122	.ASCII	/R/
(4)	002003	114	.ASCII	/L/
(4)	002004	103	.ASCII	/C/
(6)	002005	000	.BYTE	0
(6)	002006	000	.BYTE	0
(5)	002007	000	.BYTE	0
(4)	002010	102	.ASCII	/B/
(4)	002011	060	.ASCII	/O/
(4)	002012	000000	.WORD	0
(4)	002014	000001	.WORD	1
(4)	002016	031650	.WORD	L\$HARD
(4)	002020	031774	.WORD	L\$SOFT
(4)	002022	014642	.WORD	L\$HW
(4)	002024	014656	.WORD	L\$SW
(4)	002026	032514	.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	014674	.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	000001	.WORD	1
(4)	002054	000001	.WORD	1
(4)	002056	000000	.WORD	0
(5)	002060	000000	.WORD	0
(4)	002062	000000	.WORD	0
(4)	002064	002114	.WORD	L\$DV'TYP
(4)	002066	000000	.WORD	0
(4)	002070	002112	.WORD	L\$DR
(4)	002072	002112	.WORD	L\$DRST
(4)	002074	000000	.WORD	0
(4)	002076	016052	.WORD	L\$DU
(5)	002100	000014	.WORD	14
(4)	002102	000000	.WORD	0
(4)	002104	014730	.WORD	L\$INIT
(4)	002106	015732	.WORD	L\$CLEAN

```

21 002110          ENDMOD
22 002110          DEVREG
(5) 002110 000000  .WORD 0
(2) 002112 000001  .BLKW
23 002114          DEVTYP <RL01>
(3) 002114 046122 030460 000 .ASCIZ /RL01/
(2) 002114 002122  .EVEN
24
25                :COPYRIGHT (C) 1977, 1978
26                :THIS SOFTWARE IS FURNISHED UNDER LICENSE FOR USE ONLY
27                :ON A SINGLE COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH
28                :THE INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS
29                :SOFTWARE, OR ANY COPIES THEREOF, MAY NOT BE PROVIDED
30                :OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT
31                :FOR USE ON SUCH SYSTEM, AND TO ONE WHO AGREES TO THESE
32                :LICENSE TERMS. TITLE TO OWNERSHIP OF THE SOFTWARE SHALL
33                :AT ALL TIMES REMAIN IN DEC.
34
35                :THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE
36                :WITHOUT NOTICE AND SHALL NOT BE CONSTRUED AS A COMMITMENT
37                :BY DIGITAL EQUIPMENT CORPORATION.
38
39                :DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY
40                :OF ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
41
42 002122          BGNMOD  GLBEQAT
43
44 002122          EQUALS
45                : OFFSETS FOR HARDWARE P-TABLE
46                CSR      -0          :BUS ADDRESS
47                VECT     2          :VECTOR ADDRESS
48                PRIOR    -4          :PRIORITY
49                DRSB     6          :DRIVE SELECT BIT
50                CNT      10         :CONTROLLER TYPE
51
52                : OFFSET FOR SOFTWARE P-TABLE
53                MISWI    =0          :SOFTWARE PARAMETERS SWITCHES
54                LOLIM    =2          :CYLINDER LOWER LIMIT
55                HILIM    =4          :CYLINDER HIGH LIMIT
56                HEAD     =6          :SELECTED HEAD FOR RUNNING TESTS
57                ERLIM    =10         :ERROR LIMIT
58                DCLIM    =12         :DATA COMPARE ERROR LIMIT
59
60                : BIT ASSIGNMENT FOR SOFTWARE P-TABLE SWITCHES
61                ALLCYL   =BIT00      :USE ALL CYLINDERS
62                ALLSEC   =BIT01      :USE ALL SECTORS
63                DRSELT   =BIT02      :EXECUTE DRIVE SELECT TEST
64                HDALIGN  =BIT03      :EXECUTE HEAD ALIGNMENT TEST
65                AUTOSZ   =BIT04      :AUTO SIZE FOR DRIVE-DROP IF NO RESPONSE
66                HEADLM   =BIT12      :HEAD LIMIT SPECIFIED FLAG
67                HICYL    =BIT13      :HI LIMIT SPECIFIED FLAG
68                LOCYL    =BIT14      :LO LIMIT SPECIFIED
69                MITEST   =BIT15      :EXECUTE MANUAL INTERVENTION TESTS
70
71                : SUBSYSTEM FUNCTIONS
72 000102          CKDATA -102        :WRITE CHECK

```



```

73      000104      GSTAT  =104      ;GET STATUS
74      000106      SEEK    =106      ;SEEK
75      000110      RDHEAD  =110      ;READ HEADER
76      000112      WDATA   =112      ;WRITE DATA
77      000114      RDATA   =114      ;READ DATA
78      000116      RDNOHR  =116      ;READ DATA, IGNORE HEADERS
79      000100      NOOP    =100      ;NO OPERATION
80
81      ;           ; OPERATION FLAGS
82      007777      COMPOP  =7777      ;COMPOSITE OPERATION FLAGS
83      000002      HDRCMP  =BIT01     ;HEADER COMPARE OPERATION
84      000001      DATACMP =BIT00     ;DATA COMPARE OPERATION
85      000004      CYLUP   =BIT02     ;CYCLE UP OPERATION
86      000010      ULOAD   =BIT03     ;UNLOAD OPERATION
87      000020      INOUTS  =BIT04     ;IN-OUT SEEK OPERATION
88      000040      OUTINS  =BIT05     ;OUT-IN SEEK OPERATION
89      000100      FOLWRT  =BIT06     ;FOLLOWING WRITE OPERATION
90      000200      REVSKS  =BIT07     ;REV SEEK SEQ (ADJ INTERFERENCE)
91      000400      FWDSKS  =BIT08     ;FWD SEEK SEQ (ADJ INTERFERENCE)
92      001000      REVSKO  =BIT09     ;REV SEEK SEQ (OVERWRITE)
93      002000      FWDSKO  =BIT10     ;FWD SEEK SEQ (OVERWRITE)
94      004000      BADADD  =BIT11     ;BAD DISK ADDRESS
95      010000      SFEKOP  =BIT12     ;SEEK OPERATION
96      020000      RORWOP  =BIT13     ;READ OR WRITE OPERATION
97      040000      RELDWT  =BIT14     ;RELOAD WAIT
98      100000      HDR40   =BIT15     ;40 HEADER OPERATION
99      003760      MQUALS  =OUTINS!INOUTS!FOLWRT.REVSKS!FWDSKS!REVSKO.FWDSKO
100                                           ;MESSAGE QUALIFIER BITS
101
102      ;           ; ERROR FLAGS FROM SUBROUTINES
103      000001      TOSLOW  =BIT00     ;OPERATION TOOK TO LONG
104      000002      NOIRPT  =BIT01     ;NO INTERRUPT FROM OPERATION
105      000004      CONHNG  =BIT02     ;CONTROLLER HUNG
106      000010      NOCLR   =BIT03     ;BAD CONTROLLER CLEAR
107
108      000000      RLCS    =0         ;CONTROL AND STATUS REGISTER
109      000002      RLBA    =2         ;BUS ADDRESS REGISTER
110      000004      RLDA    =4         ;DISK ADDRESS REGISTER
111      000006      RLMP    =6         ;MULTI-PURPOSE REGISTER
112
113      ;           ; REGISTER BIT DEFINITIONS - CONTROL STATUS REGISTER
114      000000      RLCSR   =0         ;CONTROL AND STATUS REGISTER
115      100000      ANYERR  =100000    ;ANY ERROR BIT
116      040000      DRVERR  =40000    ;DRIVE ERROR BIT
117      020000      NXMERR  =20000    ;NON-EXISTANT MEMORY ERROR
118      010000      DLTERR  =10000    ;DATA LATE ERROR
119      010000      HNFERR  =10000    ;HEADER NOT FOUND ERROR
120      004000      DCKERR  =4000     ;DATA CHECK ERROR
121      004000      HCRERR  =4000     ;HEADER CHECK ERROR
122      002000      OPIERR  =2000     ;OPERATION INCOMPLETE ERROR
123      001400      DSMSK   =1400     ;DRIVE SELECT MASK
124      000200      CRDYMSK =200      ;CONTROLLER READY MASK
125      000100      INTEBL  =100      ;INTERRUPT ENABLE MASK
126      000060      BAMSK   =60       ;BUS ADDRESS UPPER MASK
127      000001      DRDYMSK =1        ;DRIVE READY MASK
128

```

```

129          : REGISTER BIT DEFINITIONS - DISK ADDRESS FOR DATA XFER
130          000077      SAMSK =77          :SECTOR ADDRESS MASK
131          000100      HSMSK =100         :HEAD SELECT MASK
132          077600      CAMSK =77600        :CYLINDER ADDRESS MASK
133
134          : REGISTER BIT DEFINITIONS - DISK ADDRESS FOR SEEK
135          000001      MBSETO =1          :MUST BE SET, BIT 0
136          000004      DIRBIT =4          :DIRECTION BIT
137          000020      HDSEL =20         :HEAD SELECT BIT
138          077600      DIRMSK =-77600     :CYLINDER DIFFERENCE MASK
139
140          : REGISTER BIT DEFINITIONS - DISK ADDRESS FOR GET STATUS
141          000003      GETSTAT =3         :GET STATUS SETUP
142          000010      DRSET =-10        :DRIVE RESET MASK
143
144          : REGISTER BIT DEFINITIONS - MP FOR DATA XFER
145          017777      WCMSK =17777      :WORD COUNT MASK
146          160000      WCRNG =160000     :WORD COUNT RANGE MASK
147
148          : REGISTER BIT DEFINITIONS - MP FOR READ HEADER
149          077600      HDCYL =-077600    :CYLINDER MASK
150          000077      HDSEC =77         :SECTOR MASK
151          000100      HDHSEL =100       :HEAD SELECT MASK
152
153          : REGISTER BIT DEFINITIONS - MP FOR GET STATUS
154          000007      STAMSK =7          :STATE MASK
155          000010      BHSTAT =-10        :BRUSH HOME STATUS
156          000020      HOSTAT =20         :HEADS OUT STATUS
157          000040      COSTAT =40        :COVER OPEN STATUS
158          000100      HSSIAT =100        :HEAD SELECT STATUS
159          000400      DSESTAT =-400     :DRIVE SELECT ERROR STATUS
160          001000      VCSTAT =1000      :VOLUME CHECK STATUS
161          002000      WGESTAT =2000     :WRITE GATE ERROR STATUS
162          004000      SPDSTAT =4000     :SPIN ERROR STATUS
163          010000      STOSTAT =10000    :SEEK TIMEOUT ERROR STATUS
164          020000      WLSTAT =20000     :WRITE LOCK STATUS
165          040000      HCESTAT =40000    :HEAD CURRENT ERROR STATUS
166          100000      WDESTAT =100000   :WRITE DATA ERROR STATUS
167
168          002122      ENDMOD
169          002122      BGNMOD GLBDAT
170
171          : TABLE OF OPERATION MESSAGES
172          002122 000000      OPMSG: .WORD 0          :FILLER
173          002124 005002      .WORD MWRCHK          :MESSAGE FOR WRITE CHECK
174          002126 005032      .WORD MGTSTA          :GET STATUS
175          002130 004744      .WORD MSEEK           :SEEK
176          002132 004765      .WORD MREADH         :READ HEADER
177          002134 005016      .WORD MWRITE         :WRITE DATA
178          002136 004752      .WORD MREAD          :HEAD DATA
179          002140 005127      .WORD MWRSET         :WITH RESET
180          002142 005046      .WORD MDATCP         :WITH DATA COMPARE
181          002144 005071      .WORD MHDRCP         :WITH HEADER COMPARE
182          002146 005204      .WORD MCYLUP         :LOAD HEADS
183          002150 005173      .WORD MLOAD         :UNLOAD HEADS
184          002152 005235      .WORD MINOUT         :IN-OUT SEQ

```

```
185 002154 005214 .WORD MOUTIN : OUT-IN SEQ
186 002156 005260 .WORD MFOLWRT : FOLLOWING WRITE
187 002160 005304 .WORD MREVSK : REV SEEK
188 002162 005337 .WORD MFWDSK : FWD SEEK
189 002164 005426 .WORD MRESKO : REV SEEK
190 002166 005372 .WORD MFWSKO : FWD SEEK
191 002170 005462 .WORD MBADAD : BAD DISK ADD FOR WRITE
192 002172 005113 .WORD M4OHDR : 40 HEADER OPERATION
193
194
195 002174 010722 : RESTBL: TABLE OF RESULT NAME MESSAGE ADDRESSES
196 002176 011033 .WORD MCERR : CONTROLLER ERROR
197 002200 011360 .WORD MDRERR : DRIVE ERROR
198 002202 011332 .WORD MNEERR : NON-EXISTANT MEMORY ERROR
199 002204 011315 .WORD MFLERR : HEADER NOT FOUND-DATA LATE
200 002206 011276 .WORD MHDERR : HEADER OR DATA ERROR
201 002210 011405 .WORD MOPERR : OPERATION INCOMPLETE
202 002212 000000 .WORD 0 : NO DRIVE STATUS AVAILABLE
203 002214 011261 .WORD MWDERR : WRITE DATA ERROR
204 002216 011243 .WORD MHCERR : HEAD CURRENT ERROR
205 002220 000000 .WORD 0
206 002222 011225 .WORD MSTERR : SEEK TIMECUT ERROR
207 002224 011172 .WORD MSPERR : SPINDLE ERROR
208 002226 011210 .WORD MWGERR : WRITE GATE ERROR
209 002230 000000 .WORD 0
210 002232 011142 .WORD MDSERR : DRIVE SELECT ERROR
211
212
213 002234 004466 : PATTBL: PATTERN TABLE
214 002236 004470 .WORD PAT1
215 002240 004530 .WORD PAT2
216 002242 004570 .WORD PAT3
217 002244 004630 .WORD PAT4
218 002246 004636 .WORD PAT5
219 002250 004676 .WORD PAT6
220 002252 004700 .WORD PAT7
221 002254 004740 .WORD PAT8
222 002256 004742 .WORD PAT9
223
224
225
226 002260 000000 : SUBSTK: SUBROUTINE CALLING STACK
227 002262 000000 .WORD 0 ;STACK IS 12 WORDS LONG
228 002264 000000 .WORD 0
229 002266 000000 .WORD 0
230 002270 000000 .WORD 0
231 002272 000000 .WORD 0
232 002274 000000 .WORD 0
233 002276 000000 .WORD 0
234 002300 000000 .WORD 0
235 002302 000000 .WORD 0
236
237 002304 000002 : I25TBL: .WORD 2 ;TABLE OF DIFFERENCES TO BE USED
238 002306 000006 .WORD 6 ;IN TEST 25
239 002310 000011 .WORD 9
240 002312 000014 .WORD 12.
```

241	002314	000021	.WORD	17.
242	002316	000026	.WORD	22.
243	002320	000033	.WORD	27.
244	002322	000042	.WORD	34.
245	002324	000051	.WORD	41.
246	002326	000200	.WORD	128.
247	002330	000377	.WORD	255.

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

250				
251	002332	000010	33TBL: .BLKW	10
252				
253	002352	002	CYI TBL: .BYTE	2
254	002353	007	.BYTE	7.
255	002354	016	.BYTE	14.
256	002355	024	.BYTE	20.
257	002356	033	.BYTE	27.
258	002357	041	.BYTE	33.
259	002360	046	.BYTE	38.
260	002361	055	.BYTE	45.
261	002362	064	.BYTE	52.
262	002363	072	.BYTE	58.
263	002364	101	.BYTE	65.
264	002365	110	.BYTE	72.
265	002366	115	.BYTE	77.
266	002367	124	.BYTE	84.
267	002370	133	.BYTE	91.
268	002371	141	.BYTE	97.
269	002372	146	.BYTE	102.
270	002373	154	.BYTE	108.
271	002374	161	.BYTE	113.
272	002375	170	.BYTE	120.
273	002376	177	.BYTE	127.
274	002377	206	.BYTE	134.
275	002400	213	.BYTE	139.
276	002401	222	.BYTE	146.
277	002402	230	.BYTE	152.
278	002403	235	.BYTE	157.
279	002404	244	.BYTE	164.
280	002405	252	.BYTE	170.
281	002406	261	.BYTE	177.
282	002407	270	.BYTE	184.
283	002410	275	.BYTE	189.
284	002411	303	.BYTE	195.
285	002412	312	.BYTE	202.
286	002413	317	.BYTE	207.
287	002414	326	.BYTE	214.
288	002415	334	.BYTE	220.
289	002416	343	.BYTE	227.
290	002417	352	.BYTE	234.
291	002420	361	.BYTE	241.
292	002421	367	.BYTE	247.
293	002422	375	.BYTE	253.
294	002423	000	.BYTE	0

;TABLE OF DEFAULT CYLINDERS

SSINDX: .WORD 0 ;SUBROUTINE STACK INDEX POINTER

```

297
298
299 002426 000000      : OPERATIONAL FLAGS
300 002430 000000      OPFLAG: .WORD 0      ; OPERATION FLAGS
301 002432 000000      DONE: .WORD 0        ; OPERATION COMPLETE FLAG
302 002434 000000      HADONE: .WORD 0      ; HEAD ALIGNMENT DONE FLAG
303 002436 000000      ERHEAD: .WORD 0     ; ADDRESS OF ERROR HEADER
304 002440 000000      MORECE: .WORD 0     ; MORE THAN 1 COMPARE ERROR
305 002442 000000      ERRSWI: .WORD 0     ; ERROR RETURN SWITCH
306 002444 000000      BSFLAG: .WORD 0    ; BAD SECTOR FLAGS
307 002446 000000      WRTSWI: .WORD 0    ; WRITE SWITCH
308                                TBLSTR: .WORD 0    ; TABLE STORAGE
309 002450 000000      RLBAS: .WORD 0     ; RL11 BASE ADDRESS
310 002452 000000      RLVEC: .WORD 0     ; RL11 VECTOR ADDRESS
311 002454 000000      RLDRV: .WORD 0     ; DRIVE NUMBER UNDER TEST
312
313 002456 000000      L.CS: .WORD 0      ; CONTROLLER REGISTER STORAGE
314 002460 000000      L.BA: .WORD 0      ; BEFORE OPERATION
315 002462 000000      L.DA: .WORD 0
316 002464 000000      L.MP: .WORD 0
317 002466 000000      T.CS: .WORD 0     ; CONTROLLER REGISTER STORAGE
318 002470 000000      T.BA: .WORD 0     ; AFTER OPERATION
319 002472 000000      T.DA: .WORD 0
320 002474 000000      T.MP: .WORD 0
321 002474 000000      HDWRD1: .WORD 0   ; HEADER WORD STORAGE
322 002476 000000      HDWRD2: .WORD 0
323 002500 000000      HDWRD3: .WORD 0
324
325 002502 000000      T.STAT: .WORD 0   ; DRIVE STATE STORAGE
326
327 002504 000000      RE SPARM: .WORD 0 ; PARAM BLOCK FOR REASON REPORT
328 002506 000000      .WORD 0
329 002510 000000      .WORD 0
330 002512 000000      .WORD 0
331 002514 000000      .WORD 0
332
333 002516 000000      DRVCNT: .WORD 0   ; DRIVE COUNT FOR DRIVES UNDER TEST
334 002520 000000      DIF AUG: .WORD 0  ; DIFFERENCE AUGMENT FOR SEEK
335 002522 000000      OLD CYL: .WORD 0  ; OLD CYLINDER
336 002524 000000      NEW CYL: .WORD 0  ; NEW CYLINDER
337 002526 000000      CUR CYL: .WORD 0  ; CURRENT CYLINDER
338 002530 000000      DES DIF: .WORD 0  ; DESIRED DIFFERENCE
339 002532 000000      DES SGN: .WORD 0  ; DESIRED SIGN
340 002534 000000      DES HD: .WORD 0   ; DESIRED HEAD
341 002536 000000      DES SEC: .WORD 0  ; DESIRED SECTOR
342 002540 000000      TEMP0: .WORD 0    ; TEMPORARY STORAGE
343 002542 000000      TEMP1: .WORD 0    ; TEMPORARY STORAGE
344 002544 000000      TEMP2: .WORD 0    ; TEMPORARY STORAGE
345 002546 000000      TEMP3: .WORD 0    ; TEMPORARY STORAGE
346 002550 000000      TEMP4: .WORD 0    ; TEMPORARY STORAGE
347 002552 000000      TEMP5: .WORD 0    ; TEMPORARY STORAGE
348 002554 000000      TEMP6: .WORD 0    ; TEMPORARY STORAGE
349 002556 000000      TEMP7: .WORD 0    ; TEMPORARY STORAGE
350 002560 000000      TEMP8: .WORD 0    ; TEMPORARY STORAGE
351
352                                : TIMER STORAGE

```

```

353 002562 000000 OF IN: .WORD 0 ;ONE CYLINDER FORWARD INNER
354 002564 000000 OF INU: .WORD 0 ;UPPER
355 002566 000000 OF MID: .WORD 0 ;ONE CYLINDER FORWARD MIDDLE
356 002570 000000 OF MIDU: .WORD 0 ;UPPER
357 002572 000000 OF OUT: .WORD 0 ;ONE CYLINDER FORWARD OUTER
358 002574 000000 OF OUTU: .WORD 0 ;UPPER
359 002576 000000 OR IN: .WORD 0 ;ONE CYLINDER REVERSE INNER
360 002600 000000 OR INU: .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 HF IN: .WORD 0 ;128 CYLINDER FORWARD INNER
366 002614 000000 HF INU: .WORD 0 ;UPPER
367 002616 000000 HF OUT: .WORD 0 ;128 CYLINDER FORWARD OUTER
368 002620 000000 HF OUTU: .WORD 0 ;UPPER
369 002622 000000 HR IN: .WORD 0 ;128 CYLINDER REVERSE INNER
370 002624 000000 HR INU: .WORD 0 ;UPPER
371 002626 000000 HROUT: .WORD 0 ;128 CYLINDER REVERSE OUTER
372 002630 000000 HROUTU: .WORD 0 ;UPPER
373 002632 000000 AF MID: .WORD 0 ;256 CYLINDER FORWARD
374 002634 000000 AF MIDU: .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 EXRGT: .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

465	004646	055044			.WORD	055044
466	004650	026442			.WORD	026442
467	004652	013211			.WORD	013211
468	004654	105504			.WORD	105504
469	004656	042642			.WORD	042642
470	004660	021321			.WORD	021321
471	004662	110550			.WORD	110550
472	004664	044264			.WORD	044264
473	004666	022132			.WORD	022132
474	004670	011055			.WORD	011055
475	004672	104426			.WORD	104426
476	004674	042213			.WORD	042213
477						
478	004676	177777		PAT7:	.WORD	177777
479						
480	004700	045513		PAT8:	.WORD	045513
481	004702	122645			.WORD	122645
482	004704	151322			.WORD	151322
483	004706	064551			.WORD	064551
484	004710	132264			.WORD	132264
485	004712	055132			.WORD	055132
486	004714	026455			.WORD	026455
487	004716	113226			.WORD	113226
488	004720	045513			.WORD	045513
489	004722	122645			.WORD	122645
490	004724	151322			.WORD	151322
491	004726	064551			.WORD	064551
492	004730	132264			.WORD	132264
493	004732	055132			.WORD	055132
494	004734	026455			.WORD	026455
495	004736	113226			.WORD	113226
496						
497	004740	125252		PAT9:	.WORD	125252
498						
499	004742	155555		PAT10:	.WORD	155555
500						
501	004744			ENDMOD		
502						
506	004744			BGNMOD	GLBXT	
507	004744	042523	045505	000040	MSEEK:	.ASCIZ /SEEK /
508	004752	042522	042101	042040	MREAD:	.ASCIZ /READ DATA /
509	004765	122	040505	020104	MREADH:	.ASCIZ /READ HEADER /
510	005002	051127	052111	020105	MWRCHK:	.ASCIZ /WRITE CHECK/
511	005016	051127	052111	020105	MWRITE:	.ASCIZ /WRITE DATA /
512	005032	042507	020124	052123	MGTSTA:	.ASCIZ /GET STATUS /
513	005046	044527	044124	042040	MDATCP:	.ASCIZ /WITH DATA COMPARE /
514	005071	127	052111	020110	MHDRCP:	.ASCIZ /WITH HDR COMPARE /
515	005113	106	051117	032040	M40HDR:	.ASCIZ /FOR 40 HDRS/
516	005127	127	052111	020110	MWRSET:	.ASCIZ /WITH RESET /
517	005143	117	042520	040522	MOPER:	.ASCIZ /OPERATION: /
518	005157	122	051505	046125	MRESLT:	.ASCIZ /RESULT: /
519	005173	125	046116	020104	MULOAD:	.ASCIZ /UNLD DRV/
520	005204	042114	042040	053122	MCYLUP:	.ASCIZ /LD DRV /
521	005214	047506	020114	020060	MOUTIN:	.ASCIZ /FOL 0 TO CC SEEK/
522	005235	106	046117	031040	MINOUT:	.ASCIZ /FOL 255 TO CC SEEK/
523	005260	047506	020114	051127	MFOUWRT:	.ASCIZ /FOL WRITE (NU SEEK)/

524	005304	042101	020112	054503	MREVSX:	.ASCIZ	/AD. YL WRTTN AFTER REV SK/
525	005337	101	045104	041440	MFWDK:	.ASCIZ	/ADJ CYL WRTTN AFTER FWD SK/
526	005372	045523	043040	042127	MFWSKO:	.ASCIZ	/SK FWD,WRT - SK REV,OVERWRT/
527	005426	045523	051040	053105	MRESKO:	.ASCIZ	/SK REV,WRT - SK FWD,OVERWRT/
528	005462	047117	041040	042101	MBADAD:	.ASCIZ	/ON BAD SEC FILES/
529	005503	103	047101	052047	MBADSF:	.ASCIZ	/CAN'T GET BAD SEC FILES/
530	005533	102	042101	051440	MFMTER:	.ASCIZ	/BAD SEC FILE FMT ERR/
531	005560	047524	046440	047101	MTMBS:	.ASCIZ	/TO MANY BAD SEC FOR PROG CAPACITY/
532	005622	052502	020123	042101	BASADD:	.ASCIZ	/BUS ADD-/
533	005633	104	053122	000075	DRVNAM:	.ASCIZ	/DRV-/
534	005640	051104	053111	020105	DRVNAV:	.ASCIZ	/DRIVE UNAVAILABLE FOR TEST/
535	005673	104	053122	042040	NOFWR:	.ASCIZ	/DRV DID NOT REC'R FROM PWR FAIL/
536	005733	122	041514	000123	CSNAM:	.ASCIZ	/RLCS/
537	005740	046122	040502	000	BANAM:	.ASCIZ	/RLBA/
538	005745	122	042114	000101	DANAM:	.ASCIZ	/RLDA/
539	005752	046122	050115	000	MPNAM:	.ASCIZ	/RLMP/
540	005757	117	020120	047111	LAB1:	.ASCIZ	/OP INIT = /
541	005772	050117	042040	047117	LAB2:	.ASCIZ	/OP DONE - /
542	006005	127	051117	020104	MWORD:	.ASCIZ	/WORD /
543	006013	111	052116	050122	MTOSLOW:	.ASCIZ	/INTRPT TO LATE/
544	006032	050117	020111	042523	MDRRES:	.ASCIZ	/OPI SET-NO DRV RESPONSE/
545	006062	047516	044440	052116	MNOINT:	.ASCIZ	/NO INTRPT ON CMND COMPLETE/
546	006115	103	052116	051114	MCONHNG:	.ASCIZ	/CNTLR HUNG (NO RDY)/
547	006141	105	051122	042040	MNOCLR:	.ASCIZ	/ERR DID NOT CLR/
548	006161	126	046117	041440	VCNRST:	.ASCIZ	/VOL CHK NOT RSET/
549	006202	047125	050130	052103	UNXERR:	.ASCIZ	/UNXPCTED ERR/
550	006217	040	042524	052123	TSTLAB:	.ASCIZ	/TEST/
552	006225	115	047101	044440	MISTST:	.ASCIZ	/MAN INTERVENT STAT/
553	006250	052123	052101	020105	NSTACHG:	.ASCIZ	/STATE CHG/
554	006262	050123	042116	020114	SPDERR:	.ASCIZ	/SPNDL TIMEOUT FAILED TO SET/
555	006316	040506	046111	043040	GSTER1:	.ASCIZ	/FAIL FORCING DRV SEL ERR/
556	006347	111	044516	020124	INITST:	.ASCIZ	/INIT STATE/
557	006362	051104	020126	042523	T05ERR:	.ASCIZ	/DRV SELECT/
558	006375	104	053122	051040	T09ERR:	.ASCIZ	/DRV RDY/
559	006405	123	042505	020113	T10ERR:	.ASCIZ	/SEEK SGN SWITCH/
560	006425	110	020104	053523	T12ERR:	.ASCIZ	/HD SWITCH/
561	006437	122	020104	042110	T13ERR:	.ASCIZ	/RD HDR (P1)/
562	006453	122	020104	042110	T14ERR:	.ASCIZ	/RD HDR (P2)/
563	006467	127	052122	046040	T16ERR:	.ASCIZ	/WRT LCK/
565	006477				P2T01E:		
566	006477	104	043111	020106	P2T02E:	.ASCIZ	/DIFF OF 1 SEEK/
567	006516	052517	020124	051107	P2T03E:	.ASCIZ	/OUT GRD BAND DETECT/
568	006542	047111	020103	042523	P2T04E:	.ASCIZ	/INC SEEK FWD HD 0/
569	006564	047111	020103	042523	P2T05E:	.ASCIZ	/INC SEEK REV HD 0/
570	006606	047111	020103	042523	P2T06E:	.ASCIZ	/INC SEEK FWD HD 1/
571	006630	047111	020116	051107	P2T07E:	.ASCIZ	/INN GRD BAND DETECT/
572	006654	047111	020103	042523	P2T08E:	.ASCIZ	/INC SEEK REV HD 1/
573	006676	042523	045505	000	P2T09E:	.ASCIZ	/SEEK/
574	006703	106	042127	047440	P2T10E:	.ASCIZ	/FWD OSC SEEK/
575	006720	042522	020126	051517	P2T11E:	.ASCIZ	/REV OSC SEFK/
576	006735	123	042505	020113	P2T12E:	.ASCIZ	/SEEK TIMING/
577	006751	102	051501	041511	P2T13E:	.ASCIZ	/BASIC READ DATA/
578	006771	127	052122	051057	P2T14E:	.ASCIZ	&WRT/READ DATA (P1)&
579	007014	050123	047111	046104	P2T15E:	.ASCIZ	/SPINDLE ROTATION TIMING/
580	007044	051127	027524	042522	P2T16E:	.ASCIZ	&WRT/READ DATA (P2)&
581	007067	127	052122	046040	P2T17E:	.ASCIZ	/WRT LCK ERR AND DATA PROTECTION/

582	007127	101	045104	041440	P2T18E:	.ASCIZ	/ADJ CYL INTERFERENCE/
583	007154	053117	051105	051127	P2T19E:	.ASCIZ	/OVERWRITE/
584	007166	042523	045505	052040	SKTMES:	.ASCIZ	/SEEK TIMES /
585	007202	050123	047111	046104	SRTMES:	.ASCIZ	/SPINDLE ROTATION TIME /
586	007231	050	052123	052101	VALDES:	.ASCIZ	/((STATED IN 100'S OF MICRO SEC)/
587	007270	050101	051120	054117	MAPROX:	.ASCIZ	/APPROX /
588	007300	047111	042516	000122	LABIN:	.ASCIZ	/INNER/
589	007306	044515	042104	042514	LABMID:	.ASCIZ	/MIDDLE/
590	007315	117	052125	051105	LABOUT:	.ASCIZ	/OUTER/
591	007323	105	050130	041505	LABEXP:	.ASCIZ	/EXPECTED/
592	007334	030060	020061	054503	LABOCF:	.ASCIZ	/001 CYL FWD/
593	007350	030060	020061	054503	LABOCR:	.ASCIZ	/001 CYL REV/
594	007364	031061	020070	054503	LABHCF:	.ASCIZ	/128 CYL FWD/
595	007400	031061	020070	054503	LABHCR:	.ASCIZ	/128 CYL REV/
596	007414	032462	020065	054503	LABACF:	.ASCIZ	/255 CYL FWD/
597	007430	032462	020065	054503	LABACR:	.ASCIZ	/255 CYL REV/
598	007444	042110	020123	040506	HDMOVF:	.ASCIZ	/HDS FAILED TO MOVE IN 10 TRIES/
600	007503	103	046131	050040	CYLPER:	.ASCIZ	/CYL PORTION OF HDRS DIFFER WHEN READ FROM TRK 0 & 1/
601	007567	110	040505	020104	HAMES1:	.ASCIZ	/HEAD ALIGN. RSET WRT LCK TO SEL HD 0, SET FOR HD 1/
602	007652	054524	042520	021040	HAMES2:	.ASCIZ	/TYPE 'CTL C' & 'CONT' TO CONTINUE TESTING/
603	007724	041101	053117	020105	OPR002:	.ASCIZ	/ABOVE CONDITIONS MET/
604	007751	127	051501	046040	OPR003:	.ASCIZ	/WAS LOAD DEPRESSED/
605	007774	044103	020113	051104	OPR1:	.ASCIZ	/CHK DRV IS UNLDED, COVER OPN, AND WRTE LCKED /
606	010052	046103	042523	041440	OPR2:	.ASCIZ	/CLSE COVER & RST WRT LCK /
607	010104	051120	051505	020123	OPR3:	.ASCIZ	/PRESS LOAD /
608	010120	051120	051505	020123	OPR5:	.ASCIZ	/PRESS LOAD & WAIT FOR LOAD LIGHT /
609	010162	051120	051505	020123	OPR6:	.ASCIZ	/PRESS LOAD & WAIT FOR RDY /
610	010215	122	046505	053117	OPR7:	.ASCIZ	/REMOVE ADD PLGS EXCPT /
611	010244	047111	051123	020124	OPR8:	.ASCIZ	/INSRT ADD PLG /
612	010263	111	020116	046101	OPR9:	.ASCIZ	/IN ALL DRVS /
613	010300	047111	052523	043106	OPR10:	.ASCIZ	/INSUFFICIENT DRVS FOR DRV SEL ERR TST/
614	010346	050122	041514	020105	OPR11:	.ASCIZ	/RPLCE ADD PLGS AS BEFORE/
616	010377	122	051505	052105	OPR12:	.ASCIZ	/RESET WRT LCK /
617	010416	047117	000040		OPR1A:	.ASCIZ	/ON /
618	010422	047117	042040	053122	OPR1B:	.ASCIZ	/ON DRV /
619	010432	047125	042504	020122	UNDTST:	.ASCIZ	/UNDER TEST/
620	010445	123	052105	053440	OPR004:	.ASCIZ	/SET WRT LCK /
621	010462	044504	043106	000040	DIFWD:	.ASCIZ	/DIFF /
622	010470	043523	020116	000	SGNWD:	.ASCIZ	/SGN /
623	010475	110	020104	000	HDWD:	.ASCIZ	/HD /
624	010501	123	041505	000040	SECWD:	.ASCIZ	/SEC /
625	010506	054503	020114	000	CYLWD:	.ASCIZ	/CYL /
626	010513	106	047522	020115	FRMWD:	.ASCIZ	/FROM /
627	010521	040	054502	040520	BYPNM:	.ASCIZ	/ BYPASSED /
628	010534	047522	052125	047111	SEQMES:	.ASCIZ	/ROUTINE TRACE SEQ (IN SEQ CALLED):/
629	010577	104	053122	051440	STAMES:	.ASCIZ	/DRV STAT/
630	010610	040502	020104	042523	BSNSTR:	.ASCIZ	/BAD SEC FILES NOT STPD. ALL SEC ASSUMED GOOD./
631	010666	047524	020124	047503	TCERR:	.ASCIZ	/TOT COMPARE ERRS: /
632							
633							
634	010711	104	053122	051040	MDRDY:	.ASCIZ	/DRV RDY /
635	010722	047503	052116	042440	MCERR:	.ASCIZ	/CONT ERR /
636	010734	042110	020122	051103	MHCRC:	.ASCIZ	/HDR CRC/
637	010744	040504	040524	041440	MDCRC:	.ASCIZ	/DATA CRC/
638	010755	110	051104	047040	MHNF:	.ASCIZ	/DR NOT FND/
639	010771	104	052101	020101	MDLT:	.ASCIZ	/DATA LATE/

640	011003	110	051104	047040	MHFCRC:	.ASCIZ	&HDP NOT FND/HDR CRC/OPI&
641	011033	104	053122	042440	MDRERR:	.ASCIZ	/DRV ERR /
643	011044	042523	042514	052103	MHSTA:	.ASCIZ	/SELECTED HD /
644	011061	126	046117	041440	MVOLCK:	.ASCIZ	/VOL CHK /
645	011072	047503	042526	020122	MCOSTA:	.ASCIZ	/COVER OPN /
646	011105	102	052522	044123	MBHSTA:	.ASCIZ	/BRUSH HME /
647	011120	051127	020124	041514	MWLSTA:	.ASCIZ	/WRT L K /
648	011131	110	051504	047440	MHOSTA:	.ASCIZ	/HDS OUT /
650	011142	051104	020126	042523	MDSERR:	.ASCIZ	/DRV SEL ERR /
651	011157	104	053122	051440	MDRVST:	.ASCIZ	/DRV STATE /
652	011172	050123	047111	052040	MSPERR:	.ASCIZ	/SPIN TIMEOUT /
653	011210	051127	020124	040507	MWGERR:	.ASCIZ	/WRT GAT ERR /
654	011225	123	042505	020113	MSTERR:	.ASCIZ	/SEEK TIMEOUT /
655	011243	110	040505	020104	MHCERR:	.ASCIZ	/HEAD CUR ERR /
656	011261	127	052122	042040	MWDERR:	.ASCIZ	/WRT DAT ERR /
657	011276	050117	044440	041516	MOPERR:	.ASCIZ	/OP INCOMPLETE /
658	011315	110	051104	042057	MHDERR:	.ASCIZ	&HDR/DAT ERR &
659	011332	042110	020122	047516	MFLERR:	.ASCIZ	&HDR NOT FND/DAT LATE &
660	011360	047516	026516	054105	MNEERR:	.ASCIZ	/NON-EXSTNT MEM /
661	011400	054503	020114	000	MCYLOC:	.ASCIZ	/CYL /
662	011405	103	052517	042114	MNDRST:	.ASCIZ	/COULD NOT RETRIEVE DRIVE STATUS/
663	011445	125	045516	020116	MUNDEF:	.ASCIZ	/UNKN DRV STATE-NO RDY,NO ERR,HDS OUT/
664	011512	040506	046111	052040	MRLFAL:	.ASCIZ	/FAIL TO RELD HDS AFTER ERR CLEAR/
665	011553	127	044522	042524	MWRTAB:	.ASCIZ	/WRITE ABORTED/
666	011571	040	051105	020122	MEXERS:	.ASCIZ	/ ERR LIMIT EXCEEDED - UNIT DROPPED/
667	011634	042440	051122	051117	MERRS:	.ASCIZ	/ ERROR/
668	011643	207	177777	000	BELL:	.ASCIZ	<207><377><377>
669							
670					:	RESULT SETTINGS	
671	011647	111	020123	000	RESE3:	.ASCIZ	/IS /
672	011653	040	041123	000040	RESE4:	.ASCIZ	/ SB /
673							
674					:	RESULT CONDITIONS	
675	011660	044440	020116	000	RESE5:	.ASCIZ	/ IN /
676	011665	040	043117	000040	RESE6:	.ASCIZ	/ OF /
677	011672	052123	052101	020105	STATE2:	.ASCIZ	/STATE 2/
678	011702	052123	052101	020105	STATE3:	.ASCIZ	/STATE 3/
679	011712	052123	052101	020105	STATE5:	.ASCIZ	/STATE 5/
681	011722	042523	045505	053440	CDRDY:	.ASCIZ	&SEEK W/O MOTION&
683	011742	044506	051522	020124	C10MS:	.ASCIZ	/FIRST 3 MS/
684	011755	065	030060	051515	C500MS:	.ASCIZ	/500MS/
685	011763	103	041531	042514	CCYLUP:	.ASCIZ	/CYCLE UP/
686	011774	040504	040524	054040	CAFDT:	.ASCIZ	/DATA XFER/
687	012006	020065	042523	042103	CSSEC:	.ASCIZ	/5 SECDS/
688							
689	012016	047045	052045	047045	FMTOP1:	.ASCIZ	/N%T%N%T%T%06%S%T%01%N/
690	012045	045	022516	022524	FMTOP2:	.ASCIZ	/N%T%01%S1%T%01%N/
691	012067	045	022516	022524	FMTOP3:	.ASCIZ	/N%T%01%S1%T%T%N/
692	012110	052045	052045	000	FMT1:	.ASCIZ	/T%T/
693	012115	045	022516	022524	FMT1.1:	.ASCIZ	/N%T%T/
694	012124	052045	000		FMT2:	.ASCIZ	/T/
695	012127	045	000116		FMT3:	.ASCIZ	/N/
696	012132	047045	052045	052045	FMT4:	.ASCIZ	/N%T%T%N/
697	012143	045	022516	022524	FMT5:	.ASCIZ	/N%T%06%S1%T%01/
698	012163	045	022516	030523	FMT6:	.ASCIZ	/N%S11%T%S4%T%S4%T%S4%T%S4%T%S2%T/
699	012225	045	022516	022524	FMT7:	.ASCIZ	/N%T%06%S2%06%S2%06%S2%06%S3%03%S2%01%N/

```

700 012275 045 022516 022524 FMT8: .ASCIZ /%N%T%06%S2%06%S2%06%S2%06/
701 012327 045 022516 000124 FMT9: .ASCIZ /%N%T/
702 012334 052045 047445 000061 FMT11: .ASCIZ /%T%01/
703 012342 052045 047445 000063 FMT12: .ASCIZ /%T%03/
704 012350 047045 051445 030461 FMT13: .ASCIZ /%N%S11%T%03%S1%T%03%S1%T%01%S1%T%01/
705 012414 047045 052045 052045 FMT14: .ASCIZ /%N%T%T%D3%S1%T%06%S1%T%06/
706 012446 047045 051445 030461 FMT15: .ASCIZ /%N%S11%T%D3%S1%T%06%S1%T%06/
707 012502 047045 051445 022465 FMT16: .ASCIZ /%N%S5%06/
708 012513 045 030523 022460 FMT17: .ASCIZ /%S10%T%N%S11%06%N/
709 012535 045 022516 030523 FMT18: .ASCIZ /%N%S13%T%S5%T%S4%T%S5%T%N/
710 012567 045 022524 031123 FMT19: .ASCIZ /%T%S2%D6%S4%D6%S4%D6%S4%D6%N/
711 012624 052045 051445 022462 FMT20: .ASCIZ /%T%S2%D6%S14%D6%S4%D6%N/
712 012654 052045 051445 031061 FMT21: .ASCIZ /%T%S12%D6%S14%D6%N/
713 012677 045 022516 030523 FMT22: .ASCIZ /%N%S11%T%03%S1%T%01%S1%T%02/
714 012733 045 022524 022524 FMT23: .ASCIZ /%T%T%T%01%N/
715 012747 045 022516 000124 FMT24: .ASCIZ /%N%T/
716 012754 047045 042045 022462 FMT25: .ASCIZ /%N%D2%T/
717 012764 047045 051445 022461 FMT26: .ASCIZ /%N%S1%T%D4%T%T%D3%N/
718 013010 047045 052045 042045 FMT27: .ASCIZ /%N%T%D3%T%D3%N/
719 013027 045 022516 022524 FMT28: .ASCIZ /%N%T%T%T/
720 013040 ENDMOD

```

```

725
726 013040 BGNMOD GLBERR
727 : ERR1 R3 POINTS TO RESULT MESSAGE
728 : F-SULT: (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
738 : RESULT: (R3) IS 1 SB 0 (R4)
739 :
740 : ERR5 R3 POINTS TO RESULT NAME
741 : R4 POINTS TO RESULT CONDITIONS
742 : RESULT: (R3) IS 0 SB 1 (R4)
743 :
744 : ERR6 RESULT ROUTINE DETERMINES WHICH ERROR(S) ARE SET AND
745 : REPORTS ALL
746 : RESULT: 'ERROR' IS 1 SB 0
747 :
748 : ERR7 DRIVE STATE ERROR REPORT
749 : R3 CONTAINS EXPECTED STATE
750 : T STAT CONTAINS BAD STATE
751 : RESULT: DRIVE STATE IS (T.STAT, SB (R3)
752 :
753 : ERR8 HEAD POSITIONING ERROR REPORT
754 : NEWCYL CONTAINS EXPECTED CYLINDER
755 : HDWRD1 CONTAINS BAD CYLINDER
756 : RESULT: CYLINDER IS (HDWRD1) SB (NEWCYL)
757 :
758 : ERR9 UTILITY RESULT REPORT
759 : R3 POINTS TO RESULT NAME

```

```

760      :           R4 POINTS TO VALUE 1
761      :           R5 POINTS TO VALUE 2
762      :           RESULT: (R3-NAME) IS (R4-VALUE 1) SB (R5-VALUE 2)
763      :
764      :           ERR10 COMPARE ERROR REPORT
765      :           R3 CONTAINS THE BAD WORD NUMBER
766      :           R4 POINTS TO BAD WORD
767      :           R5 POINTS TO GOOD WORD
768      :           RESULT: WORD (R3) IS (R4) SB (R5)
769      :
770
771 013040 BGNMSG  ERR1
772 013040 105737 003067 TSTB NOERCT ;TEST IF ERROR COUNTING INHIBITED
773 013044 001002 BNE 1$ ;YES - SKIP
774 013046 005277 167606 INC @ERRPOINT ;ELSE BUMP ERROR COUNT
775 013052 010146 1$: MOV R1,-(SP) ;STORE R1
776 013054 004737 021516 JSR PC,RPTOP ;REPORT OPERATION
777 013060 012721 000001 MOV #1,(R1)+ ;SET PARAM NUMBER
778 013064 010321 MOV R3,(R1)+ ;INSERT MESSAGE ADDRESS POINTER
779 013066 004737 022304 JSR PC,RPTRES ;REPORT RESULTS
780 013072 004737 022512 JSR PC,RPTREM ;REPORT REMAINDER
781 013076 012601 MOV (SP)+,R1 ;RESTORE R1
782 013100 004737 016122 JSR PC,CKERLM ;GO CHECK IF ERROR COUNT EXCEEDED
783 013104 ENDMMSG
(3) 013104 L10000:
(3) 013104 104023 EMT C$MSG
784
785 013106 BGNMSG  ERR2
786 013106 005277 167546 INC @ERRPOINT ;BUMP ERROR COUNT
787 013112 010146 MOV R1,-(SP) ;STORE R1
788 013114 004737 021516 JSR PC,RPTOP ;REPORT OPERATION
789 013120 012721 000003 MOV #3,(R1)+ ;SET PARAM NUMBER
790 013124 010321 MOV R3,(R1)+ ;INSERT NAME ADD POINTER
791 013126 012721 000001 MOV #1,(R1)+ ;SET IS VALUE
792 013132 005021 CLR (R1)+ ;SET SB VALUE
793 013134 004737 022304 JSR PC,RPTRES ;REPORT RESULTS
794 013140 004737 022512 JSR PC,RPTREM ;REPORT REMAINDER
795 013144 012601 MOV (SP)+,R1 ;RESTORE R1
796 013146 004737 016122 JSR PC,CKERLM ;GO CHECK IF ERROR COUNT EXCEEDED
797 013152 ENDMMSG
(3) 013152 L10001:
(3) 013152 104023 EMT L$MSG
798
799 013154 BGNMSG  ERR3
800 013154 005277 167500 INC @ERRPOINT ;BUMP ERROR COUNT
801 013160 010146 MOV R1,-(SP) ;STORE R1
802 013162 004737 021516 JSR PC,RPTOP ;REPORT OPERATION
803 013166 012721 000003 MOV #3,(R1)+ ;SET PARAM NUMBER
804 013172 010321 MOV R3,(R1)+ ;INSERT NAME ADD POINTER
805 013174 005021 CLR (R1)+ ;SET IS VALUE
806 013176 012721 000001 MOV #1,(R1)+ ;SET SB VALUE
807 013202 004737 022304 JSR PC,RPTRES ;REPORT RESULTS
808 013206 004737 022512 JSR PC,RPTREM ;REPORT REMAINDER
809 013212 012601 MOV (SP)+,R1 ;RESTORE R1
810 013214 004737 016122 JSR PC,CKERLM ;GO CHECK IF ERROR COUNT EXCEEDED
811 013220 ENDMMSG

```

```

(3) 013220          L10002:
(3) 013220 104023   EMT      CSMSG
812
813 013222          BGNMSG  ERR4
814 013222 005277 167432   INC      @ERRPOINT      ;BUMP ERROR COUNT
815 013226 010146          MOV      R1,-(SP)        ;STORE R1
816 013230 004737 021516   JSR      PC,RPTOP       ;REPORT OPERATION
817 013234 012721 000004   MOV      #4,(R1)+       ;SET PARAM NUMBER
818 013240 010321          MOV      R3,(R1)+       ;INSERT NAME ADD POINTER
819 013242 012721 000001   MOV      #1,(R1)+       ;SET IS VALUE
820 013246 005021          CLR      (R1)+          ;SET SB VALUE
821 013250 010411          MOV      R4,(R1)        ;INSERT ADD OF CONDITION POINTER
822 013252 004737 022304   JSR      PC,RPTRES      ;REPORT RESULTS
823 013256 004737 022512   JSR      PC,RPTREM      ;REPORT REMAINDER
824 013262 012601          MOV      (SP)+,R1       ;RESTORE R1
825 013264 004737 016122   JSR      PC,CKERLM      ;GO CHECK IF ERROR COUNT EXCEEDED
826 013270          ENDMSG
(3) 013270          L10003:
(3) 013270 104023   EMT      CSMSG
827
828 013272          BGNMSG  ERR5
829 013272 005277 167362   INC      @ERRPOINT      ;BUMP ERROR COUNT
830 013276 010146          MOV      R1,-(SP)        ;STORE R1
831 013300 004737 021516   JSR      PC,RPTOP       ;REPORT OPERATION
832 013304 012721 000004   MOV      #4,(R1)+       ;SET PARAM NUMBER
833 013310 010321          MOV      R3,(R1)+       ;INSERT NAME ADD POINTER
834 013312 005021          CLR      (R1)+          ;SET IS VALUE
835 013314 012721 000001   MOV      #1,(R1)+       ;SET SB VALUE
836 013320 010411          MOV      R4,(R1)        ;INSERT ADD OF CONDITION POINTER
837 013322 004737 022304   JSR      PC,RPTRES      ;REPORT RESULTS
838 013326 004737 022512   JSR      PC,RPTREM      ;REPORT REMAINDER
839 013332 012601          MOV      (SP)+,R1       ;RESTORE R1
840 013334 004737 016122   JSR      PC,CKERLM      ;GO CHECK IF ERROR COUNT EXCEEDED
841 013340          ENDMSG
(3) 013340          L10004:
(3) 013340 104023   EMT      CSMSG
842
843 013342          BGNMSG  ERR6
844 013342 105737 003067   TSTB     NOERCT         ;TEST IF ERROR COUNTING INHIBITED
845 013346 001002          BNE     17$             ;YES - SKIP
846 013350 005277 167304   INC      @ERRPOINT      ;ELSE BUMP ERROR COUNT
847 013354 010146          17$:  MOV      R1,-(SP)        ;STORE R1
848 013356 010346          MOV      R3,-(SP)        ;STORE R3
849 013360 010446          MOV      R4,-(SP)        ;STORE R4
850 013362 010546          MOV      R5,-(SP)        ;STORE R5
851 013364 004737 021516   JSR      PC,RPTOP       ;REPORT OPERATION
852 013370 012721 000003   MOV      #3,(R1)+       ;SET PARAM NUMBER
853 013374 012761 000001 000002   MOV      #1,2(R1)       ;INSERT IS VALUE
854 013402 005037 002546   CLR      TEMP3          ;CLEAR FOR STATUS STORAGE
855 013406 013703 002466   MOV      T,CS,R3        ;GET T.CS
856 013412 042703 177761   BIC     #177761,R3      ;AND CLEAR ALL BUT FUNCTION
857 013416 022703 000004   CMP     #4,R3           ;CHECK IF IT WAS GET STATUS
858 013422 001432          BFQ     1$             ;YES - STATUS IS IN T.MP, SKIP
859 013424 012762 000003 000004   MOV     #GETSTAT,RLDA(R2) ;ELSE DO GET STATUS
860 013432 012703 000004   MOV     #4,R3
861 013436 053703 002454   BIS     RLDRV,R3
    
```

```

862 013442 010362 000000      MOV      R3,RLCS(R2)
863 013446      WAITJS  #10.          ;WAIT FOR CONTROLLER READY
(3) 013446 012700 000012      MOV      #10.,R0
(3) 013452 104027      EMT      C$WTU
864 013454 032762 000200 000000      BIT      #CRDYMSK,RLCS(R2)      ;TEST IF READY
865 013462 001003      BNE      10$          ;YES - SKIP
866 013464 012703 001000 9$:      MOV      #BIT9,R3          ;ELSE SET NO DRIVE STATUS BIT
867 013470 000413      BR       2$          ;IN MESSAGE WORD AND SKIP
868 013472 016203 000006 10$:      MOV      RLMP(R2),R3      ;STORE STATUS FOR REPORT
869 013476 010337 002546      MOV      R3,TEMP3
870 013502 113703 002547      MOV#9    TEMP3+1,R3      ;GET ERROR BITS IN PROPER POSITION
871 013506 000402      BR       13$
872 013510 113703 002475 1$:      MOV#B    T.MP+1,R3      ;GET ERROR BITS FROM MP REG
873 013514 042703 177442 13$:      BIC      #177442,R3      ;CLEAR UNUSED BITS
874 013520 013704 002466 2$:      MOV      T.CS,R4        ;GET ERROR BITS FROM CS REG
875 013524 042704 001777      BIC      #1777,R4        ;CLEAR UNUSED BITS
876 013530 050403      BIS      R4,R3          ;MAKE ONE WORD OF POSSIBLE ERRORS
877 013532 032703 002000      BIT      #OPIERR,R3      ;TEST IF OPI SET
878 013536 001442      BEQ      115$         ;NO - SKIP
879 013540 032703 010000      BIT      #HNFERR,R3      ;TEST IF HDR NOT FOUND ERROR
880 013544 001026      BNE      107$         ;YES - SKIP
881 013546 032703 004000      BIT      #HRCRCERR,R3    ;TEST IF HDR CRC ERR
882 013552 001020      BNE      105$         ;YES - SKIP
883 013554 012704 011276      MOV      #MOPERR,R4      ;SET OPI ALONE MESSAGE
884 013560 100$:      PRINTB  #FMT28,#MRSLT,R4,#MERRS ;REPORT ERROR
(10) 013560 012746 011634      MOV      #MERRS,-(SP)
(9) 013564 010446      MOV      R4,-(SP)
(8) 013566 012746 005157      MOV      #MRSLT,-(SP)
(7) 013572 012746 013027      MOV      #FMT28,-(SP)
(6) 013576 012746 000004      MOV      #4,-(SP)
(3) 013602 010600      MOV      SP,R0
(4) 013604 104014      EMT      C$PNTB
(4) 013606 062706 000012      ADD      #12,SP
885 013612 000430      BR       120$         ;SKIP
886 013614 012704 010734 105$:      MOV      #MHCRC,R4      ;HDR CRC MESSAGE
887 013620 000757      BR       100$
888 013622 032703 004000 107$:      BIT      #HRCRCERR,R3    ;TEST IF HCRC WITH HDR NOT FND
889 013626 001003      BNE      109$         ;YES - SKIP
890 013630 012704 010755      MOV      #MHNF,R4        ;MESSAGE HEADER NOT FOUND
891 013634 000751      BR       100$
892 013636 012704 011003 109$:      MOV      #MHFCRC,R4      ;HNF AND HCRC MESSAGE
893 013642 000746      BR       100$         ;SKIP
894 013644 032703 004000 115$:      BIT      #DCKERR,R3      ;TEST IF DATA CHECK SET, NOT OPI
895 013650 001403      BEQ      118$         ;NO - SKIP
896 013652 012704 010744      MOV      #MDCRC,R4      ;SET MESSAGE DATA CHECK
897 013656 000740      BR       100$         ;SKIP
898 013660 032703 010000 118$:      BIT      #DLTERR,R3      ;TEST IF DATA LATE ERROR
899 013664 001403      BEQ      120$         ;NO - SKIP
900 013666 012704 010771      MOV      #MDLT,R4        ;SET MESSAGE DATA LATE
901 013672 000732      BR       100$         ;SKIP
902 013674 012705 100000 120$:      MOV      #BIT15,R5      ;SET BIT POINTER FOR TEST
903 013700 005004      CLR      R4            ;CLEAR R4 FOR TABLE COUNT
904 013702 030503 3$:      BIT      R5,R3          ;TEST IF BIT IS SET
905 013704 001005      BNE      6$           ;YES - SKIP TO REPORT
906 013706 005724 4$:      TST     (R1)+          ;ELSE BUMP TABLE POINTER
907 013710 000241      CLC

```

```

908 013712 006005      ROR      R5      ;SHIFT BIT POINTER TO NEXT BIT
909 013714 001372      BNE      3$      ;LOOP IF NOT 0
910 013716 000405      BR       7$      ;ELSE REPORT REMAINDER
911 013720 016411  J02174      6$:  MOV     RESTBL(R4),(R1) ;INSERT NAME ADDRESS
912 013724 004737  022304      JSR     PC,RPTRES ;REPORT RESULTS
913 013730 000766      BR       4$      ;GET NEXT BIT
914 013732 004737  022512      7$:  JSR     PC,RPTREM ;REPORT REMAINDER
915 013736 005737  002546      TST     TEMP3    ;TEST IF ANY NEW STATUS
916 013742 001414      BEQ     15$     ;NO - SKIP
917 013744      PRINTB #FMT17,#STAMES,TEMP3
(9) 013744 013746 002546      MOV     TEMP3,-(SP)
(8) 013750 012746 010577      MOV     #STAMES,-(SP)
(7) 013754 012746 012513      MOV     #FMT17,-(SP)
(6) 013760 012746 000003      MOV     #3,-(SP)
(3) 013764 010600      MOV     SP,R0
(4) 013766 104014      EMT     C$PNTB
(4) 013770 062706 000010      ADD     #10,SP
918 013774 032737 004000 002466 15$:  BIT     #DCKERR,T.CS ;TEST IF DATA CHECK ERROR
919 014002 001453      BEQ     25$     ;NO - SKIP
920 014004 032737 002000 002466      BIT     #OPIERR,T.CS ;TEST IF OPI SET
921 014012 001047      BNE     25$     ;YES - SKIP
922 014014 005037 002436      CLR     MORECE  ;CLEAR COMPARE ERROR COUNT
923 014020 012701 000200      MOV     #128,R1 ;SET COMPARE LENGTH
924 014024 012703 000001      MOV     #1,R3   ;SET WORD COUNT
925 014030 012705 004066      MOV     #0BUFF,R5 ;SET GOOD WORD POINTER
926 014034 012704 003466      MOV     #1BUFF,R4 ;SET TEST WORD POINTER
927 014040 021514      18$:  CMP     (R5),(R4) ;CHECK WORD
928 014042 001427      BEQ     19$     ;GOOD - SKIP
929 014044 023727 002436 000012      CMP     MORECE,#10 ;TEST IF COMPARE LIMIT REACHED
930 014052 003021      BGT     20$     ;YES - SKIP
931 014054      PRINTB #FMT15,#MWORD,R3,#RESE3,(R4),#RESE4,(R5)
(13) 014054 011546      MOV     (R5),-(SP)
(12) 014056 012746 011653      MOV     #RESE4,-(SP)
(11) 014062 011446      MOV     (R4),-(SP)
(10) 014064 012746 011647      MOV     #RESE3,-(SP)
(9) 014070 010346      MOV     R3,-(SP)
(8) 014072 012746 006005      MOV     #MWORD,-(SP)
(7) 014076 012746 012446      MOV     #FMT15,-(SP)
(6) 014102 012746 000007      MOV     #7,-(SP)
(3) 014106 010600      MOV     SP,R0
(4) 014110 104014      EMT     C$PNTB
(4) 014112 062706 000020      ADD     #20,SP
932 014116 005237 002436      20$:  INC     MORECE  ;BUMP ERROR COUNTER
933 014122 022524      19$:  CMP     (R5)+,(R4)+ ;BUMP POINTERS
934 014124 005203      INC     R3      ;BUMP COUNTER
935 014126 005301      DEC     R1      ;DEC LENGTH COUNT
936 014130 001343      BNE     18$     ;LOOP IF NOT DONE
937 014132 005737 002436      25$:  TST     MORECE  ;TEST IF ANY COMPARE ERRORS
938 014136 001421      BEQ     27$     ;NO - SKIP
939 014140 012701 000200      MOV     #128,R1 ;SET COMPARE LENGTH
940 014144      PRINTB #FMT27,#TCERR,MORECE,#RESE6,R1
(11) 014144 010146      MOV     R1,-(SP)
(10) 014146 012746 011665      MOV     #RESE6,-(SP)
(9) 014152 013746 002436      MOV     MORECE,-(SP)
(8) 014156 012746 010666      MOV     #TCERR,-(SP)
(7) 014162 012746 013010      MOV     #FMT27,-(SP)

```



```

(6) 014166 012746 000005      MOV    #5,-(SP)
(3) 014172 010600      MOV    SP,R0
(4) 014174 104014      EMT
(4) 014176 062706 000014      ADD    #14,SP
941 014202 012605      27$:  MOV    (SP)+,R5      ;RESTORE R5, 4, 3, 1
942 014204 012604      MOV    (SP)+,R4
943 014206 012603      MOV    (SP)+,R3
944 014210 012601      MOV    (SP)+,R1
945 014212 004737 016122      JSR    PC,CKERLM      ;GO CHECK IF ERROR COUNT EXCEEDED
946 014216      ENDMSG
(3) 014216      L10005:
(3) 014216 104023      EMT    C$MSG
948 014220      BGNMSG
949 014220 005277 166434      ERR7  INC    @ERRPOINT      ;BUMP ERROR COUNT
950 014224 010146      MOV    R1,-(SP)      ;STORE R1
951 014226 004737 021516      JSR    PC,RPTOP      ;REPORT OPERATION
952 014232 012721 000003      MOV    #3,(R1)+      ;SET PARAM NUMBER
953 014236 012721 011157      MOV    #MDRVST,(R1)+ ;INSERT NAME ADD POINTER
954 014242 013721 002502      MOV    T,STAT,(R1)+ ;INSERT IS VALUE
955 014246 010311      MOV    R3,(R1) ;INSERT SB VALUE
956 014250 004737 022304      JSR    PC,RPTRES     ;REPORT RESULTS
957 014254 004737 022512      JSR    PC,RPTREM     ;REPORT REMAINDER
958 014260 012601      MOV    (SP)+,R1      ;RESTORE R1
959 014262 004737 016122      JSR    PC,CKERLM    ;GO CHECK IF ERROR COUNT EXCEEDED
960 014266      ENDMSG
(3) 014266      L10006:
(3) 014266 104023      EMT    C$MSG
962 014270      BGNMSG
963 014270 005277 166364      ERR8  INC    @ERRPOINT      ;BUMP ERROR COUNT
964 014274 010146      MOV    R1,-(SP)      ;STORE R1
965 014276 010346      MOV    R3,-(SP)      ;STORE R3
966 014300 004737 021516      JSR    PC,RPTOP      ;REPORT OPERATION
967 014304 012721 000003      MOV    #3,(R1)+      ;SET PARAM NUMBER
968 014310 012721 011400      MOV    #MCYLOC,(R1)+ ;INSERT NAME ADD POINTER
969 014314 013711 002474      MOV    HDWRD1,(R1)   ;GET HEADER WORD
970 014320 012703 000007      MOV    #7,R3        ;SET SHIFT COUNT
971 014324 000241      3$:  CLC
972 014326 006011      ROR    (R1)          ;ALIGN CHAR FOR PRINTING
973 014330 005303      DEC    R3            ; AS IS VALUE
974 014332 001374      BNE    3$
975 014334 005721      TST    (R1)+        ;BUMP PARAM POINTER
976 014336 013711 002524      MOV    NEWCYL,(R1)  ;INSERT SB VALUE
977 014342 004737 022304      JSR    PC,RPTRES     ;REPORT RESULTS
978 014346 004737 022512      JSR    PC,RPTREM     ;REPORT REMAINDER
979 014352 012603      MOV    (SP)+,R3      ;RESTORE R3
980 014354 012601      MOV    (SP)+,R1      ;RESTORE R1
981 014356 004737 016122      JSR    PC,CKERLM    ;GO CHECK IF ERROR COUNT EXCEEDED
982 014362      ENDMSG
(3) 014362      L10007:
(3) 014362 104023      EMT    C$MSG
984 014364      BGNMSG
985 014364 005277 166270      ERR9  INC    @ERRPOINT      ;BUMP ERROR COUNT
986 014370 010146      MOV    R1,-(SP)      ;STORE R1
  
```

```

987 014372 004737 021516      JSR    PC,RPTOP      ;REPORT OPERATION
988 014376 012721 000003      MOV    #3,(R1)+     ;SET PARAM NUMBER
989 014402 010321              MOV    R3,(R1)+     ;INSERT NAME ADD POINTER
990 014404 010421              MOV    R4,(R1)+     ;SET IS VALUE
991 014406 010521              MCV   R7,(R1)+     ;SET SB VALUE
992 014410 004737 022304      JSR    PC,RPTRES    ;REPORT RESULTS
993 014414 004737 022512      JSR    PC,RPTREM    ;REPORT REMAINDER
994 014420 012601              MOV    (SP)+,R1     ;RESTORE R1
995 014422 004737 016122      JSR    PC,CKERLM    ;GO CHECK IF ERROR COUNT EXCEEDED
996 014426
(3) 014426
(3) 014426 104023      ENDMMSG
997 014430              BGNMSG
998 014430 010146              EMT    C$MSG
999 014432 005737 002436      MOV    R1,-(SP)     ;STORE R1
1000 014436 001051 002436      TST   MORECE        ;TEST IF 2ND BAD LINE
1001 014440 005277 166214      BNE   3$            ;YES - SKIP
1002 014444 004737 021516      INC   @ERRPOINT     ;BUMP ERROR COUNT
1003 014450              JSR    PC,RPTOP     ;REPORT OPERATION
(11) 014450 005046              PRINTB #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1> ;REPORT ID
(11) 014452 153716 002455      CLR   -(SP)
(10) 014456 012746 005633      BISB  RLDRV+1,(SP)
(9) 014462 013746 002450      MOV   #DRVNAM,-(SP)
(8) 014466 012746 005622      MOV   RLBAS,-(SP)
(7) 014472 012746 012143      MOV   #BASADD,-(SP)
(6) 014476 012746 000005      MOV   #FMT5,-(SP)
(3) 014502 010600              MOV   #5,-(SP)
(4) 014504 104014              MOV   SP,R0
(4) 014506 062706 000014      EMT   C$PNTB
1004 014512              ADD   #14,SP
(14) 014512 011546              PRINTB #FMT14,#MRSLT,#MWORD,R3,#RESE3,(R4),#RESE4,(R5)
(13) 014514 012746 011653      MOV   (R5),-(SP)
(12) 014520 011446              MOV   #RESE4,-(SP)
(11) 014522 012746 011647      MOV   (R4),-(SP)
(10) 014526 010346              MOV   #RESE3,-(SP)
(9) 014530 012746 006005      MOV   R3,-(SP)
(8) 014534 012746 005157      MCV   #MWORD,-(SP)
(7) 014540 012746 012414      MOV   #MRSLT,-(SP)
(6) 014544 012746 000010      MOV   #FMT14,-(SP)
(3) 014550 010600              MOV   #10,-(SP)
(4) 014552 104014              MOV   SP,R0
(4) 014554 062706 000022      EMT   C$PNTB
1005 014560 000421              ADD   #22,SP
1006 014562              BR    4$
(13) 014562 011546              3$: PRINTB #FMT15,#MWORD,R3,#RESE3,(R4),#RESE4,(R5) ;REPORT DATA
(12) 014564 012746 011653      MOV   (R5),-(SP)
(11) 014570 011446              MOV   #RESE4,-(SP)
(10) 014572 012746 011647      MOV   (R4),-(SP)
(9) 014576 010346              MOV   #RESE3,-(SP)
(8) 014600 012746 006005      MOV   R3,-(SP)
(7) 014604 012746 012446      MOV   #MWORD,-(SP)
(6) 014610 012746 000007      MOV   #FMT15,-(SP)
(3) 014614 010600              MOV   #7,-(SP)
(4) 014616 104014              MOV   SP,R0
(4) 014620 062706 000020      EMT   C$PNTB
1007 014624 005237 002436      ADD   #20,SP
(4) 014624 005237 002436      4$: INC   MORECE      ;INC COMPARE ERROR COUNT

```

```

1008 014630 012601          MOV      (SP)+,R1      ;RESTORE R1
1009 014632 004737 016122  JSR      PC,CKERLM    ;GO CHECK IF ERROR COUNT EXCEEDED
1010 014636          ENDMSG
      (3) 014636          L10011:
      (3) 014636 104023    EMT      C$MSG
1011 014640          ENDMOD
1012          .EVEN
1013
1014 014640          BGNMOD  HPTCODE
1015 014640          BGNHW
      (3) 014640 000005    .WORD   L10012-L$HW/2
1016 014642 174400    .WORD   174400      ;CSR BASE ADDRESS DEFAULT
1017 014644 000160    .WORD   160        ;VECTOR DEFAULT
1018 014646 000240    .WORD   240        ;PRIORITY DEFAULT
1019 014650 000000    .WORD   0          ;DRIVE NUMBER DEFAULT
1020 014652 000001    .WORD   1          ;RL11 CONTROLLER
1021 014654          ENDSW
      (3) 014654          L10012:
1022 014654          ENDMOD
1023
1024 014654          BGNMOD  SPTCODE
1025 014654          BGNHW
      (3) 014654 000006    .WORD   L10013-L$SW/2
1026 014656 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 014660 000000    LOLIMW: .WORD   0
1036 014662 000377    HILIMW: .WORD   255.
1037 014664 000000    HEADW:  .WORD   0
1038 014666 000024    ERLIMW: .WORD   20.   ;ERROR LIMIT
1039 014670 000012    DCLIMW: .WORD   10.   ;COMPARE ERROR LIMIT
1040 014672          ENDSW
      (3) 014672          L10013:
1041 014672          ENDMOD
1042
1043 014672          BGNMOD  DSPCODE
1045 014672          D:SPATCH 14
      (4) 014672 000016    .WORD   14
      (6) 014674 022776    .WORD   T1
      (6) 014676 023246    .WORD   T2
      (6) 014700 023450    .WORD   T3
      (6) 014702 024704    .WORD   T4
      (6) 014704 025512    .WORD   T5
      (6) 014706 026116    .WORD   T6
      (6) 014710 026770    .WORD   T7
      (6) 014712 027302    .WORD   T8
      (6) 014714 027366    .WORD   T9
      (6) 014716 027666    .WORD   T10
      (6) 014720 030210    .WORD   T11
      (6) 014722 030642    .WORD   T12
  
```

```

(6) 014724 031160          .WORD T13
(6) 014726 031372          .WORD T14
1050 014730          ENDMOD
1051
1052 014730          BGNMOD INITCODE
1053 014730          BGNINIT
1054 014730          SETPRI #340
(3) 014730 012700 000340  MOV #340,RO
(3) 014734 104041          EMT C$SPRI
1055 014736          MANUAL ;CHECK IF MANUAL INTERVENTION ALLOWED
(3) 014736 104051          EMT C$MANI
1056 014740          BCOMPLETE 1$ ;YES - SKIP
(2) 014740 103403          BCS 1$
1057 014742 042737 100014 014656 BIC #MITEST!DRSELT.HDALIGN,MISWIW ;CLEAR ALL MANUAL
1058 ; ; INTERVENTION FLAGS
1059 014750 005037 002424 1$: CLR SSINDX ;CLEAR SUBROUTINE STACK INDEX
1060 014754          READEF #EF.PWR ;POWER FAILURE
(3) 014754 012700 000071  MOV #EF.PWR,RO
(3) 014760 104050          EMT C$REFG
1061 014762          BNCOMPLETE 4$ ;NO, GO CHECK NEW PASS
(2) 014762 103004          BCC 4$
1062 014764 013737 002012 003072  MOV LSUNIT,PWRFLG ;SET POWER FAIL FLAG
1063 014772 000531          BR PWCON ;GO SERVICE POWER FAIL
1064 014774          4$: READEF #EF.START ;CHECK IF START
(3) 014774 012700 000040  MOV #EF.START,RO
(3) 015000 104050          EMT C$REFG
1065 015002          BNCOMPLETE RESTART ;NO - SKIP
(2) 015002 103043          BCC RESTART
1066 ; ON START INITIALIZE TO START AT FIRST DRIVE, CLEAR INTERNAL
1067 ; PASS COUNT, AND ERROR COUNT.
1068 015004 013737 002012 002516  MOV LSUNIT,DRV CNT ;SET UP UNIT COUNT
1069 015012 005037 003062  RSTRT: CLR PASNUM ;CLEAR PASS NUMBER
1070 015016 012700 002662  MOV #ERRCNT,RO
1071 015022 012701 000100  MOV #64,R1 ;GET A COUNT
1072 015026 005020          1$: CLR (RO)+ ;CLEAR A ERROR COUNTER STORAGE AREA
1073 015030 005301          DEC R1
1074 015032 001375          BNE 1$ ;LOOP TILL ALL CLEARED
1075 015034 012737 002660 002660  MOV #ERRCNT-2,ERRPOINT ;INIT ERROR POINTER
1076 015042 012737 177777 003064  MOV #-1,PSETNM ;SET PARAM SELECT TO INITIAL VALUE
1077 015050 012737 177777 002432  MOV #-1,HADONE ;PRESET HEAD ALIGN DONE FLAG
1078 015056 032737 020000 014656  BIT #HICYL,MISWIW ;TEST IF HI LIMIT SET
1079 015064 001003          BNE 3$ ;YES - SKIP
1080 015066 012737 000377 014662  MOV #377,HILIMW ;ELSE INIT HILIMIT
1081 015074 032737 040000 014656  3$: BIT #LOCYL,MISWIW ;TEST IF LO LIMIT SET
1082 015102 001002          BNE 5$ ;YES - SKIP
1083 015104 005037 014660          CLR LOLIMW ;ELSE CLEAR LO LIMIT
1084 015110 000432          5$: BR SETDON
1085 015112          RESTART:
1086 015112          READEF #EF.RESTART ;CHECK IF RESTART
(3) 015112 012700 000077  MOV #EF.RESTART,RO
(3) 015116 104050          EMT C$REFG
1087 015120          BCOMPLETE RSTRT ;NO - SKIP
(2) 015120 103734          BCS RSTRT
1088 015122          CONTINUE:
1089 015122          READEF #EF.CONTINUE ;TEST IF CONTINUE
(3) 015122 012700 000036  MOV #EF.CONTINUE,RO
  
```

```

(3) 015126 104050      EMT      C$REFG
1090 015130      BCOMPLETE PWCON
(2) 015130 103452      BCS      PWCON
1091      :      ON CONTINUE PICK UP UNIT LAST UNDER TEST
1092 015132      READEF   #EF.NEW      :CHECK IF STARTING NEW PASS
(3) 015132 012700 000035      MOV      #EF.NEW,R0
(3) 015136 104050      EMT      C$REFG
1093 015140      BCOMPLETE PASNEW
(2) 015140 103403      BCS      PASNEW
1094 015142      NXTPAS:
1095 015142 005737 002516      TST      DRVCNT      :TEST IF ALL UNITS CHECKED
1096 015146 001013      BNE      SETDON      :NO - SKIP
1097 015150 005237 003062      PASNEW: INC      PASNUM      :ELSE BUMP PASS COUNT
1098 015154 012737 002660 002660      MOV      #ERRCNT-2,ERRPOINT :INIT THE ERROR POINTER
1099 015162 013737 002012 002516      MOV      L$UNIT,DRVCNT :GET ALL DRIVES
1100 015170 012737 177777 003064      MOV      #-1,PSETNM      :SET PARAM SELECT TO INITIAL
1101 015176 005237 003064      LFTDON: INC      PSETNM      :NEXT SET OF PARAMETERS
1102 015202 005337 002516      DEC      DRVCNT      :DOWN COUNT DRIVE TOTAL
1103 015206 062737 000002 002660      ADD      #2,ERRPOINT :UPDATE THE ERROR POINTER
1104 015214 013700 003064      MOV      PSETNM,R0      :SET UP TO GET PARAMETERS
1105 015220 012702 002450      MOV      #RLBAS,R2
1106 015224      GPHARD   R0,R1
(3) 015224 104042      EMT      C$GPHRD
(3) 015226 010001      MOV      R0,R1
1107 015230      BCOMPLETE 7$      :SKIP IF GOOD PARAM
(2) 015230 103406      BCS      7$
1108 015232 005737 003072      TST      PWRFLG      :RECENT POWER FAILURE
1109 015236 001741      BEQ      NXTPAS      :NO
1110 015240 005337 003072      DEC      PWRFLG      :ACCOUNT FOR DRIVE
1111 015244 000736      BR       NXTPAS
1112 015246 012122      7$:      MOV      (R1)+,(R2)+      :STORE PARAMETERS CSR
1113 015250 012122      MOV      (R1)+,(R2)+      :VECTOR
1114 015252 005721      TST      (R1)+      :BUMP PAST PRIORITY
1115 015254 012122      MOV      (R1)+,(R2)+      :DRIVE
1116
1117 015256      PWCON: SETVEC   R1      VTHLR,#340      ;SET UP VECTOR
(7) 015256 012746 000340      MOV      '
(6) 015262 012746 016064      MOV      -(SP)
(5) 015266 013746 002452      MOV      SP)
(4) 015272 012746 000003      MOV      #3,-(SP)
(3) 015276 104037      EMT      C$SVEC
(2) 015300 062706 000010      ADD      #10,SP
1118 015304      SETPRI   #0      :SET PRIORITY
(3) 015304 012700 000000      MOV      #0,R0
(3) 015310 104041      EMT      C$SPRI
1119 015312 013702 002450      MOV      RLBAS,R2      :SET RL BASE ADDRESS POINTER
1120
1121
1122
1123      :      CHECK IF DOING AUTO SIZE AND DROP DRIVE IF NOT READY AND
1124      :      ERROR SETS ON GET STATUS.
1125 015316 005737 003062      TST      PASNUM      :TEST IF PASS 0
1126 015322 001135      BNE      22$      :NO - SKIP
1127 015324 032737 000020 014056      BIT      #AUTOSZ,MISWIW :TEST IF DOING AUTO SIZE
1128 015332 001531      BEQ      22$      :NO - SKIP
1129      :CHECK IF UNIBUS ADDRESS IS THERE BEFORE WE CHECK DRIVE READY

```

```

1130 015334 005037 003070 CLR TRPFLG ;TRAP OCCURANCE
1131 015340 SETVEC ERRVEC,#TRPHAN,#340 ;SET TRAP VECTOR
(7) 015340 012746 000340 MOV #340,-(SP)
(6) 015344 012746 016056 MOV #TRPHAN,-(SP)
(5) 015350 013746 002652 MOV ERRVEC,-(SP)
(4) 015354 012746 000003 MOV #3,-(SP)
(3) 015360 104037 EMT C$SVEC
(2) 015362 062706 000010 ADD #10,SP
1132 015366 005762 000000 TST RLCS(R2) ;ACCESS BUS
1133 015372 005737 003070 TST TRPFLG ;TRAP OCCUR??
1134 015376 001032 BNE 5$ ;YES, DON'T INVESTIGATE FURTHER
1135 015400 013705 002454 MOV RLDRV,R5 ;GET DRIVE NUMBER
1136 015404 052705 000200 BIS #CRDYMSK,R5 ;INSERT CONT READY
1137 015410 010562 000000 MOV R5,RLCS(R2) ;LOAD IN DRIVE NUMBER
1138 015414 032762 000001 000000 BIT #DRDYMSK,RLCS(R2) ;CHECK IF DRIVE IS READY
1139 015422 001072 BNE 20$ ;YES - GO DO TEST
1140 015424 012762 000003 000004 MOV #GETSTAT,RLDA(R2) ;ELSE INSERT GET STATUS
1141 015432 052705 000004 BIS #4,R5 ;LOAD R5 WITH GET STATUS FUNCTION
1142 015436 042705 000200 BIC #CRDYMSK,R5 ;CLEAR CONTROLLER READY
1143 015442 010562 000000 MOV R5,RLCS(R2) ;LOAD CS REG
1144 015446 WAITMS #4 ;WAIT 4 MS
(3) 015446 012700 000004 MOV #4,R0
(3) 015452 104026 EMT C$WTM
1145 015454 032762 002000 000000 BIT #OPIERR,RLCS(R2) ;TEST IF OPI SET
1146 015462 001452 BEQ 20$ ;NO - SKIP
1147 015464 5$: CLRVEC ERRVEC
(3) 015464 013700 002652 MOV ERRVEC,R0
(3) 015470 104036 EMT C$CVEC
1148 015472 PRINTF #FMT24,#DRVNAV
(8) 015472 012746 005640 MOV #DRVNAV,-(SP)
(7) 015476 012746 012747 MOV #FMT24,-(SP)
(6) 015502 012746 000002 MOV #2,-(SP)
(3) 015506 010600 MOV SP,R0
(4) 015510 104017 EMT C$PNTF
(4) 015512 062706 000006 ADD #6,SP
1149 015516 10$: PRINTF #FMT5,#BASADD,RLBAS,#DRVNAV,<B,RLDRV+1>
(11) 015516 005046 CLR -(SP)
(11) 015520 153716 002455 BISB RLDRV+1,(SP)
(10) 015524 012746 005633 MOV #DRVNAV,-(SP)
(9) 015530 013746 002450 MOV RLBAS,-(SP)
(8) 015534 012746 005622 MOV #BASADD,-(SP)
(7) 015540 012746 012143 MOV #FMT5,-(SP)
(6) 015544 012746 000005 MOV #5,-(SP)
(3) 015550 010600 MOV SP,R0
(4) 015552 104017 EMT C$PNTF
(4) 015554 062706 000014 ADD #14,SP
1150 015560 PRINTF #FMT3
(7) 015560 012746 012127 MOV #FMT3,-(SP)
(6) 015564 012746 000001 MOV #1,-(SP)
(3) 015570 010600 MOV SP,R0
(4) 015572 104017 EMT C$PNTF
(4) 015574 062706 000004 ADD #4,SP
1151 015600 DODU PSETNM ;DROP DRIVE
(3) 015600 013700 003064 MOV PSETNM,R0
(3) 015604 104053 EMT C$DODU
1152 015606 DODLN

```

```

(3) 015606 104044
1153 015610          20$:  EMT  C$DCLN
(3) 015610 013700 002652  CLRVEC ERRVEC
(3) 015614 104036          MOV  ERRVEC,R0
1154 015616          22$:  EMT  C$CVEC
1156
1157 015616          MANUAL          ;MANUAL INTERVENTION ALLOWED
(3) 015616 104051          EMT  C$MANI
1158 015620          BNCOMPLETE 4$          ;NO
(2) 015620 103004          BCC  4$
1159
1160 015622 005737 003062          TST  PASNUM          ;YES, CHECK PASS NUMBER
1161 015626 001001          BNE  4$          ;NOT FIRST PASS, NEED DRIVE UP
1162 015630 000437          BR   8$          ;FIRST PASS, PROGRAM WILL INSTRUCT USER
1163
1165          ;CHECK IF POWER FAILURE WAIT IS NEEDED
1166
1167 015632 005737 003072          4$:  TST  PWRFLG          ;NEEDED???
1168 015636 001434          BEQ  8$          ;NO, SKIP
1169
1170 015640 013705 002454          MOV  RLDY,R5          ;DRIVE SELECT
1171 015644 052705 000200          BIS  #CRDYMSK,R5          ;SET CRDY
1172 015650 010562 000000          MOV  R5,RLCS(R2)          ;SELECT DRIVE
1173 015654 012701 000074          MOV  #60,R1          ;SIXTY SECOND TIMER
1174 015660 032762 000001 000000 9$:  BIT  #DRDYMSK,RLCS(R2)          ;DRIVE UP YET
1175 015666 001020          BNE  8$          ;YES START TEST
1176
1177 015670          WAITMS #10.          ;WAIT A SECOND
(3) 015670 012700 000012          MOV  #10,R0
(3) 015674 104026          EMT  C$WTM
1178 015676 005301          DEC  R1          ;SIXTY GONE BY
1179 015700 001367          BNE  9$          ;NO
1180 015702          PRINTF #FMT24,#NOPWR
(8) 015702 012746 005673          MOV  #NOPWR,-(SP)
(7) 015706 012746 012747          MOV  #FMT24,-(SP)
(6) 015712 012746 000002          MOV  #2,-(SP)
(3) 015716 010600          MOV  SP,R0
(4) 015720 104017          EMT  C$PNTF
(4) 015722 062706 000006          ADD  #6,SP
1181 015726 000673          BR   10$
1182
1183 015730          8$:
1184
1185 015730          ENDINIT
(3) 015730          L10014:
(3) 015730 104011          EMT  C$INIT
1186 015732          ENDMOD
1187
1188 015732          BGNMOD  CLNCODE
1189 015732          BGNCLN
1190
1191 015732          SETVEC ERRVEC,#TRPHAN,#340
(7) 015732 012746 000340          MOV  #340,-(SP)
(6) 015736 012746 016056          MOV  #TRPHAN,-(SP)
(5) 015742 013746 002652          MOV  ERRVEC,-(SP)
(4) 015746 012746 000003          MOV  #3,-(SP)
  
```

```

(3) 015752 104037      EMT      C$SVEC
(2) 015754 062706 000010  ADD      #10,SP
1192
1193 015760      SETPRI #7          ;SET PRORITY TO 7
(3) 015760 012700 000007  MOV      #7,R0
(3) 015764 104041      EMT      C$SPRI
1194 015766 032762 000200 000000 2$:  BIT      #CRDYMSK,RLCS(R2) ;TEST IF CONTROLLER READY
1195 015774 001407      BEQ      3$          ;NO LOOP UNTIL READY
1196 015776 053762 002454 000000  BIS      RLDRV,RLCS(R2) ;SET DRIVE NUMBER
1197 016004 032762 000001 000000  BIT      #DRDYMSK,RLCS(R2) ;TEST IF DRIVE BUSY
1198 016012 001003      BNE      5$          ;NO - SKIP
1199 016014      WAITMS #3          ;WAIT 300 MS
(3) 016014 012700 000003  MOV      #3,R0
(3) 016020 104026      EMT      C$WTM
1200 016022      CLRVEC RLVEC          ;RELEASE VEC
(3) 016022 013700 002452 5$:  MOV      RLVEC,R0
(3) 016026 104036      EMT      C$CVLC
1201 016030 005737 003072  TST      PWRFLG ;PWR FAIL SET
1202 016034 001402      BEQ      7$          ;NO
1203 016036 005337 003072  DEC      PWRFLG
1204 016042      CLRVEC ERRVEC
(3) 016042 013700 002652 7$:  MOV      ERRVEC,R0
(3) 016046 104036      EMT      C$CVEC
1205 016050      ENDCLN
(3) 016050      L10015:
(3) 016050 104012      EMT      C$CLEAN
1206
1207 016052      BGNDU
1208 016052 000240      NOP
1209 016054      ENDDU
(3) 016054      L10016:
(3) 016054 104055      EMT      C$DU
1210
1211 016056      ENDMJD
1212 016056      BG*MOD GLBSUB
1213
1214 016056 005237 003070  TRPHAN: INC      TRPFLG
1215 016062 000002      RTI
1216
1217 016064      BGNSRV INTHLR
1218      ; INTERRUPT HANDLER. ABORTS WAIT TIMER AND STORES ALL RL11 REGS
1219 016064      ABORTWAIT
(3) 016064 104021      EMT      C$ABRT
1220 016066 012237 002466  MOV      (R2)+,T.CS ;STORE RL REGISTERS
1221 016072 012237 002470  MOV      (R2)+,T.BA
1222 016076 012237 002472  MOV      (R2)+,T.DA
1223 016102 011237 002474  MOV      (R2),T.MP
1224 016106 012737 177777 002430  MOV      #-1,DONE ;SET DONE FLAG
1225 016114 013702 002450  MOV      RL6AS,R2 ;RESTORE R2
1226 016120      ENDSRV
(3) 016120      L10017:
(2) 016120 000002      RTI
1227
1228      ; ERROR LIMIT CHECKING ROUTINE
1229      ; DROPS DRIVE IF ERROR LIMIT EXCEEDED
1230 016122 027737 164532 014666 (KERLM: CMP      @ERRPOINT,ERLIMW ;TEST IF ERROR LIMIT EXCEEDED

```



```

1260 016370 000207          2$:  RTS      PC          :RETURN
1261 016372          4$:  WAITMS  #3          :WAIT 300 MS FOR TIMEOUT
   (3) 016372 012700 000003    MOV      #3,R0
   (3) 016376 104026    EMT      C$WTM
1262 016400 032762 000200 000000  BIT      #CRDYMSK,RLCS(R2) :TEST IF READY NOW SET
1263 016406 001006    BNE      3$          :YES - SKIP
1264 016410 004737 016262    JSR      PC,READRL    :READ RL REGS
1265 016414 012766 006115 000002  MOV      #MCONHNG,2(SP) :SET MESSAGE FOR CONTROLLER HUNG
1266 016422 000762    BR       2$          :SKIP
1267 016424 005737 002430          3$:  TST      DONE      :ELSE CHECK IF INTERRUPT OCCURRED
1268 016430 001345    BNE      1$          :YES - SKIP TO SET TO SLOW
1269 016432 004737 016262          5$:  JSR      PC,READRL    :READ RL REGS
1270 016436 012766 006062 00000?  MOV      #MNOINT,2(SP) :ELSE SET NO INTERRUPT FLAG
1271 016444 000751    BR       2$          :GO TO RETURN
1272
1273
1274 016446 005037 002426  :STINT: OPERATION AND TEST INITIALIZE ROUTINE
1275 016452 105037 003067    CLR      OPFLAG      :CLEAR OPERATION FLAGS
1276 016456 005037 002436    CLR      NOERCT      :RESET INHIBIT ERROR COUNTING
1277 016462 000207    CLR      MORECE      :RESET MORE COMPARE ERRORS
1278
1279
1280 016464 013746 002550  :GSTATR: GET STATUS AND GET STATUS WITH RESET ROUTINE
1281 016470 012737 000013 002550  MOV      TEMP4,-(SP)   :STORE TEMP4
1282 016476 000412    MOV      #GETSTAT!DRSET,TEMP4 :SET FOR RESET
1283 016500 013746 002550  :GSTATC: BR       GSTATG
1284 016504 012737 000003 002550  MOV      TEMP4,-(SP)   :STORE TEMP4
1285 016512 000404    MOV      #GETSTAT,TEMP4 :SET FOR NO RESET
1286 016514 013746 002550  :GSTAT: BR       GSTATG
1287 016520 005037 002550  MOV      TEMP4,-(SP)   :STORE TEMP4
1288 016524 010346  :GSTATG: CLR      TEMP4      :SET FOR SAVE L. AND T. REGS
1289 016526 013703 002424    MOV      R3,-(SP)     :STORE R3
1290 016532 005723    MOV      SSINDX,R3    :GET SUBROUTINE INDEX
1291 016534 016663 000004 002260  TST      (R3)+        :BUMP IT FOR NEXT ENTRY
1292 016542 162763 000004 002260  MOV      4(SP),SUBSTK(R3) :INSERT THIS CALL
1293 016550 010337 002424    SUB      #4,SUBSTK(R3) :ADJUST IT TO CALLING LOCATION
1294 016554 010046    MOV      R3,SSINDX    :STORE IT BACK
1295 016556 010146    MOV      R0,-(SP)     :STORE R0
1296 016560 012737 000002 002440  MOV      R1,-(SP)     :STORE R1
1297 016566 032737 000010 002550  MOV      #2,ERRSWI    :SET FOR NO ERROR RETURN
1298 016574 001453    BIT      #DRSET,TEMP4 :TEST IF DRIVE RESET
1299 016576 032762 040000 000000  BEQ      11$          :NO - SKIP
1300 016604 001403    BIT      #DRVERR,RLCS(R2) :TEST IF DRIVE ERROR SET
1301 016606          BEQ      49$          :NO - SKIP
   (3) 016606 012700 000003    WAITMS  #3          :WAIT FOR 300 MS FOR DRIVE TO SETTLE
   (3) 016612 104026    MOV      #3,R0
1302 016614 012701 000062          49$:  EMT      C$WTM
1303 016620 004737 016514          50$:  MOV      #50,R1      :SET WAIT FOR 5 SEC
1304 016624 017260    JSR      PC,GSTAT     :GET DRIVE STATUS
1305 016626 032737 000001 002466  BIT      #DRDYMSK,T.CS :TEST IF DRIVE READY
1306 016634 001051    BNE      5$          :YES - GO DO CLEAR
1307 016636 032737 000020 002474  BIT      #HOSTAT,T.MP  :ELSE TEST IF HEADS OUT
1308 016644 001010    BNE      51$          :YES - BYPASS RELOAD WAIT FLAG SETTING
1309 016646 032737 144000 002474  BIT      #SPDSTAT!HCESTAT!WDESTAT,T.MP :TEST IF DRIVE HAS ERROR
1310
1311
;THAT CAUSED HEADS TO
;UNLOAD

```

1312	016654	001441				BEQ	5\$:NO - SKIP
1313	016656	052737	040000	002426		BIS	#RELDWT,OPFLAG		:ELSE SET WAIT FLAG
1314	016664	000435				BR	5\$:SKIP TO CLEAR
1315	016666	032737	040000	002466	51\$:	BIT	#DRVERR,T.CS		:TEST IF DRIVE EPROR NOW
1316	016674	001031				BNE	5\$:YES - SKIP TO CLEAR
1317	016676					WAITMS	#1		:WAIT FOR DRIVE TO GET ERROR, RDY, OR HU
(3)	016676	01270J	000001			MOV	#1,R0		
(3)	016702	104026				EMT	C\$WTM		
1318	016704	005301				DEC	R1		:DEC WAIT COUNTER
1319	016706	001344				BNE	50\$:IF NOT DONE, LOOP
1320	016710	012703	011445			MOV	#MUNDEF,R3		:MESSAGE FOR UNDEFINED STATE
1321	016714					ERRHRD	10001.,ERR1		
(3)	016714	104443				TRAP	T\$ERRCODE		
(5)	016716	023421				.WORD	10001		
(5)	016720	013040				.WORD	ERR1		
1322	016722	000554				BR	14\$:EXIT
1323	016724	005737	002550		1	TST	TEMP4		:TEST IF SAVE REGISTERS
1324	016730	001013				BNE	5\$:NO SKIP
1325	016732	012701	000004			MOV	#4,R1		:SET SAVE COUNT
1326	016736	012703	002466			MOV	#L,MP+2,R3		:SET ADDRESS OF FIRST SAVE
1327	016742	014346			8\$:	MOV	-(R3),-(SP)		:PUT REG ON STACK
1328	016744	005301				DEC	R1		:DEC COUNT
1329	016746	001375				BNE	8\$:LOOP UNTIL ALL SAVED
1330	016750	012737	000003	002462		MOV	#GETSTAT,L.DA		:SET FOR GET STATUS
1331	016756	000403				BR	6\$:SKIP
1332	016760	013737	002550	002462	5\$:	MOV	TEMP4,L.DA		:INSERT PRESET FOR STATUS
1333	016766				6\$:				
1334	016766	005037	002430			CLR	DONE		:CLEAR INTERRUPT FLAG
1335	016772	013737	002454	002456		MOV	RLDRV,L.CS		:SET UP TO GET STATUS
1336	017000	042737	002000	002456		BIC	#BIT10,L.CS		:CLEAR FOR DRIVE 4 - 7 SPEC'D
1337	017006	052737	000104	002456		BIS	#GTSTAT,L.CS		
1338	017014	013762	002462	000004		MOV	L.DA,RLDA(R2)		:LOAD RL REGS
1339	017022	013762	002456	000000		MOV	L.CS,RLCSR(R2)		:LOAD CS REG
1340	017030					WAITUS	#1		:WAIT 100 US FOR INTERRUPT
(3)	017030	012700	000001			MOV	#1,R0		
(3)	017034	104027				EMT	C\$WTU		
1341	017036	005737	002430			TST	DONE		:CHECK IF INTERRUPT OCCURRED
1342	017042	001476				BEQ	1\$:NO - SKIP
1343	017044	013737	002474	002502	4\$:	MOV	T.MP,T.STAT		:STORE MP REGISTER
1344	017052	042737	177770	002502		BIC	#C<STAMSK>,T.STAT		:CLEAR ALL BUT STATE
1345	017060	032737	000010	002462		BIT	#DRSET,L.DA		:TEST IF RESET WAS SPECIFIED
1346	017066	001474				BEQ	3\$:NO - SKIP TO EXIT
1347	017070	032737	040000	002426		BIT	#RELDWT,OPFLAG		:TEST IF RELOAD WAIT FLAG SET
1348	017076	001424				BEQ	12\$:NO - SKIP
1349	017100	012701	001130			MOV	#600.,R1		:SET WAIT COUNT FOR 60 SECONDS
1350	017104	032762	000001	000000	13\$:	BIT	#DRDYMSK,RLCS(R2)		:TEST IF DRIVE NOW READY
1351	017112	001016				BNE	12\$:YES - SKIP
1352	017114					WAITMS	#1		:CALL WAIT
(3)	017114	012700	000001			MOV	#1,R0		
(3)	017120	104026				EMT	C\$WTM		
1353	017122	005301				DEC	R1		:DEC COUNT
1354	017124	001367				BNE	13\$:LOOP IF NOT 0
1355	017126	004737	016514			JSR	PC,GSTAT		:GET DRIVE STATUS
1356	017132	017260				3\$:ERROR RETURN
1357	017134	012703	011512			MOV	#MUNDEF,R3		:SET RESULT MESSAGE POINTER
1358	017140					ERRHRD	10003.,ERR1		

(3)	017140	104443			TRAP	T\$ERCODE	
(5)	017142	023423			.WORD	10003	
(5)	017144	013040			.WORD	ERR1	
1359	017146	000442			BR	14\$:GO TO EXIT
1360	017150			12\$:	WAITUS	#10.	:WAIT FOR 1MS
(3)	017150	012700	000012		MOV	#10.,R0	
(3)	017154	104027			FMT	C\$WTU	
1361	017156	004737	016514		JSR	PC,GSTAT	:GET DRIVE STATUS
1362	017162	017260			3\$		
1363	017164	032737	100000	002466	BIT	#ANYERR,T.CS	:TEST IF ANY ERROR
1364	017172	001432			BEQ	3\$:NO - SKIP
1365	017174	032737	001000	002474	BIT	#VCSTAT,T.MP	:CHECK IF VOLUME CHECK RESET
1366	017202	001403			BEQ	7\$:YES SKIP
1367	017204	012703	00616		MOV	#VCNRST,R3	:SET REASON POINTER
1368	017210	000416			BR	2\$:EXIT
1369	017212	032737	040000	002466	7\$:	BIT	#DRVERR,T.CS
1370	017220	001404			BEQ	9\$:CHECK IF DRIVE ERROR
1371	017222				ERRHRD	10004...ERR6	:NO - SKIP
(3)	017222	104443			TRAP	T\$ERCODE	
(5)	017224	023424			.WORD	10004	
(5)	017226	013342			.WORD	ERR6	
1372	017230	000411			BR	14\$:EXIT
1373	017232	012703	006202		9\$:	MOV	#LXERR,R3
1374	017236	000403			BR	2\$:SET REASON POINTER
1375	017240	004737	016314		1\$:	JSR	PC,WAITIN
1376	017244	012603			MOV	(SP)+,R3	:EXIT
1377	017246				2\$:	ERRHRD	10002...ERR1
(3)	017246	104443			TRAP	T\$ERCODE	:WAIT FOR INTERRUPT
(5)	017250	023422			.WORD	10002	:STORE REASON POINTER FOR RETURN
(5)	017252	013040			.WORD	ERR1	
1378	017254	005037	002440		14\$:	CLR	ERRSWI
1379	017260	005737	002550		3\$:	TST	TEMP4
1380	017264	001007			BNE	22\$:CLEAR FOR ERROR RETURN
1381	017266	012703	002456		MOV	#L.CS,R3	:TEST IF REGISTERS WERE SAVED
1382	017272	012701	000004		MOV	#4,R1	:NO - SKIP
1383	017276	012623			20\$:	MOV	(SP)+,(R3)+
1384	017300	005301			DEC	R1	:SET POINTER TO RESTORE
1385	017302	001375			BNE	20\$:SET REGISTER COUNT
1386	017304	162737	000002	002424	22\$:	SUB	#2,SSINDX
1387	017312	012601			MOV	(SP)+,R1	:RESTORE REG
1388	017314	012600			MOV	(SP)+,R0	:DEC COUNT
1389	017316	012603			MOV	(SP)+,R3	:LOOP UNTIL ALL ARE RESTORED
1390	017320	012637	002550		MOV	(SP)+,TEMP4	:REMOVE ENTRY FROM SUBROUT STACK
1391	017324	005737	002440		TST	ERRSWI	:RESTORE R3
1392	017330	001403			BEQ	99\$:RESTORE TEMP4
1393	017332	063716	002440		ADD	ERRSWI,(SP)	:TEST IF ERROR RETURN
1394	017336	000207			RTS	PC	:YES - SKIP
1395	017340	017616	000000		99\$:	MOV	@(SP),(SP)
1396	017344	000207			RTS	PC	:ADD IN ERROR RETURN
1397							:SET ERROR RETURN ADDRESS
1398							
1497	017346	010346			SIMSEK:	MOV	R3,-(SP)
1498	017350	013703	002424		MOV	SSINDX,R3	:STORE REGISTERS
1499	017354	005723			TST	(R3)+	:GET SUBROUTINE INDEX
1500	017356	016663	000002	002260	MOV	2(SP),SUBSTK(R3)	:BUMP IT FOR NEXT ENTRY
1501	017364	162763	000004	002260	SUB	#4,SUBSTK(R3)	:INSERT THIS CALL
							:ADJUST IT TO CALLING LOCATION

```

1502 017372 010337 002424      MOV      R3,SSINDX      ;STORE IT BACK
1503 017376 010046              MOV      R0,-(SP)
1504 017400 010446              MOV      R4,-(SP)
1505 017402 012737 000002 002440  MOV      #2,ERRSWI      ;SET FOR NO ERROR RETURN
1506 017410 004737 017632      JSR      PC,RDYCHK      ;CHECK IF DRIVE READY
1507 017414 017574 65$
1508 017416 012704 002456      MOV      #L,CS,R4      ;GET POINTER TO L REGS
1509 017422 012714 000106      MOV      #SEEK,(R4)    ;SET FOR SEEK
1510 017426 053714 002454      BIS      RLDRV,(R4)    ;INSERT DRIVE NUMBER
1511 017432 042724 002000      BIC      #BIT10,(R4)+  ;CLEAR FOR DRIVE 4 - 7 SPEC'D
1512 017436 005024              CLR      (R4)+         ;CLEAR BUS ADDRESS
1513 017440 013714 002530      MOV      DESDIF,(R4)   ;LOAD DIFFERENCE
1514 017444 012703 000007      MOV      #7,R3        ;SET COUNT FOR SHIFT TO ALIGN
1515 017450 006314 3$:      ASL      (R4)         ;ALIGN DIFFERENCE IN DA
1516 017452 005303              DEC      R3
1517 017454 001375 3$:      BNE      3$
1518 017456 005737 002532      TST      DESSGN       ;TEST IF SIGN SET
1519 017462 001402 5$:      BEQ      5$          ;NO - SKIP
1520 017464 052714 000004      BIS      #DIRBIT,(R4) ;INSERT SIGN
1521 017470 005737 002534      TST      DESHD        ;TEST IF HEAD 0
1522 017474 001402 7$:      BEQ      7$          ;YES - SKIP
1523 017476 052714 000020      BIS      #HDSEL,(R4)  ;INSERT HEAD BIT
1524 017502 052724 000001 7$:      BIS      #MBSET0,(R4)+;INSERT MARKER BIT
1525 017506 005037 002430      CLR      DONE        ;CLEAR INTERRUPT FLAG
1526 017512 012701 000012      MOV      #10,R1       ;SET WAIT COUNT FOR 800US
1527 017516 014462 000004      MOV      -(R4),RLDA(R2);LOAD RL REGISTERS
1528 017522 014462 000002      MOV      -(R4),RLBA(R2)
1529 017526 014462 000000      MOV      -(R4),RLCS(R2)
1530 017532 005737 002430 10$:      TST      DONE        ;CHECK IF INTERRUPTED
1531 017536 001016 65$:      BNE      65$        ;YES - SKIP
1532 017540 005301 13$:      DEC      R1          ;DEC WAIT COUNT
1533 017542 001404              BEQ      13$         ;IF 0 - SKIP
1534 017544              WAITUS  #1
   (3) 017544 012700 000001      MOV      #1,R0
   (3) 017550 104027              EMT      CSWTU
1535 017552 000767 13$:      BR       10$        ;GO CHECK DONE
1536 017554 004737 016314      JSR      PC,WAITIN    ;GO WAIT FOR TIMEOUT
1537 017560 012603              MOV      (SP)+,R3     ;GET RESULT MESSAGE POINTER
1538 017562              ERRHRD 10011,ERR1
   (3) 017562 104443              TRAP    T$ERCODE
   (5) 017564 023433              .WORD  10011
   (5) 017566 013040              .WORD  ERR1
1539 017570 005037 002440      CLR      ERRSWI      ;CLEAR FOR ERROR ERROR RETURN
1540 017574 14$:
1541 017574 162737 000002 002424 65$:      SUB      #2,SSINDX    ;REMOVE ENTRY FROM SUBROUT STACK
1542 017602 012604              MOV      (SP)+,R4     ;RESTORE REGS
1543 017604 012600              MOV      (SP)+,R0
1544 017606 012603              MOV      (SP)+,R3
1545 017610 005737 002440      TST      ERRSWI      ;TEST IF ERROR RETURN
1546 017614 001403 99$:      BEQ      99$        ;YES - SKIP
1547 017616 063716 002440      ADD      ERRSWI,(SP)  ;ADD IN ERROR RETURN
1548 017622 000207              RTS      PC
1549 017624 017616 000000 99$:      MOV      @ (SP),(SP)  ;SET ERROR RETURN ADDRESS
1550 017630 000207              RTS      PC
1552
1628 ;      DRIVE READY TEST ROUTINE. CHECKS DEIVE IS READY. IF NOT, WAIT

```

```

1629          ; 500MS FOR READY TO SET.
1630 017632 010346          RDYCHK: MOV R3,-(SP)      ;STORE REGS
1631 017634 013703 002424  MOV SSINDX,R3    ;GET SUBROUTINE INDEX
1632 017640 005723          TST (R3)+        ;BUMP IT FOR NEXT ENTRY
1633 017642 016663 000002 002260  MOV 2(SP),SUBSTK(R3) ;INSERT THIS CALL
1634 017650 162763 000004 002260  SUB #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
1635 017656 010337 002424          MOV R3,SSINDX      ;STORE IT BACK
1636 017662 010046          MOV R0,-(SP)
1637 017664 010146          MOV R1,-(SP)
1638 017666 010446          MOV R4,-(SP)
1639 017670 012737 000002 002440  MOV #2,ERRSWI      ;SET FOR NO ERROR RETURN
1640 017676 012701 011610          MOV #5000,R1       ;SET WAIT COUNT
1641 017702 004          1$: JSR PC,GSTAT      ;GET DRIVE STATUS
1642 017706 020026 4$
1643 017710 032737 000001 002466  BIT #DRDYMSK,T.CS ;TEST IF DRIVE READY
1644 017716 001045          BNE 5$           ;YES - EXIT
1645 017720          WAITUS #1
   (3) 017720 012700 000001          MOV #1,R0
   (3) 017724 104027          EMT C$WTU
1646 017726 005301          DEC R1           ;DEC WAIT COUNT
1647 017730 001364          BNE 1$           ;LOOP IF NOT 0
1648 017732 012703 010711          MOV #MDRDY,R3     ;SET RESULT MESSAGE POINTER
1649 017736 012704 011755          MOV #C500MS,R4   ;SET CONDITION MESSAGE POINTER
1650 017742          ERRHRD 10010,ERR5
   (3) 017742 104443          TRAP T$ERCODE
   (5) 017744 023432          .WORD 10010
   (5) 017746 013272          .WORD ERR5
1651 017750 012701 000062          MOV #50,R1       ;SET WAIT COUNT FOR 5 SECONDS
1652 017754 004737 016514          2$: JSR PC,GSTAT      ;GET DRIVE STATUS
1653 017760 020026 4$
1654 017762 032737 000001 002466  BIT #DRDYMSK,T.CS ;TEST IF DRIVE READY
1655 017770 001005          BNE 3$           ;YES - SKIP
1656 017772          WAITMS #1
   (3) 017772 012700 000001          MOV #1,R0
   (3) 017776 104026          EMT C$WTM
1657 020000 005301          DEC R1           ;DEC WAIT COUNTER
1658 020002 001364          BNE 2$           ;JOP UNTIL TIME DONE
1659 020004 032737 100000 002466  3$: BIT #ANYERR,T.CS ;TEST IF ANYERR SET
1660 020012 001405          BEQ 4$           ;NO - SKIP
1661 020014          ERRHRD 10011,ERR6
   (3) 020014 104443          TRAP T$ERCODE
   (5) 020016 023433          .WORD 10011
   (5) 020020 013342          .WORD ERR6
1662 020022 005337 002662          DEC ERRCNT       ;REDUCE ERROR COUNT FOR DUAL ERRORS
1663 020026 005037 002440          4$: CLR ERRSWI     ;CLEAR FOR ERROR RETURN
1664 020032 162737 000002 002424  5$: SUB #2,SSINDX    ;REMOVE ENTRY FROM SUBROUT STACK
1665 020040 012604          MOV (SP)+,R4     ;RESTORE REGS
1666 020042 012601          MOV (SP)+,R1
1667 020044 012600          MOV (SP)+,R0
1668 020046 012603          MOV (SP)+,R3
1669 020050 005737 002440          TST ERRSWI      ;TEST IF ERROR RETURN
1670 020054 001403          BEQ 99$         ;YES - SKIP
1671 020056 063716 002440          ADD ERRSWI,(SP) ;ADD IN ERROR RETURN
1672 020062 000207          RTS PC
1673 020064 017616 000000          99$: MOV @,SP),(SP) ;SET ERROR RETURN ADDRESS
1674 020070 000207          RTS PC

```

```

1675
1718
1719 020072 012737 000001 002550 XRDHDC: MOV #1,TEMP4 ;SET FLAG TO BYPASS REG STORAGE
1720 020100 000402 BR XRDHDG ;GO DO IT
1721 020102 005037 002550 XRDHD: CLR TEMP4 ;SET FLAG TO SAVE T. AMD L. REGS
1722 020106 010346 XRDHDG: MOV R3,-(SP) ;STORE REGISTERS
1723 020110 013703 002424 MOV SSINDX,R3 ;GET SUBROUTINE INDEX
1724 020114 005723 TST (R3)+ ;BUMP IT FOR NEXT ENTRY
1725 020116 016663 000002 002260 MOV 2(SP),SUBSTK(R3) ;INSERT THIS CALL
1726 020124 162763 000004 002260 SUB #4,SUBSTK(R3) ;ADJUST IT TO CALLING LOCATION
1727 020132 010337 002424 MOV R3,SSINDX ;STORE IT BACK
1728 020136 010046 MOV R0,-(SP)
1729 020140 010146 MOV R1,-(SP)
1730 020142 010446 MOV R4,-(SP)
1731 020144 012737 000002 002440 MOV #2,ERRSWI ;SET FOR NO ERROR RETURN
1732 020152 005737 002550 TST TEMP4 ;TEST IF REGISTERS TO BE SAVED
1733 020156 001007 BNE 2$ ;NO - SKIP
1734 020160 012703 002466 MOV #L.MP+2,R3 ;SET POINTER FOR REGS
1735 020164 012701 000004 MOV #4,R1 ;SET COUNT
1736 020170 014346 1$: MOV -(R3),-(SP) ;SAVE REGISTER
1737 020172 005301 DEC R1 ;DEC COUNT
1738 020174 001375 BNE 1$ ;LOOP UNTIL ALL ARE SAVED
1739 020176 004737 017632 2$: JSR PC,RDYCHK ;CHECK DRIVE READY
1740 020202 020436 65$
1741 020204 005037 002430 CLR DONE ;CLEAR INTERRUPT FLAG
1742 020210 012701 002456 MOV #L.CS,R1 ;GET ADDRESS OF LOAD REGS
1743 020214 013711 002454 MOV RLDRV,(R1) ;LOAD DRIVE NUMBER
1744 020220 042711 002000 BIC #BIT10,(R1) ;CLEAR FOR DRIVE 4 - 7 SPEC'D
1745 020224 052721 000110 BIS #RDHEAD,(R1)+ ;INSERT COMMAND
1746 020230 005021 CLR (R1)+ ;CLEAR BA
1747 020232 005021 CLR (R1)+ ;CLEAR DA
1748 020234 014162 000004 MOV -(R1),RLDA(R2) ;LOAD RL11 REGS
1749 020240 014162 000002 MOV -(R1),RLBA(R2)
1750 020244 014162 000000 MOV -(R1),RLCSR(R2)
1751 020250 3$: WAITUS #10 ;WAIT 1MS FOR INTERRUPT
(3) 020250 012700 000012 MOV #10,RC
(3) 020254 104027 EMT CSWTU
1752 020256 005737 002430 TST DONE ;TEST IN INTERRUPT FLAG SET
1753 020262 001455 BEQ 14$ ;NO - SKIP
1754 020264 032737 000001 002466 5$: BIT #DRDYMSK,T.CS ;TEST IF DRIVE READY
1755 020272 001033 BNE 10$ ;YES - SKIP
1756 020274 012703 010711 MOV #MDRDY,R3 ;SET NO READY MESSAGE
1757 020300 012704 011774 MOV #CAFDI,R4 ;CONDITION OF AFTER DATA XFER
1758 020304 ERRHRD 10017,ERR5
(3) 020304 104443 TRAP T$ERRCODE
(5) 020306 023441 .WORD 10017
(5) 020310 013272 .WORD ERR5
1759 020312 012701 000062 MOV #50,R1 ;SET WAIT COUNT FOR 5 SECONDS
1760 020316 004737 016514 4$: JSR PC,GSTAT ;GET STATUS
1761 020322 020432 60$
1762 020324 032737 000001 002466 BIT #DRDYMSK,T.CS ;TEST IF DRIVE HAS COME READY
1763 020332 001403 BEQ 11$ ;NO - SKIP
1764 020334 005037 002440 CLR ERRSWI ;CLEAR ERROR SWITCH
1765 020340 000410 BR 10$ ;SKIP
1766 020342 005301 11$: DEC R1 ;DEC WAIT COUNT
1767 020344 001364 BNF 4$ ;LOOP UNTIL TIME DONE
    
```



```

1890      :      WAIT FOR READY ROUTINE. DURATION OF WAIT PASSED TO THE ROUTINE
1891      :      FROM THE CALLING ROUTINE IN R1.
1892 020560 010346      RDYWAIT:      MOV      R3,-(SP)      ;STORE R3
1893 020562 013703 002424      MOV      SSINDX,R3      ;GET SUBROUTINE INDEX
1894 020566 005723      TST      (R3)+      ;BUMP IT FOR NEXT ENTRY
1895 020570 016663 000002 002260      MOV      2(SP),SUBSTK(R3) ;INSERT THIS CALL
1896 020576 162763 000004 002260      SUB      #4,SUBSTK(R3) ;ADJUST IT TO CALLING LOCATION
1897 020604 010337 002424      MOV      R3,SSINDX      ;STORE IT BACK
1898 020610 010046      MOV      R0,-(SP)
1899 020612 010146      MOV      R1,-(SP)
1900 020614 010446      MOV      R4,-(SP)
1901 020616 012737 000002 002440      MOV      #2,ERRSWI      ;SET FOR NO ERROR RETURN
1902 020624 004737 016514 5$:      JSR      PC,GSTAT      ;GET DRIVE STATUS
1903 020630 020762      10$
1904 020632 032737 000001 002466      BIT      #DRDYMSK,T.CS ;CHECK IF READY
1905 020640 001052      BNE      9$      ;YES - SKIP
1906 020642 005301      DEC      R1      ;DEC WAIT COUNT
1907 020644 001404      BEQ      7$      ;SKIP IF 0
1908 020646      WAITUS #1
   (3) 020646 012700 000001      MOV      #1,R0
   (3) 020652 104027      EMT      C$WTJ
1909 020654 000763      BR      5$
1910 020656 012703 010711 7$:      MOV      #MMDRDY,R3      ;SET NAME MESSAGE PTR
1911 020662      ERRHRD 10020,,ERR3 ;REPORT READY ERROR
   (3) 020662 104443      TRAP    T$ERCODE
   (5) 020664 023444      .WORD  10020
   (5) 020666 013154      .WORD  ERR3
1912 020670 012701 000062      MOV      #50,R1      ;SET WAIT COUNT FOR 5 SECONDS
1913 020674 004737 016514 6$:      JSR      PC,GSTAT      ;GET DRIVE STATUS
1914 020700 020762      10$
1915 020702 032737 000001 002466      BIT      #DRDYMSK,T.CS ;TEST IF DRIVE READY
1916 020710 001013      BNE      8$      ;YES - SKIP
1917 020712      WAITMS #1
   (3) 020712 012700 000001      MOV      #1,R0
   (3) 020716 104026      EMT      C$WTM
1918 020720 005301      DEC      R1      ;DEC WAIT COUNT
1919 020722 001364      BNE      6$      ;LOOP UNTIL TIME DONE
1920 020724 012704 012006      MOV      #C5SEC,R4      ;SET CONDITION AFTER 5 SECDS
1921 020730      ERRHRD 10021,,ERR5
   (3) 020730 104443      TRAP    T$ERCODE
   (5) 020732 023445      .WORD  10021
   (5) 020734 013272      .WORD  ERR5
1922 020736 000407      BR      11$      ;EXIT
1923 020740 032737 100000 002466 8$:      BIT      #ANYERR,T.CS ;TEST IF ANY ERROR SET
1924 020746 001405      BEQ      10$      ;NO - SKIP
1925 020750      ERRHRD 10022,,ERR6 ;REPORT ALL ERRORS
   (3) 020750 104443      TRAP    T$ERCODE
   (5) 020752 023446      .WORD  10022
   (5) 020754 013342      .WORD  ERR6
1926 020756 005337 002662 11$:      DEC      ERRCNT      ;DEC FOR DOUBLE ERROR REPORT
1927 020762 005037 002440 10$:      CLR      ERRSWI      ;CLEAR FOR ERROR ERROR RETURN
1928 020766 162737 000002 002424 9$:      SUB      #2,SSINDX      ;REMOVE ENTRY FROM SUBROUT STACK
1929 020774 012604      MOV      (SP)+,R4      ;RESTORE REGISTERS
1930 020776 012601      MOV      (SP)+,R1
1931 021000 012600      MOV      (SP)+,R0
1932 021002 012603      MOV      (SP)+,R3      ;RESTORE R3

```

```

1933 021004 005737 002440      TST      ERRSWI      ;TEST IF ERROR RETURN
1934 021010 001403              BEQ      99$         ;YES - SKIP
1935 021012 063716 002440      ADD      ERRSWI,(SP) ;ADD IN ERROR RETURN
1936 021016 000207              RTS      PC
1937 021020 017616 000000      99$:    MOV      @ (SP),(SP) ;SET ERROR RETURN ADDRESS
1938 021024 000207              RTS      PC
1939
1940      ;      GET POSITION ROUTINE. READS A HEADER FROM CURRENT CYLINDER
1941      ;      (WHERE IT IS PRESENTLY POSITIONED) AND STORES CYLINDER
1942      ;      NUMBER IN CURCYL.
1943 021026 010346      GETPOS: MOV      R3,-(SP)      ;STORE REGISTERS
1944 021030 013703 002424      MOV      SSINDX,R3      ;GET SUBROUTINE INDEX
1945 021034 005723              TST      (R3)+          ;BUMP IT FOR NEXT ENTRY
1946 021036 016663 000002 002260      MOV      2(SP),SUBSTK(R3) ;INSERT THIS CALL
1947 021044 162763 000004 002260      SUB      #4,SUBSTK(R3)  ;ADJUST IT TO CALLING LOCATION
1948 021052 010337 002424      MOV      R3,SSINDX      ;STORE IT BACK
1949 021056 010046              MOV      R0,-(SP)
1950 021060 010546              MOV      R5,-(SP)
1951 021062 004737 020102      JSR      PC,XRDHD       ;DO READ HEADER
1952 021066 021116              65$
1953 021070 013703 002474      MOV      HDWRD1,R3      ;GET HEADER WORD
1954 021074 042703 100177      BIC      #^CHDCYL,R3    ;CLEAR ALL BUT CYLINDER
1955 021100 012705 000007      MOV      #7,R5          ;SET SHIFT COUNT
1956 021104 006203              4$:    ASR      R3          ;SHIFT TO RIGHT JUSTIFY
1957 021106 005305              DEC      R5
1958 021110 001375              BNE      4$
1959 021112 010337 002526      MOV      R3,CURCYL      ;STORE AS CURRENT CYLINDER
1960 021116 162737 000002 002424 65$:    SUB      #2,SSINDX      ;REMOVE ENTRY FROM SUBROUT STACK
1961 021124 012605              MOV      (SP)+,R5        ;RESTORE REGISTERS
1962 021126 012600              MOV      (SP)+,R0
1963 021130 012603              MOV      (SP)+,R3
1964 021132 005737 002440      TST      ERRSWI        ;TEST IF ERROR RETURN
1965 021136 001403              BEQ      99$         ;YES - SKIP
1966 021140 063716 002440      ADD      ERRSWI,(SP)   ;ADD IN ERROR RETURN
1967 021144 000207              RTS      PC
1968 021146 017616 000000      99$:    MOV      @ (SP),(SP)   ;SET ERROR RETURN ADDRESS
1969 021152 000207              RTS      PC
1970
1999      ;      READ ALL HEADERS ROUTINE. 40 HEADERS ARE READ AND STORED
2000      ;      IN Ibuff.
2001 021154 010346      RDALHD: MOV      R3,-(SP)      ;STORE REGISTERS
2002 021156 013703 002424      MOV      SSINDX,R3      ;GET SUBROUTINE INDEX
2003 021162 005723              TST      (R3)+          ;BUMP IT FOR NEXT ENTRY
2004 021164 016663 000002 002260      MOV      2(SP),SUBSTK(R3) ;INSERT THIS CALL
2005 021172 162763 000004 002260      SUB      #4,SUBSTK(R3)  ;ADJUST IT TO CALLING LOCATION
2006 021200 010337 002424      MOV      R3,SSINDX      ;STORE IT BACK
2007 021204 010046              MOV      R0,-(SP)
2008 021206 010146              MOV      R1,-(SP)
2009 021210 010446              MOV      R4,-(SP)
2010 021212 012737 000002 002440      MOV      #2,ERRSWI      ;SET FOR NO ERROR RETURN
2011 021220 012701 000050      MOV      #40,R1         ;SET HEADER COUNT
2012 021224 052737 100000 002426      BIS      #HDR40,OPFLAG  ;SET 40 HDR OP FLAG
2013 021232 012703 003466      MOV      #IBUFF,R3      ;SET POINTER TO STORE HDRS
2014 021236 013704 002450      MOV      RLBAS,R4       ;GET BASE ADDRESS
2015 021242 062704 000006      ADD      #7,MP,R4       ;MAKE IT POINT TO MP REG
2016 021246 012737 000010 002456      MOV      #10,LCS       ;LOAD FOR READ HEADER, NO INTERRUPT

```

```

2017 021254 053737 002454 002456 BIS RLDPV,L,CS ;INSERT DRIVE NUMBER
2018 021262 042737 002000 002456 BIC #BIT10,L,CS ;CLEAR FOR DRIVE 4 - 7 SPEC'D
2019 021270 005037 002460 CLR L,BA ;CLEAR BA
2020 021274 005037 002462 CLR L,DA ;CLEAR DA
2021 021300 005737 002534 TST DESHD ;TEST IF HEAD 0
2022 021304 001403 BEQ 3$ ;YES - SKIP
2023 021306 052737 000020 002462 BIS #HDSEL,L,DA ;ELSE INSERT HEAD 0
2024 021314 013762 002462 000004 3$: MOV L,DA,RLDA(R2) ;LOAD RLDA REG
2025 021322 013762 002460 000002 MOV L,BA,RLBA(R2) ;LOAD RLBA
2026 021330 032762 000200 000000 BIT #CRDYMSK,RLCS(R2) ;TEST IF CONTROLLER READY
2027 021336 001003 BNE 6$ ;YES - SKIP
2028 021340 004737 017632 JSR PC,RDYCHK ;ELSE CHECK READY
2029 021344 021456 65$: MOV L,CS,RLCS(R2) ;LOAD RLCS REG
2030 021346 013762 002456 000000 65$: MOV #77777,R0 ;SET COUNT FOR WAIT
2031 021354 012700 077777 7$: BIT #CRDYMSK,RLCS(R2) ;CHECK THAT OPERATION COMPLETED
2032 021360 032762 000200 000000 7$: BNE 8$ ;YES - SKIP
2033 021366 001015 DEC R0 ;DEC COUNT
2034 021370 005300 BNE 7$ ;SKIP IF NOT YET 0
2035 021372 001372 JSR PC,READRL ;ELSE GET ALL REGISTERS
2036 021374 004737 016262 JSR PC,WAITIN ;ELSE WAIT FOR TIMEOUT
2037 021400 004737 016314 MOV (SP)+,R3 ;GET RESULT MESSAGE POINTER
2038 021404 012603 ERRHRD 10025,,,ERR1
2039 021406 TRAP T$ERCODE
(3) 021406 104443 .WORD 10025
(5) 021410 023451 .WORD ERR1
(5) 021412 013040 CLR ERRSWI ;CLEAR FOR ERROR RETURN
2040 021414 005037 002440 BR 65$
2041 021420 000416 TST T,CS ;TEST FOR ANY ERRORS
2042 021422 005737 002466 8$: BPL 12$ ;NO - SKIP
2043 021426 100006 ERRHRD 10026,,,ERR6
2044 021430 TRAP T$ERCODE
(3) 021430 104443 .WORD 10026
(5) 021432 023452 .WORD ERR6
(5) 021434 013342 CLR ERRSWI ;CLEAR FOR ERROR RETURN
2045 021436 005037 002440 BR 65$
2046 021442 000405 12$: MOV (R4),(R3)+ ;STORF HEADER WORDS
2047 021444 011423 MOV (R4),(R3)+
2048 021446 011423 MOV (R4),(R3)+
2049 021450 011423 DEC R1 ;DEC HEADER COUNT
2050 021452 005301 BNE 6$
2051 021454 001334 SUB #2,SSINDX ;REMOVE ENTRY FROM SUBROUT STACK
2052 021456 162737 000002 002424 65$: MOV (SP)+,R4 ;RESTORE REGISTERS
2053 021464 012604 MOV (SP)+,R1
2054 021466 012601 MOV (SP)+,R0
2055 021470 012600 MOV (SP)+,R3
2056 021472 012603 TST ERRSWI ;TEST IF ERROR RETURN
2057 021474 005737 002440 BEQ 99$ ;YES - SKIP
2058 021500 001403 ADD ERRSWI,(SP) ;ADD IN ERROR RETURN
2059 021502 063716 002440 RTS PC
2060 021506 000207 99$: MOV @ (SF),(SP) ;SET ERROR RETURN ADDRESS
2061 021510 017616 000000 RTS PC
2062 021514 000207
2063
2064
2292 ; REPORT OPERATION ROUTINE. PRINTS SUBROUTINE TRACE SEQUENCE AND
2293 ; OPERATION BEING PERFORMED PORTION OF ALL
  
```

```

2294
2295 021516 010446
2296 021520 005737 002424
2297 021524 001433
2298 021526 012704 000002
2299 021532
(8) 021532 012746 010534
(7) 021536 012746 012327
(6) 021542 012746 000002
(3) 021546 010600
(4) 021550 104014
(4) 021552 062706 000006
2300 021556
(8) 021556 016446 002260
(7) 021562 012746 012502
(6) 021566 012746 000002
(3) 021572 010600
(4) 021574 104014
(4) 021576 062706 000006
2301 021602 062704 000002
2302 021606 020437 002424
2303 021612 003761
2304 021614
(9) 021614 012746 006217
(8) 021620 013746 002434
(7) 021624 012746 012132
(6) 021630 012746 000003
(3) 021634 010600
(4) 021636 104014
(4) 021640 062706 000010
2305 021644 042737 030000 002426
2306 021652 013701 002456
2307 021656 042701 177741
2308 021662 022701 000006
2309 021666 001003
2310 021670 052737 010000 002426
2311 021676 022701 000012
2312 021702 001003
2313 021704 052737 020000 002426
2314 021712 022701 000014
2315 021716 001003
2316 021720 052737 020000 002426
2317 021726
(9) 021726 016146 002122
(8) 021732 012746 005143
(7) 021736 012746 012110
(6) 021742 012746 000003
(3) 021746 010600
(4) 021750 104014
(4) 021752 062706 000010
2318 021756 020127 000004
2319 021762 001007
2320 021764 032737 000010 002462
2321 021772 001403
2322 021774 012701 000016
2323 022000 000436

```

```

: RPTOP: ERROR MESSAGES.
MOV R4,-(SP)
TST SSINDX ;TEST SUBROUTINE INDEX 0
BEQ 1$ ;SKIP IF 0
MOV #2,R4 ;SET INDEXER TO FIRST ENTRY
PRINTB #FMT9,#SEQMES ;PRINT 'SUBROUTINE CALL SEQ'
MOV #SEQMES,-(SP)
MOV #FMT9,-(SP)
MOV #2,-(SP)
MOV SP,R0
EMT C$PNTB
ADD #6,SP
3$: PRINTB #FMT16,SUBSTK(R4) ;PRINT CALLING LOCATION
MOV SUBSTK(R4),-(SP)
MOV #FMT16,-(SP)
MOV #2,-(SP)
MOV SP,R0
EMT C$PNTB
ADD #6,SP
ADD #2,R4 ;BUMP INDEX
CMP R4,SSINDX ;CHECK IF ALL PRINTED
BLE 3$ ;LOOP IF NOT ALL PRINTED YET
1$: PRINTB #FMT4,ERHEAD,#T$LAB ;PRINT ERROR HEADER
MOV #T$LAB,-(SP)
MOV ERHEAD,-(SP)
MOV #FMT4,-(SP)
MOV #3,-(SP)
MOV SP,R0
EMT C$PNTB
ADD #10,SP
BIC #SEEKOP!RORWOP,OPFLAG ;CLEAR SK & RD OR WRT FLAG
MOV L,CS,R1 ;GET COMMAND EXECUTED
BIC #177741,R1 ;STRIP ALL BUT FUNCTION CODE
CMP #6,R1 ;TEST IF SEEK OPERATION
BNE 2$ ;NO - SKIP
BIS #SEEKOP,OPFLAG ;ELSE SET SEEK FLAG
2$: CMP #12,R1 ;TEST IF WRITE
BNE 20$ ;NO - SKIP
BIS #RORWOP,OPFLAG ;SET RD OR WRT FLAG
20$: CMP #14,R1 ;TEST IF READ
BNE 22$ ;NO - SKIP
BIS #RORWOP,OPFLAG ;SET RD OR WRT FLAG
22$: PRINTB #FMT1,#MOPER,OPMSG(S(R1)) ;PRINT OPERATION
MOV OPMSG(S(R1)),-(SP)
MOV #MOPER,-(SP)
MOV #FMT1,-(SP)
MOV #3,-(SP)
MOV SP,R0
EMT C$PNTB
ADD #10,SP
CMP R1,#4 ;CHECK IF GET STATUS
BNE 4$ ;NO - SKIP
BIT #DRSET,L,DA ;TEST IF RESET INCLUDED
BEQ 4$ ;NO - SKIP
MOV #5,R1 ;SET TO PRINT WITH RESET
BR 9$

```

```

2324 022002 032737 007777 002426 4$: BIT #COMPOP,OPFLAG ;TEST IF ANY OTHER OPERATION
2325 022010 001424 BEQ 8$ ;NO - SKIP
2326 022012 013704 002426 MOV OPFLAG,R4 ;SET UP TO DETERMINE WHICH ONE
2327 022016 012701 000020 MOV #20,R1 ;PRESET THE POINTER
2328 022022 032704 000001 5$: BIT #BIT00,R4 ;CHECK THE BIT
2329 022026 001003 BNE 6$ ;IF SET - SKIP
2330 022030 005721 TST (R1)+ ;BUMP POINTER
2331 022032 006204 ASR R4
2332 022034 000772 BR 5$
2333 022036 6$: PRINTB #FMT2,OPMSG5(R1)
(8) 022036 016146 002122 MOV OPMSG5(R1),-(SP)
(7) 022042 012746 012124 MOV #FMT2,-(SP)
(6) 022046 012746 000002 MOV #2,-(SP)
(3) 022052 010600 MOV SP,R0
(4) 022054 104014 EMT C$PNTB
(4) 022056 062706 000006 ADD #6,SP
2334 022062 032737 100000 002426 8$: BIT #HDR40,OPFLAG ;TEST IF 40 HEADER OPERATION
2335 022070 001415 BEQ 10$ ;NO - SKIP
2336 022072 012701 000050 MOV #50,R1 ;ELSE PRINT IT
2337 022076 9$: PRINTB #FMT2,OPMSG5(R1)
(8) 022076 016146 002122 MOV OPMSG5(R1),-(SP)
(7) 022102 012746 012124 MOV #FMT2,-(SP)
(6) 022106 012746 000002 MOV #2,-(SP)
(3) 022112 010600 MOV SP,R0
(4) 022114 104014 EMT C$PNTB
(4) 022116 062706 000006 ADD #6,SP
2338 022122 000434 BR 15$ ;SKIP
2339 022124 032737 010000 002426 10$: BIT #SEEKOP,OPFLAG ;TEST IF SEEK
2340 022132 001430 BEQ 15$ ;NO - SKIP
2341 022134 PRINTB #FMT13,#FRMWD,OLDCYL,#DIFWD,DESDIF,#SGNWD,DESSGN,#HDWD,DESHD
(15) 022134 013746 002534 MOV DESHD,-(SP)
(14) 022140 012746 010475 MOV #HDWD,-(SP)
(13) 022144 013746 002532 MOV DESSGN,-(SP)
(12) 022150 012746 010470 MOV #SGNWD,-(SP)
(11) 022154 013746 002530 MOV DESDIF,-(SP)
(10) 022160 012746 010462 MOV #DIFWD,-(SP)
(9) 022164 013746 002522 MOV OLDCYL,-(SP)
(8) 022170 012746 010513 MOV #FRMWD,-(SP)
(7) 022174 012746 012350 MOV #FMT13,-(SP)
(6) 022200 012746 000011 MOV #11,-(SP)
(3) 022204 010600 MOV SP,R0
(4) 022206 104014 EMT C$PNTB
(4) 022210 062706 000024 ADD #24,SP
2342 022214 032737 020000 002426 15$: BIT #RORWOP,OPFLAG ;TEST IF READ OR WRITE SET
2343 022222 001424 BEQ 17$ ;NO - SKIP
2344 022224 PRINTB #FMT22,#CYLWD,CURCYL,#HDWD,DESHD,#SECWD,DESSEC
(13) 022224 013746 002536 MOV DESSEC,-(SP)
(12) 022230 012746 010501 MOV #SECWD,-(SP)
(11) 022234 013746 002534 MOV DESHD,-(SP)
(10) 022240 012746 010475 MOV #HDWD,-(SP)
(9) 022244 013746 002526 MOV CURCYL,-(SP)
(8) 022250 012746 010506 MOV #CYLWD,-(SP)
(7) 022254 012746 012677 MOV #FMT22,-(SP)
(6) 022260 012746 000007 MOV #7,-(SP)
(3) 022264 010600 MOV SP,R0
(4) 022266 104014 EMT C$PNTB

```

```

(4) 022270 062706 000020
2345 022274 004737 022746      17$:  ADD      #20,SP
2346 022300 012604                JSR      PC,CLRPARM      ;CLEAR PARAM TABLE
2347 022302 000207                MOV      (SP)+,R4        ;RESTORE R4
2348
2349
2350      :      REPORT REASON ROUTINE
2351 022304 010146      RPTRES: PRINTS REASON PORTION FOR ALL ERROR REPORTS.
2352 022306 010346      MOV      R1,-(SP)        ;STORE R1
2353 022310 010446      MOV      R3,-(SP)        ;STORE R3
2354 022312 012701 002504      MOV      R4,-(SP)        ;STORE R4
2355 022316 012103      MOV      #RESPARM,R1     ;GET START OF PARAM
2356 022320                MOV      (R1)+,R3        ;GET NUMBER OF PARAM
(9) 022320 011146      PRINTB  #FMT1.1,#MRSLT,(R1) ;PRINT NAME
(8) 022322 012746 005157      MOV      (R1),-(SP)
(7) 022326 012746 012115      MOV      #MRSLT,-(SP)
(6) 022332 012746 000003      MOV      #FMT1.1,-(SP)
(3) 022336 010600      MOV      #3,-(SP)
(4) 022340 104014      MOV      SP,R0
(4) 022342 062706 000010      EMT      C$PNTB
2357 022346 021127 011405      ADD      #10,SP
2358 022352 001453      CMP      (R1),#MNRDST    ;TEST IF MESSAGE IS NO DRV STATUS
2359 022354 012704 012334      BEQ      6$              ;YES - SKIP REST OF REPORT
2360 022360 022127 011400      MOV      #FMT11,R4       ;PRISET FOR FORMAT 11
2361 022364 001002      CMP      (R1)+,#MCYLOC   ;CHECK IF REPORTING CYLINDER LOC
2362 022366 012704 012342      BNE      3$              ;NO - SKIP
2363 022372 005303      MOV      #FMT12,R4       ;ELSE CHANGE TO FORMAT 12
2364 022374 001442      3$:  DEC      R3              ;DEC PARAM COUNT
2365 022376                BEQ      6$              ;IF 0 - EXIT
(9) 022376 012146      PRINTB  R4,#RESE3,(R1)+  ;REPORT IS VALUE
(8) 022400 012746 011647      MOV      (R1)+,-(SP)
(7) 022404 010446      MOV      #RESE3,-(SP)
(6) 022406 012746 000003      MOV      R4,-(SP)
(3) 022412 010600      MOV      #3,-(SP)
(4) 022414 104014      MOV      SP,R0
(4) 022416 062706 000010      EMT      C$PNTB
2366 022422                ADD      #10,SP
(9) 022422 012146      PRINTB  R4,#RESE4,(R1)+  ;REPORT SB VALUE
(8) 022424 012746 011653      MOV      (R1)+,-(SP)
(7) 022430 010446      MOV      #RESE4,-(SP)
(6) 022432 012746 000003      MOV      R4,-(SP)
(3) 022436 010600      MOV      #3,-(SP)
(4) 022440 104014      MOV      SP,R0
(4) 022442 062706 000010      EMT      C$PNTB
2367 022446 162703 000002      ADD      #10,SP
2368 022452 001413      SUB      #2,R3            ;DEC PARAM COUNT
2369 022454                BEQ      6$              ;IF 0 - EXIT
(9) 022454 012146      PRINTB  #FMT1,#RESE5,(R1)+ ;REPORT CONDITION
(8) 022456 012746 011660      MOV      (R1)+,-(SP)
(7) 022462 012746 012110      MOV      #RESE5,-(SP)
(6) 022466 012746 000003      MOV      #FMT1,-(SP)
(3) 022472 010600      MOV      #3,-(SP)
(4) 022474 104014      MOV      SP,R0
(4) 022476 062706 000010      EMT      C$PNTB
2370 022502 012604      6$:  ADD      #10,SP
2371 022504 012603      MOV      (R1)+,R4        ;RESTORE REGS
      MOV      (SP)+,R3
  
```

```

2372 022506 012601      MOV      (SP)+,R1
2373 022510 000207      RTS      PC                ;RETURN
2374
2375      :      REPORT PHYSICAL ADDRESS OF DEVICE UNDER TEST
2376      :      AND ALL REGISTER CONTENTS.
2377 022512      RPTREM: PRINTB #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
      (11) 022512 005046      CLR      -(SP)
      (11) 022514 153716 002455      BISB    RLDRV+1,(SP)
      (10) 022520 012746 005633      MOV     #DRVNAM,-(SP)
      (9)  022524 013746 002450      MOV     RLBAS,-(SP)
      (8)  022530 012746 005622      MOV     #BASADD,-(SP)
      (7)  022534 012746 012143      MOV     #FMT5,-(SP)
      (6)  022540 012746 000005      MOV     #5,-(SP)
      (3)  022544 010600      MOV     SP,R0
      (4)  022546 104014      EMT     C$PNTB
      (4)  022550 062706 000014      ADD     #14,SP
2378      :      REPORT R11 REGISTERS
2379 022554      PRINTB #FMT6,#CSNAM,#DANAM,#BANAM,#MPNAM,#CYLWD,#HDWD
      (13) 022554 012746 010475      MOV     #HDWD,-(SP)
      (12) 022560 012746 010506      MOV     #CYLWD,-(SP)
      (11) 022564 012746 005752      MOV     #MPNAM,-(SP)
      (10) 022570 012746 005740      MOV     #BANAM,-(SP)
      (9)  022574 012746 005745      MOV     #DANAM,-(SP)
      (8)  022600 012746 005733      MOV     #CSNAM,-(SP)
      (7)  022604 012746 012163      MOV     #FMT6,-(SP)
      (6)  022610 012746 000007      MOV     #7,-(SP)
      (3)  022614 010600      MOV     SP,R0
      (4)  022616 104014      EMT     C$PNTB
      (4)  022620 062706 000020      ADD     #20,SP
2380 022624      PRINTB #FMT8,#LAB1,L.CS,L.DA,L.BA,L.MP
      (12) 022624 013746 002464      MOV     L.MP,-(SP)
      (11) 022630 013746 002460      MOV     L.BA,-(SP)
      (10) 022634 013746 002462      MOV     L.DA,-(SP)
      (9)  022640 013746 002456      MOV     L.CS,-(SP)
      (8)  022644 012746 005757      MOV     #LAB1,-(SP)
      (7)  022650 012746 012275      MOV     #FMT8,-(SP)
      (6)  022654 012746 000006      MOV     #6,-(SP)
      (3)  022660 010600      MOV     SP,R0
      (4)  022662 104014      EMT     C$PNTB
      (4)  022664 062706 000016      ADD     #16,SP
2381 022670      PRINTB #FMT7,#LAB2,T.CS,T.DA,T.BA,T.MP,CURCYL,DESHD
      (14) 022670 013746 002534      MOV     DESHD,-(SP)
      (13) 022674 013746 002526      MOV     CURCYL,-(SP)
      (12) 022700 013746 002474      MOV     T.MP,-(SP)
      (11) 022704 013746 002470      MOV     T.BA,-(SP)
      (10) 022710 013746 002472      MOV     T.DA,-(SP)
      (9)  022714 013746 002466      MOV     T.CS,-(SP)
      (8)  022720 012746 005772      MOV     #LAB2,-(SP)
      (7)  022724 012746 012225      MOV     #FIT7,-(SP)
      (6)  022730 012746 000010      MOV     #10,-(SP)
      (3)  022734 010600      MOV     SP,R0
      (4)  022736 104014      EMT     C$PNTB
      (4)  022740 062706 000022      ADD     #22,SP
2382 022744 000207      RTS      PC
2383
2384      :      CLEAR PARAMETER BLOCK FOR REPORTING

```

```
2385 022746 010546          CLRPARM:      MOV      R5, -(SP)      :STORE R5
2386 022750 012701 002504      MOV      #RESPARM,R1   :GET ADDRESS OF BLOCK
2387 022754 012705 000005      MOV      #5,R5         :SET COUNT
2388 022760 005021          2$:      CLR      (R1)+      :CLEAR WORD
2389 022762 005305          DEC      R5            :DEC COUNT
2390 022764 001375          BNE      2$           :LOOP UNTIL 0
2391 022766 012701 002504      MOV      #RESPARM,R1   :RESET POINTER
2392 022772 012605          MOV      (SP)+,R5      :RESTORE R5
2393 022774 000207          RTS      PC
2394
2395 022776          ENDMOD
2396
```



```

2398 022776          BGNMOD  HRDWTST
2399          .SRTTL  *TEST 1          BASIC INTERFACE (PART 1)
2400
2401 022776          BGNST          :TEST01
(3) 022776
2402 022776 005737 003062          TST    PASNUM          :CHECK IF FIRST PASS          T1::
2403 023002 001120          BNE    65$             :EXIT IF NO
2404 023004 005737 014656          TST    MISWIW         :CHECK IF MANUAL INTERVENTION
2405 023010 100115          BPL    65$             :NO - EXIT TEST
2406 023012 012737 006225 002434  MOV    #MISTST,ERHEAD :LOAD ERR HEADER
2407 023020          2$: PRINTF  #FMTOP1,#OPR1,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
(13) 023020 005046          CLR    -(SP)
(13) 023022 153716 002455          BISB  RLDRV+1,(SP)
(12) 023026 012746 005633          MOV   #DRVNAM,-(SP)
(11) 023032 013746 002450          MOV   RLBAS,-(SP)
(10) 023036 012746 005622          MOV   #BASADD,-(SP)
(9)  023042 012746 010416          MOV   #OPR1A,-(SP)
(8)  023046 012746 007774          MOV   #OPR1,-(SP)
(7)  023052 012746 012016          MOV   #FMTOP1,-(SP)
(6)  023056 012746 000007          MOV   #7,-(SP)
(3)  023062 010600          MOV   SP,R0
(4)  023064 104017          EMT   C$PNTF
(4)  023066 062706 000020          ADD   #20,SP
2408 023072 005037 004066          CLR   OBUFF           :CLEAR FOR RESPONSE
2409 023076          GMANIL OPR002,OBUFF,1,NO
(3)  023076 104043          EMT   C$GMAN
(3)  023100 000404          BR    10000$
(4)  023102 004066          .WORD OBUFF
(5)  023104 000120          .WORD T$CODE
(5)  023106 007724          .WORD OPR002
(5)  023110 000001          .WORD 1
(3)  023112          10000$:
2410 023112 005737 004066          TST   OBUFF           :TEST RESPONSE YES
2411 023116 001740          BEQ   2$              :YES - SKIP
2412 023120 004737 016446          1$: JSR  PC,TSTINT     :INITIALIZE TEST
2413 023124 004737 016500          JSR  PC,GSTATC       :GO GET STATUS (NO RESET)
2414 023130 023244          65$
2415 023132 032737 000040 002474  BIT   #COSTAT,T.MP    :CHECK IF COVER OPEN SET
2416 023140 001005          BNE   7$              :YES - SKIP
2417 023142 012703 011072          MOV   #MCOSTA,R3     :SET NAME POINTER
2418 023146          ERRHRD 101...ERR3
(3)  023146 104443          TRAP T$ERRCODE
(5)  023150 000145          .WORD 101
(5)  023152 013154          .WORD ERR3
2419 023154 032737 000010 002474  7$: BIT   #BHSTAT,T.MP  :TEST IF BRUSHES HOME
2420 023162 001005          BNE   9$              :YES - SKIP
2421 023164 012703 011105          MOV   #MBHSTA,R3     :SET POINTER FOR BRUSH HOME ERROR
2422 023170          ERRHRD 102...ERR3
(3)  023170 104443          TRAP T$ERRCODE
(5)  023172 000146          .WORD 102
(5)  023174 013154          .WORD ERR3
2423 023176 032737 020000 002474  9$: BIT   #WLSTAT,T.MP  :TEST IF WRITE LOCK SET
2424 023204 001005          BNE   11$             :YES - SKIP
2425 023206 012703 011120          MOV   #MWLSTA,R3     :SET NAME POINTER
2426 023212          ERRHRD 103...ERR3
(3)  023212 104443          TRAP T$ERRCODE

```

```
(5) 023214 000147 .WORD 103
(5) 023216 013154 .WORD ERR3
2427 023220 005737 002502 11$: TST T,STAT ;TEST IF STATE ZERO
2428 023224 001404 BEQ 15$ ;YES - SKIP
2429 023226 005003 CLR R3 ;SET STATE EXPECTED
2430 023230 ERRHRD 104...ERR7
(3) 023230 104443 TRAP T$ERCODE
(5) 023232 000150 .WORD 104
(5) 023234 014220 .WORD ERR7
2431 023236 004737 016464 15$: JSR PC,GSTATR ;DO DRIVE RESET
2432 023242 023244 65$:
2433 023244 ENDTST
2434 023244 L10020:
(3) 023244 EMT C$ETST
(3) 023244 104001
2435
```

```

2437      .SBTTL *TEST 2          BASIC INTERFACE (PART 2)
2438
2439      023246      BGNTST          ;TEST 2
(3)      023246
2440      023246      005737      003062      TST      PASNUM          ;TEST IF PASS 0
2441      023252      001075      BNE      65$          ;NO - SKIP
2442      023254      005737      014656      TST      MISWIW        ;TEST IF MANUAL INTERVENTION
2443      023260      100072      BPL      65$          ;NO - SKIP
2444      023262      012737      006225      002434      MOV      #MISTST,ERHEAD ;SET ERROR HEADER
2445      023270      2$:      PRINTF   #FMTOP1,#OPR2,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1> ;REQUEST CLOSE
(13)     023270      005046      CLR      -(SP)
(13)     023272      153716      002455      BISB    RLDRV+1,(SP)
(12)     023276      012746      005633      MOV     #DRVNAM,-(SP)
(11)     023302      013746      002450      MOV     RLBAS,-(SP)
(10)     023306      012746      005622      MOV     #BASADD,-(SP)
(9)      023312      012746      010416      MOV     #OPR1A,-(SP)
(8)      023316      012746      010052      MOV     #OPR2,-(SP)
(7)      023322      012746      012016      MOV     #FMTOP1,-(SP)
(6)      023326      012746      000007      MOV     #7,-(SP)
(3)      023332      010600      MOV     SP,R0
(4)      023334      104017      EMT     C$PNTF
(4)      023336      062706      000020      ADD     #20,SP
2446
2447      023342      005037      004066      CLR     OBUFF          ;COVER AND RESET WRITE LOCK
2448      023346      GMANIL   OPR002,OBUFF,1,NO    ;CLEAR FOR RESPONSE
(3)      023346      104043      EMT     C$GMAN
(3)      023350      000404      BR      10000$
(4)      023352      004066      .WORD  OBUFF
(5)      023354      000120      .WORD  T$CODE
(5)      023356      007724      .WORD  OPR002
(5)      023360      000001      .WORD  1
(3)      023362      10000$:
2449      023362      005737      004066      TST     OBUFF          ;TEST IF RESPONSE YES
2450      023366      001740      BEQ     2$          ;NO - SKIP
2451      023370      004737      016446      1$:      JSR     PC,TSTINT      ;INITIALIZE TEST
2452      023374      004737      016464      JSR     PC,GSTATR     ;GET STATUS WITH RESET
2453      023400      023446      65$
2454      023402      032737      000040      002474      BIT     #COSTAT,T.MP   ;TEST IF COVER OPEN RESET
2455      023410      001405      BEQ     9$          ;YES - SKIP
2456      023412      012703      011072      MOV     #MCOSTA,R3    ;SET NAME MESSAGE POINTER
2457      023416      ERRHRD   201...ERR2
(3)      023416      104443      TRAP   T$ERRCODE
(5)      023420      000311      .WORD  201
(5)      023422      013106      .WORD  ERR2
2458      023424      032737      020000      002474      9$:      BIT     #WLSTAT,T.MP   ;TEST IF WRITE LOCK RESET
2459      023432      001405      BEQ     65$          ;YES - SKIP
2460      023434      012703      011120      MOV     #MWLSTA,R3    ;SET NAME MESSAGE POINTER
2461      023440      ERRHRD   202...ERR2
(3)      023440      104443      TRAP   T$ERRCODE
(5)      023442      000312      .WORD  202
(5)      023444      013106      .WORD  ERR2
2462      023446      65$:
2463      023446      FNDTST
(3)      023446      L10021:
(3)      023446      104001      EMT     C$FTST
2464

```

```

2466          .SBTTL *TEST 3          HEAD LOADING
2467 023450    RGNTST                :TEST03
(3) 023450
2468 023450    005737 003062          TST PASNUM          :TEST IF PASS 0
2469 023454    001003                BNE 4$              :NO SKIP
2470 023456    005737 014656          TST MISWIW         :TEST IF MANUAL INTERVENTION
2471 023462    10040?                BMI 5$              :YES - SKIP
2472 023464    4$: EXIT TST
(3) 023464    104032                EMT C$EXIT
(3) 023466    001214                .WORD L10022-
2473 023470    004737 016446          5$: JSR PC,TSTINT   :INITIALIZE TEST
2474 023474    004737 016464          JSR PC,GSTATR     :GET STATUS
2475 023500    024702                T365$
2476 023502    005737 002502          IST T,STAT        :TEST IF STATE ZERO
2477 023506    001440                BEQ 2$              :YES - SKIP
2478 023510    1$: PRINTF #FMTOP1,#OPR5,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1> ;REQUEST DRIVE BE
(13) 023510    005046                CLR -(SP)
(13) 023512    153716 002455          BISB RLDRV+1,(SP)
(12) 023516    012746 005633          MOV #DRVNAM,-(SP)
(11) 023522    013746 002450          MOV RLBAS,-(SP)
(10) 023526    012746 005622          MOV #BASADD,-(SP)
(9) 023532    012746 010416          MOV #OPR1A,-(SP)
(8) 023536    012746 010120          MOV #OPR5,-(SP)
(7) 023542    012746 012016          MOV #FMTOP1,-(SP)
(6) 023546    012746 000007          MOV #7,-(SP)
(3) 023552    010600                MOV SP,R0
(4) 023554    104017                EMT C$PNTF
(4) 023556    062706 000020          ADD #20,SP
2479 023562    005037 004066          CLR OBUF          :CLEAR FOR RESPONSE
2480 023566    GMANIL OPR002,OBUF,1,NO
(3) 023566    104043                EMT C$GMAN
(3) 023570    000404                BR 10000$
(4) 023572    004066                .WORD OBUF
(5) 023574    000120                .WORD T$CODE
(5) 023576    007724                .WORD OPR002
(5) 023600    000001                .WORD 1
(3) 023602    10000$:
2481 023602    005737 004066          TST OBUF          :TEST IF RESPONSE YES
2482 023606    001740                BEQ 1$              :NO - SKIP
2483 023610    2$: PRINTF #FMTOP1,#OPR3,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
(13) 023610    005046                CLR -(SP)
(13) 023612    153716 002455          BISB RLDRV+1,(SP)
(12) 023616    012746 005633          MOV #DRVNAM,-(SP)
(11) 023622    013746 002450          MOV RLBAS,-(SP)
(10) 023626    012746 005622          MOV #BASADD,-(SP)
(9) 023632    012746 010416          MOV #OPR1A,-(SP)
(8) 023636    012746 010104          MOV #OPR3,-(SP)
(7) 023642    012746 012016          MOV #FMTOP1,-(SP)
(6) 023646    012746 000007          MOV #7,-(SP)
(3) 023652    010600                MOV SP,R0
(4) 023654    104017                EMT C$PNTF
(4) 023656    062706 000020          ADD #20,SP
2484 023662    012737 000004 002426          MOV #CYLUP,OPFLAG :SET CYCLE UP FLAG
2485 023670    012703 000001                MOV #1,R3          :SET EXPECTED STATE VALUE
2486 023674    012737 006250 002434          MOV #1,STACHG,ERHEAD :SET ERROR HEADER
2487 023702    012701 000454                MOV #300.,R'       :SET WAIT COUNT FOR 30 SECONDS
  
```

```

2488 023706 004737 016500 3$: JSR PC,GSTATC ;GET STATUS
2489 023712 024702 T365$
2490 023714 005737 002502 TST T,STAT ;TEST IF STATE IS STILL 0
2491 023720 001022 BNE 10$ ;NO - SKIP
2492 023722 005301 DEC R1 ;DEC WAIT COUNT
2493 023724 001404 BEQ 6$ ;EXIT IF WAIT DONE
2494 023726 WAITMS #1
(3) 023726 012700 000001 MOV #1,R0
(3) 023732 104026 EMT C$WTM
2495 023734 000764 BR 3$
2496 023736 005037 004066 6$: CLR OBUFF ;CLEAR FOR RESPONSE
2497 023742 GMANIL OPRO03,OBUFF,1,NO
(3) 023742 104043 EMT C$GMAN
(3) 023744 000404 BR 10001$
(4) 023746 004066 .WORD OBUFF
(5) 023750 000120 .WORD T$CODE
(5) 023752 007751 .WORD OPRO03
(5) 023754 000001 .WORD 1
(3) 023756 10001$: TST OBUFF ;TEST IF RESPONSE YES
2498 023756 005737 004066 BNE 11$ ;YES - REPORT
2499 023762 001004 BR 1$
2500 023764 000651 10$: CMP R3,T,STAT ;CHECK IF NOW STATE 1
2501 023766 020337 002502 BEQ 13$ ;YES - SKIP
2502 023772 001405 11$: ERRHRD 301...ERR7
2503 023774 104443 TRAP T$ERRCODE
(3) 023774 104443 .WORD 301
(5) 023776 000455 .WORD ERR7
(5) 024000 014220 EXIT TST
2504 024002 104032 EMT C$EXIT
(3) 024004 000676 .WORD L10022-
2505 024006 012701 000454 13$: MOV #300,R1 ;SET WAIT FOR 30 SECONDS
2506 024012 012703 000002 MOV #2,R3 ;SET EXPECTED STATE VALUE
2507 024016 004737 016500 14$: JSR PC,GSTATC ;GET STATUS
2508 024022 024702 T365$
2509 024024 020337 002502 CMP R3,T,STAT ;CHECK IF STATE 2
2510 024030 001435 BEQ 20$ ;YES - SKIP
2511 024032 101005 BHI 17$ ;CHECK IF NO CHANGE - YES - SKIP
2512 024034 302...ERR7 ERRHRD 302...ERR7
(3) 024034 104443 TRAP T$ERRCODE
(5) 024036 000456 .WORD 302
(5) 024040 014220 .WORD ERR7
2513 024042 104032 EXIT TST
(3) 024042 104032 EMT C$EXIT
(3) 024044 000636 .WORD L10022-
2514 024046 005301 17$: DEC R1 ;DEC WAIT COUNT
2515 024050 001404 BEQ 18$ ;SKIP IF C
2516 024052 012700 000001 WAITMS #1
(3) 024052 012700 000001 MOV #1,R0
(3) 024056 104026 EMT C$WTM
2517 024060 000756 BR 14$
2518 024062 303...ERR7 ERRHRD 303...ERR7
(3) 024062 104443 TRAP T$ERRCODE
(5) 024064 000457 .WORD 303
(5) 024066 014220 .WORD ERR7
2519 024070 032737 004000 002474 BIT #SPDSTAT,T,MP ;TEST IF SPINDLE TIMEOUT
    
```

2520	024076	001010			BNE	19\$:YES - SKIP
2521	024100	012737	006262	002434	MOV	#SPDERR,ERHEAD		:SET ERROR HEADER
2522	024106	012703	011172		MOV	#MSPERR,R3		:SET NAME MESSAGE POINTER
2523	024112				ERRHRD	304...ERR3		
(3)	024112	104443			TRAP	T\$ERCODE		
(5)	024114	000460			.WORD	304		
(5)	024116	013154			.WORD	ERR3		
2524	024120			19\$:	EXIT	TST		
(3)	024120	104032			EMT	C\$EXIT		
(3)	024122	000560			.WORD	L10022-		
2525	024124	012737	006225	002434	20\$:	MOV	#MISTST,ERHEAD	:SET ERROR HEADER
2526	024132	012704	011672		MOV	#STATE2,R4		:SET CONDITION MESSAGE POINTER
2527	024136	012703	011105		MOV	#MBHSTA,R3		:SET NAME MESSAGE POINTER
2528	024142	032737	000010	002474	BIT	#BHSTAT,T.MP		:TEST IF BRUSH HOME STILL SET
2529	024150	001005			BNE	22\$:YES - SKIP
2530	024152				ERRHRD	305...ERR5		
(3)	024152	104443			TRAP	T\$ERCODE		
(5)	024154	000461			.WORD	305		
(5)	024156	013272			.WORD	ERR5		
2531	024160				EXIT	TST		
(3)	024160	104032			EMT	C\$EXIT		
(3)	024162	000520			.WORD	L10022-		
2532	024164	012701	000062		22\$:	MOV	#50,R1	:SET WAIT COUNT FOR 5 SECONDS
2533	024170	004737	016500		23\$:	JSR	PC,G\$STATC	:GET STATUS
2534	024174	024702			T365\$			
2535	024176	032737	000010	002474	BIT	#BHSTAT,T.MP		:TEST IF BRUSH HOME RESET
2536	024204	001413			BEQ	27\$:YES - SKIP
2537	024206	005301			DEC	R1		:DEC WAIT COUNT
2538	024210	001404			BEQ	26\$:SKIP IF ZERO
2539	024212				WAITMS	#1		
(3)	024212	012700	000001		MOV	#1,R0		
(3)	024216	104026			EMT	C\$WTM		
2540	024220	000763			BR	23\$:LOOP
2541	024222			26\$:	ERRHRD	306...ERR4		
(3)	024222	104443			TRAP	T\$ERCODE		
(5)	024224	000462			.WORD	306		
(5)	024226	013222			.WORD	ERR4		
2542	024230				EXIT	TST		
(3)	024230	104032			EMT	C\$EXIT		
(3)	024232	000450			.WORD	L10022-		
2543	024234	012701	000454		27\$:	MOV	#300,R1	:SET WAIT COUNT 30 SECONDS
2544	024240	004737	016500		28\$:	JSR	PC,G\$STATC	:GET STATUS
2545	024244	024702			T365\$			
2546	024246	032737	000010	002474	BIT	#BHSTAT,T.MP		:TEST IF BRUSH HOME SET AGAIN
2547	024254	001013			BNE	32\$:YES - SKIP
2548	024256	005301			DEC	R1		:ELSE DEC WAIT COUNT
2549	024260	001404			BEQ	30\$:SKIP IF 0
2550	024262				WAITMS	#1		
(3)	024262	012700	000001		MOV	#1,R0		
(3)	024266	104026			EMT	C\$WTM		
2551	024270	000763			BR	28\$		
2552	024272			30\$:	ERRHRD	307...ERR5		
(3)	024272	104443			TRAP	T\$ERCODE		
(5)	024274	000463			.WORD	307		
(5)	024276	013272			.WORD	ERR5		
2553	024300				EXIT	TST		

```

(3) 024300 104032 EMT C$EXIT
(3) 024302 000400 .WORD L10022-
2554 024304 012737 006250 002434 32$: MOV #NSTACHG,ERHEAD ;SET ERROR HEADER
2555 024312 012703 000003 MOV #3,R3 ;SET EXPECTED STATE VALUE
2556 024316 004737 016500 JSR PC,GSTATC ;GET STATUS
2557 024322 024702 T365$
2558 024324 020337 002502 CMP R3,T.STAT ;CHECK IF STATE 3
2559 024330 001405 BEQ 36$ ;YES - SKIP
2560 024332 ERRHRD 308...ERR7
(3) 024332 104443 TRAP T$ERCODE
(5) 024334 000464 .WORD 308
(5) 024336 014220 .WORD ERR7
2561 024340 EXIT TST
(3) 024340 104032 EMT C$EXIT
(3) 024342 000340 .WORD L10022-
2562 024344 012737 006225 002434 36$: MOV #MISTST,ERHEAD ;SET ERROR HEADER
2563 024352 012704 011702 MOV #STATE3,R4 ;SET CONDITION MESSAGE POINTER
2564 024356 012703 011131 MOV #MHOSTA,R3 ;SET NAME MESSAGE POINTER
2565 024362 004737 016500 JSR PC,GSTATC ;GET STATUS
2566 024366 024702 T365$
2567 024370 032737 000020 002474 BIT #HOSTAT,T.MP ;TEST IF HEADS OUT SET
2568 024376 001005 BNE 38$ ;YES - SKIP
2569 024400 ERRHRD 309...ERR5
(3) 024400 104443 TRAP T$ERCODE
(5) 024402 000465 .WORD 309
(5) 024404 013272 .WORD ERR5
2570 024406 EXIT TST
(3) 024406 104032 EMT C$EXIT
(3) 024410 000272 .WORD L10022-
2571 024412 032737 001000 002474 38$: BIT #VCSTAT,T.MP ;TEST IF VOLUME CHECK SET
2572 024420 001007 BNE 40$
2573 024422 012703 011061 MOV #MVOLCK,R3 ;SET NAME MESSAGE POINTER
2574 024426 ERRHRD 310...ERR5
(3) 024426 104443 TRAP T$ERCODE
(5) 024430 000466 .WORD 310
(5) 024432 013272 .WORD ERR5
2575 024434 EXIT TST
(3) 024434 104032 EMT C$EXIT
(3) 024436 000244 .WORD L10022-
2576 024440 032737 040000 002466 40$: BIT #DRVERR,T.CS ;TEST IF DRIVE ERROR SET
2577 024446 001007 BNE 42$ ;YES - SKIP
2578 024450 012703 011033 MOV #MDRERR,R3 ;SET NAME MESSAGE POINTER
2579 024454 ERRHRD 311...ERR5
(3) 024454 104443 TRAP T$ERCODE
(5) 024456 000467 .WORD 311
(5) 024460 013272 .WORD ERR5
2580 024462 EXIT TST
(3) 024462 104032 EMT C$EXIT
(3) 024464 000216 .WORD L10022-
2581 024466 012701 005670 002434 42$: MOV #3000,R1 ;SET WAIT COUNT FOR 300 MS
2582 024472 012737 006250 002434 42$: MOV #NSTACHG,ERHEAD ;SET ERROR HEADER
2583 024500 012703 000004 MOV #4,R3 ;SET EXPECTED STATE VALUE
2584 024504 004737 016500 43$: JSR PC,GSTATC ;GET STATUS
2585 024510 024702 T365$
2586 024512 020337 002502 CMP R3,T.STAT ;CHECK IF STATE 4
2587 024516 001413 BEQ 49$ ;YES - SKIP

```

```

2588 024520 005301          DEC      R1          ;DEC WAIT COUNT
2589 024522 001404          BEQ      47$         ;SKIP IF 0
2590 024524          WAITUS  #1
      (3) 024524 012700 000001  MOV     #1,R0
      (3) 024530 104027          EMT     C$WTU
2591 024532 000764          BR      43$
2592 024534          47$:  ERRHRD  312...ERR7
      (3) 024534 104443          TRAP   T$ERCODE
      (5) 024536 000470          .WORD  312
      (5) 024540 014220          .WORD  ERR7
2593 024542          EXIT    TST
      (3) 024542 104032          EMT     C$EXIT
      (3) 024544 000136          .WORD  L10022-
2594 024546 012701 000454  49$:  MOV     #300,R1      ;SET WAIT COUNT FOR 30 MS
2595 024552 012703 000005          MOV     #5,R3        ;SET EXPECTED STATE VALUE
2596 024556 004737 016500  50$:  JSR     PC,GSTATC   ;GET STATUS
2597 024562 024702          T365$
2598 024564 020337 002502          CMP     R3,T$STAT   ;CHECK IF STATE 5
2599 024570 001413          BEQ     55$         ;YES - SKIP
2600 024572 005301          DEC     R1          ;DEC WAIT COUNT
2601 024574 001404          BEQ     51$         ;ELSE SKIP
2602 024576          WAITUS  #1
      (3) 024576 012700 000001  MOV     #1,R0
      (3) 024602 104027          EMT     C$WTU
2603 024604 000764          BR      50$
2604 024606          51$:  ERRHRD  313...ERR7
      (3) 024606 104443          TRAP   T$ERCODE
      (5) 024610 000471          .WORD  313
      (5) 024612 014220          .WORD  ERR7
2605 024614          EXIT    TST
      (3) 024614 104032          EMT     C$EXIT
      (3) 024616 000064          .WORD  L10022-
2606 024620 012701 000120  55$:  MOV     #80,R1      ;SET WAIT FOR 8 MS
2607 024624 004737 016500  56$:  JSR     PC,GSTATC   ;GET STATUS
2608 024630 024702          T365$
2609 024632 032737 000001 002466  BIT     #DRDYMSK,T$CS ;CHECK IF DRIVE READY
2610 024640 001020          BNE     62$         ;YES - SKIP
2611 024642 005301          DEC     R1          ;DEC COUNT
2612 024644 001404          BEQ     60$         ;SKIP IF 0
2613 024646          WAITUS  #1
      (3) 024646 012700 000001  MOV     #1,R0
      (3) 024652 104027          EMT     C$WTU
2614 024654 000763          BR      56$
2615 024656 012737 005225 002434  60$:  MOV     #MISTST,ERHEAD ;SET ERROR HEADER
2616 024664 012704 011712          MOV     #STATE5,R4   ;SET CONDITION MESSAGE POINTER
2617 024670 012703 010711          MOV     #MDRDY,R3    ;SET NAME MESSAGE POINTER
2618 024674          ERRHRD  314...ERR5
      (3) 024674 104443          TRAP   T$ERCODE
      (5) 024676 000472          .WORD  314
      (5) 024700 013272          .WORD  ERR5
2619
2620 024702          62$:
2621 024702          T365$:
2622 024702          ENDTST
      (3) 024702          10022:
      (3) 024702 1040C1          EMT     C$WTU

```



```

2624          .SBTTL *TEST 4          HEAD UNLOADING
2625 024704    BGNSTST                ;TEST04
(3) 024704
2626 024704    005737 003062          TST    PASNUM          ;TEST IF FIRST PASS
2627 024710    001003                    BNE    8$              ;NO - SKIP
2628 024712    005737 014656          TST    MISWIW         ;TEST IF MANUAL INTERVENTION
2629 024716    100402                    BMI    10$            ;YES - SKIP
2630 024720    8$: EXIT TST
(3) 024720    104032                    EMT    C$EXIT
(3) 024722    000566                    .WORD  L10023-.
2631 024724    10$:
2632 024724    BGNSTST
(3) 024724
(3) 024724    104002                    EMT    C$BSUB          T4.1:
2633 024726    012737 006250 002434    MOV    #JSTACHG,ERHEAD ;SET ERROR HEADER
2634 024734    004737 016446                    JSR    PC,TSTINT      ;INITIALIZE TEST
2635 024740    004737 016464                    JSR    PC,GSTATR     ;GET STATUS
2636 024744    025400                    T465$
2637 024746    032737 000001 002466    BIT    #DRDYMSK,T.CS  ;CHECK IF DRIVE READY
2638 024754    001040                    BNE    3$              ;YES - SKIP
2639 024756    1$: PRINTF #FMTOP1,#OPR6,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
(13) 024756    005046                    CLR    -(SP)
(13) 024760    153716 002455                    BISS  RLDRV+1,(SP)
(12) 024764    012746 005633                    MOV    #DRVNAM,-(SP)
(11) 024770    013746 002450                    MOV    RLBAS,-(SP)
(10) 024774    012746 005622                    MOV    #BASADD,-(SP)
(9) 025000    012746 010416                    MOV    #OPR1A,-(SP)
(8) 025004    012746 010162                    MOV    #OPR6,-(SP)
(7) 025010    012746 012016                    MOV    #FMTOP1,-(SP)
(6) 025014    012746 000007                    MOV    #7,-(SP)
(3) 025020    010600                    MOV    SP,R0
(4) 025022    104017                    EMT    C$PNTF
(4) 025024    062706 000020                    ADD    #20,SP
2640 025030    005037 004066                    CLR    OBUF           ;CLEAR FOR RESPONSE
2641 025034    GMANIL OPR002,OBUF,1,NO
(3) 025034    104043                    EMT    C$GMAN
(3) 025036    000404                    BR     10000$
(4) 025040    004066                    .WORD  OBUF
(5) 025042    000120                    .WORD  T$CODE
(5) 025044    007724                    .WORD  OPR002
(5) 025046    000001                    .WORD  1
(3) 025050    10000$:
2642 025050    005737 004066                    TST    OBUF           ;TST RESPONSE YES
2643 025054    001740                    BEQ    1$              ;NO - SKIP
2644
2645 025056    052737 000010 002426 3$: BIS    #UNLOAD,OPFLAG ;SET UNLOAD OPERATION
2646 025064    4$: PRINTF #FMTOP1,#OPR3,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
(13) 025064    005046                    CLR    -(SP)
(13) 025066    153716 002455                    BISS  RLDRV+1,(SP)
(12) 025072    012746 005633                    MOV    #DRVNAM,-(SP)
(11) 025076    013746 002450                    MOV    RLBAS,-(SP)
(10) 025102    012746 005622                    MOV    #BASADD,-(SP)
(9) 025106    012746 010416                    MOV    #OPR1A,-(SP)
(8) 025112    012746 010104                    MOV    #OPR3,-(SP)
(7) 025116    012746 012016                    MOV    #FMTOP1,-(SP)
(6) 025122    012746 000007                    MOV    #7,-(SP)

```

```

(3) 025126 010600      MOV      SP,R0
(4) 025130 104017      EMT      C$PNTF
(4) 025132 062706 000020  ADD      #20,SP
2647 025136 012703 000006  MOV      #6,R3      ;SET EXPECTED STATE VALUE
2648 025142 012704 000144  MOV      #100.,R4   ;SET SECOND LEVEL COUNT
2649 025146 012701 001274  MOV      #700.,R1  ;SET WAIT COUNT FOR 30 SECONDS
2650 025152 004737 016500 5$:      JSR      PC,GSTATC ;GET STATUS
2651 025156 025400      T465$
2652 025160 020337 002502  CMP      R3,T.STAT ;CHECK IF STATE 6
2653 025164 001436      BEQ      11$        ;YES - SKIP
2654 025166 022737 000005 002502  CMP      #5,T.STAT ;TEST IF STATE 5
2655 025174 001025      BNE      9$         ;NO - REPORT WRONG STATE
2656 025176 005304 8$:      DEC      R4         ;DEC 2ND LEVEL COUNT
2657 025200 001004      BNE      6$        ;SKIP IF NOT 0
2658 025202 005301      DEC      R1         ;ELSE DEC 1ST LEVEL COUNT
2659 025204 001406      BEQ      7$        ;IF 0 - SKIP TO QUESTION
2660 025206 012704 000144  MOV      #100.,R4   ;ELSE RESET 2ND LEVEL
2661 025212 6$:      WAITUS  #1         ;WAIT 100 US
(3) 025212 012700 000001  MOV      #1,R0
(3) 025216 104027      EMT      C$WTU
2662 025220 000754      BR       5$
2663 025222 005037 004066 7$:      CLR      OBUF      ;CLEAR FOR RESPONSE
2664 025226      GMANIL  OPR003,OBUF,1,NO
(3) 025226 104043      EMT      C$GMAN
(3) 025230 000404      BR       10001$
(4) 025232 004066      .WORD   OBUF
(5) 025234 000120      .WORD   T$CODE
(5) 025236 007751      .WORD   OPR003
(5) 025240 000001      .WORD   1
(3) 025242 10001$:
2665 025242 005737 004066  TST      OBUF      ;TEST IF RESPONSE YES
2666 025246 001706      BEQ      4$        ;NO - SKIP
2667 025250 9$:      ERRHRD  401.,ERR7  ;ELSE REPORT STATE CHANGE WRONG
(3) 025250 104443      TRAP    T$ERCODE
(5) 025252 000621      .WORD   401
(5) 025254 014220      .WORD   ERR7
2668 025256      EXIT      SUB
(3) 025256 104032      EMT      C$EXIT
(3) 025260 000126      .WORD   L10024-.
2669 025262 012703 000007 11$:     MOV      #7,R3      ;SET EXPECTED STATE VALUE
2670 025266 012701 005670      MOV      #3000.,R1 ;SET COUNT FOR 300MS
2671 025272 004737 016500 12$:     JSR      PC,GSTATC ;GET STATUS
2672 025276 025400      T465$
2673 025300 020337 002502  CMP      R3,T.STAT ;CHECK IF STATE 7
2674 025304 001413      BEQ      18$        ;YES - SKIP
2675 025306 005301      DEC      R1         ;DEC WAIT COUNT
2676 025310 001404      BEQ      16$        ;SKIP IF 0
2677 025312      WAITUS  #1
(3) 025312 012700 000001  MOV      #1,R0
(3) 025316 104027      EMT      C$WTU
2678 025320 000764      BR       12$
2679 025322 16$:     ERRHRD  402.,ERR7  ;REPORT WRONG STATE CHANGE
(3) 025322 104443      TRAP    T$ERCODE
(5) 025324 000622      .WORD   402
(5) 025326 014220      .WORD   ERR7
2680 025330      EXIT      SUB

```

```

(3) 025330 104032      EMT      C$EXIT
(3) 025332 000054      .WORD    L10024-.
2681 025334 005003      18$:    CLR      R3          ;SET EXPECTED STATE VALJE
2682 025336 012701 001130    MOV      #600,R1      ;SET WAIT COUNT FOR 60 SECONDS
2683 025342 004737 016500    20$:    JSR      PC,GSTATC ;GET STATUS
2684 025346 025400      T465$
2685 025350 005737 002502    TST      T,STAT      ;CHECK IF STATE 0
2686 025354 001411      BEQ      24$         ;YES - SKIP
2687 025356 005301      DEC      R1          ;DEC WAIT COUNT
2688 025360 001404      BEQ      22$         ;SKIP IF 0
2689 025362
(3) 025362 012700 000001    WAITMS  #1
(3) 025366 104026      MOV      #1,R0
2690 025370 000764      EMT      C$WTM
2691 025372      BR       20$
(3) 025372 104443      22$:    ERRHRD 403,,ERR7 ;REPORT WRONG STATE CHANGE
(5) 025374 000623      TRAP    T$ERCODE
(5) 025376 014220      .WORD   403
2692 025400      .WORD   ERR7
2693 025400 012737 000002 002440 24$:    MOV      #2,ERRSWI ;INIT ERROR SWITCH
2694
2695 025406      ENDSUB
(3) 025406      L10024:
(3) 025406 104003      EMT      C$ESUB
2696 025410      26$:    PRINTF #FMTOP1,#OPR6,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1> ;REQUEST CYCLE UP
(13) 025410 005046      CLR      -(SP)
(13) 025412 153716 002455    BISB    RLDRV+1,(SP)
(12) 025416 012746 005633    MOV     #DRVNAM,-(SP)
(11) 025422 013746 002450    MOV     RLBAS,-(SP)
(10) 025426 012746 005622    MOV     #BASADD,-(SP)
(9) 025432 012746 010416   MOV     #OPR1A,-(SP)
(8) 025436 012746 010162   MOV     #OPR6,-(SP)
(7) 025442 012746 012016   MOV     #FMTOP1,-(SP)
(6) 025446 012746 000007   MOV     #7,-(SP)
(3) 025452 010600      MOV     SP,R0
(4) 025454 104017      EMT      C$PNTF
(4) 025456 062706 000020    ADD     #20,SP
2697 025462 005037 004066    CLR     OBUFF ;CLEAR FOR RESPONSE
2698 025466      GMANIL  OPR002,OBUFF,1,NO
(3) 025466 104043      EMT      C$GMAN
(3) 025470 000404      BR       10000$
(4) 025472 004066      .WORD   OBUFF
(5) 025474 000120      .WORD   T$CODE
(5) 025476 007724      .WORD   OPR002
(5) 025500 000001      .WORD   1
(3) 025502      10000$:
2699 025502 005737 004066    TST     OBUFF ;TEST RESPONSE YES
2700 025506 001740      BEQ     26$         ;NO - SKIP
2701 025510
2702
2703 025510      29$:
(3) 025510      ENDTST
(3) 025510 104001      L10023:
(3) 025510      EMT      C$ETST
  
```

```

2705          .SBTTL *TEST 5          DRIVE SELECT
2706 025512    BGNTST                ;TEST05
(3) 025512
2707 025512    012737 000002 002440    MOV #2,ERHSWI ;SET FOR NO ERROR RETURN
2708 025520    005737 003062          TST PASNUM ;TEST IF FIRST PASS
2709 025524    001173          BNE EXT05 ;NO - SKIP
2710 025526    032737 000004 014656    BIT #DRSELT,MISWIW ;TEST IF SELECT TESTS
2711 025534    001567          BEQ EXT05 ;NO - SKIP
2712 025536    1$: PRINTF #FMTOP1,#OPR7,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1> ,REQUEST REMOVE A
(13) 025536    005046          CLR -(SP)
(13) 025540    153716 002455          BISB RLDRV+1,(SP)
(12) 025544    012746 005633          MOV #DRVNAM,-(SP)
(11) 025550    013746 002450          MOV RLBAS,-(SP)
(10) 025554    012746 005622          MOV #BASADD,-(SP)
(9) 025560    012746 010416          MOV #OPR1A,-(SP)
(8) 025564    012746 010215          MOV #OPR7,-(SP)
(7) 025570    012746 012016          MOV #FMTOP1,-(SP)
(6) 025574    012746 000007          MOV #7,-(SP)
(3) 025600    010600          MOV SP,R0
(4) 025602    104017          EMT C$PNTF
(4) 025604    062706 000020          ADD #20,SP
2713 025610    005037 004066          CLR OBUFF ;CLEAR FOR RESPONSE
2714 025614    GMANIL OPR002,OBUFF,1,NO
(3) 025614    104043          EMT C$GMAN
(3) 025616    000404          BR 10000$
(4) 025620    004066          .WORD OBUFF
(5) 025622    000120          .WORD T$CODE
(5) 025624    007724          .WORD OPR002
(5) 025626    000001          .WORD 1
(3) 025630    10000$:
2715 025630    005737 004066          TST OBUFF ;TEST RESPONSE YES
2716 025634    001740          BEQ 1$ ;NO - SKIP
2717 025636    012737 006362 002434 3$: MOV #T05ERR,ERHEAD ;SET ERROR HEADER MESSAGE
2718 025644    004737 016446          JSR PC,TSTINT ;INITIALIZE TEST
2719 025650    004737 016500          JSR PC,GSTATC ;DO SELECT AND GET STATUS
2720 025654    026036          T504$
2721 025656    013737 002454 002540    MOV RLDRV,TEMPO ;STORE ORIGINAL DRIVE NUMBER
2722 025664    013701 002454          MOV RLDRV,R1 ;PUT IT IN R1
2723 025670    012704 000004          MOV #4,R4 ;SET COUNT FOR NUMBER OF PLUGS
2724 025674    062701 000400          LPT05: ADD #400,R1 ;BUMP TO NEXT DRIVE
2725 025700    022701 002000          CMP #2000,R1 ;CHECK IF TOO LARGE
2726 025704    001001          BNE 4$ ;NO - SKIP
2727 025706    005001          CLR R1 ;ELSE CLEAR TO DRIVE 0
2728 025710    010137 002454          4$: MOV R1,RLDRV ;PUT IT BACK IN RLDRV
2729 025714    5$: PRINTF #FMTOP3,#OPR8,<B,RLDRV+1>,#OPR1B,#UNDTST
(11) 025714    012746 010432          MOV #UNDTST,-(SP)
(10) 025720    012746 010422          MOV #OPR1B,-(SP)
(9) 025724    005046          CLR -(SP)
(9) 025726    153716 002455          BISB RLDRV+1,(SP)
(8) 025732    012746 010244          MOV #OPR8,-(SP)
(7) 025736    012746 012067          MOV #FMTOP3,-(SP)
(6) 025742    012746 000005          MOV #5,-(SP)
(3) 025746    010600          MOV SP,R0
(4) 025750    104017          EMT C$PNTF
(4) 025752    062706 000014          ADD #'',SP
2730          ;INSERT PLUG REQUEST
  
```

```

2731 025756 005037 004066 CLR OBUFF ;CLEAR FOR RESPONSE
2732 025762 GMANIL OPR002,OBUFF,1,NO
(3) 025762 104043 EMT C$GMAN
(3) 025764 000404 BR 10001$
(4) 025766 004066 .WORD OBUFF
(5) 025770 000120 .WORD T$CODE
(5) 025772 007724 .WORD OPR002
(5) 025774 000001 .WORD 1
(3) 025776 10001$:
2733 025776 005737 004066 TST OBUFF ;TEST RESPONSE YES
2734 026002 001744 BEQ 5$ ;NO - SKIP
2735 026004 BGNSUB
(3) 026004 T5.1:
(3) 026004 104002 EMT C$BSUB
2736 026006 004737 016500 JSR PC,GSTATC ;GET STATUS - REPORT ANY FRROR
2737 026012 026014 60$ MOV #2,ERRSWI ;INIT ERROR SWITCH
2738 026014 012737 000002 002440
2739 026022 ENDSUB
(3) 026022 L10020:
(3) 026022 104003 EMT C$ESJB
2741 026024 005304 DEC R4 ;DEC COUNT
2742 026026 001322 BNE LPT05 ;LOOP IF NOT ZERO
2743 026030 013737 002540 002454 MOV TEMPO,RLDRV ;ELSE RESTORE RLDRV
2744 026036 T504$:
2745 026036 4$: PRINTF #FMT1,#OPR8,#OPR9
(9) 026036 012746 010263 MOV #OPR9,-(SP)
(8) 026042 012746 010244 MOV #OPR8,-(SP)
(7) 026046 012746 012110 MOV #FMT1,-(SP)
(6) 026052 012746 000003 MOV #3,-(SP)
(3) 026056 010600 MOV SP,R0
(4) 026060 104017 EMT C$PNTF
(4) 026062 062706 000010 ADD #10,SP
2746 026066 005037 004066 CLR OBUFF ;CLEAR FOR RESPONSE
2747 026072 GMANIL OPR002,OBUFF,1,NO
(3) 026072 104043 EMT C$GMAN
(3) 026074 000404 BR 10000$
(4) 026076 004066 .WORD OBUFF
(5) 026100 000120 .WORD T$CODE
(5) 026102 007724 .WORD OPR002
(5) 026104 000001 .WORD 1
(3) 026106 10000$:
2748 026106 005737 004066 TST OBUFF ;TEST RESPONSE YES
2749 026112 001751 BEQ 4$ ;NO - SKIP
2750 026114 EXT05:
2751 026114 ENDTST
(3) 026114 L10025:
(3) 026114 104001 EMT C$ETST
2752

```

```

2754          .SBTTL *TEST 6          DRIVE SELECT TEST
2755 026116   BGNTST                  ;TEST06
(3) 026116
2756 026116   065737 003062          TS-   PASNUM          ;CHECK IF FIRST PASS
2757 026122   001004          BNE    1$           ;NO - SKIP
2758 026124   032737 000004 014656  BIT    #DRSELT,MISWIW ;CHECK IF TEST DRIVE SELECT
2759 026132   001002          BNE    4$           ;YES - SKIP
2760 026134   1$:
(3) 026134   104032          EXIT   TST
(3) 026136   000630          EMT   C$EXIT
2761 026140   012737 006316 002434  4$:   .WORD    L10027-
2762 026146   004737 016446          MOV   #GSTER1,ERHEAD ;SET ERROR HEADER
2763 026152   013703 003064          JSR   PC,TSTINT      ;INITIALIZE TEST
2764 026156   023727 002012 000001  MOV   PS$TM,R3       ;GET PARAM SET NUMBER
2765 026164   101450          CMP   L$UNIT,#1      ;TEST IF MORE THAN 1 UNIT
2766 026166   005203          BLOS  5$           ;NO - SKIP
2767 026170   020337 002012          INC   R3            ;BUMP PARAMETER SET NUMBER
2768 026174   101401          CMP   R3,L$UNIT     ;CHECK IF PAST VALID PARAMETER TABLE
2769 026176   005003          BLOS  3$           ;NO - SKIP
2770 026200   3$:   GPHARD R3,R0        ;ELSE CLEAR TO POINT TO ENTRY 0
(3) 026200   010300          MOV   R3,R0
(3) 026202   104042          EMT   C$GPHRD
2771 026204   103370          BNCOMPLETE 2$      ;SKIP IF NOT AVAILABLE
(2) 026204   010004          BCC   2$
2772 026210   021437 002450          MOV   R0,R4         ;PUT POINTER INTO R4
2773 026214   001364          CMP   (R4),RLBAS    ;CHECK IF SAME CONTROLLER
2774 026216   005037 002430          BNE   2$           ;NO - SKIP
2775 026222   012737 016514 002456  CLR   DONE          ;CLEAR DONE FLAG
2776 026230   056437 000006 002456  MOV   #GSTAT,L$CS   ;LOAD GET STATUS
2777 026236   012737 000013 002462  BIS   6(R4),L$CS    ;INSERT DRIVE
2778 026244   013762 002462 000004  MOV   #GETSTAT!DRSET,L$DA ;SET UP TO CLEAR DRIVE
2779 026252   013762 002456 000000  MOV   L$DA,RLDA(R2) ;LOAD DA REG
2780 026260   012700 000003          MOV   L$CS,RLCS(R2) ;LOAD CS REG
(3) 026260   104026          WAITMS #3          ;WAIT 300 MS
(3) 026264   005737 002430          MOV   #3,R0
2782 026266   001735          EMT   C$WTM
2783 026272   032737 100000 002466  TST   DONE          ;TEST IF INTEPRUPT
2784 026274   001415          BEQ   2$           ;NO - SKIP
2785 026302   000730          BIT   #ANYERR,T$CS ;TEST IF ANY ERROR SET
2786 026304   012746 010300          BEQ   7$           ;NO - GO TEST
2787 026306   012746 012327          BR    2$          ;ELSE CHECK NEXT DRIVE
(8) 026306   012746 000002          PRINTF #FMT9,#OPR10 ;REPORT CAN'T FIND 2ND DRIVE
(7) 026312   012746 000002          MOV   #OPR10,-(SP)
(6) 026316   010600          MOV   #FMT9,-(SP)
(3) 026322   104017          MOV   #2,-(SP)
(4) 026324   02706 000006          MOV   SP,R0
(4) 026326   000137 026766          EMT   C$PNTF
2788 026332   016437 000006 002542  7$:   ADD   #6,SP
2789 026336   000006 002542          JMP   LC$EXT
2790          MOV   6(R4),TEMP1 ;STORE NEW ADDRESS
2791 026344   013700 002454          ;ASK FOR PLUG CHANGE
2792 026350   013705 002542          MOV   RLDRV,R0     ;GET DRIVE UNDER TEST
2793 026354   042700 002000          MOV   TEMP1,R5    ;GET NEW ADDRESS
2794 026360   042700 002000          BIC   #2000,R0     ;CLEAR FOR ADDRESS 0 TO 3
2795 026364   020527 001400          BIC   #2000,R0
20$:   CMP   R5,#1400    ;TEST IF DRIVE NUMBER 3
  
```

```

2796 026370 001001      BNE      21$          ;NO - SKIP
2797 026372 005005      CLR      R5          ;ELSE SET TO DRIVE NUMBER 0
2798 026374 062705 000400 21$:      ADD      #400,R5     ;BUMP TO NEXT ADDRESS
2799 026400 020500      CMP      R5,R0      ;THIS EQUAL TO NEW ADDRESS?
2800 026402 001770      BEQ      20$          ;YES - SKIP
2801 026404 052705 000200      BIS      #CRDYMSK,R5 ;ELSE SET CONTROLLER READY BIT
2802 026410 010562 000000      MOV      R5,RLCS(R2) ;AND LOAD CS REG
2803 026414      PRINTF  #FMTOP2,#OPR8,<B,RLDRV+1>,#OPR1B,<B,TEMP1+1>
(11) 026414 005046      CLR      -(SP)
(11) 026416 153716 002543      BISB    TEMP1+1,(SP)
(10) 026422 012746 010422      MOV      #OPR1B,-(SP)
(9) 026426 005046      CLR      -(SP)
(9) 026430 153716 002455      BISB    RLDRV+1,(SP)
(8) 026434 012746 010244      MOV      #OPR8,-(SP)
(7) 026440 012746 012045      MOV      #FMTOP2,-(SP)
(6) 026444 012746 000005      MOV      #5,-(SP)
(3) 026450 010600      MOV      SP,R0
(4) 026452 104017      EMT      C$PNTF
(4) 026454 062706 000014      ADD      #14,SP
2804 026460 005037 004066      CLR      OBUFF      ;CLEAR FOR RESPONSE
2805 026464      GMANIL  OPR002,OBUFF,1,NO
(3) 026464 104043      EMT      C$GMAN
(3) 026466 000404      BR       10000$
(4) 026470 004066      .WORD   OBUFF
(5) 026472 000120      .WORD   T$CODE
(5) 026474 007724      .WORD   OPR002
(5) 026476 000001      .WORD   1
(3) 026500      10000$:
2806 026500 005737 004066      TST      OBUFF      ;TEST IF RESPONSE YES
2807 026504 001717      BEQ      9$          ;NO - SKIP
2808 026506 012704 000012      MOV      #10.,R4    ;SET COUNT
2809 026512      BGNSUB
(3) 026512      T6.1:
(3) 026512 104002      EMT      C$BSUB
2810 026514 013737 002454 002456 8$:      MOV      RLDRV,L.CS ;SET UP TO SELECT MULTIPLE DRIVES
2811 026522 013762 002456 000000      MOV      L.CS,RLCSR(R2) ;DO IT
2812 026530      WAITMS #10.
(3) 026530 012700 000012      MOV      #10.,R0
(3) 026534 104026      EMT      C$WTM
2813 026536 052737 000104 002456      BIS      #GTSTAT,L.CS ;SET GET STATUS
2814 026544 012737 000003 002462      MOV      #GETSTAT,L.DA
2815 026552 013762 002462 000004      MOV      L.DA,RLDA(R2)
2816 026560 005037 002430      CLR      DONE
2817 026564 013762 002456 000000      MOV      L.CS,RLCSR(R2) ;DO GET STATUS
2818 026572      WAITUS #1          ;WAIT FOR INTERRUPT
(3) 026572 012700 000001      MOV      #1,R0
(3) 026576 104027      EMT      C$WTU
2819 026600 005737 002430      TST      DONE      ;CHECK IF INTERRUPTED
2820 026604 001012      BNE      12$          ;YES - SKIP
2821 026606 004737 016314      JSR      PC,WAITIN  ;WAIT FOR TIMEOUT
2822 026612 012603      MOV      (SP)+,R3    ;GET ERROR POINTER
2823 026614 001406      BEQ      12$          ;SKIP IF 0
2824 026616      ERRHRD 601.,GSTER1,ERR1
(3) 026616 104463      TRAP    T$ERCODE
(5) 026620 001131      .WORD   601
(5) 026622 006316      .WORD   GSTER1
  
```

```

(5) 026624 013040          .WORD  ERR1
2825 026626          EXIT   SUB
(3) 026626 104032          EMT   C$EXIT
(3) 026630 000062          .WORD  L10030-.
2826 026632          12$:  WAITMS #2          ;WAIT FOR DSE TO SET
(3) 026632 012700 000002  MOV   #2,R0
(3) 026636 104026          EMT   C$WTM
2827 026640 004737 016514  JSR   PC,GSTAT          ;GET STATUS
2828 026644 026704          60$:  BIT   #DSESTAT,T.MP          ;TEST IF DRIVE SELECT ERROR SET
2829 026646 032737 000400 002474  BNE   16$          ;YES - SKIP
2830 026654 001007          MOV   #MDSERR,R3          ;SET NAME MESSAGE POINTER
2831 026656 012703 011142  ERRHRD 602...ERR3
2832 026662          TRAP  T$ERCODE
(3) 026662 104443          .WORD  602
(5) 026664 001132          .WORD  ERR3
(5) 026666 013154          EXIT  SUB
2833 026670          EMT   C$EXIT
(3) 026670 104032          .WORD  L10030-.
(3) 026672 000020          16$:  MOV   R5,RLCS(R2)          ;LOAD IN DIFFERENT ADDRESS
2834 026674 010562 000000  DEC   R4          ;DEC COUNT
2835 026700 005304          BNE   8$          ;LOOP IF NOT ZERO
2836 026702 001304          60$:  MOV   #2,ERRSWI          ;INIT ERROR SWITCH
2837 026704 012737 000002 002440  ENDSUB
2838 026712          L10030:
(3) 026712 104003          EMT   C$ESUB
2839 026714          15$:  PRINTF #FMT9,#OPR11          ;REQUEST PLUG CHANGE
(8) 026714 012746 010346  MOV   #OPR11,-(SP)
(7) 026720 012746 012327  MOV   #FMT9,-(SP)
(6) 026724 012746 000002  MOV   #2,-(SP)
(3) 026730 010600          MOV   SP,R0
(4) 026732 104017          EMT   C$PNTF
(4) 026734 062706 000006  ADD   #6,SP
2840 026740 005037 004066  CLR   OBUF          ;CLEAR FOR RESPONSE
2841 026744          GMANIL OPR002,OBUF,1,NO
(3) 026744 104043          EMT   C$GMAN
(3) 026746 000404          BR   10000$
(4) 026750 004066          .WORD  OBUF
(5) 026752 000120          .WORD  T$CODE
(5) 026754 007724          .WORD  OPR002
(5) 026756 000001          .WORD  1
(3) 026760          10000$:
2842 026760 005737 004066  TST   OBUF          ;TEST RESPONSE YES
2843 026764 001753          BEQ   15$          ;NO - SKIP
2844 026766          LCLEXT:
2845 026766          ENDTST
(3) 026766          L10027:
(3) 026766 104001          EMT   C$ETST
  
```



```

2847          .SBTTL *TEST 7          INITIAL STATE
2848 026770    BGNTST                ;TEST 07
(3) 026770
2849 026770    012737 006347 002434    MOV    #INITST,ERHEAD ;SET ERROR HEADER      T7::
2850 026776    004737 016446          JSR    PC,TSTINT      ;INITIALIZE TEST
2851 027002    WAITUS #10.           ;WAIT 1 MS
(3) 027002    012700 000012          MOV    #10.,R0
(3) 027006    104027          EMT    C$WTU
2852 027010    004737 016500          JSR    PC,GSTATC     ;GET STATUS
2853 027014    027300          65$
2854 027016    032737 000001 002466    BIT    #DRDYMSK,T.CS ;CHECK IF DRIVE READY
2855 027024    001003          BNE    3$           ;YES-SKIP
2856 027026    012703 010711          MOV    #MDRDY,R3    ;SET NAME MESSAGE POINTER
2857 027032    000427          BR     9$           ;GO REPORT
2858 027034    012703 000005          3$: MOV    #5,R3       ;SET EXPECTED STATE VALUE
2859
2860 027040    020337 002502          CMP    R3,T.STAT    ;CHECK IF STATE OK
2861 027044    001405          BEQ    5$           ;YES-SKIP
2862 027046    ERRHRD 701...ERR7      ;ELSE REPORT STATE ERROR
(3) 027046    104443          TRAP  T$ERRCODE
(5) 027050    001275          .WORD 701
(5) 027052    014220          .WORD ERR7
2863 027054    EXIT TST                ;EXIT
(3) 027054    104032          EMT    C$EXIT
(3) 027056    000222          .WORD L10031-.
2864 027060    013701 002474          5$: MOV    T.MP,R1    ;GET MP REG
2865 027064    032701 000020          BIT    #HOSTAT,R1  ;CHECK HEADS OUT
2866 027070    001003          BNE    7$           ;YES-SKIP
2867 027072    012703 011131          MOV    #MHOSTA,R3  ;SET NAME MESSAGE PTR
2868 027076    000405          BR     9$           ;GO REPORT
2869 027100    032701 000010          7$: BIT    #BHSTAT,R1 ;CHECK BRUSH HOME SET
2870 027104    001007          BNE    10$          ;YES-SKIP
2871 027106    012703 011105          MOV    #MBHSTA,R3  ;SET NAME MESSAGE PTR
2872 027112    9$: ERRHRD 702...ERR3      ;REPORT ERROR
(3) 027112    104443          TRAP  T$ERRCODE
(5) 027114    001276          .WORD 702
(5) 027116    013154          .WORD ERR3
2873 027120    EXIT TST                ;EXIT
(3) 027120    104032          EMT    C$EXIT
(3) 027122    000156          .WORD L10031-.
2874 027124    005737 014656          10$: TST    MISWIW    ;TEST IF MANUAL INTERVENTION RUN
2875 027130    100034          BPL    16$          ;NO-SKIP
2876 027132    005737 003062          TST    PASNUM      ;CHECK IF FIRST PASS
2877 027136    001031          BNE    16$          ;NO-SKIP
2878 027140    032701 000100          BIT    #HSSTAT,R1  ;ELSE CHECK HD 0 SELECTED
2879 027144    001411          BEQ    13$          ;YES-SKIP
2880 027146    012703 011044          MOV    #MHSTA,R3   ;SET NAME MESSAGE PTR
2881 027152    012704 011763          MOV    #CCYLJP,R4  ;SET CONDITION POINTER
2882 027156    ERRHRD 703...ERR4      ;REPORT ERROR
(3) 027156    104443          TRAP  T$ERRCODE
(5) 027160    001277          .WORD 703
(5) 027162    013222          .WORD ERR4
2883 027164    EXIT TST                ;EXIT
(3) 027164    104032          EMT    C$EXIT
(3) 027166    000112          .WORD L10031-.
2884 027170    032701 001000          13$: BIT    #VOSTAT,R1 ;CHECK VOL CHECK SET
  
```

2885	027174	001003			BNE	15\$:YES-SKIP
2886	027176	012703	011061		MOV	#MVLCK,R3	:ELSE SET NAME MESSAGE PTR
2887	027202	000743			BR	9\$:GO REPORT
2888	027204	032737	040000	002466	15\$:	BIT	#DRVEAR,T.CS
2889	027212	001003			BNE	16\$:TEST DRIVE ERROR SET
2890	027214	012703	011033		MOV	#MDRERR,R3	:YES-SKIP
2891	027220	000734			BR	9\$:ELSE SET NAME MESSAGE PTR
2892	027222	032701	020000		16\$:	BIT	#WLSTAT,R1
2893	027226	001405			BEQ	17\$:CHECK WRITE LOCK STATUS
2894	027230	012703	011120		MOV	#MWLSTA,R3	:SKIP IF RESET
2895	027234				ERRHRD	705...ERR2	:ELSE SET NAME MESSAGE PTR
(3)	027234	104443			TRAP	T\$ERCODE	
(5)	027236	001301			.WORD	705	
(5)	027240	013106			.WORD	ERR2	
2896	027242	042701	021177		17\$:	BIC	#21177,R1
2897	027246	005701			TST	R1	:CLEAR STATUS EXCEPT FOR ERROR BITS
2898	027250	001405			BEQ	19\$:CHECK IF ANY ERROR SET
2899	027252				18\$:	ERRHRD	704...ERR6
(3)	027252	104443			TRAP	T\$ERCODE	:NO-SKIP
(5)	027254	001300			.WORD	704	:ELSE REPORT ALL ERRORS
(5)	027256	013342			.WORD	ERR6	
2900	027260				EXIT	TST	:EXIT
(3)	027260	104032			EMT	C\$EXIT	
(3)	027262	000016			.WORD	L10031-	
2901	027264	013701	002466		19\$:	MOV	T.CS,R1
2902	027270	042701	141777		BIC	#141777,R1	:GET CS REG
2903	027274	005701			TST	R1	:CLEAR ALL BUT ERROR BITS
2904	027276	001365			BNE	18\$:TEST IF ANY ERROR SET
2905	027300				25\$:		:YES-SKIP TO REPORT
2906	027300				65\$:		
2907	027300				ENDTST		
(3)	027300				L10031:		
(3)	027300	104001			EMT	C\$ETST	

```

2909
2910
2911          .SBTTL *TEST 8          INITIAL RESET STATE
2912 027302          BGNTST          ;TEST 8
(3) 027302
2913 027302 012737 006347 002434      MOV #INITST,ERHEAD
2914 027310 004737 016446              JSR PC,TSTINT          ;INITIALIZE TEST
2915
2916 027314 004737 016464              JSR PC,GSTATR          ;GET STATUS WITH RESET
2917 027320 027364
2918 027322 005737 014656              65$
2919 027326 100016              TST MISWIW            ;CHECK IF MAN INTERVENTION WAS RUN
2920 027330 005737 003062              BPL 4$                ;NO-SKIP
2921 027334 001013              TST PASNUM            ;CHECK IF 1ST PASS
2922 027336 032737 000100 002474      BNE 4$                ;NO-SKIP
2923 027344 001407              BIT #HSSTAT,T.MP      ;CHECK HD SELECT STILL 0
2924 027346 012703 011044              BEQ 4$                ;YES-SKIP
2925 027352 012704 011763              MOV #MHSTA,R3          ;SET NAME MESSAGE PTR
2926 027356          ERRHRD 80!.,ERR4    ;SET CONDITION POINTER
(3) 027356 104443              TRAP T$ERCODE         ;REPORT ERROR
(5) 027360 001441
(5) 027362 013222
2927 027364          4$:
2928 027364          65$:
2929 027364          ENDTST
(3) 027364          L10032:
(3) 027364 104001          EMT C$ETST
2930
  
```

```

2932
2933
2934          .SBTTL *TEST 9          DRIVE READY
2935          BGNTST                   :TEST 9
2936          (3) 027366                T9::
2937          027366 012737 006375 002434 MOV #T09ERR,ERHEAD ;SET ERROR HEADER
2938          027374 012701 002524 MOV #NEWCYL,R1 ;GET POINTER TO DESIRED LOC
2939          027400 005021 CLR (R1)+ ;CLEAR NEW CYL
2940          027402 005021 CLR (R1)+ ;CLEAR CURRENT CYL
2941          027404 005021 CLR (R1)+ ; DIFFERENCE
2942          027406 005011 CLR (R1) ; SIGN
2943          027410 004737 016446 JSR PC,TSTINT ;INITIALIZE TEST
2944          027414 004737 016464 JSR PC,GSTATR ;GET STATUS WITH RESET
2945          027420 027664 65$
2946          027422 004737 020530 JSR PC,POSHSB ;POSITION HEAD SELECTED BIT
2947          027426 010537 002534 MOV R5,DESHD ;STORE AS DESIRED HEAD
2948          027432 004737 017346 JSR PC,SIMSEK ;EXECUTE SIMPLE SEEK
2949          027436 027664 65$
2950          027440 012703 010711 MOV #MDRDY,R3 ;SET NAME MESSAGE PTR
2951          027444 012704 011722 MOV #CDRDY,R4 ;SET CONDITION POINTER
2952          027450 004737 016514 JSR PC,GSTAT ;GET STATUS
2953          027454 027664 65$
2954          027456 032737 000001 002466 BIT #DRDYMSK,T.CS ;TEST READY SET
2955          027464 001405 BEQ 4$ ;NO-SKIP
2956          (3) 027466 104443 ERRHRD 901...ERR4 ;REPORT READY ERROR
2957          (5) 027470 001605 TRAP T$ERRCODE
2958          (5) 027472 013222 .WORD 901
2959          027474 .WORD ERR4
2960          (3) 027474 104032 EXIT TST ;EXIT
2961          (3) 027476 000166 EMT C$EXIT
2962          027500 012701 000121 4$: MOV #81,R1 ;SET WAIT COUNT
2963          027504 004737 016514 5$: JSR PC,GSTAT ;GET STATUS
2964          027510 027664 65$
2965          027512 012703 000005 MOV #5,R3 ;SET EXPECTED STATE VALUE
2966          027516 023703 002502 CMP T,STAT,R3 ;CHECK STATE IS 5
2967          027522 001405 BEQ 7$ ;YES-SKIP
2968          027524 ERRHRD 902...ERR7 ;ELSE REPORT
2969          (3) 027524 104443 TRAP T$ERRCODE
2970          (5) 027526 001606 .WORD 902
2971          (5) 027530 014220 .WORD ERR7
2972          027532 EXIT TST
2973          (3) 027532 104032 EMT C$EXIT
2974          (3) 027534 000130 .WORD L10033-
2975          027536 012703 010711 7$: MOV #MDRDY,R3
2976          027542 032737 000001 002466 BIT #DRDYMSK,T.CS ;CHECK READY SET
2977          027550 001013 BNE 12$ ;YES-SKIP
2978          027552 005301 DEC R1 ;ELSE DEC WAIT COUNT
2979          027554 001404 BEQ 9$ ;SKIP IF 0
2980          027556 WAITUS #1
2981          (3) 027556 012700 000001 MOV #1,R0
2982          (3) 027562 104027 EMT C$WTU
2983          027564 000747 BR 5$
2984          027566 001404 9$: ERRHRD 903...ERR5 ;REPORT READY ERROR
2985          (3) 027566 104443 TRAP T$ERRCODE
2986          (5) 027570 001607 .WORD 903
  
```

```

(5) 027572 013272          .WORD  ERR5
2973 027574                EXIT    TST
(3) 027574 104032          EMT    C$EXIT
(3) 027576 000066          .WORD  L10033-.
2974
2975 027600 005737 002466 12$:  TST    T.CS          ;TEST IF ANY ERROR
2976 027604 100005                BPL    15$          ;NO-SKIP
2977 027606                ERRHRD 904...ERR6
(3) 027606 104443                TRAP  T$ERCODE
(5) 027610 001610                .WORD 904
(5) 027612 013342                .WORD ERR6
2978 027614                EXIT    TST
(3) 027614 104032          EMT    C$EXIT
(3) 027616 000046          .WORD  L10033-.
2979 027620 012703 011044 15$:  MOV    MMHSTA,R3      ;SET NAME MESSAGE PTR
2980 027624 004737 020530  JSR    PC,POSHSB    ;POSITION HEAD SELECT BIT FOR TEST
2981 027630 020537 002534  CMP    R5,DESHD    ;CHECK IF CORRECT HEAD SELECTED
2982 027634 001413                BEQ    20$          ;YES-SKIP
2983 027636 005737 002534  TST    DESHD      ;ELSE TEST IF 1 DESIRED
2984 027642 001405                BEQ    17$          ;NO-REPORT SB 0
2985 027644                ERRHRD 905...ERR3  ;ELSE REPORT SB 1
(3) 027644 104443                TRAP  T$ERCODE
(5) 027646 001611                .WORD 905
(5) 027650 013154                .WORD ERR3
2986 027652                EXIT    TST
(3) 027652 104032          EMT    C$EXIT
(3) 027654 000010          .WORD  L10033-.
2987 027656                ERRHRD 906...ERR2 17$:
(3) 027656 104443                TRAP  T$ERCODE
(5) 027660 001612                .WORD 906
(5) 027662 013106                .WORD ERR2
2988 027664                20$:
2989 027664                65$:
2990 027664                ENDTST
(3) 027664                L10033:
(3) 027664 104001          EMT    C$EXIT
  
```

```

2992          .SBTTL *TEST 10          SEEK SIGN SWITCH
2993 027666    BGNTST                    ;TEST 10
(3) 027666
2994 027666    012737 006405 002434    MOV      #T10ERR,ERHEAD ;SET ERROR HEADER          T10::
2995 027674    012701 002524          MOV      #NEWCYL,R1
2996 027700    005021          CLR      (R1)+          ;CLEAR NEW CYL
2997 027702    005021          CLR      (R1)+          ;CLEAR CURRENT CYLINDER
2998 027704    005021          CLR      (R1)+          ;CLEAR DIFFERENCE
2999 027706    052721 000001          BIS      #BIT0,(R1)+    ;SET FOR SIGN OF 1
3000 027712    004737 020530          JSR      PC,POSHSB      ;GET SELECTED HEAD
3001 027716    010521          MOV      R5,(R1)+      ;SET AS DESIRED HEAD
3002 027720
3003 027720          T104$:
(3) 027720          BGNSUB
(3) 027720    104002          EMT      CSBSUB          T10.1:
3004 027722    004737 016446          JSR      PC,TSTINT      ;INITIALIZE TEST
3005 027726    004737 016464          JSR      PC,GSTATR      ;GET STATUS
3006 027732    030166          60$
3007 027734    004737 017346          JSR      PC,SIMSEK      ;DO SEEK
3008 027740    030166          60$
3009 027742    012703 010711          MOV      #MDRDY,R3      ;SET NAME MESSAGE PTR
3010 027746    012704 011722          MOV      #CDRDY,R4      ;SET CONDITION MESSAGE PTR
3011 027752    004737 016514          JSR      PC,GSTAT      ;GET STATUS
3012 027756    030166          60$
3013 027760    032737 000001 002466    BIT      #DRDYMSK,T.CS  ;CHECK READY RESET
3014 027766    001405          BEQ      4$              ;YES-SKIP
3015 027770          ERRHRD 1001...ERR4          ;REPORT READY ERROR
(3) 027770    104443          TRAP    T$ERCODE
(5) 027772    001751          .WORD   1001
(5) 027774    013222          .WORD   ERR4
3016 027776          EXIT    SUB              ;EXIT SUBTEST
(3) 027776    104032          EMT      C$EXIT
(3) 030000    000166          .WORD   L10035-.
3017
3018
3019 030002    012701 000121          4$: MOV      #81,R1          ;SET WAIT COUNT
3020 030006    004737 016514          5$: JSR      PC,GSTAT      ;GET STATUS
3021 030012    030166          60$
3022 030014    012703 000005          MOV      #5,R3          ;SET EXPECTED STATE
3023 030020    020337 002502          CMP      R3,T.STAT      ;CHECK STATE IS 5
3024 030024    001405          BEQ      7$              ;YES-SKIP
3025 030026          ERRHRD 1002...ERR7          ;REPORT STATE ERROR
(3) 030026    104443          TRAP    T$ERCODE
(5) 030030    001752          .WORD   1002
(5) 030032    014220          .WORD   ERR7
3026 030034          EXIT    SUB              ;EXIT
(3) 030034    104032          EMT      C$EXIT
(3) 030036    000130          .WORD   L10035-.
3027 030040    012703 010711          7$: MOV      #MDRDY,R3      ;SET NAME MESSAGE PTR
3028 030044    032737 000001 002466    BIT      #DRDYMSK,T.CS  ;CHECK READY SET
3029 030052    001013          BNE     12$              ;YES-SKIP
3030 030054    005301          DEC      R1              ;DO WAIT COUNT
3031 030056    001404          BEQ      9$              ;SKIP IF 0
3032 030060          WAITUS #1
(3) 030060    012700 000001          MOV      #,R0
(3) 030064    104027          EMT      C$WTJ
  
```

```

3033 030066 000747          BR      5$
3034
3035 030070          9$:  ERRHRD 1003...ERR5 ;REPORT READY ERROR
(3) 030070 104443      TRAP   T$ERCODE
(5) 030072 001753      .WORD 1003
(5) 030074 013272      .WORD ERR5
3036 030076          EXIT   SUB ;EXIT
(3) 030076 104032      EMT   C$EXIT
(3) 030100 000066      .WORD L10035-.
3037 030102 005737 002466 12$:  TST   T.CS ;TEST IF ANY OTHER ERROR
3038 030106 100005      BPL   15$ ;NO-SKIP
3039 030110          ERRHRD 1004...ERR6 ;REPORT ALL ERRORS
(3) 030110 104443      TRAP   T$ERCODE
(5) 030112 001754      .WORD 1004
(5) 030114 013342      .WORD ERR6
3040 030116          EXIT   SUB ;EXIT
(3) 030116 104032      EMT   C$EXIT
(3) 030120 000046      .WORD L10035-.
3041
3042 030122 012703 011044 15$:  MOV   #MHSTA,R3 ;SET NAME MESSAGE PTR
3043 030126 004737 020530      JSR   FC,POSHSB ;GET SELECTED HEAD BIT
3044 030132 020537 002534      CMP   R5,DESHD ;CHECK IF CORRECT
3045 030136 001413      BEQ   20$ ;YES - SKIP
3046 030140 005737 002534      TST   DESHD ;WAS IT SET
3047 030144 001405      BEQ   17$ ;NO-SKIP
3048 030146          ERRHRD 1005...ERR3 ;REPORT SB 1
(3) 030146 104443      TRAP   T$ERCODE
(5) 030150 001755      .WORD 1005
(5) 030152 013154      .WORD ERR3
3049 030154          EXIT   SUB
(3) 030154 104032      EMT   C$EXIT
(3) 030156 000010      .WORD L10035-.
3050 030160          17$:  ERRHRD 1006...ERR2 ;REPORT SB 0
(3) 030160 104443      TRAP   T$ERCODE
(5) 030162 001756      .WORD 1006
(5) 030164 013106      .WORD ERR2
3051
3052 030166          20$:
3053 030166          60$:
3054 030166          ENDSUB
(3) 030166          L10035:
(3) 030166 104003      EMT   C$ESUB
3055 030170 005737 002532      TST   DESSGN ;CHECK IF BOTH SIGN USED
3056 030174 001404      BEQ   25$ ;YES-SKIP
3057 030176 005037 002532      CLR   DESSGN ;SET OR SIGN OF 0
3058 030202 000137 027720      JMP   T104$ ;DO TEST AGAIN
3059 030206          25$:
3060 030206          ENDTST
(3) 030206          L10034:
(3) 030206 104001      EMT   C$ETST
  
```

```

3062
3063
3064          .SBTTL *TEST 11          HEAD ALIGNMENT SUPPORT
3065          BGNTST                    ;TEST 11
(3) 030210
3066 030210 032737 000010 014656      BIT    #HDALIGN,MISWIW ;CHECK IF RUN HEAD ALIGNMENT
3067 030216 001411                      BEQ    1$           ;NO-EXIT
3068 030220 005737 003062              TST    PASNUM      ;TEST IF PASS 0
3069 030224 001006                      BNE    1$           ;NO-EXIT
3070 030226 023737 002454 002432      CMP    RLDRV,HADONE ;TEST IF HEAD ALIGN DONE THIS DRIVE
3071 030234 001004                      BNE    2$           ;NO - SKIP
3072 030236 000137 030540              JMP    T115$       ;GO CHECK WRITE LOCK
3073 030242          1$:              EXIT   TST
(3) 030242 104032                      EMT    C$EXIT
(3) 030244 000374                      .WORD  L10036-
3074 030246 013737 002454 002432 2$:  MOV    RLDRV,HADONE ;SET HEAD ALIGN DONE FLAG
3075 030254          PRINTF #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
(11) 030254 005046                      CLR    -(SP)
(11) 030256 153716 002455              BISB  RLDRV+1,(SP)
(10) 030262 012746 005633              MOV    #DRVNAM,-(SP)
(9) 030266 013746 002450              MOV    RLBAS,-(SP)
(8) 030272 012746 005622              MOV    #BASADD,-(SP)
(7) 030276 012746 012143              MOV    #FMT5,-(SP)
(6) 030302 012746 000005              MOV    #5,-(SP)
(3) 030306 010600                      MOV    SP,R0
(4) 030310 104017                      EMT    C$PNTF
(4) 030312 062706 000014              ADD    #14,SP
3076 030316          PRINTF #FMT9,#HAMES1 ;TYPE INSTRUCTIONS
(8) 030316 012746 007567              MOV    #HAMES1,-(SP)
(7) 030322 012746 012327              MOV    #FMT9,-(SP)
(6) 030326 012746 000002              MOV    #2,-(SP)
(3) 030332 010600                      MOV    SP,R0
(4) 030334 104017                      EMT    C$PNTF
(4) 030336 062706 000006              ADD    #6,SP
3077 030342          PRINTF #FMT9,#HAMES2
(8) 030342 012746 007652              MOV    #HAMES2,-(SP)
(7) 030346 012746 012327              MOV    #FMT9,-(SP)
(6) 030352 012746 000002              MOV    #2,-(SP)
(3) 030356 010600                      MOV    SP,R0
(4) 030360 104017                      EMT    C$PNTF
(4) 030362 062706 000006              ADD    #6,SP
3078 030366          BGNSUB
(3) 030366
(3) 030366 104002
3079 030370 004737 016446          3$:  EMT    C$BSUB ;INITIALIZE TEST
3080 030374 005037 002430          JSR    PC,TSTINT ;INITIALIZE TEST
3081 030400 013737 002454 002456      CLR    DONE       ;CLEAR DONE
3082 030406 052737 000104 002456      MOV    RLDRV,L.CS ;SET UP FOR GET STATUS
3083 030414 012737 000013 002462      BIS    #GTSTAT,L.CS
3084 030422 013762 002462 000004      MOV    #GETSTAT!DRSET,L.DA
3085 030430 013762 002456 000000      MOV    L.DA,RLDA(R?)
3086 030436          WAITMS #50. ;DO GET STATUS
(3) 030436 012700 000062          WAITMS #50. ;WAIT FOR INTERRUPT
(5) 030442 104026
3087 030444 005737 002430          MOV    #50.,R0
3088 030450 001747          EMT    C$WTM
          TST    DONE ;CHECK IF DONE
          BEQ    -3 ;NO-GO CLR CONTROLLER
  
```



```

3089
3090
3091 030452 012737 000021 002462 10$: MOV #HDSEL!MBSETO,L.DA;LOAD FOR HEAD 1
3092 030460 032737 020000 002474 BIT #WLSTAT,T.MP ;CHECK IF WRITE LOCK SET
3093 030466 C01003 BNE 12$ ;YES-SKIP
3094 030470 042737 000020 002462 BIC #HDSEL,L.DA ;ELSE CLEAR TO HEAD 0
3095 030476 013737 002454 002456 12$: MOV RLDRV,L.CS ;LOAD IN DRIVE NUMBER
3096 030504 052737 000106 002456 BIS #SEEK,L.CS ;SET FOR SEEK
3097 030512 013762 002462 000004 MOV L.DA,RLDA(R2) ;LOAD & EXECUTE SEEK
3098 030520 013762 002456 000000 MOV L.CS,RLCSR(R2)
3099 030526 WAITMS #30. ;WAIT FOR INTERRUPT
(3) 030526 012700 000036 MOV #30.,R0
(3) 030532 104026 EMT C$WTM
3100 030534 000715 BR 3$ ;LOOP
3101 030536 59$:
3102 030536 ENDSUB
(3) 030536 L10037:
(3) 030536 104003 EMT C$ESUB
3103 030540 T115$:
3104 030540 BGNSUB
(3) 030540 104002 EMT C$BSUB T11.2:
3105 030542 004737 016446 JSR PC,TSTINT ;INITIALIZE TEST
3106 030546 004737 016464 JSR PC,GSTATR ;CLEAR DRIVE
3107 030552 030636 60$:
3108 030554 032737 020000 002474 BIT #WLSTAT,T.MP ;CHECK WRITE LOCK RESET
3109 030562 001425 BEQ 19$ ;YES-SKIP
3110 030564 18$: PRINTF #FMT9,#OPR12 ;REQUEST WRITE LOCK RESET
(8) 030564 012746 010377 MOV #OPR12,-(SP)
(7) 030570 012746 012327 MOV #FMT9,-(SP)
(6) 030574 012746 000002 MOV #2,-(SP)
(3) 030600 010600 MOV SP,R0
(4) 030602 104017 EMT C$PNTF
(4) 030604 062706 000006 ADD #6,SP
3111 030610 005037 004066 CLR OBUFF ;CLEAR FOR RESPONSE
3112 030614 GMANIL OPR002,OBUFF,1,NO ;GET RESPONSE
(3) 030614 104043 EMT C$GMAN
(3) 030616 000404 BR 10000$
(4) 030620 004066 .WORD OBUFF
(5) 030622 000120 .WORD T$CODE
(5) 030624 007724 .WORD OPR002
(5) 030626 000001 .WORD 1
(3) 030630 10000$:
3113 030630 005737 004066 TST OBUFF ;WAS ANSWER YES
3114 030634 001753 BEQ 18$ ;NO-REPEAT REQUEST
3115 030636 19$:
3116 030636 60$:
3117 030636 ENDSUB
(3) 030636 L10040:
(3) 030636 104003 EMT C$ESUB
3118 030640 20$:
3119 030640 ENDTST
(3) 030640 L10036:
(3) 030640 104001 EMT C$ETST

```

```

3121
3122
3123          .SBTTL *TEST 12          HEAD SWITCHING
3124 030642          BGNTST          ;TEST 12
(3) 030642
3125 030642 012737 006425 002434          MCV #T12ERR,ERHEAD ;SET ERROR HEADER
3126 030650 012701 002524          MOV #NEWCYL,R1 ;GET POINTER TO DESIRED LOCATION
3127 030654 005021          CLR (R1)+ ;CLEAR NEW CYLINDER
3128 030656 005021          CLR (R1)+ ;CLEAR CURRENT CYL.
3129 030660 005021          CLR (R1)+ ;CLEAR DIFFERENCE
3130 030662 005021          CLR (R1)+ ;CLEAR SIGN
3131 030664 012721 000001          MOV #1,(R1)+ ;SET FOR HEAD 1
3132 030670          T124$:
3133 030670          BGNSUB
(3) 030670
(3) 030670 104002          EMT CSBSUB ;INITIALIZE TEST
3134 030672 004737 016446          JSR PC,TSTINT ;INITIALIZE TEST
3135 030676 004737 016464          JSR PC,GSTATR ;GET STATUS WITH RESET
3136 030702 031136          60$
3137 030704 004737 017346          JSR PC,SIMSEK ;DO SEEK
3138 030710 031136          60$
3139 030712 012703 010711          MOV #MDRDY,R3 ;SET NAME MESSAGE PTR
3140 030716 012704 011722          MOV #CDRDY,R4 ;SET CONDITION POINTER
3141 030722 004737 016514          JSR PC,GSTAT ;GET STATUS
3142 030726 031136          60$
3143 030730 032737 000001 002466          BIT #DRDYMSK,T.CS ;CHECK IF READY
3144 030736 001405          BEQ 5$ ;NO-SKIP
3145 030740          ERRHRD 1201...ERR4 ;REPORT READY ERROR
(3) 030740 104443          TRAP T$ERCODE
(5) 030742 002261          .WORD 1201
(5) 030744 013222          .WORD ERR4
3146 030746          EXIT SUB ;EXIT
(3) 030746 104032          EMT C$EXIT
(3) 030750 000166          .WORD L10042-.
3147
3148 030752 012701 000121          5$: MOV #81,R1 ;SET WAIT COUNT
3149 030756 004737 016514          6$: JSR PC,GSTAT ;GET STATUS
3150 030762 031136          60$
3151 030764 012703 000005          MOV #5,R3 ;SET EXPECTED STATE VALUE
3152 030770 020337 002502          CMP R3,T.STAT ;CHECK IF STATE IS 5
3153 030774 001405          BEQ 7$ ;YES-SKIP
3154 030776          ERRHRD 1202...ERR7 ;REPORT STATE ERROR
(3) 030776 104443          TRAP T$ERCODE
(5) 031000 002262          .WORD 1202
(5) 031002 014220          .WORD ERR7
3155 031004          EXIT SUB
(3) 031004 104032          EMT C$EXIT
(3) 031006 000130          .WORD L10042-.
3156
3157 031010 012703 010711          7$: MOV #MDRDY,R3 ;SET NAME MESSAGE PTR
3158 031014 032737 000001 002466          BIT #DRDYMSK,T.CS ;CHECK DRIVE READY
3159 031022 001013          BNE 12$ ;YES-SKIP
3160 031024 005301          DEC R1 ;DEC WAIT COUNT
3161 031026 001404          BEQ 9$ ;SKIP IF 0
3162 031030          WAITUS #
(3) 031030 012700 000001          MOV #1,R0
    
```

```

(3) 031034 104027          EMT      C$WTU
3163 031036 000747          BR      6$
3164
3165 031040          9$:  ERRHRD  1203...ERR5  ;REPORT READY ERROR
(3) 031040 104443          TRAP    T$ERCODE
(5) 031042 002263          .WORD  1203
(5) 031044 013272          .WORD  ERR5
3166 031046          EXIT    SUB          ;EXIT
(3) 031046 104032          EMT     C$EXIT
(3) 031050 000066          .WORD  L10042-.
3167
3168 031052 005737 002466    12$:  TST      T.CS          ;TEST IF ANY ERROR
3169 031056 100005          BPL     15$          ;NO-SKIP
3170 031060          ERRHRD  1204...ERR6  ;REPORT ALL ERRORS
(3) 031060 104443          TRAP    T$ERCODE
(5) 031062 002264          .WORD  1204
(5) 031064 013342          .WORD  ERR6
3171 031066          EXIT    SUB
(3) 031066 104032          EMT     C$EXIT
(3) 031070 000046          .WORD  L10042-.
3172 031072 012703 011044    15$:  MOV      #MHSTA,R3  ;SET NAME MESSAGE PTR
3173 031076 004737 020530          JSR     PC,POSHSB  ;POSITION HEAD SELECT BIT
3174 031102 023705 002534          CMP     DESHD,R5   ;CHECK IF CORRECT HEAD SELECTED
3175 031106 001413          BEQ     20$          ;YES-SKIP
3176 031110 005737 002534          TST     DESHD      ;WAS HEAD 0 SELECTED
3177 031114 001405          BEQ     17$          ;YES-SKIP
3178 031116          ERRHRD  1205...ERR3  ;REPORT HEAD SB 1
(3) 031116 104443          TRAP    T$ERCODE
(5) 031120 002265          .WORD  1205
(5) 031122 013154          .WORD  ERR3
3179 031124          EXIT    SUB          ;EXIT
(3) 031124 104032          EMT     C$EXIT
(3) 031126 000010          .WORD  L10042-.
3180 031130          17$:  ERRHRD  1206...ERR2  ;ELSE REPORT HEAD SB 0
(3) 031130 104443          TRAP    T$ERCODE
(5) 031132 002266          .WORD  1206
(5) 031134 013106          .WORD  ERR2
3181
3182 031136          20$:
3183 031136          60$:
3184 031136          ENDSUB
(3) 031136 104003          L10042:
(3) 031136 005737 002534          EMT     C$ESUB
3185 031140 001404          TST     DESHD      ;CHECK IF HD 0 WAS DONE
3186 031144 005037 002534          BEQ     25$          ;YES--SKIP
3187 031152 000137 030670          CLR     DESHD      ;ELSE SET TO HEAD 0
3188 031156          JMP     T124$      ;REDO TEST
3189 031156          25$:
3190 031156          ENDTST
(3) 031156 104001          L10041:
(3) 031156          EMT     C$ETST
  
```

```

3192
3193
3194
3195      031160      .SBTTL *TEST 13      READ HEADER (PART 1)
          (3) 031160      BGNSTST      ;TEST 13
3196      031160      012737 006437 002434      MOV      #T13ERR,ERHEAD      ;SET ERROR HEADER
3197      031166      012701 002524      MOV      #NEWCYL,R1      ;GET ADDRESS OF DESIRED LOCATIONS
3198      031172      005021      CLR      (R1)+      ;CLEAR NEW CYL
3199      031174      005021      CLR      (R1)+      ;CLEAR CURRENT CYL
3200      031176      005021      CLR      (R1)+      ;CLEAR DIFF
3201      031200      005021      CLR      (R1)+      ;CLEAR SIGN
3202      031202      005021      CLR      (R1)+      ;CLEAR HEAD
3203      031204      T134$:
3204      031204      BGNSTST
          (3) 031204
          (3) 031204      104002      EMT      CSBSUB      T13.1:
3205      031206      004737 016446      JSR      PC,TSTINT      ;INITIALIZE TEST
3206      031212      004737 016464      JSR      PC,GSTATR      ;GET STATUS W/RESET
3207      031216      031304      60$
3208      031220      004737 017346      JSR      PC,SIMSEK      ;DO SEEK
3209      031224      031304      60$
3210      031226      012701 000121      MOV      #81,,R1      ;SET WAIT COUNT
3211      031232      004737 020560      JSR      PC,RDYWAIT      ;WAIT FOR READY
3212      031236      031304      60$
3213
3214      031240      004737 020072      10$: JSR      PC,XRDHDC      ;DO READ HEADER
3215      031244      031304      60$
3216      031246      012703 011044      MOV      #MHSTA,R3      ;SET NAME MESSAGE PTR
3217      031252      004737 020522      JSR      PC,POSHW1      ;POSITION HS BIT IN HD WRD 1
3218      031256      020537 002534      CMP      R5,DESHD      ;CHECK IF HEAD CORRECT
3219      031262      001410      BEQ      15$      ;YES-SKIP
3220      031264      ERRHRD 1301,,,ERR3      ;REPORT SB 1
          (3) 031264      104443      TRAP      T$ERRCODE
          (5) 031266      002425      .WORD    1301
          (5) 031270      013154      .WORD    ERR3
3221      031272      EXIT      SUB
          (3) 031272      104032      EMT      C$EXIT
          (3) 031274      000010      .WORD    L10044-
3222      031276      17$: ERRHRD 1302,,,ERR2      ;REPORT SB 0
          (3) 031276      104443      TRAP      T$ERRCODE
          (5) 031300      002426      .WORD    1302
          (5) 031302      013106      .WORD    ERR2
3223
3224      031304      15$:
3225      031304      60$:
3226      031304      ENDSUB
          (3) 031304      L10044:
          (3) 031304      104003      EMT      C$ESUB
3227      031306      005737 002534      TST      DE$HD      ;TEST IF HEAD 1 DONE
3228      031312      001007      BNE      20$      ;YES-SKIP
3229      031314      012737 000001 002534      MOV      #1,DESHD      ;ELSE SET TO HEAD 1
3230      031322      013737 002474 002540      MOV      HDWRD1,TEMPO      ;STORE HDR WORD 1
3231      031330      000725      BR      T134$      ;DO TEST AGAIN
3232      031332      042737 100177 002540      20$: BIC      #^CHDCYL,TEMPO      ;CLEAR ALL BUT CYLINDER IN 1ST HEADER
3233      031340      042737 100177 002474      BIC      #^CHDCYL,HDWRD1      ;CLEAR ALL BY CYL IN 2ND HEADER
3234      031346      023737 002540 002474      CMP      TEMPO,HDWRD1      ;COMPARE IF EQUAL
  
```

3235	031354	001405		BEO	22\$:YES-SKIP
3236	031356	012703	007503	MOV	#CYLPER,R3	:SET NAME MESSAGE PTR
3237	031362			ERRHRD	1306...ERR1	:REPORT HEAD ALIGNMENT PROBLEM
(3)	031362	104443		TRAP	T\$ERCODE	
(5)	031364	002432		.WORD	1306	
(5)	031366	013040		.WORD	ERR1	
3238	031370					
3239	031370					
(3)	031370					
(3)	031370	104001				

22\$:
FNDTST
L10043:
EMT C\$ETST

```

3241          .SBTTL *TEST 14      READ HEADER (PART 2)
3242 031372    BGNTST              ;TEST 14
(3) 031372
3243 031372    012737 006453 002434  MOV #T14ERR,ERHEAD ;SET ERROR HEADER      T14::
3244 031400    012701 002526          MOV #CURCYL,R1      ;GET ADDRESS OF DESIRED VALUE
3245 031404    005021          CLR (R1)+           ;CLEAR CURRENT CYL
3246 031406    005021          CLR (R1)+           ;CLEAR DESIRED DIFF
3247 031410    005021          CLR (R1)+           ;CLEAR SIGN
3248 031412    005021          CLR (R1)+           ;CLEAR DESIRED HEAD
3249 031414
3250 031414    T153$:           BGNSUB
(3) 031414
(3) 031414    104002          EMT CSBSUB          T14.1:
3251 031416    004737 016446          JSR PC,TSTINT      ;INITIALIZE TEST
3252 031422    004737 016464          JSR PC,GSTATR     ;CLEAR DRIVE
3253 031426    031622          60$
3254 031430    004737 017346          JSR PC,SIMSEK     ;DO SEEK
3255 031434    031622          60$
3256 031436    012701 000310          MOV #200,R1       ;SET WAIT COUNT FOR 20 MS
3257 031442    004737 020560          JSR PC,RDYWAIT    ;WAIT FOR READY
3258 031446    031622          60$
3259 031450    004737 021154          JSR PC,RDALHD     ;DO READ HEADER ALL HEADERS
3260 031454    031622          60$
3261 031456    005037 002436          CLR MORECE        ;CLEAR MORE COMPARE ERRORS FOR REPORT
3262 031462    052737 000002 002426  BIS #HDCMP,OPFLAG ;SET HDR COMPARE FLAG
3263 031470    005003          CLR R3            ;CLEAR FOR HDR COUNT
3264 031472    012704 003466          MOV #IBUFF,R4     ;GET POINTER FOR HDR TO BE CHECKED
3265 031476    012705 002540          MOV #TEMPO,R5     ;GET POINTER TO TEST AREA
3266 031502    012701 000050          MOV #40,R1        ;SET HDR COUNT
3267 031506    011415          MOV (R4),(R5)     ;GET FIRST HEADER WORD
3268 031510    042715 100100          BIC #BIT15:HDHSEL,(R5);CLEAR BIT 15 AND HEAD SEL BIT
3269 031514    005737 002534          TST DESHD         ;TEST IF HD 0 DESIRED
3270 031520    001404          BEQ 10$           ;YES-SKIP
3271 031522    052715 000100          BIS #HDHSEL,(R5) ;ELSE SET HEAD BIT
3272 031526    005065 000002          CLR 2(R5)         ;CLEAR 2ND WORD OF TEST AREA
3273 031532    021524          10$: CMP (R5),(R4)+    ;COMPARE HEADER WORD
3274 031534    001405          BEQ 13$           ;SKIP IF OK
3275 031536    005744          TST -(R4)         ;ELSE POSITION R4 TO BAD WORD
3276 031540          ERRHRD 1501...ERR10 ;REPORT ERROR
(3) 031540    104443          TRAP T$ERRCODE
(5) 031542    002735          .WORD 1501
(5) 031544    014430          .WORD ERR10
3277 031546    005724          TST (R4)+         ;BUMP R4 TO NEXT WORD
3278 031550    005203          14$: INC R3           ;BUMP WORD COUNT
3279 031552    005724          TST (R4)+         ;TEST 2ND WORD IS 0
3280 031554    001405          BEQ 15$           ;YES - SKIP
3281 031556    022544          CMP (R5)+,-(R4)  ;POSITION PTRS FOR REPORT
3282 031560          ERRHRD 1501...ERR10 ;REPORT ERROR
(3) 031560    104443          TRAP T$ERRCODE
(5) 031562    002735          .WORD 1501
(5) 031564    014430          .WORD ERR10
3283 031566    024524          15$: CMP -(R5),(R4)+  ;REPOSITION POINTER
3284 031570    005724          TST (R4)+         ;POSITION R4 PAST ECC WORD
3285 031572    005203          INC R3            ;BUMP WORD COUNT
3286 031574    005215          INC (R5)          ;BUMP SECTOR COUNT
3287 031576    011500          MOV (R5),R0       ;CHECK IF SECTOR IS PAST LAST SECTOR
  
```

3288	031600	042700	177700		BIC	#^CHDSEC,RO	
3289	031604	022700	000050		CMP	#40.,RO	
3290	031610	001002			BNE	17\$:NO-SKIP
3291	031612	042715	000077		BIC	#HDSEC,(R5)	:ELSE CLEAR SECTOR TO 0
3292	031616	005301		17\$:	DEC	R1	:DEC HDR COUNT
3293	031620	001344			BNE	10\$:YES-SKIP
3294							
3295	031622			60\$:			
3296	031622			ENDJOB			
(3)	031622			L10046:			
(3)	031622	104003			EMT	C\$ESUB	
3297	031624	005737	002534		TST	DESHD	:CHECK IF HD 1 TESTED
3298	031630	001005			BNE	20\$:YES-SKIP
3299	031632	012737	000001 002534		MOV	#1,DESHD	:ELSE SET TO HEAD 1
3300	031640	000 7	031414		JMP	T153\$:REDO TEST
3301	031644			20\$:			
3302	031644			ENDTST			
(3)	031644			L10045:			
(7)	031644	104001			EMT	C\$FTST	
3303	031646			ENDMOD			

3306	031646				BGNMOD	HRDPRM	
3307	031646				BGNHRD		
(3)	031646	000025				.WORD	L10047-L\$HARD/2
3308	031650				GPRML	CNTYPE,CNT,1,YES	
(4)	031650	004130				.WORD	T\$CODE
(4)	031652	031764				.WORD	CNTYPE
(4)	031654	000001				.WORD	1
3309	031656				GPRMA	CSRMSG,CSR,0,160000,177776,YES	
(4)	031656	000031				.WORD	T\$CODE
(4)	031660	031722				.WORD	CSRMSG
(4)	031662	160000				.WORD	T\$LLOLIM
(4)	031664	177776				.WORD	T\$HILIM
3310	031666				GPRMA	VECMMSG,VECT,0,0,776,YES	
(4)	031666	001031				.WORD	T\$CODE
(4)	031670	031736				.WORD	VECMMSG
(4)	031672	000000				.WORD	T\$LLOLIM
(4)	031674	000776				.WORD	T\$HILIM
3311	031676				GPRMD	BRMSG,PRIOR,0,340,0,7,YES	
(4)	031676	002032				.WORD	T\$CODE
(4)	031700	031745				.WORD	BRMSG
(4)	031702	000340				.WORD	340
(4)	031704	000000				.WORD	T\$LLOLIM
(4)	031706	000007				.WORD	T\$HILIM
3312	031710				GPRMD	DRMSG,DRSB,0,3400,0,7,YES	
(4)	031710	003032				.WORD	T\$CODE
(4)	031712	031756				.WORD	DRMSG
(4)	031714	003400				.WORD	3400
(4)	031716	000000				.WORD	T\$LLOLIM
(4)	031720	000007				.WORD	T\$HILIM
3313							
3314	031722				ENDHRD		
(2)						.EVEN	
(3)	031722				L10047:		
3315							
3316	031722	052502	020123	042101	CSRMSG:	.ASCIZ	/BUS ADDRESS/
	031730	051104	051505	000123			
3317	031736	042526	052103	051117	VECMMSG:	.ASCIZ	/VECTOR/
	031744	000					
3318	031745	102	020122	042514	BRMSG:	.ASCIZ	/BR LEVEL/
	031752	042526	000114				
3319	031756	051104	053111	000105	DRMSG:	.ASCIZ	/DRIVE/
3320	031764	046122	030461	000	CNTYPE:	.ASCIZ	/RL11/
3321	031771				ENDMOD		
3322		031772				.EVEN	
3323							
3324	031772				BGNMOD	SF*PRM	
3325	031772				BGNSFT		
(3)	031772	000021				.WORD	L10050-L\$SOFT/2
3326							
3332	031774				GPRML	SELO,MISWI,4,YES	
(4)	031774	000130				.WORD	T\$CODE
(4)	031776	032036				.WORD	SELO
(4)	032000	000004				.WORD	4
3333	032002				GPRML	ALGNQ,MISWI,10,YES	
(4)	032002	000130				.WORD	T\$CODE
(4)	032004	032071				.WORD	ALGNQ


```

(4) 032006 000010
3335 032010          GPRML  .WORD 10
      (4) 032010 000130          MANQ,MISWI,100000,YES
      (4) 032012 032130          .WORD T$CODE
      (4) 032014 100000          .WORD MANQ
3336          .WORD 100000
3348 032016          3$:  GPRMD  ERLIMQ,ERLIM,D,377,0,377,YES
      (4) 032016 004052          .WORD T$CODE
      (4) 032020 032172          .WORD ERLIMQ
      (4) 032022 000377          .WORD 377
      (4) 032024 000000          .WORD T$LOLIM
      (4) 032026 000377          .WORD T$HILIM
3352 032030          GPRML  AUTOQ,MISWI,20,YES
      (4) 032030 000130          .WORD T$CODE
      (4) 032032 032216          .WORD AUTOQ
      (4) 032034 000020          .WORD 20
3353 032036          ENDSFT
      (2)          .EVEN
      (3) 032036          L10050:
3354
3360 032036 054105 041505 052125 SELQ:  .ASCIZ /EXECUTE DRIVE SELECT TESTS/
      032044 020105 051104 053111
      032052 020105 042523 042514
      032060 052103 052040 051505
      032066 051524 000
3361 032071 105 042530 052503 ALGNQ:  .ASCIZ /EXECUTE HEAD ALIGNMENT SUPPORT/
      032076 042524 044040 040505
      032104 020104 046101 043511
      032112 046516 047105 020124
      032120 052523 050120 051117
      032126 000124
3363 032130 054105 041505 052125 MANQ:  .ASCIZ /EXECUTE MANUAL INTERVENTION TESTS/
      032136 020105 040515 052516
      032144 046101 044440 052116
      032152 051105 042526 052116
      032160 047511 020116 042524
      032166 052123 000123
3371 032172 050123 041505 043111 ERLIMQ: .ASCIZ /SPECIFY ERROR LIMIT/
      032200 020131 051105 047522
      032206 020122 044514 044515
      032214 000124
3372 032216 051104 050117 042040 AUTOQ:  .ASCIZ /DROP DRIVE IF NO RESPONSE/
      032224 044522 042526 044440
      032232 020106 047516 051040
      032240 051505 047520 051516
      032246 000105

```

```

3376
3377 032250
3378
3380
3381 032514
3382
3383
3384
3385
3386

```

```

      .EVEN
      ENDMOD

```

```

      . 32514

```

```

; AREA RESERVED AS PATCH AREA FOR DIAGNOSTICS.
; -32514 WAS SELECTED AS 'LASTAD' TO PROVIDE APT TO LSI-11 COMPATIBILITY.
; BIT 7 OF 'LASTAD' MUST BE CLEARED TO ACHIEVE A VALID MAILBOX ADDRESS
; WHEN RUNNING ON THE LSI-11 UNDER APT.

```

3387
3387
3399
3400 032514
(2)
(3) 032514

LASTAD
LAST:: .EVEN

```
3402  
14273 063310 000000 .SBTTL DIAGNOSTIC SUPERVISOR -- LOW CORE SET UP  
14274 063312 000000 .WORD 0 ;SPACE FOR USER POOL POINTER  
14275 063314 000000 .WORD 0 ;SIZE  
14276 063316 000000 .WORD 0 ;CHECKSUM (NOT CURRENTLY USED)  
14277 063322 .END.SUPV-.+2 ;SIZE OF H.W. PTAB. ALLOCATION  
14278 00020C .END 200
```

ABOFLA 033040 G	BIT6 = 000100 G	CONTCL 061672 G	C\$LOOP= 000100	DRVNAM 005633
ABOPAS 032756 G	BIT7 = 000200 G	CONTIN 015122	C\$MANI= 000051	DRVNAV 005640
ABO.FM 035320	BIT8 = 000400 G	COSTAT= 000040	C\$MSG = 000023	DSESTA= 000400
AFMID 002632	BIT9 = 001000 G	COUNT 002656	C\$PNTB= 000014	DSMSK = 001400
AFMIDU 002634	BLD.MW 040202	CRDYMS= 000200	C\$PNTF= 000017	DSPCOD 014672 G
AFSI 032546 G	BLOCK 055614	CRLF 052172	C\$PNTS= 000016	DUNIT. 032762 G
ALGNO 032071	BRMSG 031745	CSNAM 005733	C\$PNTX= 000015	DVC.FT 046042
ALLCYL= 000001	BSFLAG 002442	CSR = 000000	C\$POIN= 000040	DSAAG 046746
ALLOC 053460	BSFVAL 003074	CSRMSG 031722	C\$QIO = 000377	DSAAG 046764
ALI SEC= 000002	BSNSTR 010610	CURCYL 002526	C\$RDBU= 000007	DSAAI 051532
ANYERR= 100000	BYPNM 010521	CURR.S 032522 G	C\$REFG= 000050	DSAAJ 051536
APT.ER 034450	B\$AAB 041604	CURR.T 032524 G	C\$REQT= 000045	DSAAK 051554
ARMID 002636	B\$AAF 041516	CYLPER 007503	C\$RESE= 000033	DSAAL 051572
ARMIDU 002640	CAFDT 011774	CYLTBL 002352	C\$REVI= 000002	DSAAM 051602
ASSEMB= 000010	CALLPC= 000022	CYLUP = 000004	C\$RPT = 000025	EF.CON= 000036 G
AUTOQ 032216	CALLPS= 000024	CYLWD 010506	C\$SEFG= 000047	EF.NEW= 000035 G
AUTOSZ= 000020	CALLSP= 000026	C\$AAD 045062	C\$SPRI= 000041	EF.PWR= 000034 G
ASA AV 037316	CALLTC= 000030	C\$AAE 045074	C\$SVEC= 000037	EF.RES= 000037 G
ASA AW 037332	CAL.CL 060202	C\$AAK 046072	C\$TPRI= 000013	EF.STA= 000040 G
ASA AX 037344	CAL.TI 060240 G	C\$AAL 046236	C\$UNBU= 000031	EF01 = 000001 G
ASA AY 037352	CAMSK = 077600	C\$ABRT= 000021	C\$WTM = 000026	EF02 = 000002 G
ASA AZ 037366	CCYLUP 011763	C\$ADR = 000020	C\$WTU = 000027	EF03 = 000003 G
ASABA 037376	CDRDY 011722	C\$AJ = 000054	C1OMS 011742	EF04 = 000004 G
BADADD= 004000	CHKLUP 041620	C\$DRK = 000022	C5SEC 012006	EF05 = 000005 G
BAMSK = 000060	CHKSTR 054022	C\$BSEG= 000004	C5OMS 011755	EF06 = 000006 G
BANAM 005740	CHKTTY 052110	C\$BSUB= 000002	DANAM 005745	EF07 = 000007 G
BASADD 005622	CHK.MA 037760	C\$BUFF= 000030	DATA CM= 000001	EF08 = 000010 G
BELL 011643	CHK.PC 045110	C\$CEFG= 000046	DCKERR= 004000	EF09 = 000011 G
BGN.SU= 032514	CHK.SW 034150	C\$CLEA= 000012	DCLIM = 000012	EF10 = 000012 G
BHSTAT= 000010	CHRCNT 053342	C\$CLP1= 000006	DCLIMW 014670	EF11 = 000013 G
BINMSG 051770	CH.FLA 037466	C\$CVEC= 000036	DECMG 052004	EF12 = 000014 G
BIT0 = 000001 G	CH.PAS 037504	C\$DCLN= 000044	DESDIF 002530	EF13 = 000015 G
BIT00 = 000001 G	CKDATA= 000102	C\$DODU= 000053	DESHD 002534	EF14 = 000016 G
BIT01 = 000002 G	CKERM 016122	C\$DRPT= 000024	DESSEC 002536	EF15 = 000017 G
BIT02 = 000004 G	CLEAR. 041102	C\$DU = 000055	DESSGN 002532	EF16 = 000020 G
BIT03 = 000010 G	CLKACC 032754 G	C\$EDIT= 000002	DEV.CO 032526 G	EMT.TR 033044 G
BIT04 = 000020 G	CLKBFR 060204	C\$ERDF= 000002	DIAGMC= 000000	END.OF 041070
BIT05 = 000040 G	CLKCNT 032752 G	C\$ERHR= 000003	DIAG.T 033046 G	END.SU= 063322
BIT06 = 000100 G	CLKJUM 060610 G	C\$ERSF= 000001	DIFAUG 002520	ENVIRO 032566 G
BIT07 = 000200 G	CLKRES 061612 G	C\$ERSO= 000004	DIFWD 010462	EOP.CH 061770 G
BIT08 = 000400 G	CLKSER 061746 G	C\$ESCA= 000010	DIRBIT= 000004	EOP.FM 035334
BIT09 001000 G	CLKSON 033012 G	C\$ESEG= 000005	DIRMSK= 077600	EOP.IN 037500
BIT1 000002 G	CLK.SE 037562	C\$ESUB= 000003	DLTERR= 010000	ERHEAD 002434
BIT10 = 002000 G	CLNCOD 015732 G	C\$ETST= 000001	DONE 002430	ERLIM = 000010
BIT11 = 004000 G	CLRPAR 022746	C\$EXIT= 000032	DPDVD 062456 G	ERLIMQ 032172
BIT12 = 010000 G	CLR.MA 040036	C\$GMAN= 000043	DPMUL 062344 G	ERLIMW 014666
BIT13 = 020000 G	CNT = 000010	C\$GPHR= 000042	DRDYMS= 000001	ERRCNT 002662
BIT14 = 040000 G	CNTYPE 031764	C\$GPRI= 000040	DRMSG 031756	ERRFOR 046314
BIT15 = 100000 G	CNVT 056260	C\$GTIM= 000052	DRSB = 000006	ERRHAN 045114
BIT2 = 000004 G	COMMAN 032564 G	C\$INIT= 000011	DRSET = 000004	ERRPOI 002660
BIT3 000010 G	COMMTA 056074	C\$INLP= 000020	DRSET = 000010	ERRSWI 002440
BIT4 = 000020 G	COMPOP 007777	C\$KWOF= 000035	DRVCNT 002516	ERRVEC 002652
BIT5 000040 G	CONHNG= 000004	C\$KWON= 000034	DRVERR= 040000	ERR.HR 046052

ERR.NU	032516	G	FMT3	012127	G\$LOLI=	000001	HPTCOD	014640	G	LABOUT	007315	
ERR.SF	046056		FMT4	012132	G\$NO	= 000000	HRDPRM	031646	G	LAB1	005757	
ERR1	013040	G	FMT5	012143	G\$OFFS=	000400	HRDWTS	022776	G	LAB2	005772	
ERR1FO	046400		FMT6	012163	G\$OFFSI=	000376	HRIN	002622		LCLEXT	026766	
ERR10	014430	G	FMT7	012225	G\$PRMA=	000001	HRINU	002624		LINE.F	033042	G
ERR2	013106	G	FMT8	012275	G\$PRMD=	000002	HROUT	002626		LOAD.F	037502	
ERR3	013154	G	FMT9	012327	G\$PRML=	000000	HROUTU	002630		LOCERR	003066	
ERR4	013222	G	FOLWRT=	000100	G\$RADA=	000140	HSMSK =	000100		LOCYL =	040000	
ERR5	013272	G	FORM.T	046410	G\$RADB=	000000	HSSTAT=	000100		LOGMSG	052012	
ERR6	013342	G	FREE	053716	G\$RADD=	000040	HW.ADR	032550	G	LOLIM =	000002	
ERR7	014220	G	FRMWD	010513	G\$RADF=	000200	H\$AAB	056606		LOLIMW	014660	
ERR8	014270	G	FWDSKO=	002000	G\$RADL=	000120	IBUFF	003466		LPBFR	032622	G
ERR9	014364	G	FWDSKS=	000400	G\$RADO=	000020	INIT	032772	G	LPCNTR	032620	G
ESC.PC	045106		F\$AU =	000015	G\$RADT=	000100	INITCO	014730	G	LPT.AD	037144	
EV.COU	032520	G	F\$BGN =	000040	G\$XFER=	000004	INITIA	052020		LPT.RE	037140	
EXACYL	002646		F\$CLEA=	000007	G\$YES =	000010	INITST	006347		LPT05	025674	
EXHCYL	002644		F\$DIJ =	000016	HADONE	002432	INIT.M	040104		LSI.RE	037134	
EXOCYL	002642		F\$END =	000041	HAMES1	007567	INIT.R	032606	G	LUP	060106	
EXROT	002650		F\$HARD=	000004	HAMES2	007652	INOUTS=	000020		LUP.AD	045112	
EXT05	026114		F\$HW =	000013	HCESTA=	040000	INPUTA	052746		L\$APT	002036	G
FBSFIL	003272		F\$INIT=	000006	HCORED	037256	INTEBL=	000100		L\$AUT	002074	G
FILL	052640		F\$JMP =	000050	HCOREQ	037166	INTFCR	046244		L\$CCP	002106	G
FILL.C	000204	G	F\$MOD =	000000	HCORET	033002	INTHLR	016064	G	L\$CLEA	015732	G
FLAGS	032560	G	F\$MSG =	000011	HCR CER=	004000	INVAL.	037052		L\$CO	002032	G
FLAGS1	032562	G	F\$PWR =	000017	HC.ADR	032552	INVINT	046102		L\$DEPO	002011	G
FLAGTA	056012		F\$RPT =	000012	HC.DEF	032544	INV.SW	034104		L\$DESC	002102	G
FLAG.I	037546		F\$SEG =	000003	HC.DIA	032542	IN.SUF	041054		L\$DEVP	002064	G
FLA.SE	055760		F\$SOFT=	000005	HDALIG=	000010	ISAU =	000041		L\$DISP	014674	G
FLG.MA	037506		F\$SRV =	000010	HDCYL =	077600	ISCLN =	000041		L\$DR	002112	G
FMTOP1	012016		F\$SUB =	000002	HDHSEL=	000100	ISDU =	000041		L\$DRCT	002070	G
FMTOP2	012045		F\$SW =	000014	HDMOVF	007444	ISHPD =	000041		L\$DRS	002072	G
FMTOP3	012067		F\$TEST=	000001	HDRCMP=	000002	ISINIT=	000041		L\$DRST	002112	G
FMT1	012110		GARBAG	053344	HDR40 =	100000	ISMOD =	000041		L\$DTP	002040	G
FMT1.1	012115		GETCHR	052050	HDSEC =	000077	ISMSG =	000041		L\$DU	015052	G
FMT11	012334		GETCMN	055434	HDSEL =	000020	ISPWR =	000041		L\$DUT	002076	G
FMT12	012342		GETPAR	047126	HDWD	010475	ISRPT =	000041		L\$DVTY	002114	G
FMT13	012350		GETPOS	021026	HDWRD1	002474	ISSEG =	000041		L\$EF	002056	G
FMT14	012414		GETSTA=	000003	HDWRD2	002476	ISSFT =	000041		L\$EFLG	002034	G
FMT15	012446		GETSWI	054430	HDWRD3	002500	ISSRV =	000041		L\$EXP1	002042	G
FMT16	012502		GET.TW	054200	HEAD =	000006	ISSUB =	000041		L\$EXP2	002044	G
FMT17	012513		GLBDAT	002122	HEADLM=	010000	ISTST =	000041		L\$EXP3	002046	G
FMT18	012535		GLBECA	002122	HEADW	014664	JSJMP =	000167		L\$HARD	031650	G
FMT19	012567		GLBERR	013040	HERTZ.	037126	KBPTR	032624	G	L\$HPCP	002016	G
FMT2	012124		GLBSUB	016056	HFIN	002612	KBUF	032626	G	L\$HPTP	002022	G
FMT20	012624		GLBXTX	004744	HFINU	002614	LABACF	007414		L\$HW	014642	G
FMT21	012654		GSTAT	016514	HFOUT	002616	LABACR	007430		L\$ICP	002104	G
FMT22	012677		GSTATC	016500	HFOUTU	002620	LABEXP	007323		L\$INIT	014730	G
FMT23	012733		GSTATG	016524	HICYL =	020000	LABHCF	007364		L\$LADP	002026	G
FMT24	012747		GSTATR	016464	HILIM -	000004	LABHCR	007400		L\$LAST	032514	G
FMT25	012754		GSTER1	006316	HILIMW	014662	LABIN	007300		L\$MREV	002050	G
FMT26	012764		GTSTAT-	000104	HNFERR=	010000	LABMID	007306		L\$NAME	002000	G
FMT27	013010		G\$EXCP=	000400	HOLDSP=	000020	LABOCF	007334		L\$REPP	002066	G
FMT28	013027		G\$HILI=	000002	HOSTAT=	000020	LABOCR	007350		L\$REV	002010	G

LSSOFT	031774	G	L10045	031644	MISWIW	014656	NUMBIN	046434	PASCNT	002654
LSSPC	002062	G	L10046	031622	MITEST=	100000	NUM.LA	046602	PASVEW	015150
LSSPCP	002020	G	L10047	031722	MNDRST	011405	NUM.NO	032556	PASNUM	003062
LSSPTP	002024	G	L10050	032036	MNEERR	011360	NUM.UN	033164	PASS.C	032530
LSSSTA	002030	G	MAJ.IN	032576	MNOCLR	006141	NUNITS	041572	PATTBL	002234
LSSW	014656	G	MAJ.LO	060206	MNOINT	006062	NXMERR=	020000	PAT1	004466
LSTIML	002014	G	MAJ.US	032600	MODR	062256	NXTFOR	056252	PAT10	004742
LSTIMU	002054	G	MANQ	032130	MOPER	005143	NXTPAS	015142	PAT2	004470
LSTIM1	002052	G	MAN.TI	001244	MOPERR	011276	OBUFF	004066	PAT3	004530
LSTSTI	002100	G	MAPROX	007270	MORECE	002436	OCTMSG	051776	PAT4	004570
LSUNIT	002012	G	MAP16	062714	MOUTIN	005214	OF IN	002562	PAT5	004630
L.BA	002460		MASK.B	041616	MPNAM	005752	OF INU	002564	PAT6	004636
L.CLK.	037112		MASK.W	041614	MQUALS=	003760	OFMID	002566	PAT7	004676
L.CS	002456		MBADAD	005462	MREAD	004752	OFMIDU	002570	PAT8	004700
L.DA	002462		MBADSF	005503	MREADH	004765	OFOUT	002572	PAT9	004740
L.MP	002464		MBHSTA	011105	MRESKO	005426	OFOUTU	002574	POSHDO	020534
L10000	013104		MBSETO=	000001	MREVS	005304	OLD CYL	002522	POSHSB	020530
L10001	013152		MCERR	010722	MRLFAL	011512	OPFLAG	002426	POSHW1	020522
L10002	013220		MCONHN	006115	MRSLT	005157	OPIERR=	002000	PRINTC	053320
L10003	013270		MCOSTA	011072	MSEEK	004744	OPMSG	002122	PRINTF	056626
L10004	013340		MCYLOC	011400	MSG.AD	032540	OPR02	007724	PRIOR =	000004
L10005	014216		MCYLUP	005204	MSG.TY	032514	OPR03	007751	PRI00 =	000000
L10006	014266		MDATCP	005046	MSPERR	011172	OPR04	010445	PRI01 =	000040
L10007	014362		MDCRC	010744	MSTERR	011225	OPR1	007774	PRI02 =	000100
L10010	014426		MDHEDR	002000	MTMBS	005560	OPR1A	010416	PRI03 =	000140
L10011	014636		MDLT	010771	MTOSLO	006013	OPR1B	010422	PRI04 =	000200
L10012	014654		MDRDY	010711	MUL	062212	OPR10	010300	PRI05 =	000240
L10013	014672		MDRERR	011033	MULOAD	005173	OPR11	010346	PRI06 =	000300
L10014	015730		MDRRES	006032	MUNDEF	011445	OPR12	010377	PRI07 =	000340
L10015	016050		MDRVST	011157	MVOLCK	011061	OPR2	010052	PRNTST	053210
L10016	016054		MDSERR	011142	MWDERR	011261	OPR3	010104	PRO.CM	037460
L10017	016120		MEM.SI	037154	MWGERR	011210	OPR5	010120	PSETNM	003064
L10020	023244		MERRS	011634	MWLSTA	011120	OPR6	010162	PTAB.S	033000
L10021	023446		MEXERS	011571	MWORD	006005	OPR7	010215	PUTCHR	052024
L10022	024702		MFLERR	011332	MWRCHK	005002	OPR8	010244	PWCON	015256
L10023	025510		MFMTER	005533	MWRITE	005016	OPR9	010263	PWRFLG	003072
L10024	025406		MFOLWR	005260	MWRSET	005127	ORIN	002576	PWR.FA	063150
L10025	026114		MFWDSK	005337	MWRTAB	011553	ORINU	002600	PWR.FL	032604
L10026	026022		MFWSKO	005372	M4QHDR	005113	ORMID	002602	PWR.MS	063276
L10027	026766		MGTSTA	005032	NEWCYL	002524	ORMIDU	002604	PWR.SA	063272
L10030	026712		MHCERR	011243	NEWPRI	061736	OROUT	002606	PWR.UP	063274
L10031	027300		MHCRC	010734	NEXAR	056176	OROUTU	002610	P.CLK.	037120
L10032	027364		MHDERR	011315	NOCLR =	000010	OUTINS=	000040	P2T01E	006477
L10033	027664		MHDRCP	005071	NOERCT	003067	OSAPTS=	000000	P2T02E	006477
L10034	030206		MHFRC	011003	NOIRPT-	000002	OSAU =	000000	P2T03E	006516
L10035	030166		MHNF	010755	NOOP =	000100	OSBGNR=	000000	P2T04E	006542
L10036	030640		MHOSTA	011131	NOGWR	005673	OSBGNS=	000001	P2T05E	006564
L10037	030536		MHSTA	011044	NO.CLK	037102	OSDU =	000001	P2T06E	006606
L10040	030636		MINOUT	005235	NO.FLA	055772	OSGNSW=	000001	P2T07E	006630
L10041	031156		MIN.IN	032572	NO.LPT	053310	OSPOIN=	000001	P2T08E	006654
L10042	031136		MIN.US	032574	NO.PTA	037306	PARSES	055506	P2T09E	006676
L10043	031370		MISTST	006225	NR =	000000	PART1 =	000000	P2T10E	006703
L10044	031304		MISWI =	000000	NSTACH	006250	PAR.LA	051474	P2T11E	006720

P2T12E 006735
P2T13E 006751
P2T14E 006771
P2T15E 007014
P2T16E 007044
P2T17E 007067
P2T18E 007127
P2T19E 007154
RDALHD 021154
RDDATA= 000114
RDHEAD= 000110
RDNOHR= 000116
RDYCHK 017632
RDYWAI 020560
READRL 016262
READ.P 060210 G
REGBAC 062700 G
REGSAV 062664 G
RELDWT= 040000
REQN.P 032570 G
REQN.T 037462
RESE3 011647
RESE4 011653
RESE5 011660
RESE6 011665
RESPAR 002504
RESTAR 015112
RESTBL 002174
REVSKE= 001000
REVSKS= 000200
RE.SET 034252
RLBA = 000002
RLBAS 002450
RLCS = 000000
RLCSR = 000000
RLDA = 000004
RLDRV 002454
RLMP = 000000
RLVEC 002452
RORWOP= 020000
RPTOP 021516
RPTREM 022512
RPTRES 022304
RSTACK 062140 G
RSTRT 015012
SAMSK = 000077
SAVEDO= 034450
SBSFIL 003076
SEARCH 054146
SECWD 010501
SEEK = 000106
SEEKOP= 010000
SEGSTA 033014 G

SELO 032036
SEQMES 010534
SETDON 015176
SET.MA 037672
SFTPRM 031772 G
SGNWD 010470
SHIFT 062776 G
SIMSEK 017346
SKTMES 007166
SPDERR 006262
SPDSTA= 004000
SPEC.U 037406
SPTCOD 014654 G
SPV.SE 000400
SRTMES 007202
SSINDX 002424
STAMES 010577
STAMSK= 000007
STARTC 061666 G
STATE2 011672
STATE3 011702
STATE5 011712
STOSTA= 010000
STRCHR 052700
STRT.T 037464
ST.SET 034316
SUBSTK 002260
SUNIT. 037470
SUPERV 035352
SUPFLA 032760 G
SUPV.T 033132 G
SUP.PR 034070
SVCBGL= 000001
SVCGBL= 000000
SVCHAN 042006
SVCINS= 000000
SVC SUB= 000001
SVCTAG= 000000
SVCTST= 000001
SWCHAN 037300
SWITCH 056152
SW.ADR 032554 G
SW.PTA 037264
SYS.FT 046032
SSLSYM= 010000
TBLSTR 002446
TCERR 010666
TEMPO 002540
TEMP1 002542
TEMP2 002544
TEMP3 002546
TEMP4 002550
TEMP5 002552

TEMP6 002554
TEMP7 002556
TEMP8 002560
TERMI 060176
TERMLI 056000
TERMTA 051762
TEST.M 037420
TIMFLG 032750 G
TIM.CO 032602 G
TIM.OP 046406
TOO.MA 051742
TOSLOW= 000001
TRPFLG 003070
TRPHAN 016056
TSTINT 016446
TSTLAB 006217
TST.AB 041730
TST.TO 034132
TYPEC 052336
TYPEPC 046232
TYPFLA 055654
TYPLIN 052234
TYPNUM 051616
TYPSTR 052254
TYP.ER 046062
TY.UNI 041074
T\$ARGC= 000002
T\$CODE= 000130
T\$ERCO= 000043
T\$ERRN= 002735
T\$EXCP= 000000
T\$FLAG= 000040
T\$HILI= 000377
T\$LOLI= 000000
T\$LSYM= 010000
T\$NEST= 177777
T\$NSKO= 000000
T\$NSK1= 000005
T\$NSK2= 000002
T\$SAVL= 177777
T\$SEGL= 177777
T\$SUBN= 000001
T\$TAGL= 177777
T\$TAGN= 010051
T\$TEMP= 000000
T\$TEST= 000016
T\$TSTM= 177777
T\$TSTS= 000001
T\$SCLE= 010015
T\$SDU = 010016
T\$SHAR= 010047
T\$SHW = 010012
T\$SINI= 010014

T\$MSG= 010011
T\$SOF= 010050
T\$SRV= 010017
T\$SUB= 010046
T\$SW = 010013
T\$TES= 010045
T.BA 002470
T.CS 002466
T.DA 002472
T.MP 002474
T.STAT 002502
TOSERR 006362
TO9ERR 006375
T1 022776 G
T10 027666 G
T10ERR 006405
T10.1 027720
T104\$ 027720
T11 030210 G
T11.1 030366
T11.2 030540
T115\$ 030540
T12 030642 G
T12ERR 006425
T12.1 030670
T124\$ 030670
T13 031160 G
T13ERR 006437
T13.1 031204
T134\$ 031204
T14 031372 G
T14ERR 006453
T14.1 031414
T153\$ 031414
T16ERR 006467
T2 023246 G
T25TBL 002304
T3 023450 G
T33TBL 002332
T365\$ 024702
T4 024704 G
T4.1 024724
T465\$ 025400
T5 025512 G
T5.1 026004
T504\$ 026036
T6 026116 G
T6.1 026512
T7 026770 G
T8 027302 G
T9 027366 G
ULOAD = 000010
UNDTST 010432

UNIT.D 032532 G
UNI.MA 037410
UNXERR 006202
USER.P 032774 G
USER.T 032776 G
VALDES 007231
VALID. 033234
VAL.LA 034054
VAL.SW 037520
VCNRST 006161
VCSTAT= 001000
VECMSG 031736
VECT = 000002
WAITIN 016314
WCMSK = 017777
WCRNG = 160000
WDESTA= 100000
WGESTA= 002000
WIDTH 047002
WLSTAT= 020000
WRTSWI 002444
WTDATA= 000112
XEQDIA 062024 G
XFQSUB 062012 G
XEQ.CL 041534
XEQ.CM 037044
XEQ.IN 041216
XEQ.LA 035306
XEQ.OP 041310
XEQ.PR 034510
XEQ.TE 041354
XRDHD 020102
XRDHDC 020072
XRDHDG 020106
XTIME 060676 G
XTIMEN 061522
XTIMST 060720
XXDP.D 037064
XSALWA= 000000
XSALS= 000040
XSOFFS= 000400
XSTRUF= 000020
\$BREG 037560
\$ENDAD 061776 G
\$SAV2 063042 G
\$SAV3 063056 G
\$SAV4 063074 G
\$SAV5 063114 G
: = 063320

ERRORS DETECTED: 0

DSKZ:CZRLCB,DSKZ:CZRLCB/EQ:PART1=CZRLCB/ML,CZRLCB.PT1,CZRLCB.P11,CZRLCB.PT2,CZRLCB.SUP
RUN-TIME: 50 49 1 SECONDS
RUN-TIME RATIO: 177/101-1.7
CORE USED: 16K (31 PAGES)