

IDENTIFICATION

PRODUCT CODE: AC-E044B-MC
PRODUCT NAME: CZRLCPO RLC1 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 SVNTAX
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 COPE" 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 RL01 DRIVES
1 - 8 RL01K CARTRIDGES WITH BAD SECTOR FILE
KW11P, KW11L (OPTIONAL)
LINEPRINTER(OPTIONAL)

1.2.2 SOFTWARE REQUIREMENTS

CZRLCRO RL01 DRIVE TEST PART 1
(FORMERLY MD-11-DZRLC-A)

1.3 RELATED DOCUMENTS AND STANDARDS

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

1.4 DIAGNOSTIC HIERARCV PREREQUISITES

THE RL01 SUBSYSTEM SHOULD HAVE SUCCESSFULLY RUN THE FOLLOWING PROGRAMS:

CVRLAAQ RLV11 RL01 DISKLESS TEST (RLV11)
CZRLABO RL11/RLV11 CONTROLLER TEST PART 1
CZRLBBO RL11/RLV11 CONTROLLER TEST PART 2

1.5 ASSUMPTIONS

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

2.0 OPERATING INSTRUCTIONS

2.1 HOW TO RUN THIS DIAGNOSTIC

2.1.1 THE SIX STEPS OF EXECUTION

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

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

* STEP 1 *

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

QUESTION MEANING -----

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

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

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

* STEP 2 *

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

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

STA/PASS:1/FLAGS:HOE

THINGS TO NOTE HERE:

1. ONLY THE FIRST THREE CHARACTERS OF THIS OR ANY COMMAND AT THE "DS-B>" LEVEL NEED TO BE TYPED.
2. THE "PASS" SWITCH SPECIFIES HOW MANY PASSES YOU DESIRE. A PASS CONSISTS OF RUNNING THE FULL DIAGNOSTIC AGAINST ALL UNITS BEING TESTED (THIS WILL BE EXPLAINED SHORTLY). ONE PASS IS SPECIFIED IN THE ABOVE EXAMPLE.
3. THE "FLAGS" SWITCH MAY SPECIFY ANY OF A NUMBER OF FLAGS, BUT THE MAIN USEFUL ONES ARE:

LOE	LOOP ONE ERROR
HOE	HALT ON ERROR
IER	INHIBIT ERROR PRINTOUT

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

* STEP 3 *

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

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

 * STEP 4 *

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

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

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

 * STEP 5 *

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

 * STEP 6 *

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

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

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

HOE SET: THE ERROR WILL BE REPORTED ON THE CONSOLE AND THE DIAGNOSTIC WILL RETURN TO COMMAND MODE.
 LOE SET: THE DIAGNOSTIC WILL LOOP ENDLESSLY ON THE BLOCK OF CODE THAT DETECTED THE ERROR.
 NEITHER HOE NOR LOE SET: THE ERROR WILL BE REPORTED ON THE CONSOLE AND NORMAL EXECUTION WILL RESUME AS IF NO ERROR HAD OCCURED.

2.1.2 SAMPLE RUN-THROUGH

LET'S SEE HOW ALL THIS WORKS IN A REAL SITUATION. RECALL THAT WE ENTERED THE COMMAND "STA/PASS:1/FLAGS:HOE". THIS WOULD BE A VERY TYPICAL WAY TO RUN THE DIAGNOSTIC. IF NO ERRORS ARE ENCOUNTERED, THE SINGLE REQUESTED PASS WILL BE EXECUTED AND THE PROMPT WILL BE 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/FLAGS:IER:LOE:HOE=0

THIS WILL DO THE FOLLOWING:

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

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

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

1. START
2. RESTART
3. CONTINUE

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

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

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

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

1. START
2. RESTART

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

2.2 HOW TO CREATE A CHAINABLE FILE

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

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

HERE IS A SAMPLE DIALOGUE TO ACCOMPLISH THIS:

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

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

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

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

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

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

```
.R DIAG.BIC
```

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

2.3 DETAILS OF COMMANDS AND SYNTAX

2.3.1 TABLE OF COMMAND VALIDITY

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

HOW ENTERED	LEGAL COMMANDS
1. OPERATOR ENTERED "RUN DIAG"	START PRINT DISPLAY FLAGS ZFLAGS
2. DIAGNOSTIC HAS FINISHED ALL ITS REQUESTED 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 COLONS, WHERE <FLAG> HAS ONE OF THE FOLLOWING VALUES:

HOE	HALT ON ERROR, CAUSING COMMAND MODE TO BE ENTERED WHEN AN ERROR IS ENCOUNTERED
LOE	LOOP ON ERROR, CAUSING THE DIAGNOSTIC TO LOOP CONTINUOUSLY WITHIN THE SMALLEST DEFINED BLOCK OF CODING (SEGMENT, SURTEST, OR TEST) CONTAINING THE ERROR
IER	INHIBIT ERROR REPORTING
IBE	INHIBIT BASIC ERROR REPORTS
IXE	INHIBIT EXTENDED ERROR REPORTS
PRI	DIRECT ALL MESSAGES TO A LINE PRINTER
PNT	PRINT NUMBER OF TES BEING EXECUTED
BOE	BELL ON ERROR
UAM	RUN IN UNATTENDED MODE, BYPASSING MANUAL INTERVENTION TESTS
ISR	INHIBIT STATISTICAL REPORTS
IDU	INHIBIT DROPPING OF UNITS BY DIAGNOSTIC

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

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

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

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

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

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

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

1. UNSPECIFIED FLAG SETTINGS ARE UNCHANGED

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

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

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

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

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

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

 DRO(P)/UNITS:UNIT-LIST

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

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

 ADD/UNITS:UNIT-LIST

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

 PRI(NT)

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

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

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

 FLA(GS)

THE CURRENT SETTINGS OF ALL FLAGS ARE PRINTED.

 ZFL(AGS)

ALL FLAGS ARE CLEARED.

2.4 EXTENDED P-TABLE DIALOGUE

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

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

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

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

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

IN GIVING A STRING OF VALUES, COMMAS WITHOUT INTERVENING VALUES MAY BE USED TO INDICATE A REPETITION OF THE LAST NAMED VALUE.

A STRING OF VALUES MAY BE GIVEN AS A RANGE (6-10 FOR EXAMPLE). IF THE VALUES REPRESENT PURE NUMERICAL DATA, THIS SAMPLE RANGE 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 INTTERFACE 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 FURTHUR 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 PPROGRAM 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   SURTEST 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      CVL      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 AS INCREMENTAL SEKS, 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/DLT	DCRC/HCRC	OPI	MESSAGE
1	1	1	HDR NOT FND/HDR CRC/OPI ERROR
0	1	1	HDR CRC ERROR
1	0	1	HDR NOT FND ERROR
0	1	0	DATA CRC ERROR
1	0	0	DATA LATE ERROR

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

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

LINE 8 PROVIDES THE 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 HEADER - READ HEADER FOR 40 HEADERS - READ HEADER FOR 40 HEADERS WITH HEADER COMPARE - HAVE THE SAME GENERAL MEANING AS THE READ DATA AND READ DATA WITH DATA COMPARE. MESSAGES HAVING THE OPERATION OF READ HEADER OR READ HEADER FOR 40 HEADERS ARE THE RESULT OF ERRORS DETECTED IN THE ACTUAL OPERATION WHILE THE READ HEADER FOR 40 HEADERS WITH HEADER COMPARE INDICATES NO ERROR IN THE ACTUAL OPERATION BUT THE HEADER DATA ITSELF WAS IN ERROR.

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

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

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

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

THE ABOVE OPERATIONS CAN BE REPORTED WITH ANY OF THE

QUALIFIERS. THE QUALIFIERS IN THESE TESTS ARE AN ATTEMPT TO MAKE THE REPORT MORE MEANINGFUL BY PROVIDING INFORMATION ABOUT THE SEQUENCE OF OPERATIONS BEING DONE.

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

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

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

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

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

3.1.2 SPECIFIC RESULT MESSAGES

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

THE GENERAL FORMAT FOR THE RESULT LINE IS

RESULT:(VAR 1) IS (VAR 2) SB (VAR 3) (OPTIONAL QUALIFIER)

WHERE VARIABLE 1 CAN BE ONE OF THE FOLLOWING:

CONT ERR	(CONTROLLER ERROR)
DRV ERR	(DRIVE ERROR)
NON-EXSTNT MEM	(NON-EXISTANT MEMORY)
HDR CRC	(HEADER CRC ERROR)
DATA CRC	
HDR NOT FND	(HEADER NOT FOUND)
DATA LATE	
HDR NOT FND/HDR CRC/OPI	(ALL 3 BITS SET)
DRV RDY	(DRIVE READY)
SELECTED HEAD	
VOL CHK	(VOLUME CHECK)
COVER OPEN	
BRUSH HME	(BRUCH HOME)
WRT LCK	(WRITE LOCK)
HDS OUT	(HEADER OUT)
DRV SEL ERR	(DRIVE SELECT ERROR)
DRV STATE	(DRIVE STATE)

SPIN TIMEOUT	(SPINDLE TIMEOUT SPD ERROR)
WRT GAT ERR	(WRITE GATE ERROR)
SEEK TIMEOUT	(SKTO ERROR)
CUR HEAD ERR	(CURRENT IN HEAD ERROR)
WRT DAT ERR	(WRITE DATA ERROR)
OP INCOMPLETE	(OPI ERROR)
HDR/DAT ERR	(HEADER CRC OR DATA CRC ERROR BIT 11 OF CS REGISTER)
HDR NOT FND/DAT LATE	(HEADER NOT FOUND OR DATA LATE ERROR BIT 12 OF CS REGISTER)
CVL	(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 HVE IS 1 SB 0 IN STATE 2
HEADS OUT IS 0 SB 1 IN STATE 3
DRV RDY IS 0 SB 1 IN DATA XFER
SELECTED HEAD IS 1 SB 0 IN CYCLE UP
DRV RDY IS 0 SB 1 IN STATE 5
DRV RDY IS 1 SB 0 IN SEEK W/O MOTION
DRV RDY IS 0 SB 1 IN 10MS
DRV RDY IS 0 SB 1 IN 500MS
DRV RDY IS 0 SB 1 IN 5SECONDS
```

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

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

"INTERRUPT TO LATE" WHICH INDICATES THAT THE OPERATION BEING PERFORMED DID NOT

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

"FAIL TO RELOAD 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 NO CORRESPOND TO THE EXPECTED FORMAT. (REFER TO DEC STANDARD 144 FOR FORMAT SPECIFICS.)

3.1.3 OTHER MESSAGES

OTHER INFORMATION IS REPORTED UNDER VARIOUS CIRCUMSTANCES. 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 COMPLIMENT)

FOR READ HEADER FUNCTION

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

FOR GET STATUS FUNCTION

HAS DRIVE STATUS

BIT 15 - WRITE DATA ERROR
 BIT 14 - CURRENT HEAD ERROR(CHE)
 BIT 13 - WRITE LOCK STATUS(WL)
 BIT 12 - SEEK TIME OUT(SKTO)
 BIT 11 - SPIN ERROR(SPE)
 BIT 10 - WRITE GATE ERROR(WGE)
 BIT 9 - VOLUME CHECK(VC)
 BIT 8 - DRIVE SELECT ERROR(DSE)
 BIT 7 - RESERVED(0)
 BIT 6 - SURFACE

BIT 5 - COVER OPEN
 BIT 4 - HEADS HOME
 BIT 3 - BRUSHES HOME
 BIT 2-0 - STATE BITS
 0 - LOAD STATE
 1 - SPIN UP
 2 - BRUSH CYCLE
 3 - LOAD HEADS
 4 - SEEK - TRACK COUNTING
 5 - SEEK - LINEAR MODE
 6 - UNLOAD HEADS
 7 - SPIN DOWN

6.0 TEST SUMMARIES

TEST 1 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 TO 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 OPI 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 OPI 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 OUT 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. %


```

23 002110 FNDMOD DEVRG
(5) 002110 000000 .WORD 0
(2) 002112 000001 .RPLW
(2) 002114 046122 030460 000 .DEVTP <PL01>
(2) 002114 002122 .ASCIZ /RLC1/
    .EVEN

;COPYRIGHT (C) 1977, 1978
;THIS SOFTWARE IS FURNISHED UNDER LICENSE FOR USE ONLY
;ON A SINGLE COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH
;THE INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS
;SOFTWARE, OR ANY COPIES THEREOF, MAY NOT BE PROVIDED
;OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT
;FOR USE ON SUCH SYSTEM, AND TO ONE WHO AGREES TO THESE
;LICENSE TERMS. TITLE TO OWNERSHIP OF THE SOFTWARE SHALL
;AT ALL TIMES REMAIN IN DEC.

;THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE
;WITHOUT NOTICE AND SHALL NOT BE CONSTRUED AS A COMMITMENT
;BY DIGITAL EQUIPMENT CORPORATION.

;DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY
;OF ITS SOFTWARE OR EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.

002122 BGNMOD GLBEQAT
002122 EQUALS
    000006 OFFSETS FOR HARDWARE P-TABLE
    000002 CSR =0 ;BUS ADDRESS
    000004 PRIOR =4 ;VECTOR ADDRESS
    000006 DRSR =4 ;PRIORITY
    000010 CNT =10 ;DRIVE SELECT BIT
    ;CONTROLLER TYPE

    ; OFFSET FOR SOFTWARE P-TABLE
    MISW =0 ;SOFTWARE PARAMETERS SWITCHES
    L0LIM =2 ;CYLINDER LOWER LIMIT
    H0LIM =4 ;CYLINDER HIGH LIMIT
    HEAD =6 ;CYLINDER HEAD FOR RUNNING TESTS
    ERLIM =10 ;ERROR LIMIT
    DCLIM =12 ;DATA COMPARE ERROR LIMIT

    ; BIT ASSIGNMENT FOR SOFTWARE P-TABLE SWITCHES
    ALLCVL =BIT00 ;USE ALL CYLINDERS
    ALLSEC =BIT01 ;USE ALL SECTORS
    DRSELC =BIT02 ;EXECUTE DRIVE SELECT TEST
    HDALIGN =BIT03 ;EXECUTE HEAD ALIGNMENT TEST
    AUTOSZ =BIT04 ;AUTO SIZE FOR DRIVE-DROP IF NO RESPONSE
    HEADLW =BIT05 ;HEAD LIMIT SPECIFIED FLAG
    H0LIMT =BIT06 ;H0 LIMIT SPECIFIED FLAG
    L0CYL =BIT07 ;H0 LIMIT SPECIFIED
    WITEST =BIT15 ;EXECUTE MANUAL INTERVENTION TESTS

    ; SUBSYSTEM FUNCTIONS
    CKDATA =102 ;WRITE CHECK
    
```

```

73 000104 GTSTAT =104 ;GET STATUS
74 000102 SEEK =100 ;SEEK
75 000110 RDHEAD =100 ;SEEK HEADER
76 000112 WTDATA =112 ;WRITE DATA
77 000114 RDDATA =114 ;READ DATA
78 000112 RDNHR =100 ;READ DATA, IGNORE HEADERS
79 000100 NOOP =100 ;NO OPERATION

; OPERATION FLAGS
80 007777 COMPOP =7777 ;COMPOSITE OPERATION FLAGS
81 000002 HDRCMP =BIT01 ;HEADER COMPARE OPERATION
82 000001 DATACMP =BIT00 ;DATA COMPARE OPERATION
83 000004 CVLUP =BIT02 ;CYCLE UP OPERATION
84 000008 ULOAD =BIT03 ;UNLOAD OPERATION
85 000010 INOUTS =BIT04 ;IN-OUT SEEK OPERATION
86 000020 OUTINS =BIT05 ;OUT-IN SEEK OPERATION
87 000040 FOLWRT =BIT06 ;FOLLOWING WRITE OPERATION
88 000100 REVSKS =BIT07 ;REV SEEK SEQ (ADJ INTERFERENCE)
89 000200 FWDSKS =BIT08 ;FWD SEEK SEQ (ADJ INTERFERENCE)
90 000400 REVSKO =BIT09 ;REV SEEK SEQ (OVERWRITE)
91 001000 FWDSKO =BIT10 ;FWD SEEK SEQ (OVERWRITE)
92 002000 RADADD =BIT11 ;RAD DISK ADDRESS
93 010000 SEEKOP =BIT12 ;SEEK OPERATION
94 020000 ROPWOP =BIT13 ;READ OP WRITE OPERATION
95 040000 RELOAD =BIT14 ;RELOAD WAIT
96 100000 HBR40 =BIT15 ;40 HEADER OPERATION
97 003760 MQUALS =OUTINS|INOUTS|FOLWRT|REVSKS|FWDSKS|REVSKO|FWDSKO
    ;MESSAGE QUALIFIER BITS

; ERROR FLAGS FROM SUBROUTINES
100 000001 TOSLOW =BIT00 ;OPERATION TOOK TO LONG
101 000002 NOIRPT =BIT01 ;NO INTERRUPT FROM OPERATION
102 000004 CONHNG =BIT02 ;CONTROLLER HUNG
103 000010 NOCLR =BIT03 ;RAD CONTROLLER CLEAR

; CONTROL AND STATUS REGISTER
104 000000 RLCS =0 ;CONTROL AND STATUS REGISTER
105 100000 ANVERR =10000 ;ANY ERROR BIT
106 040000 DRVERR =40000 ;DRIVE ERROR BIT
107 020000 NVWERR =20000 ;NON-EXISTANT MEMORY ERROR
108 010000 DLTERR =10000 ;DATA LATE ERROR
109 010000 HNGERR =10000 ;DATA CHECK ERROR
110 040000 DCKERR =40000 ;HEADER NOT FOUND ERROR
111 040000 HRCERR =40000 ;HEADER CHECK ERROR
112 020000 RPTERR =20000 ;OPERATION INCOMPLETE ERROR
113 010000 DSMSK =10000 ;DRIVE SELECT MASK
114 0A2000 CRDYSK =20000 ;CONTROLLER READY MASK
115 001000 INTEBL =100 ;INTERPRET ENABLE MASK
116 001000 BMSK =100 ;BUS ADDRESS UPPER MASK
117 000001 DRDYSK =1 ;DRIVE READY MASK
    
```

```

129          000077          ; SANSK REGISTER PIT DEFINITIONS - DISK ADDRESS FOR DATA XFER
130          000100          ; SANSK REGISTER PIT DEFINITIONS - DISK ADDRESS FOR DATA XFER
131          077600          ; SANSK REGISTER PIT DEFINITIONS - DISK ADDRESS FOR DATA XFER
132          077600          ; SANSK REGISTER PIT DEFINITIONS - DISK ADDRESS FOR DATA XFER
133          000001          ; MRSET0 REGISTER RIT DEFINITIONS - DISK ADDRESS FOR SEEK
134          000000          ; DIRBIT REGISTER RIT DEFINITIONS - DISK ADDRESS FOR SEEK
135          000000          ; HDSBL REGISTER RIT DEFINITIONS - DISK ADDRESS FOR SEEK
136          077600          ; DIRMSK REGISTER RIT DEFINITIONS - DISK ADDRESS FOR SEEK
137          000003          ; GETSTAT REGISTER RIT DEFINITIONS - DISK ADDRESS FOR GET STATUS
138          000010          ; DRSET REGISTER RIT DEFINITIONS - DISK ADDRESS FOR GET STATUS
139          017777          ; WMSK REGISTER RIT DEFINITIONS - WP FOR DATA XFER
140          160000          ; WCRNG REGISTER RIT DEFINITIONS - WP FOR DATA XFER
141          077600          ; HDCYL REGISTER RIT DEFINITIONS - WP FOR READ HEADER
142          000077          ; HDSEC REGISTER RIT DEFINITIONS - WP FOR READ HEADER
143          000100          ; HDHSEL REGISTER RIT DEFINITIONS - WP FOR READ HEADER
144          000007          ; STANSK REGISTER RIT DEFINITIONS - WP FOR GET STATUS
145          000010          ; RRSTAT REGISTER RIT DEFINITIONS - WP FOR GET STATUS
146          000020          ; HSTAT REGISTER RIT DEFINITIONS - WP FOR GET STATUS
147          000040          ; CSTAT REGISTER RIT DEFINITIONS - WP FOR GET STATUS
148          000100          ; HSSSTAT REGISTER RIT DEFINITIONS - WP FOR GET STATUS
149          000400          ; DSESTAT REGISTER RIT DEFINITIONS - WP FOR GET STATUS
150          000100          ; VCSSTAT REGISTER RIT DEFINITIONS - WP FOR GET STATUS
151          000200          ; WCSSTAT REGISTER RIT DEFINITIONS - WP FOR GET STATUS
152          004000          ; SPDSTAT REGISTER RIT DEFINITIONS - WP FOR GET STATUS
153          010000          ; STOSTAT REGISTER RIT DEFINITIONS - WP FOR GET STATUS
154          010000          ; MLSTAT REGISTER RIT DEFINITIONS - WP FOR GET STATUS
155          040000          ; HCSSTAT REGISTER RIT DEFINITIONS - WP FOR GET STATUS
156          100000          ; WDESTAT REGISTER RIT DEFINITIONS - WP FOR GET STATUS
157          002122          ; ENDMOD
158          002122          ; BGNMOD
159          002122          ; CLRDAT
160          002122          ; CLRDAT
161          000000          ; OPMSGS: TABLE OF OPERATION MESSAGES
162          002124          ; .WORD 0 ; FILLER FOR WRITE CHECK
163          002126          ; .WORD MFTSTA ; MESSAGE FOR GET STATUS
164          002130          ; .WORD MSEEK ; SEEK
165          002134          ; .WORD MREADH ; READ HEADER
166          002136          ; .WORD MWRITE ; WRITE DATA
167          002140          ; .WORD MWRSET ; WITH RESET
168          002144          ; .WORD MDATECP ; WITH DATA COMPARE
169          002146          ; .WORD MHDPCP ; WITH HEADER COMPARE
170          002150          ; .WORD MCLUP ; LOAD HEADS
171          002152          ; .WORD MLOAD ; UNLOAD HEADS
172          002152          ; .WORD MINOUT ; IN-OUT SEQ
    
```

```

185          002154          ; .WORD MOUTIN ; OUT-IN SEQ
186          002156          ; .WORD MFWLWRT ; FOLLOWING WRITE
187          002160          ; .WORD MREVSK ; REV SEEK
188          002162          ; .WORD MREVSK ; REV SEEK
189          002164          ; .WORD MREVSK ; REV SEEK
190          002166          ; .WORD MREVSK ; REV SEEK
191          002170          ; .WORD MREVSK ; REV SEEK
192          002172          ; .WORD MREVSK ; REV SEEK
193          002172          ; .WORD MREVSK ; REV SEEK
194          002172          ; .WORD MREVSK ; REV SEEK
195          002174          ; .WORD MREVSK ; REV SEEK
196          002176          ; .WORD MREVSK ; REV SEEK
197          002178          ; .WORD MREVSK ; REV SEEK
198          002180          ; .WORD MREVSK ; REV SEEK
199          002182          ; .WORD MREVSK ; REV SEEK
200          002184          ; .WORD MREVSK ; REV SEEK
201          002186          ; .WORD MREVSK ; REV SEEK
202          002188          ; .WORD MREVSK ; REV SEEK
203          002190          ; .WORD MREVSK ; REV SEEK
204          002192          ; .WORD MREVSK ; REV SEEK
205          002194          ; .WORD MREVSK ; REV SEEK
206          002196          ; .WORD MREVSK ; REV SEEK
207          002198          ; .WORD MREVSK ; REV SEEK
208          002200          ; .WORD MREVSK ; REV SEEK
209          002202          ; .WORD MREVSK ; REV SEEK
210          002204          ; .WORD MREVSK ; REV SEEK
211          002206          ; .WORD MREVSK ; REV SEEK
212          002208          ; .WORD MREVSK ; REV SEEK
213          002210          ; .WORD MREVSK ; REV SEEK
214          002212          ; .WORD MREVSK ; REV SEEK
215          002214          ; .WORD MREVSK ; REV SEEK
216          002216          ; .WORD MREVSK ; REV SEEK
217          002218          ; .WORD MREVSK ; REV SEEK
218          002220          ; .WORD MREVSK ; REV SEEK
219          002222          ; .WORD MREVSK ; REV SEEK
220          002224          ; .WORD MREVSK ; REV SEEK
221          002226          ; .WORD MREVSK ; REV SEEK
222          002228          ; .WORD MREVSK ; REV SEEK
223          002230          ; .WORD MREVSK ; REV SEEK
224          002232          ; .WORD MREVSK ; REV SEEK
225          002234          ; .WORD MREVSK ; REV SEEK
226          002236          ; .WORD MREVSK ; REV SEEK
227          002238          ; .WORD MREVSK ; REV SEEK
228          002240          ; .WORD MREVSK ; REV SEEK
229          002242          ; .WORD MREVSK ; REV SEEK
230          002244          ; .WORD MREVSK ; REV SEEK
231          002246          ; .WORD MREVSK ; REV SEEK
232          002248          ; .WORD MREVSK ; REV SEEK
233          002250          ; .WORD MREVSK ; REV SEEK
234          002252          ; .WORD MREVSK ; REV SEEK
235          002254          ; .WORD MREVSK ; REV SEEK
236          002256          ; .WORD MREVSK ; REV SEEK
237          002260          ; .WORD MREVSK ; REV SEEK
238          002262          ; .WORD MREVSK ; REV SEEK
239          002264          ; .WORD MREVSK ; REV SEEK
240          002266          ; .WORD MREVSK ; REV SEEK
241          002268          ; .WORD MREVSK ; REV SEEK
242          002270          ; .WORD MREVSK ; REV SEEK
243          002272          ; .WORD MREVSK ; REV SEEK
244          002274          ; .WORD MREVSK ; REV SEEK
245          002276          ; .WORD MREVSK ; REV SEEK
246          002278          ; .WORD MREVSK ; REV SEEK
247          002280          ; .WORD MREVSK ; REV SEEK
248          002282          ; .WORD MREVSK ; REV SEEK
249          002284          ; .WORD MREVSK ; REV SEEK
250          002286          ; .WORD MREVSK ; REV SEEK
251          002288          ; .WORD MREVSK ; REV SEEK
252          002290          ; .WORD MREVSK ; REV SEEK
253          002292          ; .WORD MREVSK ; REV SEEK
254          002294          ; .WORD MREVSK ; REV SEEK
255          002296          ; .WORD MREVSK ; REV SEEK
256          002298          ; .WORD MREVSK ; REV SEEK
257          002300          ; .WORD MREVSK ; REV SEEK
258          002302          ; .WORD MREVSK ; REV SEEK
259          002304          ; .WORD MREVSK ; REV SEEK
260          002306          ; .WORD MREVSK ; REV SEEK
261          002308          ; .WORD MREVSK ; REV SEEK
262          002310          ; .WORD MREVSK ; REV SEEK
263          002312          ; .WORD MREVSK ; REV SEEK
    
```


353	002562	000000	OPIN:	.WORD	0	;ONE CYLINDER FORWARD INNER
354	002564	000000	OPINU:	.WORD	0	;UPPER
355	002566	000000	OPID:	.WORD	0	;ONE CYLINDER FORWARD MIDDLE
356	002570	000000	OPIDU:	.WORD	0	;UPPER
357	002572	000000	OPOUT:	.WORD	0	;ONE CYLINDER FORWARD OUTER
358	002574	000000	OPOUTU:	.WORD	0	;UPPER
359	002576	000000	ORIN:	.WORD	0	;ONE CYLINDER REVERSE INNER
360	002580	000000	ORINU:	.WORD	0	;UPPER
361	002582	000000	ORMID:	.WORD	0	;ONE CYLINDER REVERSE MIDDLE
362	002584	000000	ORMIDU:	.WORD	0	;UPPER
363	002586	000000	OROUT:	.WORD	0	;ONE CYLINDER REVERSE OUTER
364	002590	000000	OROUTU:	.WORD	0	;UPPER
365	002592	000000	HRIN:	.WORD	0	;128 CYLINDER FORWARD INNER
366	002594	000000	HRINU:	.WORD	0	;UPPER
367	002596	000000	HRMID:	.WORD	0	;128 CYLINDER FORWARD OUTER
368	002598	000000	HRMIDU:	.WORD	0	;UPPER
369	002600	000000	HROUT:	.WORD	0	;128 CYLINDER REVERSE INNER
370	002602	000000	HROUTU:	.WORD	0	;UPPER
371	002604	000000	HRIN:	.WORD	0	;128 CYLINDER REVERSE OUTER
372	002606	000000	HRINU:	.WORD	0	;UPPER
373	002608	000000	HRMID:	.WORD	0	;128 CYLINDER REVERSE OUTER
374	002610	000000	HRMIDU:	.WORD	0	;UPPER
375	002612	000000	HROUT:	.WORD	0	;256 CYLINDER FORWARD
376	002614	000000	HROUTU:	.WORD	0	;UPPER
377	002616	000000	ARMID:	.WORD	0	;256 CYLINDER REVERSE
378	002618	000000	ARMIDU:	.WORD	0	;UPPER
379	002620	000000	EXOCVL:	.WORD	150	;EXPECTED TIME ONE CYLINDER
380	002622	000000	EXOCVL:	.WORD	150	;EXPECTED TIME 128 CYLINDER
381	002624	000000	EXACVL:	.WORD	1000	;EXPECTED TIME 156 CYLINDER
382	002626	000000	EXROT:	.WORD	250	;EXPECTED ROTATION TIME
383	002628	000000	EPRVEC:	.WORD	4	;ERROR VECTOR USED WHEN AUTO SIZING
384	002630	000000				; MISCELLANEOUS COUNTERS
385	002632	000000	PASCNT:	.WORD	0	;PASS COUNTER (LOCAL TO A TEST)
386	002634	000000	COUNT:	.WORD	0	;A COUNTER (LOCAL TO A TEST)
387	002636	000000	ERRPNT:	.WORD	0	;ERROR POINTER
388	002638	000100	ERRCNT:	.BLKW	64	;STORAGE FOR ERROR COUNTERS
389	002640	000000	PASNUM:	.WORD	0	;PASS NUMBER FOR PROGRAM
390	002642	000000	PSETNM:	.WORD	0	;COUNTER FOR PARAMETER SET NUMBER IN USE
391	002644	000000	LOCERR:	.BYTE	0	;LOCAL ERROR COUNTER
392	002646	000000	NOERCT:	.BYTE	0	;INHIBIT ERROR COUNTING FLAG
393	002648	000000	TRPFLG:	.WORD	0	;HARDWARE TRAP OCCURANCE
394	002650	000000	PWRFLG:	.WORD	0	;POWER FAILURE OCCURANCE
395	002652	000000				; RAD SECTOR TABLES AND POINTERS
396	003074	000000	BSFVAL:	.WORD	0	;RAD SECTORS FILES VALID FLAG
397	003076	000076	SBSFIL:	.BLKW	76	;SOFTWARE BAD SECTOR FILE
398	003272	000076	FBSFIL:	.BLKW	76	;FACTORY RAD SECTOR FILE
399	003466	000200	IBUFF:	.BLKW	200	;INPUT BUFFER
400	004068	000200	OBUFF:	.BLKW	200	;OUTPUT BUFFER
401	004466	000000	PAT1:	.WORD	0	;PATTERN 1 (ALL ZEROS)
402	004470	177777	PAT2:	.WORD	177777	
403	004474	177777		.WORD	177777	
404	004478	177777		.WORD	177777	

409	004476	052525		.WORD	052525	
410	004480	052525		.WORD	052525	
411	004484	052525		.WORD	052525	
412	004504	177777		.WORD	177777	
413	004508	177777		.WORD	177777	
414	004512	052525		.WORD	052525	
415	004516	052525		.WORD	052525	
416	004514	177777		.WORD	177777	
417	004516	052525		.WORD	052525	
418	004520	177777		.WORD	177777	
419	004524	177777		.WORD	177777	
420	004524	177777		.WORD	177777	
421	004526	177777		.WORD	177777	
422	004530	000000		.WORD	000000	
423	004532	000000	PAT3:	.WORD	000000	
424	004534	000000		.WORD	000000	
425	004536	000000		.WORD	000000	
426	004538	177777		.WORD	177777	
427	004540	177777		.WORD	177777	
428	004542	177777		.WORD	177777	
429	004544	000000		.WORD	000000	
430	004546	000000		.WORD	000000	
431	004548	000000		.WORD	000000	
432	004550	177777		.WORD	177777	
433	004552	177777		.WORD	177777	
434	004554	000000		.WORD	000000	
435	004556	000000		.WORD	000000	
436	004560	177777		.WORD	177777	
437	004564	000000		.WORD	000000	
438	004566	177777		.WORD	177777	
439	004570	052525		.WORD	052525	
440	004572	052525	PAT4:	.WORD	052525	
441	004574	052525		.WORD	052525	
442	004576	052525		.WORD	052525	
443	004578	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	125252		.WORD	125252	
451	004616	052525		.WORD	052525	
452	004620	052525		.WORD	052525	
453	004622	125252		.WORD	125252	
454	004624	125252		.WORD	125252	
455	004626	125252		.WORD	125252	
456	004630	155555		.WORD	155555	
457	004632	155555	PAT5:	.WORD	155555	
458	004634	155555		.WORD	155555	
459	004636	155555		.WORD	155555	
460	004638	155555		.WORD	155555	
461	004636	121105	PAT6:	.WORD	121105	
462	004640	156442		.WORD	156442	
463	004644	156442		.WORD	156442	
464	004644	132210		.WORD	132210	

465	0J4646	055044				.WORD	055044
466	004650	022442				.WORD	022442
467	004651	011321				.WORD	011321
468	004654	105504				.WORD	105504
469	004656	042642				.WORD	042642
470	004660	011321				.WORD	011321
471	004660	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			PATR:	.WORD	045513
481	004702	122645				.WORD	122645
482	004704	151322				.WORD	151322
483	004706	064555				.WORD	064555
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	064555				.WORD	064555
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							
503	004744				RGNM0D	GLBTYT	
504	004744	042523	045505	000040	MSEK:	.ASCIZ	/SEEK /
505	004752	042522	042101	042040	WPEAD:	.ASCIZ	/READ DATA /
506	004755	051122	040565	020104	MREADH:	.ASCIZ	/READ HEADED /
507	005006	051122	052111	020109	MWRCH:	.ASCIZ	/WRITE CHECK /
508	005007	042507	020124	052123	MGTST:	.ASCIZ	/GET STATUS /
509	005036	044527	024124	042040	MDATCP:	.ASCIZ	/WITH DATA COMPARE /
510	005113	1066	052111	032040	MHDRS:	.ASCIZ	/WITH HDR COMPARE /
511	005113	1066	052111	032040	MHDRS:	.ASCIZ	/FOR 40 HDRS /
512	005127	127	052111	020110	MWRSET:	.ASCIZ	/WITH RESET /
513	005143	117	042522	040522	MOPPR:	.ASCIZ	/OPERATION: /
514	005143	117	042522	040522	MOPPR:	.ASCIZ	/OPERATION: /
515	005173	125	046109	026164	MRSCL:	.ASCIZ	/RESULTS /
516	005204	042114	042040	053122	MULDR:	.ASCIZ	/UNLD DRV /
517	005214	047066	020114	020060	MOUTIN:	.ASCIZ	/LD DRV /
518	005214	047066	020114	020060	MOUTIN:	.ASCIZ	/FOL 0 TO CC SEEK /
519	005266	047066	020114	051127	MINDUT:	.ASCIZ	/FOL 255 TO CC SEEK /
520					MPDLWPT:	.ASCIZ	/FOL WRITE (NO SEEK) /

524	005304	042101	020112	054503	MREVS:	.ASCIZ	/ADJ CVL WRITN AFTER REV SK /
525	005337	045104	041444	042127	MPWDSK:	.ASCIZ	/ADJ CVL WRITN AFTER FWD SK /
526	005372	045233	053040	042127	MPWSD:	.ASCIZ	/SK FWD, WRT - SK REV, OVERWRT /
527	005426	045233	053040	042105	MPWSD:	.ASCIZ	/SK REV, WRT - SK FWD, OVERWRT /
528	005462	047117	041040	042101	MWRADP:	.ASCIZ	/WRAD SEC FILE /
529	005506	1033	047101	052047	MBADSF:	.ASCIZ	/CAN'T GET RAD SEC FILES /
530	005560	047524	048240	051440	MPWTEP:	.ASCIZ	/RAD SEC FILE FMT ERR /
531	005622	052502	020123	042101	BASADD:	.ASCIZ	/TO MANY BAD SEC FOR PROG CAPACITY /
532	005630	051104	053122	000075	DRVNAM:	.ASCIZ	/DRVS ADD= /
533	005630	051104	053122	020105	DRVNAV:	.ASCIZ	/DRIVE UNAVAILABLE FOR TEST /
534	005636	122	041514	006123	CSNAM:	.ASCIZ	/DRV DID NOT REC'R FROM PWR FAIL /
535	005673	046122	040505	000000	RANAM:	.ASCIZ	/RRA /
536	005740	046122	052111	000100	DANAM:	.ASCIZ	/RLDA /
537	005752	046122	052111	000100	HPNAM:	.ASCIZ	/RLDA /
538	005757	117	020120	047111	LAR1:	.ASCIZ	/RLWP /
539	005776	050117	042040	047111	LAR2:	.ASCIZ	/OP DONE = /
540	006019	111	053116	050104	MWORD:	.ASCIZ	/WORD /
541	006032	050117	020111	042523	MOSLW:	.ASCIZ	/INTRPT TO LATE /
542	006032	047506	044440	052116	MORRES:	.ASCIZ	/OPT SET NO DRV RESPONSE /
543	006115	0406	052116	042114	MNOINT:	.ASCIZ	/NO INTRPT ON CWRD COMPLETE /
544	006141	105	053122	042040	MCONHNG:	.ASCIZ	/CNTLP HUNG (NO RDV) /
545	006161	126	046117	041440	VCMRST:	.ASCIZ	/ERR DID NOT CLR /
546	006219	0406	042524	052123	UNXERR:	.ASCIZ	/VOL CHK NOT RSET /
547	006250	115	047122	052123	TSLAR:	.ASCIZ	/UNXPCTD EPR /
548	006250	052123	052101	020105	WSTSTG:	.ASCIZ	/TEST /
549	006250	052123	052101	020105	NSTACHG:	.ASCIZ	/MAN INTERVENT STAT /
550	006312	052123	042116	020114	SPDRPP:	.ASCIZ	/SPNDL TIMEOUT CRG /
551	006347	111	046516	043040	GSTERL:	.ASCIZ	/FAIL FORCING DRV SEL EPR /
552	006362	051104	020126	042523	T05ERR:	.ASCIZ	/INIT STATE /
553	006375	104	053122	051040	T09ERR:	.ASCIZ	/DRV SELCT /
554	006425	110	020104	053523	T10ERR:	.ASCIZ	/DRV RDV /
555	006437	127	020104	042110	T12ERR:	.ASCIZ	/SEEK SGN SWITCH /
556	006445	127	020104	042110	T13ERR:	.ASCIZ	/HD SWITCH /
557	006477	127	020104	042110	T14ERR:	.ASCIZ	/RD HDR (P1) /
558	006477	104	043111	020106	T16ERR:	.ASCIZ	/WRT LCK /
559	006516	052517	020124	051107	P201E:	.ASCIZ	/DIFF OF 1 SEEK /
560	006524	047111	020103	042523	P203E:	.ASCIZ	/OUT GRD RAND DETECT /
561	006566	047111	020103	042523	P204E:	.ASCIZ	/INC SEEK FWD HD 0 /
562	006606	047111	020103	042523	P205E:	.ASCIZ	/INC SEEK REV HD 0 /
563	006630	047111	020116	051167	P206E:	.ASCIZ	/INC SEEK FWD HD 1 /
564	006654	042523	020103	042523	P207E:	.ASCIZ	/INC SEEK REV HD 1 /
565	006675	106	042127	047000	P208E:	.ASCIZ	/SEEK /
566	006703	042522	047066	051517	P210E:	.ASCIZ	/FWD OSC SEEK /
567	006720	106	042522	020113	P211E:	.ASCIZ	/REV OSC SEEK /
568	006735	106	042522	020113	P212E:	.ASCIZ	/SEEK TIMING /
569	006771	106	052122	051057	P213E:	.ASCIZ	/ASIC /
570	006771	106	052122	051057	P213E:	.ASCIZ	/ASIC /
571	007014	051127	047111	046104	P215E:	.ASCIZ	/ASIC READ DATA (P1) /
572	007014	051127	047111	046104	P215E:	.ASCIZ	/ASIC READ DATA (P1) /
573	007014	051127	047111	046104	P216E:	.ASCIZ	/ASIC READ DATA (P1) /
574	007014	051127	047111	046104	P216E:	.ASCIZ	/ASIC READ DATA (P1) /
575	007014	051127	047111	046104	P217E:	.ASCIZ	/ASIC READ DATA (P1) /
576	007014	051127	047111	046104	P217E:	.ASCIZ	/ASIC READ DATA (P1) /
577	007014	051127	047111	046104	P217E:	.ASCIZ	/ASIC READ DATA (P1) /
578	007014	051127	047111	046104	P217E:	.ASCIZ	/ASIC READ DATA (P1) /
579	007014	051127	047111	046104	P217E:	.ASCIZ	/ASIC READ DATA (P1) /
580	007014	051127	047111	046104	P217E:	.ASCIZ	/ASIC READ DATA (P1) /
581	007014	051127	047111	046104	P217E:	.ASCIZ	/ASIC READ DATA (P1) /


```

700 012275 045 022516 022524 PMT8: .ASCIZ /*N*T*06*S2*06*S2*06*S2*06/
701 012337 045 022516 006124 PMT9: .ASCIZ /*N*/
702 012334 052 0445 047445 PMT10: .ASCIZ /*R1/
703 012342 052 0445 000063 PMT12: .ASCIZ /*T*03/
704 012414 047 0445 051445 PMT13: .ASCIZ /*N*S11*T*03*S1*T*03*S1*T*01*S1*T*01/
705 012414 047 0445 052045 PMT14: .ASCIZ /*N*S11*T*03*S1*T*06*S1*T*06/
706 012446 047 0445 051445 PMT15: .ASCIZ /*N*S11*T*03*S1*T*06*S1*T*06/
707 012502 047 0445 022465 PMT16: .ASCIZ /*N*S5*06/
708 012513 045 030523 022460 PMT17: .ASCIZ /*S10*T*N*S11*06*N/
709 012525 045 022516 030523 PMT18: .ASCIZ /*N*S13*T*S5*T*S4*T*S5*T*N/
710 012525 045 022516 031123 PMT19: .ASCIZ /*N*S13*T*S5*T*S4*T*S4*D6*N/
711 012624 052 0445 051445 PMT20: .ASCIZ /*T*S2*D6*S14*D6*S4*D6*N/
712 012654 052 0445 051445 PMT21: .ASCIZ /*T*S12*D6*S14*D6*N/
713 012677 045 022516 030523 PMT22: .ASCIZ /*N*S11*T*03*S1*T*01*S1*T*02/
714 012733 045 022516 022524 PMT23: .ASCIZ /*T**T*01*N/
715 012747 045 022516 000124 PMT24: .ASCIZ /*N*/
716 012754 047 0445 042045 PMT25: .ASCIZ /*N*D2*T/
717 012784 047 0445 022461 PMT26: .ASCIZ /*N*S11*T*04*T*03*N/
718 013010 047 0445 042045 PMT27: .ASCIZ /*N*S11*T*04*T*03*N/
719 013027 045 022516 022524 PMT28: .ASCIZ /*N*T*T*T/
720 013040 045 022516 022524 ENDMOD
721
722
723
724
725
726
727 013040 RGNMOD GLRFRP
728 ; ERR1 R3 POINTS TO RESULT MESSAGE
729 ; RESULT: (R3)
730
731 ; ERR2 R3 POINTS TO RESULT NAME
732 ; RESULT: (R3) IS 1 SB 0
733 ;
734 ; ERR3 R3 POINTS TO RESULT NAME
735 ; RESULT: (R3) IS 0 SR 1
736 ;
737 ; ERR4 R3 POINTS TO RESULT NAME
738 ; R4 POINTS TO RESULT CONDITIONS
739 ; RESULT: (R3) IS 1 SB 0 (R4)
740 ;
741 ; ERR5 R3 POINTS TO RESULT NAME
742 ; R4 POINTS TO RESULT CONDITIONS
743 ; RESULT: (R3) IS 0 SR 1 (R4)
744 ;
745 ; ERR6 RESULT ROUTINE DETERMINES WHICH ERROR(S) ARE SET AND
746 ; REPORTS ALL
747 ; RESULT: "ERROR" IS 1 SB 0
748 ;
749 ; ERR7 DRIVE STATE ERROR REPORT
750 ; R3 CONTAINS EXPECTED STATE
751 ; T*STAT CONTAINS BAD STATE
752 ; RESULT: DRIVE STATE IS (T*STAT) SR (R3)
753 ;
754 ; ERR8 HEAD POSITIONING ERROR REPORT
755 ; NEWCVL CONTAINS EXPECTED CYLINDER
756 ; HDWRD1 CONTAINS BAD CYLINDER
757 ; RESULT: CYLINDER IS (HDWRD1) SR (NEWCVL)
758 ;
759 ; ERR9 UTILITY RESULT REPORT
760 ; R3 POINTS TO RESULT NAME
    
```

```

761 ; R4 POINTS TO VALUE 1
762 ; R5 POINTS TO VALUE 2
763 ; RESULT: (R3-NAME) IS (R4-VALUE 1) SR (R5-VALUE 2)
764 ;
765 ; ERR10 COMPARE ERROR REPORT
766 ; R3 CONTAINS THE BAD WORD NUMBER
767 ; R4 POINTS TO BAD WORD
768 ; R5 POINTS TO GOOD WORD
769 ; RESULT: WORD (R3) IS (R4) SR (R5)
770
771 013040 RGNMSG ERR1
772 013040 105737 003067 BSTR NQERCT ;TEST IF ERROR COUNTING INHIBITED
773 013040 005737 167606 INC IERRPOINT ;YES - SKIP
774 013046 005737 167606 INC IERRPOINT ;ELSE BUMP ERROR COUNT
775 013052 010146 1S: MOV R1,(SP) ;STORE R1
776 013054 004737 021516 JSE RPTOP ;REPORT OPERATION
777 013060 012721 000001 MOV R3,(R1)+ ;SET PARAM NUMBER
778 013064 012721 000001 MOV R4,(R1)+ ;INSERT MESSAGE ADDRESS POINTER
779 013066 004737 022304 JSR PC,RPTRES ;REPORT RESULTS
780 013072 004737 022512 JSP PC,RPTREM ;REPORT REMAINDER
781 013076 012601 016122 MOV (SP),R1 ;RESTORE R1
782 013100 004737 016122 JSR PC,CKERLM ;GO CHECK IF ERROR COUNT EXCEEDED
783 013104
784 013104 L1000C: EMT C$MSG
785 013106 BGNMSG ERR2
786 013106 005277 167546 INC IERRPOINT ;RUMP ERROR COUNT
787 013114 012721 021516 MOV R1,(SP) ;STORE R1
788 013114 005277 167546 JSE RPTOP ;REPORT OPERATION
789 013120 012721 000003 MOV R3,(R1)+ ;SET PARAM NUMBER
790 013124 010321 000001 MOV R4,(R1)+ ;INSERT NAME ADD POINTER
791 013130 012721 000001 MOV R1,(R1)+ ;SET IS VALUE
792 013132 012721 000001 CLR R1 ;SET SB VALUE
793 013134 004737 022304 JSR PC,RPTRES ;REPORT RESULTS
794 013140 004737 022512 JSP PC,RPTREM ;REPORT REMAINDER
795 013142 012601 016122 MOV (SP),R1 ;RESTORE R1
796 013146 004737 016122 JSR PC,CKERLM ;GO CHECK IF ERROR COUNT EXCEEDED
797 013152
798 013152 L10001: EMT C$MSG
799 013154 RGNMSG ERR3
800 013154 005277 167500 INC IERRPOINT ;RUMP ERROR COUNT
801 013160 012721 021516 MOV R1,(SP) ;STORE R1
802 013162 005277 167500 JSE RPTOP ;REPORT OPERATION
803 013166 012721 000003 MOV R3,(R1)+ ;SET PARAM NUMBER
804 013172 010321 000001 MOV R4,(R1)+ ;INSERT NAME ADD POINTER
805 013174 012721 000001 CLR (R1)+ ;SET IS VALUE
806 013176 012721 000001 MOV R1,(R1)+ ;SET SB VALUE
807 013202 004737 022304 JSR PC,RPTRES ;REPORT RESULTS
808 013206 004737 022512 JSP PC,RPTREM ;REPORT REMAINDER
809 013214 012601 016122 MOV (SP),R1 ;RESTORE R1
810 013214 004737 016122 JSR PC,CKERLM ;GO CHECK IF ERROR COUNT EXCEEDED
811 013220 ENDMMSG
    
```



```
(3) 013220 L10002: EMT CSMSG
(3) 013220 104023
813 013222 BGNMSG ERR4
814 013222 INC 4ERRPOINT ;BUMP ERROR COUNT
815 013222 MOV R1, -(SP) ;STORE R1
816 013222 JSP PC, RPTOP ;REPORT OPERATION
817 013222 MOV R4, (R1)+ ;SET PARAM NUMBER
818 013222 MOV R3, (R1)+ ;INSERT NAME ADD POINTER
819 013222 MOV R1, (R1)+ ;SET IS VALUE
820 013222 CLR (P1)+ ;SET SB VALUE
821 013222 MOV R4, (R1) ;INSERT ADD OF CONDITION POINTER
822 013222 JSR PC, RPTRES ;REPORT RESULTS
823 013222 JSP PC, RPTREM ;REPORT REMAINDER
824 013222 MOV (SP)+, R1 ;RESTORE R1
825 013222 JSP PC, CKERLW ;GO CHECK IF ERROR COUNT EXCEEDED
826 (3) 013270 FNDMSG L10003: EMT CSMSG
(3) 013270 104023
827 013272 BGNMSG ERR5
828 013272 INC 4ERRPOINT ;BUMP ERROR COUNT
829 013272 MOV R1, -(SP) ;STORE R1
830 013272 JSP PC, RPTOP ;REPORT OPERATION
831 013272 MOV R4, (R1)+ ;SET PARAM NUMBER
832 013272 MOV R3, (R1)+ ;INSERT NAME ADD POINTER
833 013272 MOV R1, (R1)+ ;SET IS VALUE
834 013272 CLR (P1)+ ;SET SB VALUE
835 013272 MOV R4, (R1) ;INSERT ADD OF CONDITION POINTER
836 013272 JSR PC, RPTRES ;REPORT RESULTS
837 013272 JSP PC, RPTREM ;REPORT REMAINDER
838 013272 MOV (SP)+, R1 ;RESTORE R1
839 013272 JSP PC, CKERLW ;GO CHECK IF ERROR COUNT EXCEEDED
840 (3) 013340 FNDMSG L10004: EMT CSMSG
(3) 013340 104023
841 013342 BGNMSG ERR6
842 013342 TSTB 175, CS ;TEST IF ERROR COUNTING INHIBITED
843 013342 JZ 175, CS ;SKIP
844 013342 INC 4ERRPOINT ;ELSE BUMP ERROR COUNT
845 013342 MOV R1, -(SP) ;STORE R1
846 013342 MOV R3, -(SP) ;STORE R3
847 013342 MOV R4, -(SP) ;STORE R4
848 013342 MOV R4, -(SP) ;STORE R4
849 013342 MOV R4, -(SP) ;STORE R4
850 013342 JSE PC, RPTOP ;REPORT OPERATION
851 013342 MOV R3, (R1)+ ;SET PARAM NUMBER
852 013342 MOV R4, (R1)+ ;INSERT NAME ADD POINTER
853 013342 CLR REMP, (R1) ;CLEAR FOP STATUS STORAGE
854 013342 MOV T, CS, R3 ;GET T, CS
855 013342 BIC #177761, R3 ;AND CLEAR ALL BUT GET STATUS
856 013342 CVP #4, R3 ;CHECK IF IT WAS GET STATUS
857 013342 BEQ 15, R3 ;YES - STATUS IS IN T, MP, SKIP
858 013342 MOV #GETSTAT, RLD(A(R2)) ;ELSE DO GET STATUS
859 013342 MOV #4, R3
860 013342 MOV #RDRV, R3
861 013342 BIT
```

```
862 013442 MOV R3, RLCS(R2)
863 013446 WAITUS #10, R0 ;WAIT FOR CONTROLLER READY
(3) 013452 EMT 104027
864 013454 BIT #RDRVMSK, RLCS(R2) ;TEST IF READY
865 013454 BNE 109S ;YES - SKIP
866 013454 MOV R3, RIT9, R3 ;ELSE SET NO DRIVE STATUS BIT
867 013454 BR 2 ;CLEAR USED BITS
868 013454 MOV R4, RMP(P2), R3 ;STORE STATUS FOR REPORT
869 013454 MOV R3, TEM3
870 013454 MOV R4, R3+1, R3 ;GET ERROR BITS IN PROPER POSITION
871 013454 BR 15S ;MP+1, R3
872 013454 MOV R4, R3+1, R3 ;GET ERROR BITS FROM MP REG
873 013454 BIC #177442, R3 ;CLEAR UNUSED BITS
874 013454 MOV R4, R3+1, R3 ;GET ERROR BITS FROM CS REG
875 013454 BIC #1777, R4 ;CLEAR UNUSED BITS
876 013454 MOV R4, R3 ;WAKE ONE WORD OF POSSIBLE ERRORS
877 013454 BIT #OPIERR, R3 ;TEST IF OPI SET
878 013454 BNE 107S ;NO - SKIP
879 013454 BIT #HRCERR, R3 ;TEST IF HDR CRC ERR
880 013454 BNE 107S ;YES - SKIP
881 013454 BIT #HRCERR, R3 ;TEST IF HDR CRC ERR
882 013454 BNE 107S ;YES - SKIP
883 013454 MOV #HRCERR, R4 ;SET OPI ALONE MESSAGE
884 013454 PRINTR #EMT28, #MRSLT, R4 ;EMPRS, REPORT ERROR
(3) 013466 MOV R4, -(SP)
(3) 013472 MOV #SLT, -(SP)
(3) 013478 MOV #MRSLT, -(SP)
(3) 013484 MOV #4, -(SP)
(3) 013490 EMT CSBNT
(3) 013496 ADD #12, SP
887 013498 BR 100S ;SKIP
888 013498 MOV #HRCRC, R4 ;HDR CRC MESSAGE
889 013498 BR 100S
890 013498 BIT #HRCERR, R3 ;TEST IF HRCRC WITH HDR NOT FND
891 013498 BNE 109S ;YES - SKIP
892 013498 MOV #HRCRC, R4 ;MESSAGE HEADER NOT FOUND
893 013498 BR 100S
894 013498 MOV #HRCPC, R4 ;HNF AND HRCRC MESSAGE
895 013498 BR 100S ;SKIP
896 013498 BIT #DCERR, R3 ;TEST IF DATA CHECK SET, NOT OPI
897 013498 BNE 109S ;YES - SKIP
898 013498 MOV #HRCRC, R4 ;SET MESSAGE DATA CHECK
899 013498 BR 100S ;SKIP
900 013498 BIT #OPIERR, R3 ;TEST IF DATA LATE ERROR
901 013498 BNE 109S ;NO - SKIP
902 013498 MOV #VDT, R4 ;SET MESSAGE DATA LATE
903 013498 BR 100S ;SKIP
904 013498 CLR R4 ;SET BIT POINTER FOR TEST
905 013498 BIT R4, R3 ;TEST IF BIT IS SET
906 013498 BNE 6S ;YES - SKIP TO REPORT
907 013498 TST (P4)+ ;ELSE BUMP TABLE POINTER
908 013498 CLC ;CLEAR CARRY
```

```

908 013712 006005 ROR P5 ;SHIFT BIT POINTER TO NEXT BIT
909 013714 001372 BNE 36 ;LOOP IF NOT 0
910 013716 000405 BR 76 ;ELSE REPORT REMAINDER
911 013718 016411 MOV PC,RPTRES ;INSERT NAME ADDRESS
912 013724 034937 JSP PC,RPTRES ;REPORT RESULTS
913 013730 000766 BR 45 ;GET NEXT BIT
914 013732 004737 JSR PC,RPTREM ;REPORT REMAINDER
915 013735 002548 TST R#P3 ;TEST IF ANY NEW STATUS
916 013739 003414 BGT 16 ;NO - SKIP
917 013744 PRINTR #FMT17,#STAMES,TEMP3
(0) 013744 013746 MOV TEMP3,-(SP)
(1) 013750 013746 MOV #STAMES,-(SP)
(2) 013754 013746 MOV #FMT17,-(SP)
(3) 013760 013746 MOV #3,-(SP)
(4) 013764 010600 MOV SP,R0
(5) 013770 083914 EMT CSMMSG
(6) 013774 032737 ADD #10,SP
918 013774 032737 BIT #DCKERR,T.CS ;TEST IF DATA CHECK ERROR
919 014002 001453 BEQ 26 ;NO - SKIP
920 014004 032737 BIT #DPIERR,T.CS ;TEST IF DPI SET
921 014010 001447 BNE 26 ;YES - SKIP
922 014014 055337 CLR MDRECE ;CLEAR COMPARE ERROR COUNT
923 014020 012701 MOV #128,R1 ;SET COMPARE LENGTH
924 014024 012701 MOV #DREFF,R5 ;SET WORD COUNT
925 014030 012704 MOV #IBUFF,R4 ;SET TEST WORD POINTER
926 014040 021514 CMP (R5),(R4) ;CHECK WORD
927 014044 023727 BCS 18 ;GOOD - SKIP
928 014052 002436 MDRECE,#10. ;TEST IF COMPARE LIMIT REACHED
929 014054 003021 BGT 20 ;YES - SKIP
930 014054 PRINTR #FMT15,#WORD,R3,#RESE3,(R4),#RESE4,(R5)
(0) 014054 011546 MOV #RESE4,-(SP)
(1) 014056 011446 MOV (R4),-(SP)
(2) 014060 011647 MOV #RESE3,-(SP)
(3) 014064 011647 MOV (R4),-(SP)
(4) 014072 012746 MOV #WORD,-(SP)
(5) 014076 012746 MOV #FMT15,-(SP)
(6) 014102 012746 MOV #7,-(SP)
(7) 014110 104014 EMT CSMMSG
(8) 014112 062706 ADD #20,SP
(9) 014116 005237 INC MDRECE ;BUMP ERROR COUNTER
934 014124 005263 INC (R5)+,(R4)+ ;BUMP POINTERS
935 014126 005301 DEC R1 ;DEC LENGTH COUNT
936 014130 001449 RNE 18 ;LOOP IF NOT DONE
937 014134 001449 TST MDRECE ;TEST IF ANY COMPARE ERRORS
938 014136 001421 BGT 27 ;NO - SKIP
939 014140 012701 MOV #128,R1 ;SET COMPARE LENGTH
940 014144 PRINTR #FMT17,#TCERR,MDRECE,#RESE6,R1
(0) 014144 010146 MOV #RESE6,-(SP)
(1) 014146 012746 MOV MDRECE,-(SP)
(2) 014152 012746 MOV #TCERR,-(SP)
(3) 014156 012746 MOV #FMT17,-(SP)
(4) 014162 013010 MOV #FMT17,-(SP)
    
```

```

(6) 014166 012746 MOV #5,-(SP)
(7) 014174 010600 SP,R0
(8) 014178 104014 EMT CSMMSG
(9) 014176 062706 ADD #14,SP
941 014202 012605 MOV (SP)+,R5 ;RESTORE R5, 4, 3, 1
942 014204 012605 MOV (SP)+,R4
943 014206 012605 MOV (SP)+,R3
944 014210 012601 MOV (SP)+,R1
945 014212 004737 JSR PC,CKERLM ;GO CHECK IF ERROR COUNT EXCEEDED
(0) 014216 004737 ENDMMSG L10005:
(1) 014218 104023 EMT CSMMSG
946 014220 BGNMSG ERR7
(2) 014220 005277 166434 INC #ERRPOINT ;BUMP ERROR COUNT
(3) 014224 010146 MOV R1,-(SP) ;STORE R1
(4) 014226 004737 JSR PC,RPTOP ;REPORT OPERATION
(5) 014230 000093 MOV #3,(R1)+ ;SET PARAM NUMBER
(6) 014234 012721 MDRECE,(R1)+ ;INSERT NAME ADD POINTER
(7) 014238 002502 MOV R3,(R1)+ ;INSERT SR VALUE
(8) 014242 010319 MOV R3,(R1)+ ;INSERT SR VALUE
(9) 014246 004737 JSR PC,RPTRES ;REPORT RESULTS
955 014254 004737 JSR PC,RPTREM ;REPORT REMAINDER
956 014258 012601 MOV (R5)+,R1 ;RESTORE R1
957 014262 004737 JSR PC,CKERLM ;GO CHECK IF ERROR COUNT EXCEEDED
(0) 014266 004737 ENDMMSG L10006:
(1) 014268 104023 EMT CSMMSG
958 014270 BGNMSG ERR8
(2) 014270 005277 166364 INC #ERRPOINT ;BUMP ERROR COUNT
(3) 014274 010146 MOV R1,-(SP) ;STORE R1
(4) 014276 010346 MOV R3 ;STORE R3
(5) 014300 004737 JSR PC,RPTOP ;REPORT OPERATION
(6) 014304 012721 MOV #3,(R1)+ ;SET PARAM NUMBER
(7) 014308 012721 MDRECE,(R1)+ ;INSERT NAME ADD POINTER
(8) 014312 012721 MOV #4,WORD,(R1)+ ;GET HEADER WORD
(9) 014316 012703 MOV #7,R3 ;SET SHIFT COUNT
967 014324 002411 CLC ;ALIGN CHAR FOR PRINTING
968 014328 006011 ROR R3 ;AS IS VALUE
969 014332 005373 DEC R3
970 014336 005373 BNE 35
971 014340 005721 TST (R1)+ ;BUMP PARAM POINTER
972 014344 004737 MOV #R1,SR ;INSERT SR VALUE
973 014348 004737 JSR PC,RPTRES ;REPORT RESULTS
974 014352 004737 JSR PC,RPTREM ;REPORT REMAINDER
975 014356 012605 MOV (SP)+,R3 ;RESTORE R3
976 014360 004737 JSR PC,CKERLM ;GO CHECK IF ERROR COUNT EXCEEDED
(0) 014364 004737 ENDMMSG L10007:
(1) 014368 104023 EMT CSMMSG
977 014370 BGNMSG ERR9
(2) 014370 005277 166270 INC #ERRPOINT ;BUMP ERROR COUNT
(3) 014374 010146 MOV R1,-(SP) ;STORE R1
    
```

```

987 014372 004737 021516 JSF PC,RPSTOP ;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 014410 004737 022304 JSP PC,PPRES ;SET SR VALUE
992 014414 004737 022512 JSP PC,PPTEW ;REPORT RESULTS
993 014420 004737 016122 MOV #2,R1 ;RESTORE R1
994 014426 004737 JSP PC,CKERLM ;GO CHECK IF ERROR COUNT EXCEEDED
995 014426 004737
996 014426 004737
997 014426 004737
998 014430 104023 ENDMMSG L10010:
999 014430 010146 RCNMSG EMT CSMSG
1000 014432 005737 MOV R1,-(SP) ;STORE R1
1001 014434 001051 TST M0RECE ;TEST IF 2ND RAD LINE
1002 014444 004737 166214 RNF 3S ;YES - SKIP
1003 014450 004737 021516 JSP PC,RPSTOP ;RAMP ERROR COUNT
1004 014450 004737 021516 PRINTB #FMT5,#BASADD,RLBAS,#DRVNAM,<R,RLDRV+1>;REPORT ID
1005 014450 004737 021516 CLR -(SP)
1006 014452 005046 BISR RDRV+1,(SP)
1007 014454 005633 MOV DRVNAM,-(SP)
1008 014456 002455 MOV RLBAS,-(SP)
1009 014458 005123 MOV #BASADD,-(SP)
1010 014460 000005 MOV #5,-(SP)
1011 014502 010600 MOV SP,R0
1012 014504 004014 EMT CSPTB
1013 014510 000014 PRINTB #FMT14,#MRSLT,#WORD,P3,#RESE3,(R4),#RESE4,(R5)
1014 014512 011546 MOV #5,-(SP)
1015 014514 011453 MOV #RESE4,-(SP)
1016 014516 011647 MOV #RESE3,-(SP)
1017 014518 000005 MOV R3,-(SP)
1018 014520 005157 MOV #MWORD,-(SP)
1019 014522 002414 MOV #FMT11,-(SP)
1020 014524 000010 MOV #10,-(SP)
1021 014526 004014 MOV SP,R0
1022 014528 000022 EMT CSPTB
1023 014530 000421 ADD #22,SP
1024 014532 011546 BR 4S
1025 014534 011653 3S: PRINTB #FMT15,#WORD,P3,#RESE3,(R4),#RESE4,(R5);REPORT DATA
1026 014536 011446 MOV #RESE4,-(SP)
1027 014538 011647 MOV #RESE3,-(SP)
1028 014540 006005 MOV R3,-(SP)
1029 014542 000006 MOV #MWORD,-(SP)
1030 014544 000007 MOV #FMT15,-(SP)
1031 014546 000007 MOV #7,-(SP)
1032 014548 104014 MOV SP,R0
1033 014550 002706 EMT CSPTB
1034 014552 002706 ADD #20,SP
1035 014554 002706 INC M0RECE ;TNC COMPARE ERROR COUNT
1036 014556 002706
1037 014624 005237
    
```

```

1008 014630 012601 MOV (SP)+,R1 ;RESTORE R1
1009 014632 004737 016122 JSP PC,CKERLM ;GO CHECK IF ERROR COUNT EXCEEDED
1010 014636 004737 ENDMMSG L10011:
1011 014640 104023 ENDMOD EMT CSMSG
1012 014640 .EVEN
1013 014640 BGNMOD HPTCCDE
1014 014640 BGNHW
1015 014640 .WORD L10012-LSHW/2
1016 014642 000005 .WORD 174400 ;CSR BASE ADDRESS DEFAULT
1017 014644 000160 .WORD 140 ;VECTOR DEFAULT
1018 014646 000240 .WORD 40 ;PRIORITY DEFAULT
1019 014650 000300 .WORD 40 ;DRIVE NUMBER DEFAULT
1020 014654 000001 .WORD 1 ;RL11 CONTROLLER
1021 014654 ENDMH L10012:
1022 014654 ENDMOD
1023 014654 BGNMOD SPTCCDE
1024 014654 BGNHW
1025 014654 MISWIW: .WORD L10013-LSSW/2
1026 014656 .WORD 0
1027 014656
1028 014656
1029 014656
1030 014656
1031 014656
1032 014656
1033 014656
1034 014656
1035 014660 000000 L0LIMW: .WORD 0
1036 014662 000377 H0LIMW: .WORD 255.
1037 014664 000000 HEADW: .WORD 20.
1038 014666 000023 ERLIMW: .WORD 20.
1039 014668 000012 DCLIMW: .WORD 10.
1040 014672 ENDSW L10013:
1041 014672 ENDMOD
1042 014672
1043 014672
1044 014672
1045 014672 BGNMOD DSPCCDE
1046 014674 DISPATCH .WORD 14
1047 014676 .WORD 14
1048 014678 .WORD 11
1049 014680 .WORD 13
1050 014682 .WORD 13
1051 014684 .WORD 14
1052 014686 .WORD 14
1053 014688 .WORD 16
1054 014690 .WORD 16
1055 014692 .WORD 18
1056 014694 .WORD 10
1057 014696 .WORD 110
1058 014698 .WORD 110
1059 014700 .WORD 112
1060 014702 .WORD 112
    
```

```

(6) 014724 031160          .WORD T13
(7) 014730 031372          .WORD T14
1050 014730
1051 014730 ENDMOD
1052 014730 BGNMOD INITCODE
1053 014730 RGNINIT
1054 014730
(3) 014730 012700 000340 SETPRI #340
(3) 014734 104041 MOV #340,R0
1055 014736 104051 MANUAL CSSPRI ;CHECK IF MANUAL INTERVENTION ALLOWED
(3) 014736 104051 FMT CSMAINT ;CHECK IF MANUAL INTERVENTION ALLOWED
1056 014740 103403 BCOMPLETE 1S ;YES - SKIP
(7) 014740 103403 RCS 1S
1057 014742 100014 014656 RIC #WITESTIDRSELTIHODALIGN,WISWIW ;CLEAR ALL MANUAL
; INTERVENTION FLAGS
1058
1059 014750 005037 002424 1S: CLR SSINDX ;CLEAR SUBROUTINE STACK INDEX
1060 014754 012700 000034 READEF #EF,PWR ;POWER FAILURE
(3) 014760 104050 FMT CSREFFG
1061 014762 103004 BCOMPLETE 4S ;NO, GO CHECK NEW PASS
(7) 014762 103004 HCC 4S
1062 014772 002012 003072 MOV LSUNIT,PWRFLG ;SET POWER FAIL FLAG
1063 014772 000531 BR PWCON ;GO SERVICE POWER FAIL
1064 014774 012700 000040 4S: READEF #EF,START ;CHECK IF START
(3) 014774 104050 MOV #EF,START,R0
1065 015002 103043 FMT CSREFFG
(2) 015002 103043 BCOMPLETE RESTART ;NO - SKIP
1066
1067 ; ON START INITIALIZE TO START AT FIRST DRIVE, CLEAR INTERNAL
1068 ; PASS COUNT AND ERROR COUNT.
1069 RSTR: CLR LSUNIT,DRVCNT ;SET UP UNIT COUNT
1070 MOV #R0,R1 ;CLEAR PASS NUMBER
1071 MOV #ERRCNT,R0
1072 CLR #R0,R1 ;GET A COUNT
1073 1S: CLP #R0 ;CLEAR ERROR COUNTER STORAGE AREA
1074 DEC R1
1075 RNE 1S ;LOOP TILL ALL CLEARED
1076 MOV #RRCNT-2,ERRPNT ;INIT ERRR POINTER
1077 MOV #-1,PSETNM ;SET PARAM SELECT TO INITIAL VALUE
1078 MOV #-1,HADONE ;PRESET HEAD ALIGN DONE FLAG
1079 MOV #HICVL,WISWIW ;TEST IF HI LIMIT SET
1080 RIT ;YES - SKIP
1081 MOV #R77,HILIMW ;ELSE INIT HILIMIT
1082 RIT #LOCVL,WISWIW ;TEST IF LO LIMIT SET
1083 RFE SS ;YES - SKIP
1084 CLR #R0,R1 ;ELSE CLEAR LO LIMIT
1085 RR SETDN
1086 RESTART:
1087 READEF #EF,RESTART ;CHECK IF RESTART
(3) 015120 104050 FMT CSREFFG
1088 015120 103734 BCOMPLETE RSTPT ;NO - SKIP
1089 ;
(3) 015122 012700 000036 CONTINUE: READEF #EF,CONTINUE ;TEST IF CONTINUE
MOV #EF,CONTINUE,R0
    
```

```

(3) 015126 104050          FMT CSREFFG
1090 015130 103452          BCOMPLETE PWCON
(3) 015130 103452 RCS PWCON
1091 ;
1092 015132 012700 000035 ; ON CONTINUE PICK UP UNIT LAST UNDER TEST
(3) 015132 104050 READEF #EF,NEW,R0 ;CHECK IF STARTING NEW PASS
1093 015136 104050 MOV #EF,NEW,R0
(2) 015140 103403 FMT CSREFFG
1094 015142 103403 BCOMPLETE PASNEW PASNEW
1095
1096 NXPAS:
1097 TST DRVCNT ;TEST IF ALL UNITS CHECKED
1098 RMC PASDN ;NO - SKIP
1099 PASNEW: INC #ERRCNT ;ELSE BUMP PASS COUNT
1100 MOV #RRCNT-2,ERRPNT ;INIT THE ERROR POINTER
1101 MOV LSUNIT,DRVCNT ;GET ALL DRIVES
1102 MOV #R0,PSETNM ;SET PARAM SELECT TO INITIAL
1103 SETDN: INC #R0 ;NEXT SET OF PARAMETERS
1104 DEC DRVCNT ;DOWN COUNT DRIVE TOTAL
1105 ADD #2,ERRPNT ;UPDATE THE ERROR POINTER
1106 MOV #R0,PSETNM,R0 ;SET UP TO GET PARAMETERS
1107 MOV #R0,R2
1108 GPHARD R0,R1
1109 FMT CSREFFG
(3) 015224 104042 EMT #R0,R1
1110 015224 106001 BCOMPLETE R0,R1
1111 015230 103406 RCS 7S ;SKIP IF GOOD PARAM
1112 015232 005737 TST PWRFLG ;RECENT POWER FAILURE
1113 015232 001741 RMC NXPAS ;NO
1114 015240 005337 DEC PWRFLG ;ACCOUNT FOR DRIVE
1115 015244 000736 BR NXPAS
1116 7S: MOV (R1)+(R2)+ ;STORE PARAMETERS CSR
MOV (R1)+(R2)+ ;VECTOR
TST (R1)+ ;BUMP PAST PRIORITY
MOV (R1)+(R2)+ ;DRIVE
1117
1118 PWCON: SETVEC RLVEC,#INTHLR,#340 ;SET UP VECTOR
(7) 015256 012746 000340 MOV #340,-(SP)
(5) 015262 012746 016064 MOV #INTHLR,-(SP)
(5) 015266 013746 002452 MOV #RLVEC,-(SP)
(3) 015276 104037 MOV #3,-(SP)
(2) 015300 062706 000010 ADD #10,SP
1119 SETPRI #0 ;SET PRIORITY
(3) 015304 012700 000000 MOV #0,R0
(3) 015310 104041 EMT CSW
1119 015312 013702 002450 MOV RLBAS,R2 ;SET RL BASE ADDRESS POINTER
1120
1121 ;
1122 ; CHECK IF DOING AUTO SIZE AND DROP DRIVE IF NOT READY AND
1123 ; ERROR SETS ON GET STATUS.
1124 TST PASNM ;TEST IF PASS 0
1125 015322 001135 RMC 2S ;NO - SKIP
1126 015324 032737 000020 014656 BIT #AUTOSZ,WISWIW ;TEST IF DOING AUTO SIZE
1127 015332 001531 BEQ 22S ;NO - SKIP
1128 ;CHECK IF UNTRUS ADDRESS IS THERE BEFORE WE CHECK DRIVE READY
    
```

```

CZRLCB.PT1 25-OCT-78 13:11
1130 0156333 005037 003070 CLR TRPFLG ;TRAP OCCURANCE
1131 0156334 012746 000340 SETVEC ERRVEC,#TRPHAN,#340 ;SET TRAP VECTOR
1132 0156335 012746 000340 MOV #340,(SP)
1133 0156336 012746 0160562 MOV TRPHAN,-(SP)
1134 0156337 012746 002652 MOV ERRVEC,-(SP)
1135 0156338 104036 000003 MOV #3,-(SP)
1136 0156339 062706 000010 EMT #10,SP
1137 0156340 005276 000000 ADD #10,SP
1138 0156341 005276 000000 TST RLCS(R2) ;ACCESS BUS
1139 0156342 001032 000000 TRPFLG ;TRAP OCCUR??
1140 0156343 013705 002454 MOV RLDV,R5 ;GET DRIVE NUMBER
1141 0156344 013705 000200 BIS #CRDYMSK,R5 ;INVESTIGATE FURTHER
1142 0156345 013705 000200 MOV R5,RLCS(R2) ;GET DRIVE NUMBER
1143 0156346 032762 000001 000000 MOV #RDVMSK,RLCS(R2) ;INSERT CONT READY
1144 0156347 001072 000000 BNE #0,S ;LOAD IN DRIVE NUMBER
1145 0156348 023762 000004 000004 MOV #RDVMSK,RLCS(R2) ;CHECK IF DRIVE IS READY
1146 0156349 023762 000003 BIC #0,S ;YES - GO TO TEST
1147 0156350 023762 000200 MOV #CETSTAT,PLDA(R2) ;ELSE INSERT GET STATUS
1148 0156351 023762 000200 BIS #4,R5 ;LOAD R5 WITH GET STATUS FUNCTION
1149 0156352 023762 000200 MOV #CRDYMSK,R5 ;CLEAR CONTROLLER READY
1150 0156353 010562 000000 MOV R5,RLCS(R2) ;LOAD CS REG
1151 0156354 012700 000004 WAITMS #4 ;WAIT 4 MS
1152 0156355 104036 000000 MOV #4,R0
1153 0156356 022762 002000 000000 BIT #OPTERR,RLCS(R2);TEST IF OPTI SET
1154 0156357 001452 000000 BFO #0,S ;NO - SKIP
1155 0156358 013700 002652 5S: MOV ERRVEC,R0
1156 0156359 104036 000000 EMT #CSCVEC
1157 0156360 012746 005640 PRINTF #FMT24,#DRVNAM
1158 0156361 012746 012747 MOV #DRVNAM,-(SP)
1159 0156362 012746 000002 MOV #2,-(SP)
1160 0156363 010500 000000 MOV SP,R0
1161 0156364 062706 000006 EMT #CSPNTF
1162 0156365 062706 000006 10S: PRINTF #FMT5,#BASADD,RLBAS,#DRVNAM,<R,RLDRV+1>
1163 0156366 065046 CLR -(SP)
1164 0156367 012746 002455 CLR RLDV+1,(SP)
1165 0156368 013746 005233 MOV #RDVMSK,(SP)
1166 0156369 013746 002450 MOV RLBAS,(SP)
1167 0156370 012746 005622 MOV #BASADD,-(SP)
1168 0156371 012746 000005 MOV #FMT5,-(SP)
1169 0156372 010600 MOV #5,-(SP)
1170 0156373 104017 MOV SP,R0
1171 0156374 062706 000014 EMT #CSPNTF
1172 0156375 012746 012127 PRINTF #FMT3,-(SP)
1173 0156376 010600 MOV #1,-(SP)
1174 0156377 104017 MOV SP,R0
1175 0156378 062706 000004 EMT #CSPNTF
1176 0156379 013700 003664 ADD #4,SP
1177 0156380 104053 DDDI# PSSTNM,R0 ;DROP DRIVE
1178 0156381 104053 EMT #PSSTNM,R0
1179 0156382 104053 DOCLN EMT #CDDDU

```

```

CZRLCB.PT1 25-OCT-78 13:12
1180 015606 104044 (3) 20S: EMT #CSDCLN
1181 015610 013700 002652 CLRVEC ERRVEC
1182 015610 104036 002652 MOV ERRVEC,R0
1183 015616 104036 000000 EMT #CSCVEC
1184 015616 104051 22S: MANUAL ;MANUAL INTERVENTION ALLOWED
1185 015616 104051 EMT #CSPNTF
1186 015620 103004 BCC #4,S ;NO
1187 015622 005737 003062 TST #PASHUM ;YES, CHECK PASS NUMBER
1188 015622 001001 BNE #2,S ;NO, FIRST PASS, NEED DRIVE UP
1189 015630 006437 ;FIRST PASS, PROGRAM WILL INSTRUCT USER
1190 015632 005737 003072 ;CHECK IF POWER FAILURE WAIT IS NEEDED
1191 015636 001434 4S: TST #PWRFLG ;NEEDED??
1192 015640 013705 002454 BEQ #S ;NO, SKIP
1193 015644 013705 002454 MOV #RLDV,R5 ;DRIVE SELECT
1194 015650 010562 000000 BIS #RDVMSK,R5 ;SET CRDS
1195 015650 012762 000000 MOV R5,RLCS(R2) ;SELECT DRIVE
1196 015650 032762 000074 000000 9S: MOV #R0,R1 ;SIXTY SECOND TIMER
1197 015666 00102A BNE #RDVMSK,RLCS(R2) ;DRIVE UP YFT
1198 015670 #10,S ;YES START TEST
1199 015670 012700 000012 WAITMS #10 ;WAIT A SECOND
1200 015674 104026 MOV #10,R0
1201 015675 005301 EMT #CWT4
1202 015700 001367 DEC R1 ;SIXTY GONE BY
1203 015700 BNE #0,S ;NO
1204 015702 012746 005673 PRINTF #FMT24,#NOPWR
1205 015705 012746 012747 MOV #NOPWR,-(SP)
1206 015711 012746 000002 MOV #FMT24,-(SP)
1207 015720 104017 MOV #2,-(SP)
1208 015722 104017 EMT #CSPNTF
1209 015722 006673 ADD #4,SP
1210 015730 BR #10,S
1211 015730 8S: ENDINIT
1212 015730 L10014: EMT #CSINIT
1213 015730 104011 ENMOD
1214 015732 RGMOD #CLNCODE
1215 015732 BGNCLN
1216 015733 012746 000340 SETVEC ERRVEC,#TRPHAN,#340
1217 015733 012746 0160562 MOV #340,(SP)
1218 015742 012746 002652 MOV TRPFLG,-(SP)
1219 015746 012746 000003 MOV #3,-(SP)

```

```

(3) 015752 104037      EMT      C$CVEC
(2) 015754 062706      ADD      #10,SP
1193 015760              SFTPRI  #7          ;SET PRIORITY TO 7
(3) 015760 012700 000007      MOV      #7,R0
(3) 015764 104041      EMT      C$SPRI
1194 015766 032762 000200 000000 2S:      BIT      #DRDVM$K,RLCS(R2) ;TEST IF CONTROLLER READY
1195 015778 053762 002454 000000      BEQ      3S          ;NO LOOP UNTIL READY
1197 016004 032762 000001 000000      BIT      #DRDVM$K,RLCS(R2) ;SET DRIVE NUMBER
1198 016012 001003      RNE      5S          ;TEST IF DRIVE BUSY
(3) 016014 012700 000003      WAITMS  #3          ;NO - SKIP
(3) 016020 104026      MOV      #3,RC      ;WAIT 300 MS
1200 016022 013700 002452      EMT      C$WFM
(3) 016026 104036      CLPVEC  RLVEC,R0    ;RELEASE VEC
1203 016030 005737 003072      EMT      C$CVEC
1204 016034 001402 003072      TST     PWRFLG      ;PWR FAIL SET
1204 016036 005337 003072      BEQ      7S          ;NO
(3) 016042 013700 002652      DEC     PWRFLG
1205 016046 104036      MOV      #ERRVEC,R0
(3) 016050      EMT      C$CVEC
(3) 016050      ENDCLN  L10015:
(3) 016050 104012      EMT      C$CLEAN
1207 016052      RGNDU  NOP
1208 016052 000240      ENDDU  L10016:
(3) 016054      EMT      C$DU
(3) 016054 104055      ENDMOD  RGNMOD  GLBSUR
1214 016056 005237 003070      TRPHAN: INC  TRPFLG
1215 016062 000002      RTI
1216 016064      BGN$RV INT$PL  INTERRUPT HANDLER. ABOPTS WAIT TIMEP AND STORES ALL RL11 REGS
(3) 016064      ABORTWAIT
1219 016064 104021      EMT      C$ABRT
(3) 016064 012237 002466      MOV      (R2)+,T,CS    ;STORE RL REGISTERS
1221 016072 012237 002470      MOV      (R2)+,T,DA
1222 016076 012237 002472      MOV      (R2)+,T,MP
1223 016082 011737 002474      MOV      (R2)+,T,MP
1224 016086 011737 002474      MOV      #1,DONE
1225 016114 013702 002450      MOV      RLBAS,R2    ;SET DONE FLAG
1226 016120      ENDSRV  L10017:
(3) 016120 000002      RTI
1227 016120      ;
1228 016120      ; ERROR LIMIT CHECKING ROUTINE
1229 016120      DROPS  DRIVE IF ERROR LIMIT EXCEEDED
1230 016122 027737 164532 014666  CKERLM: CMP  #ERRPOINT,ERLIMW ;TEST IF ERROR LIMIT EXCEEDED
    
```

```

1231 016130 002453      BLT     1S          ;NO - SKIP
1232 016132 104020      EMT      C$INLP      ;CHECK IF IN ERROR LOOP
1233 016134 103451      RCOMPLETE 1S       ;YES - SKIP
1234 016134      BCS     1S
(3) 016136 012746 011571      PRINTF  #FMT25,ERLIMW,#MEXERS
(3) 016142 013746 014666      MOV      #MEXERS,-(SP)
(3) 016146 012746 012754      MOV      #MEXERS,-(SP)
(3) 016150 010500      MOV      #MEXERS,-(SP)
(3) 016156 010500      MOV      #3,-(S6)
(3) 016160 104017      SP,PC
1235 016166 062706 000010      EMT      C$PNTF
(3) 016166 005046      ADD     #10,SP
(3) 016170 153716 002455      PRINTF  #FMT5,#RASADD,RLBAS,#DRVNAM,<R,RLDRV+1>
(3) 016174 005633      CLR     D1,SP
(3) 016178 013746 005633      BLSB   #D1,SP
(3) 016206 013746 002450      MOV      #DRVNAM,-(SP)
(3) 016210 013746 002450      MOV      RLBAS,-(SP)
(3) 016214 012746 000005      MOV      #RASADD,-(SP)
(3) 016218 012746 000005      MOV      #MEXERS,-(SP)
(3) 016222 104017      MOV      #5,-(S6)
(3) 016224 062706 000014      EMT      C$PNTF
1236 016230 012746 012127      PRINTF  #FMT3
(3) 016234 012746 000001      MOV      #MEXERS,-(SP)
(3) 016240 010500      MOV      #1,-(S6)
(3) 016244 104017      EMT      C$PNTF
1237 016250 062706 000004      ADD     #4,SP
(3) 016250 013700 003064      DDDU   #SETNMW,R0    ;DROP DRIVE
1238 016254 104053      EMT      C$DDDU
(3) 016256 104044      DOCLN
1239 016260 000207      EMT      C$DCLN      ;GO TO CLEAN UP
1240 016260      RTS
1241 016262 016237 000000 002466  READL: READ AND STORE ALL RL11 REGISTERS
1242 016270 016237 000002 002470      MOV      #ERRVEC,R0 ;GET CS REG
1243 016278 016237 000004 002472      MOV      RLBAS,R0    ;GET BUS ADDRESS REG
1244 016286 016237 000006 002474      MOV      #RDA(R2),T,DA ;GET DISK ADDRESS
1245 016312 000207      MOV      #RMP(R2),T,MP ;GET MULTI-PURPOSE REG
1246 016312      RTS
1247 016312      ;
1248 016314 011646      ; WAIT FOR CONTROLLER TIMEOUT TO FORCE INTERRUPT ROUTINE
1249 016316 005006 000002      CLP     2(SP),-1(SP) ;MAKE ROOM FOR ERROR POINTER
1250 016322 032762 000200 000000      BIT      #CDVMSK,RLCSR(R2) ;CLEAR FOR POINTER
1251 016330 001420      BEQ     4S          ;TEST IF CONTROLLER READY
1252 016336 005737 002430      JSR     PC,READL    ;READ ALL RL REGS
1253 016342 001433      BEQ     5S          ;NO - SKIP TO WAIT
1254 016348 012756 000002      BNE     1S          ;TEST IF INTERRUPT OCCURRED
1255 016354 002000 002468      MOV      #MTO$LOW,2(SP) ;GO SET NO INTERRUPT ERR FLAG
1256 016360 001403 002000      BIT      #OPIERP,T,CS ;TEST IF OPTI SET
1257 016362 012766 006032 000002      BEQ     2S          ;TEST IF OPTI SET
1258 016362      MOV      #M$PRES,2(SP) ;SET MESSAGE #OR NO DRIVE RESPONSE
    
```

```

1260 016370 000207 2$: RTS PC ;RETURN
1261 016372 000003 4$: #3 RO ;WAIT 300 MS FOR TIMEOUT
(3) 016376 104026 EMT #3,RO
1263 016408 000000 000000 BNE BCRDYSK,RLCS(R2) ;TEST IF READY NOW SET
1264 016410 004737 ;YES - SKIP
1265 016414 006115 000002 MOV PC,READRL ;READ RL REGS
1266 016424 002430 3$: #MCONHG,2(SP) ;SET MESSAGE FOR CONTROLLER HUNG
1267 016430 001345 BNE BR ;SKIP
1268 016432 004737 5$: #ONE ;ELSE CHECK IF INTERRUPT OCCURRED
1269 016434 006062 000002 TEST SET TO SLOW ;READ RL REGS
1270 016438 000751 MOV #NOINT,2(SP) ;ELSE SET NO INTERRUPT FLAG
1271 016444 005037 002426 ;GO TO RETURN
1272 016452 005037 002436 ;STINT: OPERATION AND TEST INITIALIZE ROUTINE
1273 016456 005037 002436 CLR OPFLAG ;CLEAR OPERATION FLAGS
1274 016458 005037 002436 CLR NOERCT ;RESET INHIBIT ERROR COUNTING
1275 016462 000207 000207 RTS WORECE ;RESET WORE COMPARE ERRORS
1276 016464 000207 002550 ;GSTATR: GET STATUS AND GET STATUS WITH RESET ROUTINE
1277 016470 000013 002550 MOV TEMP4,-(SP) ;STORE TEMP4
1278 016472 000442 ;GSTATC: #GETSTAT,DRSET,TEMP4 ;SET FOR RESET
1279 016474 000003 002550 GSTATC: #GSTATG ;STORE TEMP4
1280 016476 000404 ;GSTAT: MOV #GSTATG,TEMP4 ;SET FOR NO RESET
1281 016478 000250 GSTAT: MOV TEMP4,-(SP) ;STORE TEMP4
1282 016480 000250 GSTATC: MOV #GSTATG,TEMP4 ;SET FOR NO RESET
1283 016482 000250 GSTAT: MOV TEMP4,-(SP) ;STORE TEMP4
1284 016484 000250 GSTATC: MOV #GSTATG,TEMP4 ;SET FOR NO RESET
1285 016486 000250 GSTAT: MOV #GSTATG,TEMP4 ;STORE TEMP4
1286 016488 000250 GSTATC: MOV #GSTATG,TEMP4 ;SET FOR NO RESET
1287 016490 000250 GSTAT: MOV #GSTATG,TEMP4 ;STORE TEMP4
1288 016492 000250 GSTATC: MOV #GSTATG,TEMP4 ;SET FOR NO RESET
1289 016494 000250 GSTAT: MOV #GSTATG,TEMP4 ;STORE TEMP4
1290 016496 000250 GSTATC: MOV #GSTATG,TEMP4 ;SET FOR NO RESET
1291 016498 000250 GSTAT: MOV #GSTATG,TEMP4 ;STORE TEMP4
1292 016500 000250 GSTATC: MOV #GSTATG,TEMP4 ;SET FOR NO RESET
1293 016502 000250 GSTAT: MOV #GSTATG,TEMP4 ;STORE TEMP4
1294 016504 000250 GSTATC: MOV #GSTATG,TEMP4 ;SET FOR NO RESET
1295 016506 000250 GSTAT: MOV #GSTATG,TEMP4 ;STORE TEMP4
1296 016508 000250 GSTATC: MOV #GSTATG,TEMP4 ;SET FOR NO RESET
1297 016510 000250 GSTAT: MOV #GSTATG,TEMP4 ;STORE TEMP4
1298 016512 000250 GSTATC: MOV #GSTATG,TEMP4 ;SET FOR NO RESET
1299 016514 000250 GSTAT: MOV #GSTATG,TEMP4 ;STORE TEMP4
1300 016516 000250 GSTATC: MOV #GSTATG,TEMP4 ;SET FOR NO RESET
1301 016518 000250 GSTAT: MOV #GSTATG,TEMP4 ;STORE TEMP4
1302 016520 000250 GSTATC: MOV #GSTATG,TEMP4 ;SET FOR NO RESET
1303 016522 000250 GSTAT: MOV #GSTATG,TEMP4 ;STORE TEMP4
1304 016524 000250 GSTATC: MOV #GSTATG,TEMP4 ;SET FOR NO RESET
1305 016526 000250 GSTAT: MOV #GSTATG,TEMP4 ;STORE TEMP4
1306 016528 000250 GSTATC: MOV #GSTATG,TEMP4 ;SET FOR NO RESET
1307 016530 000250 GSTAT: MOV #GSTATG,TEMP4 ;STORE TEMP4
1308 016532 000250 GSTATC: MOV #GSTATG,TEMP4 ;SET FOR NO RESET
1309 016534 000250 GSTAT: MOV #GSTATG,TEMP4 ;STORE TEMP4
1310 016536 000250 GSTATC: MOV #GSTATG,TEMP4 ;SET FOR NO RESET
1311 016538 000250 GSTAT: MOV #GSTATG,TEMP4 ;STORE TEMP4
    
```

```

1312 016654 001441 BEQ 5$ ;NO - SKIP
1313 016656 052737 040000 002426 PIS #RELDT,OPFLAG ;ELSE SET WAIT FLAG
1314 016658 000435 BR 5$ ;NO SKIP
1315 016660 000435 51$: BNE #ORVERP,T.CS ;SKIP TO CLEAR
1316 016662 001031 ;TEST IF DRIVE ERROR NOW
1317 016664 001031 ;YES - SKIP TO CLEAR
1318 016666 001031 ;WAIT FOR DRIVE TO GET ERROR, RDY, OR HC
(3) 016670 012700 000001 MOV #1,PO
1319 016672 104026 EMT #1,PO
1320 016674 005301 DEC C$MTH ;DEC WAIT COUNTER
1321 016676 001344 BNE #ONE ;IF NOT DONE, LOOP
1322 016678 012703 011445 MOV #UNDEF,R3 ;MESSAGE FOR UNDEFINED STATE
1323 016680 10001,ERR1 ERRHPD ;ERR1
1324 016682 10001,ERR1 *WORD ERR1
1325 016684 10001,ERR1 *WORD ERR1
1326 016686 005737 002550 BR 14$ ;EXIT
1327 016688 001031 11$: TEST 5$ ;TEST IF SAVE REGISTERS
1328 016690 001031 BNE 5$ ;NO SKIP
1329 016692 001031 MOV #4,R1 ;SET SAVE COUNT
1330 016694 001031 MOV #L.MP+2,R3 ;SET ADDRESS OF FIRST SAVE
1331 016696 001031 MOV -(R3),-(SP) ;PUT REG ON STACK
1332 016698 001031 DEC R1 ;DEC COUNT
1333 016700 001031 BNE 8$ ;LOOP UNTIL ALL SAVED
1334 016702 001031 MOV #CETSTAT,L.DA ;SET FOR GET STATUS
1335 016704 001031 BR 2$ ;SKIP
1336 016706 001031 MOV TEMP4,L.DA ;INSERT PRESET FOR STATUS
1337 016708 001031 6$: CLR DONE ;CLEAR INTERRUPT FLAG
1338 016710 001031 MOV #RLDVL,CS ;SET UP TO GET STATUS
1339 016712 001031 BIC #RTSTAT,L.CS ;CLEAR FOR DRIVE 4 - 7 SPEC'D
1340 016714 001031 MOV #L.DA,RLDR(R2) ;LOAD RL REGS
1341 016716 001031 MOV #L.CS,RLCSR(R2) ;LOAD CS REG
1342 016718 001031 DEATHS #1,RO ;WAIT 100 US FOR INTERRUPT
1343 016720 001031 EMT #1,RO
1344 016722 001031 TEST C$MTH ;CHECK IF INTERRUPT OCCURRED
1345 016724 001031 BNE 4$ ;NO - SKIP
1346 016726 001031 MOV #MP,T.STAT ;STORE MP REGISTER
1347 016728 001031 BIC #C$STATSK,T.STAT ;CLEAR ALL BUT STATE
1348 016730 001031 #DRSET,L.DA ;TEST IF RESET WAS SPECIFIED
1349 016732 001031 BNE 3$ ;NO - SKIP TO EXIT
1350 016734 001031 BIC #RELDT,OPFLAG ;RESET RELOAD WAIT FLAG SET
1351 016736 001031 BEQ 12$ ;NO - SKIP
1352 016738 001031 MOV #60,R1 ;SET WAIT COUNT FOR 60 SECONDS
1353 016740 001031 BNE 13$ ;TEST IF DRIVE NOW READY
1354 016742 001031 WAITMS #1 ;CALL WAIT
1355 016744 001031 MOV #1,RO
1356 016746 001031 EMT #1,RO
1357 016748 001031 DEC R1 ;DEC COUNT
1358 016750 001031 BNE 13$ ;LOOP IF NOT 0
1359 016752 001031 JSR PC,GSTAT ;GET DRIVE STATUS
1360 016754 001031 MOV #R1,FAL,R3 ;ERROR RETURN
1361 016756 001031 ERRHPD 10003,ERR1 ;SET RESULT MESSAGE POINTER
    
```

```

(3) 017140 104443 TRAP TSERCODE
(5) 017142 023423 .WORD 10003
(5) 017144 013040 BR ERR1
1350 017148 906442 BR #4
1350 017150 012700 000012 12S: MOV #10, R0 ;GO TO EXIT
1350 017154 164927 EMT #4 ;WAIT FOR IMS
1350 017159 004927 016514 JSR PC, GSTAT ;GET DRIVE STATUS
1350 017164 032737 BIT #ANVERR, T.CS ;TEST IF ANY ERROR
1350 017172 001432 BEQ 3S ;NO - SKIP
1350 017174 032737 BIT #VSTAT, T.MP ;CHECK IF VOLUME CHECK RESET
1350 017179 001403 BEQ 7S ;YES - SKIP
1350 017204 012703 MOV #VCNRST, R3 ;SET REASON POINTER
1350 017210 000416 RP 2S ;EXIT
1350 017210 012737 BEQ #DRVERR, T.CS ;CHECK IF DRIVE ERROR
1370 017210 001404 BEQ 9S ;NO - SKIP
1371 017222 ERHRD 10004, ERR6
(3) 017222 104443 TRAP TSERCODE
(5) 017224 023423 .WORD 10004
1372 017230 000411 BR 14S
1373 017232 012703 MOV #UNKERR, R3 ;EXIT
1375 017240 004927 JSR PC, GSTAT ;SET REASON POINTER
1376 017244 012603 MOV (SP)+, R3 ;WAIT FOR INTERRUPT
1377 017246 ERHRD 10002, ERR1 ;STORE REASON POINTER FOR RETURN
(3) 017248 104443 TRAP TSERCODE
(5) 017252 013040 .WORD ERR1
1378 017254 005037 CLR ERPSWI ;CLEAR FOR ERROR RETURN
1380 017259 001067 TST TEMP4 ;TEST IF REGISTERS WERE SAVED
1381 017266 012703 MOV #L, CS, R3 ;SET POINTER TO RESTORE
1383 017272 012703 MOV #4, R1 ;SET REGISTER COUNT
1384 017306 015301 MOV (SP)+, (R3)+ ;RESTORE REG
1385 017302 001375 DEC R1 ;DEC COUNT
1386 017304 162737 BNE 20S ;LOOP UNTIL ALL ARE RESTORED
1388 017314 012603 MOV (SP)+, R1 ;REMOVE ENTRY FROM SUBROUT STACK
1389 017316 012603 MOV (SP)+, R3 ;RESTORE R3
1390 017320 012637 MOV (SP)+, TEMP4 ;RESTORE TEMP4
1392 017330 005737 TST ERPSWI ;TEST IF ERROR RETURN
1393 017332 063716 BEQ 99S ;YES - SKIP
1394 017336 000207 ADD ERPSWI, (SP) ;ADD IN ERROR RETURN
1396 017340 000000 PC ;SET ERROR RETURN ADDRESS
1397 017344 RTS
1497 017356 013703 SIMSEK: MOV R3, -(SP) ;STORE REGISTERS
1498 017358 013703 MOV #4, R3 ;SET SUBROUTINE INDEX
1499 017354 005723 TST (R3)+ ;BUMP IT FOR NEXT ENTRY
1500 017356 016663 MOV 2(SP), SUBSTK(R3) ;INSERT THIS CALL
1501 017364 162763 SUB #4, SUBSTK(R3) ;ADJUST IT TO CALLING LOCATION
    
```

```

1502 017372 010337 MOV R3, SSINDX ;STORE IT BACK
1503 017400 010446 MOV R4, -(SP)
1504 017402 012737 MOV #2, ERPSWI
1506 017410 004737 MOV PC, RDVCHK ;CHECK IF DRIVE READY
1508 017416 012704 MOV #L, CS, R4 ;GET POINTER TO L REGS
1509 017422 012714 MOV #SEEK, (R4) ;SET FOR SEEK
1510 017426 053714 RTS RLDV, (R4) ;INSERT DRIVE NUMBER
1511 017436 005024 BIT #BIT10, (R4)+ ;CLEAR FOR DRIVE 4 - 7 SPEC'D
1513 017440 013714 CLR R4 ;CLEAR BUS ADDRESS
1514 017444 012703 MOV #SDIF, (R4) ;LOAD DIFFERENCE
1516 017450 005303 ASL (P4) ;SET COUNT DIFFERENCE TO ALIGN
1517 017454 001375 DEC R3 ;ALIGN DIFFERENCE IN DA
1518 017456 005737 BNE 3S
1520 017462 014402 TST DESGN ;TEST IF SIGN SET
1521 017464 000004 BEQ 5S ;NO - SKIP
1522 017470 005737 BIT #DIRBIT, (R4) ;INSERT SIGN
1523 017474 001402 TST DESHD ;TEST IF HEAD 0
1524 017478 002714 BEQ 7S ;YES - SKIP
1525 017502 005737 RTS #RSEL, (P4) ;INSERT HEAD BIT
1526 017506 005477 RTS #RSET0, (R4)+ ;INSERT MARKER BIT
1528 017512 012701 MOV #10, R1 ;CLEAR INTERRUPT FLAG
1529 017516 014402 MOV -(R4), RLDA(R2) ;SET WAIT COUNT FOR 800US
1530 017522 014402 MOV -(R4), RLBA(R2) ;LOAD RL REGISTERS
1531 017526 001375 MOV -(R4), RLCS(R2)
1532 017532 005737 TST DONE ;CHECK IF INTERRUPTED
1533 017536 001375 BNE 65S ;YES - SKIP
1534 017540 013714 DEC R1 ;DEC WAIT COUNT
1535 017544 001404 BEQ 13S ;IF 0 - SKIP
1536 017544 WAITUS #1
1537 017548 MOV #1, R0
1538 017552 EMT 10 ;GO CHECK DONE
1539 017554 004737 JSP PC, WAITIN ;GO WAIT FOR TIMEOUT
1540 017560 012603 MOV (SP)+, R3 ;GET RESULT MESSAGE POINTER
1541 017562 ERHRD 10006, ERR1
(3) 017564 104443 TRAP TSERCODE
(5) 017566 023433 .WORD 10011
1549 017574 005037 CLR ERPSWI ;CLEAR FOR ERROR ERROR RETURN
1551 017574 162737 SUB #2, SSINDX ;REMOVE ENTRY FROM SUBROUT STACK
1552 017602 012604 MOV (SP)+, R4 ;RESTORE REGS
1553 017604 012604 MOV (SP)+, R0
1554 017606 012603 MOV (SP)+, R3
1555 017610 005737 TST ERPSWI ;TEST IF ERROR RETURN
1556 017614 001375 BEQ 99S ;YES - SKIP
1557 017622 000207 ADD ERPSWI, (SP) ;ADD IN ERROR RETURN
1558 017624 000207 PC ;SET ERROR RETURN ADDRESS
1559 017630 000000 MOV #4(SP), (SP)
1560 017630 RTS
1562 ;
1563 ;
1564 ;
1565 ;
1566 ;
1567 ;
1568 ;
1569 ;
1570 ;
1571 ;
1572 ;
1573 ;
1574 ;
1575 ;
1576 ;
1577 ;
1578 ;
1579 ;
1580 ;
1581 ;
1582 ;
1583 ;
1584 ;
1585 ;
1586 ;
1587 ;
1588 ;
1589 ;
1590 ;
1591 ;
1592 ;
1593 ;
1594 ;
1595 ;
1596 ;
1597 ;
1598 ;
1599 ;
1600 ;
1601 ;
1602 ;
1603 ;
1604 ;
1605 ;
1606 ;
1607 ;
1608 ;
1609 ;
1610 ;
1611 ;
1612 ;
1613 ;
1614 ;
1615 ;
1616 ;
1617 ;
1618 ;
1619 ;
1620 ;
1621 ;
1622 ;
1623 ;
1624 ;
1625 ;
1626 ;
1627 ;
1628 ;
1629 ;
1630 ;
1631 ;
1632 ;
1633 ;
1634 ;
1635 ;
1636 ;
1637 ;
1638 ;
1639 ;
1640 ;
1641 ;
1642 ;
1643 ;
1644 ;
1645 ;
1646 ;
1647 ;
1648 ;
1649 ;
1650 ;
1651 ;
1652 ;
1653 ;
1654 ;
1655 ;
1656 ;
1657 ;
1658 ;
1659 ;
1660 ;
1661 ;
1662 ;
1663 ;
1664 ;
1665 ;
1666 ;
1667 ;
1668 ;
1669 ;
1670 ;
1671 ;
1672 ;
1673 ;
1674 ;
1675 ;
1676 ;
1677 ;
1678 ;
1679 ;
1680 ;
1681 ;
1682 ;
1683 ;
1684 ;
1685 ;
1686 ;
1687 ;
1688 ;
1689 ;
1690 ;
1691 ;
1692 ;
1693 ;
1694 ;
1695 ;
1696 ;
1697 ;
1698 ;
1699 ;
1700 ;
1701 ;
1702 ;
1703 ;
1704 ;
1705 ;
1706 ;
1707 ;
1708 ;
1709 ;
1710 ;
1711 ;
1712 ;
1713 ;
1714 ;
1715 ;
1716 ;
1717 ;
1718 ;
1719 ;
1720 ;
1721 ;
1722 ;
1723 ;
1724 ;
1725 ;
1726 ;
1727 ;
1728 ;
1729 ;
1730 ;
1731 ;
1732 ;
1733 ;
1734 ;
1735 ;
1736 ;
1737 ;
1738 ;
1739 ;
1740 ;
1741 ;
1742 ;
1743 ;
1744 ;
1745 ;
1746 ;
1747 ;
1748 ;
1749 ;
1750 ;
1751 ;
1752 ;
1753 ;
1754 ;
1755 ;
1756 ;
1757 ;
1758 ;
1759 ;
1760 ;
1761 ;
1762 ;
1763 ;
1764 ;
1765 ;
1766 ;
1767 ;
1768 ;
1769 ;
1770 ;
1771 ;
1772 ;
1773 ;
1774 ;
1775 ;
1776 ;
1777 ;
1778 ;
1779 ;
1780 ;
1781 ;
1782 ;
1783 ;
1784 ;
1785 ;
1786 ;
1787 ;
1788 ;
1789 ;
1790 ;
1791 ;
1792 ;
1793 ;
1794 ;
1795 ;
1796 ;
1797 ;
1798 ;
1799 ;
1800 ;
    
```

DRIVE READY TEST ROUTINE. CHECKS DEIVE IS READY. IF NOT, WAIT


```

1768 020346 012704 012006 MOV #C5SEC,R4 ;SET CONDITION AFTER 5 SECONDS
1769 020347 10014 ERR5
(3) 020352 104443 ERHRD TRAP TSECCODE
(5) 020354 023436 TRAP TSECCODE
1770 020360 008424 .WORD ERR5
1771 020362 005737 002466 10S: BIT T.CS ;EXIT
1772 020365 100004 RPL T.CS ;CHECK FOR ANY ERRORS
(3) 020370 104443 ERHRD TRAP TSECCODE ERR6 ;NO - SKIP
(5) 020372 023440 .WORD ERR6 ;REPORT ALL ERRORS
(5) 020374 013342 .WORD ERR6
1774 020376 008424 .WORD ERR6
1775 020380 012701 002476 12S: MOV #HDWRD2,R1 ;GET POINTER
1776 020404 016221 000006 RIMP(R2),(R1)+ ;STORE LAST TWO HEADER WORDS
1777 020410 016221 000006 MOV RIMP(R2),(R1)+
1778 020414 008424 .WORD ERR6
1779 020416 004737 016314 14S: BP ;EXIT
1780 020422 012603 JSR PC,WAITN ;WAIT FOR INTERRUPT
1781 020424 104443 MOV (SP)+,R3 ;GET RESULTS
1782 020426 023437 ERHRD TRAP TSECCODE ERR1 ;REPORT
(3) 020428 023437 TRAP TSECCODE
(5) 020430 013040 .WORD ERR1
1784 020432 005337 60S: CLR ERRSWI ;CLEAR FOR ERROR RETURN
1785 020442 001007 65S: BNE ;TEST IF REGISTERS WERE SAVED
1786 020444 012703 002456 MOV #L.CS,R3 ;SET POINTER TO RESTORE REGS
1787 020446 012703 000004 MOV #4,R1 ;SET COUNT
1788 020456 005301 20S: MOV (SP)+,(R3)+ ;RESTORE REGISTER
1789 020460 001375 DEC ;DEC COUNT
1790 020462 162737 000002 002424 22S: SUP #2,SSINDX ;LOOP UNTIL ALL ARE RESTORED
1791 020464 012601 MOV (SP)+,R1 ;REMOVE ENTRY FROM SUBROUT STACK
1792 020472 012601 MOV (SP)+,R1 ;RESTORE REGS
1793 020474 012600 MOV (SP)+,R0
1794 020476 012600 MOV (SP)+,R3
1795 020478 005301 TRAP TSECCODE
1796 020504 001403 BEQ ERRSWI ;TEST IF ERROR RETURN
1797 020506 063716 002440 ADD IN,ERRSWI,(SP) ;YES - SKIP
1798 020514 006297 000000 RTS PC ;ADD IN ERROR RETURN
1800 020520 006207 99S: MOV #0,(SP),(SP) ;SET ERROR RETURN ADDRESS
1801 020522 012705 002474 ;POSHW1: POSITION HEAD BIT FROM HEADER OR MULTIPURPOSE REGISTER TO LSB.
1802 020526 000402 MOV #HDWRD1,R5 ;START FOR POSITION HD BIT IN WD 1
1803 020530 012705 002474 POSHSB: MOV T.MP,R5 ;START FOR POSITION HD BIT IN MP
1804 020534 012705 POSHDD: MOV #1,(SP) ;STORE R1
1805 020536 012705 BIT #6,STAT,R5 ;CLEAR ALL HD BIT SEL BIT
1806 020542 012701 177677 MOV #6,R1 ;SET SHIFT COUNT
1807 020544 006205 1S: ASP #5,R1 ;SHIFT FOR RIGHT JUSTIFY
1808 020546 005301 DEC ;DEC COUNT
1809 020548 005301 BNE
1810 020554 012601 MOV (SP)+,R1 ;RESTORE R1
1811 020556 006207 RTS ;RETURN
    
```

```

1890 ; WAIT FOR READY ROUTINE DURATION OF WAIT PASSED TO THE ROUTINE
1891 ; FLOW THE CALLING ROUTINE IN R1.
1892 020560 012346 002424 ;RDYWAIT: MOV #R3,(SP) ;STORE R3
1893 020562 005737 6S: MOV SSINDX,R3 ;GET SUBROUTINE INDEX
1894 020564 016663 000002 002260 TST (R3)+ ;RUMP IT FOR NEXT ENTRY
1895 020570 162763 000004 002260 SUR #4,SUBSTK(R3) ;INSERT THIS CALL
1896 020604 010337 002424 MOV #R3,SSINDX ;STORE IT BACK
1897 020610 010446 MOV #0,(SP)
1898 020614 010446 MOV #0,(SP)
1899 020616 010446 MOV #0,(SP)
1900 020618 010446 MOV #0,(SP)
1901 020622 012737 000002 002440 5S: MOV #2,ERRSWI ;SET FOR NO ERROR RETURN
1902 020624 012737 016514 JSR PC,GSTAT ;GET DRIVE STATUS
1903 020630 020767 BIT #DRDVMASK,T.CS ;CHECK IF READY
1904 020632 032737 000001 002466 RNE #0 ;YES - SKIP
1905 020640 001352 DEC ;DEC WAIT COUNT
1906 020642 005301 BNE ;SKIP IF 0
1907 020644 001404 WAITUS #1
1908 020646 012700 MOV #1,R0
1909 020648 104927 EMT CS&TM
1910 020654 006763 MOV #HDWRD1,R3 ;SET NAME MESSAGE PTR
1911 020656 012703 7S: ERHRD TRAP TSECCODE ERR3 ;REPORT READY ERROR
(3) 020662 104443 TRAP TSECCODE
(5) 020664 023444 .WORD ERR3
(5) 020666 013154 .WORD ERR3
1912 020670 012701 000062 MOV #5C,R1 ;SET WAIT COUNT FOR 5 SECONDS
1913 020674 004737 016514 JSP PC,GSTAT ;GET DRIVE STATUS
1914 020700 020762 6S: BIT #DRDVMASK,T.CS ;TEST IF DRIVE READY
1915 020702 032737 000001 002466 RNE #0 ;YES - SKIP
1916 020710 001013 WAITMS #1 ;WAIT 100 MS
1917 020712 012700 MOV #1,R0
(3) 020716 104926 EMT CS&TM
(5) 020720 005301 DEC ;DEC WAIT COUNT
1918 020724 012704 012006 BNE #6SSEC,R4 ;LOOP UNTIL TIME DONE
1919 020730 104443 ERHRD TRAP TSECCODE ERR5 ;SET CONDITION AFTER 5 SECDS
(3) 020732 023440 TRAP TSECCODE
(5) 020734 013342 .WORD ERR5
1922 020736 000407 .WORD ERR5
1923 020740 032737 100000 002466 8S: BP ;EXIT
1924 020742 012603 REG #ANVRR,T.CS ;TEST IF ANY ERROR SET
(3) 020744 012603 ERHRD TRAP TSECCODE ERR6 ;NO - SKIP
(5) 020746 023446 TRAP TSECCODE ;REPORT ALL ERRORS
(5) 020748 013040 .WORD ERR6
1926 020756 005337 11S: CLR ERRSWI ;CLEAR FOR DOUBLE ERROR REPORT
1927 020762 005301 10S: SUP #2,SSINDX ;REMOVE ENTRY FROM SUBROUT STACK
1928 020766 162737 000002 002424 9S: MOV (SP)+,R4 ;RESTORE REGISTERS
1929 020774 012604 MOV (SP)+,R1
1930 020776 012601 MOV (SP)+,R0
1931 020778 012600 MOV (SP)+,R3
1932 021002 012603 MOV (SP)+,R3 ;RESTORE R3
    
```

```

1933 021064 005737 TST ERRSWI ;TEST IF ERROR RETURN
1934 021010 001403 REO 99 ;YES - SKIP
1935 021012 003716 ADD ERRSWI,(SP) ;ADD IN ERROR RETURN
1936 021020 000207 RTS PC ;SET ERROR RETURN ADDRESS
1937 021024 000207 99S: MOV R3,(SP),(SP)
1938 021024 000207 RTS PC
;
; GET POSITION ROUTINE. READS A HEADER FROM CURRENT CYLINDER
; (WHERE IT IS PRESENTLY POSITIONED) AND STORES CYLINDER
; NUMBER IN CURCYL.
1943 021022 010346 GETPOS: MOV R3,(SP) ;STORE REGISTERS
1944 021030 013703 MOV SSINDX,R3 ;GET SUBROUTINE INDEX
1945 021034 005737 TST (R3)+ ;BUMP IT FOR NEXT ENTRY
1946 021036 016653 MOV 2(SP),SUBSTK(R3) ;INSERT THIS CALL
1947 021044 162763 SUP #4,SUBSTK(R3) ;ADJUST IT TO CALLING LOCATION
1948 021052 010337 MOV R3,SSINDX ;STORE IT BACK
1949 021056 010046 MOV PC,-(SP)
1950 021062 004737 JSP PC,XRDHD ;DO READ HEADER
1951 021066 021116 65S: MOV HDWRD1,R3 ;GET HEADER WORD
1952 021074 042703 BIT #CHCYL,R3 ;CLEAR ALL BUT CYLINDER
1953 021100 017705 MOV #7,R5 ;SET SHIFT COUNT
1954 021104 006203 ASK R3 ;SHIFT TO RIGHT JUSTIFY
1955 021110 001375 BFC R5
1956 021112 010337 MOV R3,CURCYL ;STORE AS CURRENT CYLINDER
1957 021116 162737 SUB #2,SSINDX ;REMOVE ENTRY FROM SUBROUT STACK
1958 021124 012600 MOV (SP)+,R5 ;RESTORE REGISTERS
1959 021126 012600 MOV (SP)+,R4
1960 021130 012600 MOV (SP)+,R3
1961 021132 005737 TST ERRSWI ;TEST IF ERROR RETURN
1962 021140 003716 ADD ERRSWI,(SP) ;YES - SKIP
1963 021144 000207 RTS PC ;ADD IN ERROR RETURN
1964 021146 000000 99S: MOV R3,(SP),(SP) ;SET ERROR RETURN ADDRESS
1965 021152 000207 RTS PC
;
; READ ALL HEADERS ROUTINE. 40 HEADERS ARE READ AND STORED
; IN IBOFF.
1970 021154 010346 RDLHD: MOV R3,-(SP) ;STORE REGISTERS
1971 021156 013703 MOV SSINDX,R3 ;GET SUBROUTINE INDEX
1972 021162 005737 TST (R3)+ ;BUMP IT FOR NEXT ENTRY
1973 021164 166653 MOV 2(SP),SUBSTK(R3) ;INSERT THIS CALL
1974 021172 010337 MOV #4,SUBSTK(R3) ;ADJUST IT TO CALLING LOCATION
1975 021174 010337 MOV R3,SSINDX ;STORE IT BACK
1976 021204 010046 MOV R0,-(SP)
1977 021206 010046 MOV R1,-(SP)
1978 021210 010046 MOV R2,-(SP)
1979 021214 012737 MOV R3,-ERRSWI ;SET FOR NO ERROR RETURN
1980 021220 012737 MOV #40,R1 ;SET HEADER COUNT
1981 021224 010000 MOV #HDR40,OPFLAG ;SET 40 HDR OP FLAG
1982 021228 022737 BIS #IBOFF,R3 ;SET POINTER TO STORE HDRS
1983 021232 015774 MOV #R3,SSINDX ;SET BASE ADDRESS
1984 021236 000000 ADD #R1,R4 ;WAKE UP POINT TO MP REG
1985 021242 000000 ADD #R1,R4 ;LOAD FOR READ HEADER, NO INTERRUPT
1986 021246 000010 MOV #10,LCS
    
```

```

2017 021254 053737 002454 002456 RIS RLDW,LCS ;INSERT DRIVE NUMBER
2018 021256 005037 002460 002456 PIC #BIT10,LCS ;CLEAR FOR DRIVE 4 - 7 SPEC'D
2019 021260 005037 CLR L,R4 ;CLEAR BA
2020 021274 005037 CLR L,R4 ;CLEAR DA
2021 021300 005737 TST DRSHD ;TEST IF HEAD 0
2022 021304 005737 BREQ 35 ;YES - SKIP
2023 021314 013762 MOV #WSEL,L,DA ;ELSE INSERT HEAD 0
2024 021322 013762 MOV L,DA,RLDA(R2) ;LOAD RLDA REG
2025 021326 000000 BIT #RDMASK,RLCS(R2) ;LOAD RLDA REG
2026 021330 000000 BNE 6S ;TEST IF CONTROLLER READY
2027 021340 004737 JSP PC,PDVCHK ;YES - SKIP
2028 021344 021456 JCS PC,PDVCHK ;ELSE CHECK READY
2029 021348 013762 MOV L,CS,RLCS(R2) ;LOAD RLCS REG
2030 021352 032762 MOV #7777,R0 ;SET COUNT FOR WAIT
2031 021360 032762 BIT #CRDYSK,RLCS(R2) ;CHECK THAT OPERATION COMPLETED
2032 021366 000000 BNE 7S ;YES - SKIP
2033 021370 000300 DEC R0 ;DEC COUNT
2034 021374 004737 BNE 7S ;SKIP IF NOT YET 0
2035 021378 004737 JSP PC,READR1 ;ELSE GET ALL REGISTERS
2036 021400 004737 JSP PC,WAITN ;ELSE WAIT FOR TIMEOUT
2037 021404 016314 MOV (SP)+,R3 ;GET WAIT FOR TIMEOUT
2038 021408 012603 MOV #16025,ERRR1 ;GET RESULT MESSAGE POINTER
2039 (3) 021406 104443 ERPHRD ;ERRR1
2040 (5) 021410 023453 TRAP ;ERRR1
2041 (7) 021414 003037 *WORD ;ERRR1
2042 (9) 021418 003037 *WORD ;ERRR1
2043 021420 000416 CLR ERRSWI ;CLEAR FOR ERROR RETURN
2044 021422 005737 RTT 6S ;SET FOR ANY ERRORS
2045 021424 000000 BPL 15S ;NO - SKIP
2046 (3) 021430 104443 ERPHRD ;ERRR1
2047 (5) 021434 023453 TRAP ;ERRR1
2048 (7) 021438 003037 *WORD ;ERRR1
2049 (9) 021442 003037 *WORD ;ERRR1
2050 021444 000405 CLR ERRSWI ;CLEAR FOR ERROR RETURN
2051 021448 011443 MOV (R3),(R3)+ ;STORE HEADER WORDS
2052 021452 011443 MOV (R4),(R3)+
2053 021456 005301 MOV (R4),(R3)+
2054 021460 005301 DEC R1 ;DEC HEADER COUNT
2055 021464 011443 SUB #2,SSINDX ;REMOVE ENTRY FROM SUBROUT STACK
2056 021468 012600 MOV (SP)+,R4 ;RESTORE REGISTERS
2057 021472 012600 MOV (SP)+,R3
2058 021476 005737 TST ERRSWI ;TEST IF ERROR RETURN
2059 021480 003716 ADD ERRSWI,(SP) ;YES - SKIP
2060 021484 000207 RTS PC ;ADD IN ERROR RETURN
2061 021488 000000 99S: MOV R3,(SP),(SP) ;SET ERROR RETURN ADDRESS
2062 021492 000207 RTS PC
;
; REPORT OPERATION ROUTINE. PRINTS SUBROUTINE TRACE SEQUENCE AND
; OPERATION BEING PERFORMED PORTION OF ALL
    
```

```
2294 021516 010446  
2295 021520 005737 002424  
2296 021524 001433  
2297 021526 012704 000002  
2298 021532 012746 010534  
2299 021536 012746 012327  
2300 021542 012746 000002  
2301 021546 010600  
2302 021550 104014  
2303 021552 062766 000006  
2304 021556 016446 002260  
2305 021558 012746 012502  
2306 021556 012746 000002  
2307 021572 010600  
2308 021574 104014  
2309 021576 062766 000006  
2310 021602 062766 000002  
2311 021606 020437 002424  
2312 021612 003761  
2313 021614 012746 006217  
2314 021620 013746 002434  
2315 021624 012746 012132  
2316 021630 012746 000003  
2317 021634 010600  
2318 021636 104014  
2319 021638 062766 000010  
2320 021644 030000 002426  
2321 021652 013701 002456  
2322 021656 027011 017741  
2323 021658 000006  
2324 021666 001003  
2325 021670 052737 010090  
2326 021676 022701 000312  
2327 021702 011603  
2328 021704 052737 020000  
2329 021712 027011 000314  
2330 021716 001003  
2331 021720 052737 020000  
2332 021726 016146 002122  
2333 021732 012746 005143  
2334 021736 010600 000003  
2335 021742 012746 000003  
2336 021746 010600  
2337 021750 104014  
2338 021756 000010  
2339 021762 001067 000004  
2340 021764 032737 000010  
2341 021772 027011 002462  
2342 021774 012701 000016  
2343 022000 006436  
RPTOP: ERPOP MESSAGES  
MOV #4,-(SP)  
TST SSINDX ;TEST SUBROUTINE INDEX 0  
BFO 15 ;SKIP IF 0  
MOV #2,R4 ;SET INDEXXED TO FIRST ENTRY  
PRINTR #FMT0,#SEGNES ;PRINT "SUBROUTINE CALL SEQ"  
MOV #SSGNES,-(SP)  
MOV #FMT9,-(SP)  
MOV #2,-(SP)  
MOV #2,-(SP)  
EWT CSFNTR  
ADD #6,SP  
PRINTR #FMT16,SUBSTK(R4) ;PRINT CALLING LOCATION  
MOV SUBSTK(R4)-(SP)  
MOV #FMT16,-(SP)  
MOV #2,-(SP)  
MOV #2,-(SP)  
EWT CSFNTR  
ADD #6,SP  
ADD #2,R4 ;RUMP INDEX  
MOV #2,R4 ;CHECK IF ALL PRINTED  
CMP R4,SSINDX ;LOOP IF NOT ALL PRINTED YET  
BLF 35 ;PRINT ERROR HEADER  
PRINTB #FMT4,ERHEAD,#TSTLAR  
MOV #TSTLAR,-(SP)  
MOV #EPHEAD,-(SP)  
MOV #FMT4,-(SP)  
MOV #3,-(SP)  
MOV #3,-(SP)  
MOV #3,-(SP)  
EWT CSFNTR  
ADD #1,SP  
;SEEKOP,OPFLAG ;CLEAR SK & RD OR WRT FLAG  
MOV L,CS,R1 ;RET COMMAND EXECUTED  
BIC #17741,R1 ;STRIP ALL RUT FUNCTION CODE  
MOV #R1 ;TEST IF SEEK OPERATION  
RNE #2 ;NO - SKIP  
RIS #SEEKOP,OPFLAG ;ELSE SET SEEK FLAG  
CMP #1,R1 ;TEST IF WRITE  
RNE #2 ;NO - SKIP  
;POPWOP,OPFLAG ;SET PD OR WRT FLAG  
CMP #14,R1 ;TEST IF READ  
RNE #2 ;NO - SKIP  
;RWOP,OPFLAG ;SET RD OR WRT FLAG  
PRINTR #FMT7,#OPPER,OPMSG(R1) ;PRINT OPERATION  
MOV OPMSG(R1)-(SP)  
MOV #MOPER,-(SP)  
MOV #FMT7,-(SP)  
MOV #3,-(SP)  
MOV #3,-(SP)  
MOV #3,-(SP)  
EWT CSFNTR  
ADD #10,SP  
CMP #R4,#4 ;CHECK IF GET STATUS  
RNE #4 ;NO - SKIP  
BIT #0,RSET,L,DA ;TEST IF RESET INCLUDED  
BFO #5 ;NO - SKIP  
MOV #6,R1 ;SET TO PRINT WITH RESET  
BR 95
```

```
2324 022002 032737 007777 002426 45: BIT #COMPPOP,OPFLAG ;TEST IF ANY OTHER OPERATION  
2325 022006 013701 002426 55: BEQ 15 ;NO - SKIP  
2326 022012 013704 006020 MOV OPFLAG,R4 ;SET UP TO DETERMINE WHICH ONE  
2327 022016 012701 000001 MOV #2,R1 ;PRESET THE POINTER  
2328 022022 032704 000001 BIT #R100,R4 ;CHECK THE BIT  
2329 022026 010003 BNE #6 ;IF SET - SKIP  
2330 022030 005143 TST (R1)+ ;RUMP POINTER  
2331 022032 006244 ASP R4  
2332 022034 000772 BR 55  
2333 022036 016146 002122 65: PRINTB #FMT2,OPMSG(R1)  
2334 022042 012746 012124 MOV OPMSG(R1)-(SP)  
2335 022046 012746 000002 MOV #FMT2,-(SP)  
2336 022052 010600 MOV #2,-(SP)  
2337 022054 104014 MOV #R0  
2338 022056 062766 000006 EWT CSFNTR  
2339 022062 032737 100000 85: ADD #6,SP  
2340 022070 001415 BIT #R40,OPFLAG ;TEST IF 40 HEADER OPERATION  
2341 022072 012701 000050 BEQ 105 ;NO - SKIP  
2342 022076 016146 002122 95: PRINTR #FMT2,OPMSG(R1) ;ELSE PRINT IT  
2343 022082 012746 000002 MOV OPMSG(R1)-(SP)  
2344 022086 010600 MOV #FMT2,-(SP)  
2345 022092 104014 MOV #6,R4  
2346 022094 062766 000006 EWT CSFNTR  
2347 022096 104014 ADD #6,SP  
2348 022102 013746 002534 105: BIT #SEEKOP,OPFLAG ;SKIP  
2349 022106 012746 012124 BEQ 15 ;TEST IF SEEK  
2350 022112 010600 PRINTR #FMT13,#FBWD,OLDCYL,#DIFWD,DESDIF,#SGNWD,DESSGN,#HDWD,DESHD  
2351 022114 010600 MOV #HDWD,-(SP)  
2352 022116 013746 002532 MOV #DIFWD,-(SP)  
2353 022118 013746 010470 MOV #DESSGN,-(SP)  
2354 022120 013746 012530 MOV #DESDIF,-(SP)  
2355 022122 012746 010462 MOV #IFWD,-(SP)  
2356 022124 013746 002522 MOV #OLDCYL,-(SP)  
2357 022126 013746 010530 MOV #FBWD,-(SP)  
2358 022128 012746 000011 MOV #FMT13,-(SP)  
2359 022130 010600 MOV #11,-(SP)  
2360 022132 104014 MOV #R0  
2361 022134 000024 EWT CSFNTR  
2362 022136 032737 002000 155: ADD #6,SP  
2363 022142 001424 BIT #POPWOP,OPFLAG ;TEST IF READ OR WRITE SET  
2364 022146 013746 002536 PRINTR #FMT22,#CVLWD,CURCYL,#HDWD,DESHD,#SECWD,DESSEC  
2365 022150 010501 BEQ 175 ;NO - SKIP  
2366 022152 013746 002534 MOV #DESSEC,-(SP)  
2367 022154 013746 010475 MOV #SECWD,-(SP)  
2368 022156 013746 010530 MOV #DESHD,-(SP)  
2369 022158 013746 010506 MOV #HDWD,-(SP)  
2370 022160 012746 010566 MOV #CURCYL,-(SP)  
2371 022162 012746 012677 MOV #CVLWD,-(SP)  
2372 022164 012746 000007 MOV #FMT22,-(SP)  
2373 022166 010600 MOV #7,-(SP)  
2374 022168 104014 MOV #R0  
2375 022170 EWT CSFNTR
```

```

CZRLCB.PT1 25-OCT-78 13:12
(4) 022270 062706 000020
2345 022274 004737 022746
2346 022300 012604
2347 022302 000207
2348
2349
2350
2351 022330 017146
2352 022336 010346
2353 022340 010346
2354 022344 012701 002504
2355 022348 012103
2356 022352 011146
(0) 022356 011146
(1) 022360 005157
(2) 022364 012115
(3) 022368 000003
(4) 022372 010600
(5) 022376 104514
(6) 022380 062706
(7) 022384 021127 000010
(8) 022388 001453 011405
(9) 022392 012704 012334
2359 022396 022127 011400
2360 022400 012704 012342
2361 022404 005303
2362 022408 001442
2363
2364
2365 022412 012146
(0) 022416 011647
(1) 022420 010346
(2) 022424 010600
(3) 022428 104014
(4) 022432 062706 000010
2366 022436 062706 000010
(0) 022440 012146
(1) 022444 011653
(2) 022448 010346
(3) 022452 010600
(4) 022456 104014
(5) 022460 062706 000010
2367 022464 062706 000010
2368 022468 001413
2369
2370 022472 012146
(0) 022476 011660
(1) 022480 011110
(2) 022484 011110
(3) 022488 000003
(4) 022492 010600
(5) 022496 104014
(6) 022500 062706 000010
2371 022504 012603

```

```

175: ADD #20,SP
JSP PC,LRPARM ;CLEAR PARAM TABLE
MOV (SP)+,R4 ;RESTORE R4
RTS PC

;
REPORT REASON ROUTINE
PRINTS REASON PORTION FOR ALL ERROR REPORTS.
RPTRES: MOV R1,-(SP) ;STORE R1
MOV R2,-(SP) ;STORE R2
MOV R3,-(SP) ;STORE R3
MOV #2,SPARM,R1 ;GET PART OF PARAM
MOV (R1)+,R3 ;GET NUMBER OF PARAM
PRINTB #FMT1,LRSLT,(R1) ;PRINT NAME
MOV SP,RO
EVT C$PNTB
ADD #10,SP
BEQ #6,SP ;TEST IF MESSAGE IS NO DRV STATUS
CMP (R1),#MNDRST ;YES - SKIP REST OF REPORT
MOV #FMT11,R4 ;SET FOR FORMAT 11
CMP (R1)+,#MCVLOC ;CHECK IF REPORTING CYLINDER LOC
RNE #3,SP ;NO - SKIP
MOV #FMT12,R4 ;GET CHANGE TO FORMAT 12
DEC #3,SP ;DEC PARAM COUNT
BEQ #6,SP ;IF 0 - EXIT
PRINTB #FMT3,LRSE3,(R1)+ ;REPORT IS VALUE
MOV #FMT4,LRSE4,(R1)+
MOV R4,-(SP)
MOV #3,-(SP)
MOV SP,RO
EVT C$PNTB
ADD #10,SP
PRINTB #FMT4,LRSE4,(R1)+ ;REPORT SB VALUE
MOV #FMT5,LRSE5,(R1)+
MOV R4,-(SP)
MOV #3,-(SP)
MOV SP,RO
EVT C$PNTB
ADD #10,SP
SUB #2,R3 ;DEC PARAM COUNT
PRINTB #FMT1,LRSE5,(R1)+ ;IF 0 - EXIT
MOV (R1)+,-(SP) ;REPORT CONDITION
MOV #FMT5,LRSE5,(R1)+
MOV #3,-(SP)
MOV SP,RO
EVT C$PNTB
ADD #10,SP
MOV (SP)+,R4 ;RESTORE REGS
MOV (SP)+,R3

```

```

CZRLCB.PT1 25-OCT-78 13:12
2372 022506 012604
2373 022510 000207
2374
2375
2376
2377 022512 005046
(11) 022514 133716 002455
(12) 022516 012746 005633
(0) 022520 012746 005633
(1) 022524 012746 005633
(2) 022528 012746 005633
(3) 022532 012746 012143
(4) 022536 012746 000005
(5) 022540 010600
(6) 022544 104014
(7) 022548 062706 000014
2378 022552 062706
2379 022554 012746 010475
(13) 022556 012746 010506
(14) 022560 012746 005752
(15) 022564 012746 005740
(16) 022568 012746 005733
(17) 022572 012746 012163
(18) 022576 010600 000007
(19) 022580 104014
(20) 022584 062706 000020
2380 022588 013746 002464
(21) 022592 013746 002460
(22) 022596 013746 002462
(23) 022600 013746 002459
(24) 022604 012746 012775
(25) 022608 012746 000006
(26) 022612 010600
(27) 022616 104014
(28) 022620 062706 000016
2381 022624 013746 002534
(29) 022628 013746 002526
(30) 022632 013746 002474
(31) 022636 013746 002470
(32) 022640 013746 002472
(33) 022644 012746 002468
(34) 022648 012746 012225
(35) 022652 010600 000010
(36) 022656 104014
(37) 022660 062706 000022
2382 022664 000207
2383
2384

```

```

MOV (SP)+,R1
RTS PC ;RETURN

;
REPORT PHYSICAL ADDRESS OF DEVICE UNDER TEST
AND ALL REGISTER CONTENTS
RPTRES: PRINTB #FMT5,#BASADD,RLBAS,#DRVNAM,CR,RLDRV*1>
CLF #1,SP
MOV #DRVNAM+1,(SP)
MOV #DRVNAM,-(SP)
MOV #RLBAS,-(SP)
MOV #BASADD,-(SP)
MOV #FMT5,-(SP)
MOV #5,-(SP)
MOV SP,RO
EVT C$PNTB
ADD #14,SP
REPORT RL1,REGISTERS
PRINTB #FMT6,#CSNAM,#DANAM,#BANAM,#MPNAM,#CYLWD,#HDWD
MOV #HDWD,-(SP)
MOV #CYLWD,-(SP)
MOV #MPNAM,-(SP)
MOV #BANAM,-(SP)
MOV #DANAM,-(SP)
MOV #CSNAM,-(SP)
MOV #FMT6,-(SP)
MOV #7,-(SP)
MOV SP,RO
EVT C$PNTB
ADD #20,SP
PRINTB #FMT7,#LAB1,L.CS,L.DA,L.RA,L.WP
MOV #LAB1,-(SP)
MOV #L.CS,-(SP)
MOV #L.DA,-(SP)
MOV #L.RA,-(SP)
MOV #L.WP,-(SP)
MOV #LAB2,-(SP)
MOV #T.CS,-(SP)
MOV #T.DA,-(SP)
MOV #T.PA,-(SP)
MOV #CURCYL,-(SP)
MOV #DESHD,-(SP)
MOV #FMT7,-(SP)
MOV #10,-(SP)
MOV SP,RO
EVT C$PNTB
ADD #22,SP
RTS PC

;
CLEAR PARAMETER BLOCK FOR REPORTING

```

```

2385 022776 010546          CLRPARM:  MOV     R5, -(SP)      ;STORE R5
2386 022776 012701          ;R5,R5      ;GET ADDRESS OF RLOCK
2387 022776 012705          ;SET COUNT
2388 022760 005021          2s:      CLR     (R1)+          ;CLEAR WORD
2389 022762 005305          ;DEC COUNT
2390 022764 001376          ;LOOP UNTIL ?
2391 022766 012701          ;R5,R5      ;RESET POINTER
2392 022772 012605          ;R5,R5      ;RESTORE R5
2393 022774 000207          MOV     (SP)+,R5
2394 022774 000207          RTS
2395 022776          ENDMOD
2396
    
```

```

2398 022776          BGNMOD  HRDHTST
2399          .SBTTL *TEST1          BASIC INTERFACE (PART 1)
2400          RGNTST              ;TEST01
2401 (3) 022776          TST     PASNUM              ;CHECK IF FIRST PASS      T1::
2402 022776          BNE     65S                ;EXIT IF NO
2403 023002 001120          TST     MISWIW             ;CHECK IF MANUAL INTERVENTION
2404 023004 005337          BNE     65S                ;NO - EXIT TEST
2405 023012 012737          MOV     #1,STST,ERHEAD    ;LOAD EPR HEADER
2406 023012 012737          PPINTF #FMTOPI,#OPRI,#OPRIA,#RASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
2407 023020 005946          CLR     - (SP)
2408 023026 012746          BTR     RLDV+1, (SP)
2409 023032 013746          MOV     #RDRNAM, - (SP)
2410 023032 013746          MOV     RLRAS, - (SP)
2411 023032 013746          MOV     #RASADD, - (SP)
2412 023032 013746          MOV     #OPRIA, - (SP)
2413 023032 013746          MOV     #OPRI, - (SP)
2414 023032 013746          MOV     #FMTOPI, - (SP)
2415 023032 013746          MOV     #7, - (SP)
2416 023032 013746          MOV     SP, R0
2417 023032 013746          EMT     C5PNTF
2418 023032 013746          ADD     #20, SP
2419 023032 013746          CLR     DRUFF              ;CLEAR FOR RESPONSE
2420 023032 013746          GWRNIL DRUFF, DRUFF, 1, NO
2421 023032 013746          EMT     C5G*AN
2422 023032 013746          RP     1000CS
2423 023032 013746          .WORD  DRUFF
2424 023032 013746          .WORD  TSCODE
2425 023032 013746          .WORD  DRP002
2426 023032 013746          .WORD  I
2427 023032 013746          10000S: TST     DRUFF              ;TEST RESPONSE YES
2428 023032 013746          BEQ     10000S            ;YES - SKIP
2429 023032 013746          JSR     PC, TSTINT        ;INITIALIZE TEST
2430 023032 013746          JSR     PC, CSTATC        ;GO GET STATUS (NO RESET)
2431 023032 013746          RSC
2432 023032 013746          BTT     #COSTAT, T.MP     ;CHECK IF COVER OPEN SET
2433 023032 013746          BNE     7S                ;YES - SKIP
2434 023032 013746          MOV     #MCOSTA, R3       ;SET NAME POINTER
2435 023032 013746          ERPHRD 102, ERR3
2436 023032 013746          .WORD  ICI
2437 023032 013746          .WORD  ERR3
2438 023032 013746          BTT     #RHSTAT, T.MP    ;TEST IF BRUSHES HOME
2439 023032 013746          BNE     9S                ;YES - SKIP
2440 023032 013746          MOV     #RHSTA, R3       ;SET POINTER FOR BRUSH HOME ERROR
2441 023032 013746          ERPHRD 102, ERR3
2442 023032 013746          .WORD  ICI
2443 023032 013746          .WORD  ERR3
2444 023032 013746          BTT     #WLSTAT, T.MP    ;TEST IF WRITE LOCK SET
2445 023032 013746          BNE     11S              ;YES - SKIP
2446 023032 013746          MOV     #WLSTA, R3       ;SET NAME POINTER
2447 023032 013746          ERPHRD 103, ERR3
2448 023032 013746          TRAP   TSERCODE
2449 023032 013746
2450 023032 013746
2451 023032 013746
2452 023032 013746
2453 023032 013746
2454 023032 013746
2455 023032 013746
2456 023032 013746
2457 023032 013746
2458 023032 013746
2459 023032 013746
2460 023032 013746
2461 023032 013746
2462 023032 013746
2463 023032 013746
2464 023032 013746
2465 023032 013746
2466 023032 013746
2467 023032 013746
2468 023032 013746
2469 023032 013746
2470 023032 013746
2471 023032 013746
2472 023032 013746
2473 023032 013746
2474 023032 013746
2475 023032 013746
2476 023032 013746
2477 023032 013746
2478 023032 013746
2479 023032 013746
2480 023032 013746
2481 023032 013746
2482 023032 013746
2483 023032 013746
2484 023032 013746
2485 023032 013746
2486 023032 013746
2487 023032 013746
2488 023032 013746
2489 023032 013746
2490 023032 013746
2491 023032 013746
2492 023032 013746
2493 023032 013746
2494 023032 013746
2495 023032 013746
2496 023032 013746
2497 023032 013746
2498 023032 013746
2499 023032 013746
2500 023032 013746
    
```

```

(2) 0233214 001147 .WORD 103
(3) 0233220 013154 .WORD ERR2
2427 0233220 005737 11S: .TST T,STAT ;TEST IF STATE ZERO
2428 0233224 001404 .REG R3 ;YES - SKIP
2429 0233220 005003 .CLR R3 ;SET STATE EXPECTED
(3) 0233230 104443 .ERRHRD 104,ERR7
(5) 0233230 000150 .TAP T,ERRCODE
2431 0233230 014420 .WORD 104
2432 0233232 023244 15S: .WORD ERR7
2433 0233244 016464 .JSR PC,GSTATP ;DO DRIVE RESET
(3) 0233244 104001 .ENDST 65S
2434 0233244 10020: .L10020: EMT CSETST
2435
    
```

```

2437 .SBTTL *TEST 2 BASIC INTERFACE (PART 2)
2438
2439 0233246 005737 003062 BGNSTST ;TEST 2
2440 0233246 005737 014656 TST PASNUM ;TEST IF PASS 0 T2::
2441 0233252 001075 .BNE PAS ;NO - SKIP
2442 0233254 005737 014656 TST MISWIV ;TEST IF MANUAL INTERVENTION
2443 0233260 000074 .RPL 65S ;NO - SKIP
2444 0233262 012737 006225 002434 2S: .MOV #MTEST,ERRHEAD ;SET ERROR HEADER
2445 0233276 005046 .PRINTF #MTR01,#OPR2,#OPR1A,#RASADD,RLHAS,#DRVNAM,<R,RLDRV+1> ;REQUEST CLOSE
(13) 0233270 005046 .CLR -(SP)
(12) 0233272 015376 002455 .BTSP RLDV+1,(SP)
(11) 0233300 015746 002450 .MOV #DRVNAM,(SP)
(10) 0233300 012746 005622 .MOV #RLAS,(SP)
(9) 0233314 012746 010415 .MOV #RASA0D,(SP)
(8) 0233322 012746 010052 .MOV #OPR1A,(SP)
(7) 0233322 012746 012016 .MOV #OPR2,(SP)
(6) 0233322 012746 000007 .MOV #MTR01,(SP)
(5) 0233322 016600 .MOV SP,R0
(4) 0233322 014817 .CMTF
(3) 0233322 062766 000020 .ADD #20,SP
2446 0233342 005037 004066 CLP ORUFF ;COVER AND RESET WRITE LOCK
2447 0233346 104043 .CMANIL OPRC0,ORHFF,1,NO ;CLEAR FOR RESPONSE
(2) 0233350 000464 .EMT CSETST
(1) 0233350 004066 .FP 10000S
(0) 0233350 004066 .WORD ORUFF
(0) 0233350 007724 .WORD TSCODE
(0) 0233350 000001 .WORD OPR002
(0) 0233350 000001 .WORD I
2448 0233370 005737 004366 10000S: .TST ORUFF ;TEST IF RESPONSE YES
2449 0233370 001740 .REG R3 ;NO - SKIP
2450 0233370 004737 016445 1S: .JSR PC,TSTINT ;INITIALIZE TEST
2451 0233374 004737 016464 .JSR PC,GSTATP ;GET STATUS WITH RESET
2452 0233374 000000 .BTSP ;TEST IF COVER OPEN RESET
2453 0233374 001405 002474 .REG R3 ;YES - SKIP
2454 0233374 011072 .MOV #COSTAT,T.MP ;SET NAME MESSAGE POINTER
(12) 0233374 104443 .ERRHRD 201,ERR2
(11) 0233374 000311 .TAP T,ERRCODE
(10) 0233374 000311 .WORD 201
(9) 0233374 000311 .WORD ERR2
2455 0233374 022000 002474 9S: .WORD HMLSTA,R3 ;TEST IF WRITE LOCK RESET
2456 0233374 001405 011120 .REG R3 ;YES - SKIP
2457 0233374 012703 .ERRHRD 202,ERR2 ;SET NAME MESSAGE POINTER
(8) 0233374 104443 .TAP T,ERRCODE
(7) 0233374 000312 .WORD 202
(6) 0233374 013106 .WORD EPR2
2458 0233374 000312 .WORD EPR2
2459 0233374 000312 .WORD EPR2
2460 0233374 000312 .WORD EPR2
2461 0233374 000312 .WORD EPR2
2462 0233374 000312 .WORD EPR2
2463 0233374 000312 .WORD EPR2
2464 0233374 000312 .WORD EPR2
    
```



```

2520 024076 001010 BNE 195 ;YES - SKIP
2521 024100 012737 MOV #SPERR,ERHEAD ;SET ERROR HEADER
2522 024106 011772 MOV #MSPERR,R3 ;SET NAME MESSAGE POINTER
2523 (3) 104443 ERPHRD TSPRCODE
2524 (5) 024112 000460 .WORD 304
2525 (5) 024114 013154 .WORD ERR3
2526 (3) 024120 104032 EXIT TST
2527 (3) 024122 000560 EMT CSEXIT
2528 (5) 024124 012737 .WORD L10022-
2529 024132 012704 MOV #MSTAT,ERHEAD ;SET ERROR HEADER
2530 024136 012703 MOV #MHOSTA,R3 ;SET CONDITION MESSAGE POINTER
2531 024142 032737 BIT #RHSTAT,T.MP ;SET NAME MESSAGE POINTER
2532 024150 051065 BNE 285 ;TEST IF BRUSH HOME STILL SET
2533 (3) 024152 104443 ERPHRD 305,ERR5 ;YES - SKIP
2534 (5) 024154 000461 .WORD 305
2535 (5) 024156 013272 .WORD ERR5
2536 (3) 024160 104032 EXIT TST
2537 (3) 024162 000520 EMT CSEXIT
2538 (5) 024164 024777 .WORD L10022-
2539 024170 016500 JSR PC,GSTATC ;SET WAIT COUNT FOR 5 SECONDS
2540 024174 024702 T365S ;GET STATUS
2541 024176 032737 BIT #PHSTAT,T.MP ;TEST IF BRUSH HOME RESET
2542 024180 005301 REQ 275 ;YES - SKIP
2543 024184 001404 BEQ 285 ;DECR WAIT COUNT
2544 (3) 024186 000001 WAITMS #1 ;SKIP IF ZERO
2545 (3) 024188 104026 MOV #R0 ;
2546 (5) 024190 000763 EMT CSMTM
2547 (3) 024200 000763 BR 285 ;LOOP
2548 (3) 024202 104443 ERPHRD 306,ERR4
2549 (5) 024204 000462 .WORD 306
2550 (5) 024206 013222 .WORD ERR4
2551 (3) 024210 104032 EXIT TST
2552 (3) 024212 000450 EMT CSEXIT
2553 (5) 024214 012701 .WORD 300,21-
2554 (5) 024216 004737 JSR PC,GSTATC ;SET WAIT COUNT 30 SECONDS
2555 024220 032737 T365S ;GET STATUS
2556 024224 000010 BIT #RHSTAT,T.MP ;TEST IF BRUSH HOME SET AGAIN
2557 024228 001013 BNE 325 ;YES - SKIP
2558 024232 005301 DEC R1 ;ELSE DEC WAIT COUNT
2559 024236 001404 PEQ 305 ;SKIP IF 0
2560 (3) 024240 012700 WAITMS #1
2561 (3) 024242 104026 MOV #R0
2562 (5) 024244 000763 EMT CSMTM
2563 (3) 024246 104443 ERPHRD 307,ERR5
2564 (5) 024248 000463 .WORD 307
2565 024250 013272 .WORD ERR5
2566 (3) 024252 104443 ERPHRD 308,ERR5
2567 (5) 024254 000463 .WORD 308
2568 024256 013272 .WORD ERR5
2569 (3) 024260 104032 EXIT TST
2570 (3) 024262 000763 EMT CSMTM
2571 (3) 024264 104443 ERPHRD 309,ERR5
2572 (5) 024266 000463 .WORD 309
2573 024268 013272 .WORD ERR5
2574 (3) 024270 104032 EXIT TST
2575 (3) 024272 000763 EMT CSMTM
2576 (3) 024274 104443 ERPHRD 310,ERR5
2577 (5) 024276 000463 .WORD 310
2578 024278 013272 .WORD ERR5
2579 (3) 024280 104032 EXIT TST
2580 (3) 024282 000763 EMT CSMTM
2581 (3) 024284 104443 ERPHRD 311,ERR5
2582 (5) 024286 000463 .WORD 311
2583 024288 013272 .WORD ERR5
2584 (3) 024290 104032 EXIT TST
2585 (3) 024292 000763 EMT CSMTM
2586 (3) 024294 104443 ERPHRD 312,ERR5
2587 (5) 024296 000463 .WORD 312
2588 024298 013272 .WORD ERR5
2589 (3) 024300 104032 EXIT TST
    
```

```

2590 (3) 024300 104032 EMT CSEXIT
2591 (3) 024302 000450 .WORD L10022-
2592 024304 012703 MOV #MSTAT,ERHEAD ;SET ERROR HEADER
2593 024306 000009 MOV #MHOSTA,R3 ;SET EXPECTED STATE VALUE
2594 024308 004737 JSR PC,GSTATC ;GET STATUS
2595 024310 024702 T365S R3,T,STAT
2596 024312 001406 CMP #0,R3 ;CHECK IF STATE 3
2597 024314 000462 ERPHRD 308,ERR7 ;YES - SKIP
2598 (3) 024316 104443 TRAP TSPRCODE
2599 (5) 024318 000462 .WORD 308
2600 (5) 024320 014220 .WORD ERR7
2601 (3) 024322 104032 EXIT TST
2602 (3) 024324 000320 EMT CSEXIT
2603 024326 012737 .WORD L10022-
2604 024328 012704 MOV #MSTAT,ERHEAD ;SET ERROR HEADER
2605 024330 011772 MOV #MHOSTA,R3 ;SET CONDITION MESSAGE POINTER
2606 024332 011131 MOV #MHOSTA,R3 ;SET NAME MESSAGE POINTER
2607 024334 004737 JSR PC,GSTATC ;GET STATUS
2608 024336 024702 T365S R3,T,STAT
2609 024338 001406 CMP #0,R3 ;CHECK IF STATE 4
2610 024340 000462 ERPHRD 309,ERR5 ;YES - SKIP
2611 (3) 024342 104443 TRAP TSPRCODE
2612 (5) 024344 000462 .WORD 309
2613 (5) 024346 013272 .WORD ERR5
2614 (3) 024348 104032 EXIT TST
2615 (3) 024350 000763 EMT CSMTM
2616 024352 012737 .WORD L10022-
2617 024354 012704 MOV #MSTAT,ERHEAD ;SET ERROR HEADER
2618 024356 011772 MOV #MHOSTA,R3 ;SET CONDITION MESSAGE POINTER
2619 024358 011131 MOV #MHOSTA,R3 ;SET NAME MESSAGE POINTER
2620 024360 004737 JSR PC,GSTATC ;GET STATUS
2621 024362 032737 BIT #HSTAT,T.MP ;TEST IF HEADS OUT SET
2622 024364 001005 BNE 385 ;YES - SKIP
2623 (3) 024366 104443 ERPHRD 309,ERR5
2624 (5) 024368 000465 .WORD 309
2625 (5) 024370 013272 .WORD ERR5
2626 (3) 024372 104032 EXIT TST
2627 (3) 024374 000763 EMT CSMTM
2628 024376 012737 .WORD L10022-
2629 024378 012704 MOV #MSTAT,ERHEAD ;SET ERROR HEADER
2630 024380 011772 MOV #MHOSTA,R3 ;SET CONDITION MESSAGE POINTER
2631 024382 011131 MOV #MHOSTA,R3 ;SET NAME MESSAGE POINTER
2632 024384 004737 JSR PC,GSTATC ;GET STATUS
2633 024386 032737 BIT #VOLCK,R3 ;TEST IF VOLUME CHECK SET
2634 024388 012703 BNE 405 ;YES - SKIP
2635 024390 000462 ERPHRD 310,ERR5 ;SET NAME MESSAGE POINTER
2636 (3) 024392 104443 TRAP TSPRCODE
2637 (5) 024394 000462 .WORD 310
2638 (5) 024396 013272 .WORD ERR5
2639 (3) 024398 104032 EXIT TST
2640 (3) 024400 000763 EMT CSMTM
2641 024402 032737 .WORD L10022-
2642 024404 001007 RNE #DRVERR,T.CS ;TEST IF DRIVE ERROR SET
2643 024406 011033 MOV #R3 ;YES - SKIP
2644 024408 012703 MOV #MSPERR,R3 ;SET NAME MESSAGE POINTER
2645 (3) 024410 104443 ERPHRD 311,ERR5
2646 (5) 024412 000467 .WORD 311
2647 (5) 024414 013272 .WORD ERR5
2648 (3) 024416 104032 EXIT TST
2649 (3) 024418 000763 EMT CSMTM
2650 024420 012737 .WORD L10022-
2651 024422 005670 MOV #R0,R1 ;SET WAIT COUNT FOR 300 MS
2652 024424 006950 MOV #MSTAT,ERHEAD ;SET ERROR HEADER
2653 024426 000004 JSR #R3 ;SET EXPECTED STATE VALUE
2654 024428 004737 JSR PC,GSTATC ;GET STATUS
2655 024430 024702 T365S R3,T,STAT
2656 024432 001413 BEQ 445 ;CHECK IF STATE 4
2657 024434 000001 ;YES - SKIP
    
```

```

2588 024520 005301 DEC R1 ;DEC WAIT COUNT
2589 024524 001404 REG #1 ;SKIP IF 0
2590 (3) 024524 000001 WAITUS #1,RO
2591 (3) 024524 012700 MOV #1,RO
2592 (3) 024530 104027 EMT CSWTU
2593 (3) 024534 000764 BR #1
2594 (3) 024534 104443 47S: ERPHRD #1,ERR7
2595 (5) 024536 000476 TRAP TSERCODE
2596 (3) 024540 014220 .WORD 312
2597 (3) 024542 104032 EMT CSEXIT
2598 (3) 024544 000136 .WORD L10022-
2599 (3) 024552 000454 49S: MOV #30,R1 ;SET WAIT COUNT FOR 30 MS
2600 (3) 024556 000035 .WORD #3,R1 ;SET EXPECTED STATE VALUE
2601 (3) 024560 004737 50S: JSE PC,GSTATC ;GET STATUS
2602 (3) 024564 020339 CMP T365S
2603 (3) 024570 001413 REQ R3 T.STATE ;CHECK IF STATE 5
2604 (3) 024572 005301 DEC R1 ;YES - SKIP
2605 (3) 024574 001404 REG #1 ;DEC WAIT COUNT
2606 (3) 024576 000001 WAITUS #1,RO ;ELSE SKIP
2607 (3) 024602 104027 MOV #1,RO
2608 (3) 024604 000764 EMT CSWTU
2609 (3) 024606 104443 BR #1
2610 (5) 024610 000471 51S: ERPHRD #1,ERR7
2611 (3) 024614 014220 TRAP TSERCODE
2612 (3) 024614 104032 EMT CSEXIT
2613 (3) 024616 000064 .WORD L10022-
2614 (3) 024620 001120 55S: MOV #8,R1 ;SET WAIT FOR 8 MS
2615 (3) 024624 016500 56S: JSE PC,GSTATC ;GET STATUS
2616 (3) 024630 024702 T365S
2617 (3) 024632 032737 002466 BIT #DRDVMASK,T.CS ;CHECK IF DRIVE READY
2618 (3) 024640 001280 BFC 62S ;YES - SKIP
2619 (3) 024644 001304 DEC R0 ;DEC COUNT
2620 (3) 024646 000001 WAITUS #1,RO ;SKIP IF 0
2621 (3) 024648 012700 MOV #1,RO
2622 (3) 024654 000764 EMT CSWTU
2623 (3) 024656 000764 BR #1
2624 (3) 024664 000764 60S: MOV #W1STST,EPHEAD ;SET ERROR HEADER
2625 (3) 024666 012703 MOV #STATES,SR4 ;SET CONDITION MESSAGE POINTER
2626 (3) 024670 012703 MOV #DRDVMASK,SR4 ;SET NAME MESSAGE POINTER
2627 (3) 024674 104443 ERPHRD #1,ERR5
2628 (5) 024676 000472 TRAP TSERCODE
2629 (3) 024700 013272 .WORD 314
2630 024702 62S:
2631 024704 T365S
2632 (3) 024706 ENDTST
2633 (3) 024708 L10022: EMT CSFTST
2634 (3) 024710 104001
    
```

```

2624 024704 005737 003062 SRTTL *TEST 4 HEAD UNLOADING
2625 (3) 024704 001404 BGNSTST ;TEST 4
2626 (3) 024710 001003 TST PASNUW ;TEST IF FIRST PASS T4::
2627 (3) 024712 005737 RNE BS ;NO - SKIP
2628 (3) 024716 104042 TST #1SWIW ;TEST IF MANUAL INTERVENTION
2629 (3) 024720 104032 BMT 10S ;YES - SKIP
2630 (3) 024722 000586 EMT TST
2631 (3) 024724 104032 .WORD L10023-
2632 (3) 024726 000001 BGN SUB 10S:
2633 (3) 024728 104002 EMT C$BSUR T4.1:
2634 (3) 024730 002737 MOV #INSTACHG,ERHEAD ;SET ERROR HEADER
2635 (3) 024732 016446 JSE PC,STATM ;INITIALIZE TEST
2636 (3) 024734 025400 JSE PC,GSTATC ;GET STATUS
2637 (3) 024736 032737 000001 002466 BIT #DRDVMASK,T.CS ;CHECK IF DRIVE READY
2638 (3) 024738 001040 BFC 1S: ;YES - SKIP
2639 (3) 024740 005046 PRINTF #WTOP1,#OPR6,#OPR1A,#BASADD,RLBAS,#DRVNAM,<R,RLDRV+1>
2640 (3) 024742 005046 CLR -(SP)
2641 (3) 024744 153716 002455 RTS PLDRV+1,(SP)
2642 (3) 024746 013746 MOV #DRVNAM,-(SP)
2643 (3) 024748 002450 MOV #RLBAS,-(SP)
2644 (3) 024750 005622 MOV #BASADD,-(SP)
2645 (3) 024752 010416 MOV #OPR1A,-(SP)
2646 (3) 024754 010156 MOV #OPR1B,-(SP)
2647 (3) 024756 000007 MOV #WTOP1,-(SP)
2648 (3) 024758 000007 MOV #7,-(SP)
2649 (3) 024760 016500 MOV SP,RC
2650 (3) 024762 104017 EMT C$WTF
2651 (3) 024764 062706 ADD #20,SP
2652 (3) 024766 004066 CLR DRUFF ;CLEAR PCR RESPONSE
2653 (3) 024768 000404 EMT C$CAN
2654 (3) 024770 004966 BR 1000G
2655 (3) 024772 000120 .WORD DRUFF
2656 (3) 024774 000001 .WORD TSCODE
2657 (3) 024776 000001 .WORD OPR002
2658 (3) 024778 000001 .WORD 1
2659 (3) 024780 005737 004066 1000G: TST DRUFF ;TST RESPONSE YES
2660 (3) 024782 001740 REG #1 ;NO - SKIP
2661 (3) 024784 052737 000010 002426 3S: RTS #UNLOAD,OPFLAG ;SET UNLOAD OPERATION
2662 (3) 024786 035346 4S: PRINTF #WTOP1,#OPR3,#OPR1A,#BASADD,RLBAS,#DRVNAM,<R,RLDRV+1>
2663 (3) 024788 153716 CLR -(SP)
2664 (3) 024790 024555 RTS PLDRV+1,(SP)
2665 (3) 024792 005633 MOV #DRVNAM,-(SP)
2666 (3) 024794 002450 MOV #RLBAS,-(SP)
2667 (3) 024796 005622 MOV #BASADD,-(SP)
2668 (3) 024798 010416 MOV #OPR1A,-(SP)
2669 (3) 024800 010164 MOV #OPR1B,-(SP)
2670 (3) 024802 010164 MOV #OPR3,-(SP)
2671 (3) 024804 012746 MOV #WTOP1,-(SP)
2672 (3) 024806 000007 MOV #7,-(SP)
    
```

```

(3) 025126 010600      MOV    SP,R0
(4) 025130 104317      EMT    C$ENFF
(4) 025132 022706      ADD    #20,SP
26848 025135 000020      MOV    #R3
26849 025136 000066      MOV    #R3
26850 025137 001274      MOV    #R0,,R4
26851 025138 016500      MOV    #700,,R4
26852 025139 004737      JSR    PC,C$STATC
26853 025140 022502      T465$
26854 025141 001436      R3,T,STAT
26855 025142 000005      RFO    #R3
26856 025143 002502      CMP    #R3,T,STAT
26857 025144 000005      BNE    #R3,T,STAT
26858 025145 005304      DEC    #R3
26859 025146 001004      DEC    #R3
26860 025147 005301      DEC    #R3
26861 025148 001274      BEQ    #R3,,R4
26862 025149 012700      MOV    #1,,R0
26863 025150 104317      EMT    C$WTU
26864 025151 004066      CLR    #R3
26865 025152 005737      GMANIL OPR003,ORUFF,1,NO
26866 025153 001274      EMT    C$GMAN
26867 025154 004066      R3
26868 025155 000120      .WORD ORUFF
26869 025156 000751      .WORD T$CODE
26870 025157 000001      .WORD OPR003
26871 025158 005737      TST    ORUFF
26872 025159 001706      BEQ    #R3,,ERR7
26873 025160 104443      ERPHRD 402,,ERR7
26874 025161 000422      T$PCODE
26875 025162 000422      .WORD 401
26876 025163 014220      .WORD ERR7
26877 025164 104032      EXIT
26878 025165 000126      .WORD L10024-
26879 025166 012703      MOV    #R3
26880 025167 005670      MOV    #3000,,R1
26881 025168 004737      JSR    PC,C$STATC
26882 025169 022502      T465$
26883 025170 001436      R3,T,STAT
26884 025171 000005      RFO    #R3
26885 025172 005304      DEC    #R3
26886 025173 001004      DEC    #R3
26887 025174 005301      DEC    #R3
26888 025175 001274      BEQ    #R3,,R4
26889 025176 012700      MOV    #1,,R0
26890 025177 104317      EMT    C$WTU
26891 025178 000764      BP    #R3
26892 025179 104443      ERPHRD 402,,ERR7
26893 025180 000622      T$PCODE
26894 025181 004220      .WORD ERR7
26895 025182 014220      .WORD ERR7
26896 025183 000001      EXIT    SUR
    
```

```

(3) 025230 104032      EMT    C$EXIT
(3) 025231 000126      .WORD L10024-
26901 025232 001270      MOV    #R3
26902 025233 001130      MOV    #R0,,R1
26903 025234 016500      JSR    PC,C$STATC
26904 025235 022502      T465$
26905 025236 001436      R3,T,STAT
26906 025237 000005      RFO    #R3
26907 025238 005304      DEC    #R3
26908 025239 001004      DEC    #R3
26909 025240 005301      DEC    #R3
26910 025241 001404      BEQ    #R3,,R4
26911 025242 012700      MOV    #1,,R0
26912 025243 104317      EMT    C$WTU
26913 025244 000001      BP    #R3
26914 025245 104443      ERPHRD 402,,ERR7
26915 025246 000622      T$PCODE
26916 025247 014220      .WORD 402
26917 025248 014220      .WORD ERR7
26918 025249 025400      MOV    #2,FRPSWI
26919 025250 000002      MOV    #2,FRPSWI
26920 025251 002440      ENDSUB
26921 025252 104063      L10024:
26922 025253 005046      EMT    C$ESUR
26923 025254 005046      PRINTF #PNT01,#OPR6,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
26924 025255 002455      CLR    -(SP)
26925 025256 005433      BISR  #RLDRV+1,(SP)
26926 025257 002450      MOV    #DRVNAM,-(SP)
26927 025258 005622      MOV    #RLBAS,-(SP)
26928 025259 012746      MOV    #BASADD,-(SP)
26929 025260 010416      MOV    #OPR6,-(SP)
26930 025261 012746      MOV    #PNT01,-(SP)
26931 025262 000007      MOV    #7,-(SP)
26932 025263 010600      MOV    SP,R0
26933 025264 005037      EMT    C$ENFF
26934 025265 000020      ADD    #20,SP
26935 025266 004066      CLR    #R3
26936 025267 104043      GMANIL OPR002,ORUFF,1,NO
26937 025268 000404      EMT    C$GMAN
26938 025269 004066      R3
26939 025270 000120      .WORD ORUFF
26940 025271 000754      .WORD T$CODE
26941 025272 000001      .WORD OPR002
26942 025273 005737      TST    ORUFF
26943 025274 001740      BEQ    #R3,,ERR7
26944 025275 025510      ENDTST
26945 025510      L10023:
(3) 025510      EMT    C$SETST
    
```

```

2705      025512      012737      000002      002440      MOV      #2,ERRSWI      ;SET FOR NO ERROR RETURN T5:1
2706      025512      005737      003062      TST      PASMUM      ;TEST IF FIRST PASS
2707      025512      005737      003062      BNE      PXT05      ;NO - SKIP
2708      025512      005737      003062      BIT      #DPSLT,MISWIW ;TEST IF SELECT TESTS
2709      025512      005737      003062      BEO      EXT05      ;NO - SKIP
2710      025512      005737      003062      BNE      PXT05      ;NO - SKIP
2711      025512      005737      003062      PRINTF  #FMT0P1,#OPR7,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>;REQUEST REMOVE A
2712      025512      005737      003062      CLR      -1(SP)
2713      025512      005737      003062      BLSR    RLDV+1,(SP)
2714      025512      005737      003062      MOV      #DRVNAM,-(SP)
2715      025512      005737      003062      MOV      RLBAS,(SP)
2716      025512      005737      003062      MOV      #BASADD,-(SP)
2717      025512      005737      003062      MOV      #OPR1A,-(SP)
2718      025512      005737      003062      MOV      #OPR7,-(SP)
2719      025512      005737      003062      MOV      #FMT0P1,-(SP)
2720      025512      005737      003062      MOV      #7,-(SP)
2721      025512      005737      003062      MOV      SP,RO
2722      025512      005737      003062      FMT     C$PNTF
2723      025512      005737      003062      ADD     #20,SP
2724      025512      005737      003062      CLR     OBUFF      ;CLEAR FOR RESPONSE
2725      025512      005737      003062      GMANIL  OPR002,OBUFF,1,NO
2726      025512      005737      003062      EMT     C$GMAN
2727      025512      005737      003062      BR     10000$
2728      025512      005737      003062      .WORD  OBUFF
2729      025512      005737      003062      .WORD  TSCODE
2730      025512      005737      003062      .WORD  OPR002
2731      025512      005737      003062      .WORD  1
2732      025512      005737      003062      10000$: TST     OBUFF      ;TEST RESPONSE YES
2733      025512      005737      003062      BEQ     1$        ;NO - SKIP
2734      025512      005737      003062      MOV     #T05ERR,ERHEAD ;SET ERROR HEADER MESSAGE
2735      025512      005737      003062      JSR     PC,GSTATC  ;DO SELECT AND GET STATUS
2736      025512      005737      003062      JSR     PC,GSTATC
2737      025512      005737      003062      T504$: T504$
2738      025512      005737      003062      MOV     RLDV,TEMPO  ;STORE ORIGINAL DRIVE NUMBER
2739      025512      005737      003062      MOV     RLDV,R1     ;PUT IT IN R1
2740      025512      005737      003062      MOV     #4,R4       ;SET COUNT FOR NUMBER OF PLUGS
2741      025512      005737      003062      LPT05: ADD     #400,R1  ;BUMP TO NEXT DRIVE
2742      025512      005737      003062      CMP     #2000,R1   ;CHECK IF TOO LARGE
2743      025512      005737      003062      BNE     4$        ;NO - SKIP
2744      025512      005737      003062      CLR     R1         ;ELSE CLEAR TO DRIVE 0
2745      025512      005737      003062      MOV     R1,RLDRV   ;PUT IT BACK IN RLDV
2746      025512      005737      003062      PRINTF  #FMT0P3,#OPR8,<B,RLDRV+1>,#OPR1B,#UNDTST
2747      025512      005737      003062      MOV     #OPR1B,-(SP)
2748      025512      005737      003062      MOV     #OPR1B,-(SP)
2749      025512      005737      003062      BLSR    RLDV+1,(SP)
2750      025512      005737      003062      MOV     #FMT0P3,-(SP)
2751      025512      005737      003062      MOV     #5,-(SP)
2752      025512      005737      003062      MOV     SP,RO
2753      025512      005737      003062      FMT     C$PNTF
2754      025512      005737      003062      ADD     #14,SP
2755      025512      005737      003062      ;INSERT PLUG REQUEST
    
```

```

2731      025756      005037      004066      CLR     OBUFF      ;CLEAR FOR RESPONSE
2732      025756      005037      004066      GMANIL  OPR002,OBUFF,1,NO
2733      025756      005037      004066      EMT     C$GMAN
2734      025756      005037      004066      BR     10001$
2735      025756      005037      004066      .WORD  OBUFF
2736      025756      005037      004066      .WORD  TSCODE
2737      025756      005037      004066      .WORD  OPR002
2738      025756      005037      004066      .WORD  1
2739      025756      005037      004066      10001$: TST     OBUFF      ;TEST RESPONSE YES
2740      025756      005037      004066      BEQ     5$        ;NO - SKIP
2741      025756      005037      004066      BGNSUP  5$
2742      025756      005037      004066      EMT     C$RSUB      T5.1:
2743      025756      005037      004066      JSR     PC,GSTATC  ;GET STATUS - REPORT ANY ERROR
2744      025756      005037      004066      MOV     #2,ERRSWI  ;INIT ERROR SWITCH
2745      025756      005037      004066      60$:   MOV     #2,ERRSWI
2746      025756      005037      004066      ENDSUB  L10025:
2747      025756      005037      004066      DEC     R4         ;DEC COUNT
2748      025756      005037      004066      RNE     LPT05     ;LOOP IF NOT ZERO
2749      025756      005037      004066      MOV     #FPMPO,RLDRV ;ELSE RESTORE RLDV
2750      025756      005037      004066      T504$: T504$:
2751      025756      005037      004066      4$:   PRINTF  #FMT1,#OPR8,#OPR9
2752      025756      005037      004066      MOV     #OPR9,-(SP)
2753      025756      005037      004066      MOV     #OPR8,-(SP)
2754      025756      005037      004066      MOV     #FMT1,-(SP)
2755      025756      005037      004066      MOV     #3,-(SP)
2756      025756      005037      004066      MOV     SP,RO
2757      025756      005037      004066      FMT     C$PNTF
2758      025756      005037      004066      ADD     #10,SP
2759      025756      005037      004066      CLR     OBUFF      ;CLEAR FOR RESPONSE
2760      025756      005037      004066      GMANIL  OPR002,OBUFF,1,NO
2761      025756      005037      004066      EMT     C$GMAN
2762      025756      005037      004066      BR     10000$
2763      025756      005037      004066      .WORD  OBUFF
2764      025756      005037      004066      .WORD  TSCODE
2765      025756      005037      004066      .WORD  OPR002
2766      025756      005037      004066      .WORD  1
2767      025756      005037      004066      10000$: TST     OBUFF      ;TEST RESPONSE YES
2768      025756      005037      004066      BEQ     4$        ;NO - SKIP
2769      025756      005037      004066      EXT05: EXT05:
2770      025756      005037      004066      ENDTST  L10025:
2771      025756      005037      004066      FMT     C$SETST
    
```

```

2754 0 026116
2755 0 026116
2756 0 026116
2757 0 026116
2758 0 026116
2759 0 026116
2760 0 026116
2761 0 026116
2762 0 026116
2763 0 026116
2764 0 026116
2765 0 026116
2766 0 026116
2767 0 026116
2768 0 026116
2769 0 026116
2770 0 026116
2771 0 026116
2772 0 026116
2773 0 026116
2774 0 026116
2775 0 026116
2776 0 026116
2777 0 026116
2778 0 026116
2779 0 026116
2780 0 026116
2781 0 026116
2782 0 026116
2783 0 026116
2784 0 026116
2785 0 026116
2786 0 026116
2787 0 026116
2788 0 026116
2789 0 026116
2790 0 026116
2791 0 026116
2792 0 026116
2793 0 026116
2794 0 026116
2795 0 026116
    
```

```

2796 0 026370
2797 0 026370
2798 0 026370
2799 0 026370
2800 0 026370
2801 0 026370
2802 0 026370
2803 0 026370
2804 0 026370
2805 0 026370
2806 0 026370
2807 0 026370
2808 0 026370
2809 0 026370
2810 0 026370
2811 0 026370
2812 0 026370
2813 0 026370
2814 0 026370
2815 0 026370
2816 0 026370
2817 0 026370
2818 0 026370
2819 0 026370
2820 0 026370
2821 0 026370
2822 0 026370
2823 0 026370
2824 0 026370
2825 0 026370
2826 0 026370
2827 0 026370
2828 0 026370
2829 0 026370
2830 0 026370
2831 0 026370
2832 0 026370
2833 0 026370
2834 0 026370
2835 0 026370
2836 0 026370
2837 0 026370
2838 0 026370
2839 0 026370
2840 0 026370
2841 0 026370
2842 0 026370
2843 0 026370
2844 0 026370
2845 0 026370
2846 0 026370
2847 0 026370
2848 0 026370
2849 0 026370
2850 0 026370
2851 0 026370
2852 0 026370
2853 0 026370
2854 0 026370
2855 0 026370
2856 0 026370
2857 0 026370
2858 0 026370
2859 0 026370
2860 0 026370
2861 0 026370
2862 0 026370
2863 0 026370
2864 0 026370
2865 0 026370
2866 0 026370
2867 0 026370
2868 0 026370
2869 0 026370
2870 0 026370
2871 0 026370
2872 0 026370
2873 0 026370
2874 0 026370
2875 0 026370
2876 0 026370
2877 0 026370
2878 0 026370
2879 0 026370
2880 0 026370
2881 0 026370
2882 0 026370
2883 0 026370
2884 0 026370
2885 0 026370
2886 0 026370
2887 0 026370
2888 0 026370
2889 0 026370
2890 0 026370
2891 0 026370
2892 0 026370
2893 0 026370
2894 0 026370
2895 0 026370
2896 0 026370
2897 0 026370
2898 0 026370
2899 0 026370
2900 0 026370
    
```

```

(5) 026624 013040          .WORD   ERR1
(3) 026625          .EXIT   CSXIT
(3) 026630 000062          .WORD   L10030--
2826 026632          .WAITMS  #2          ;WAIT FOR DSE TO SET
(3) 026636 004026          .MOV    #2,R0
2828 026640 004737 016514          .FMT    #2,M
2829 026644 026704          .JSR    PC,GSTAT          ;GET STATUS
2830 026646 001077 002474          .GOS    #DSESTAT,T.MP      ;TEST IF DRIVE SELECT ERROR SET
2831 026656 012703 011142          .BNE    #DSEERR,R3        ;YES - SKIP
2832 026662          .ERRHRD  #02,ERR3        ;SET NAME MESSAGE POINTER
(5) 026664 001133          .TRAP   TSERCODE
(5) 026666 013154          .WORD   ERR3
2833 026670          .EXIT   CSXIT
(3) 026672 004032          .MOV    #0,R0
2834 026674 010562 000000          .MOV    R5,RLCS(R2)      ;LOAD IN DIFFERENT ADDRESS
2835 026700 005304          .DEC    R4                ;DEC COUNT
2836 026702 001124          .BNE    #0,R5            ;LOOP IF NOT ZERO
2838 026712 012739 000002 002440 60S: .MOV    #2,ERRSWI        ;INIT ERROR SWITCH
(3) 026714 104003          .ENDSUB L10030:
2839 026714 104003          .EXIT   CSXIT
(8) 026714 012746          .PRINTF CSESUB #OPR11 ;REQUEST PLUG CHANGE
(7) 026720 012746 012327          .MOV    #OPR1, -(SP)
(5) 026724 012746 000002          .MOV    #FMT9, -(SP)
(5) 026726 012746 000002          .MOV    #2, -(SP)
(4) 026732 104017          .FMT    #2,M
(4) 026734 062706 000006          .EXIT   CSXIT
2841 026740 005037 004966          .ADD    #6,SP
(3) 026744 104043          .CLP    OROFF
(3) 026746 000404          .GAWIL  OROFF,ORUFF,1,NO ;CLEAR FOR RESPONSE
(5) 026750 004066          .EMT    CSESTAT
(5) 026754 007724          .BR     10000S
(5) 026756 000001          .WORD   ORUFF
(5) 026760 005737 004066          .TST   ORUFF
2843 026764 001753          .REQ   15S               ;TEST RESPONSE YES
2844 026766          .LCLEXT:
2845 026768          .ENDTST:
(3) 026766 104001          .EMT    CSETST
    
```

```

2847 026770          .SRITL  *TEST 7          INITIAL STATE
(3) 026770          .RGWST  *TEST 07
2849 026770 012737 006347 002434          .MOV    #INITST,EPHEAD   ;SET ERROR HEADER T7::
2850 026776 004737 016446          .PC     TSTINT          ;INITIALIZE TEST
(3) 027002 012700 000012          .WAITMS #1,PC          ;WAIT 1 MS
(3) 027006 104027          .FMT    CSWTO
2852 027010 004737 016500          .JSR    PC,GSTAT          ;GET STATUS
2853 027014 032737 000001 002466          .GOS    #DRDVMSK,T.CS    ;CHECK IF DRIVE READY
2854 027016 032737 000001 002466          .BNE    #DRDRV,R3        ;YES-SKIP
2855 027024 001003          .MOV    #DSEERR,R3        ;SET NAME MESSAGE POINTER
2856 027026 012703 010711          .BR     #2,R3            ;GO REPORT
2857 027028 012703 000005          .MOV    #R3              ;SET EXPECTED STATE VALUE
2859 027040 020337 002502          .CMP    R3,T.STAT        ;CHECK IF STATE OK
2860 027044 001405          .BFO    #0,ERR7         ;YES-SKIP
2862 027046          .ERRHRD  #01,ERR7        ;ELSE REPORT STATE ERROR
(3) 027046 104443          .TRAP   TSERCODE
(5) 027050 001275          .WORD   ERR7
(5) 027054 014220          .WORD   ERR7
2863 027054          .EXIT   CSXIT
(3) 027054 104032          .EMT    TSERCODE
(3) 027056 000471          .EXIT   CSXIT
2864 027060 032701 002474          .WORD   L10031--
2865 027064 032701 000020          .MOV    #HOSTAT,R1       ;GET WP RFG
2866 027070 001003          .BNE    #HOSTAT,R1       ;CHECK HEADS OUT
2867 027072 012703 011131          .MOV    #HOSTA,R3        ;YES-SKIP
2868 027076 000406          .BR     #9,R1            ;SET NAME MESSAGE PTR
2869 027100 032701 000010          .BNE    #HSTAT,R1        ;GO REPORT
2870 027104 001007          .BNE    #10,S            ;CHECK BRUSH HOME SET
2871 027106 012703 011105          .MOV    #HSTA,R3         ;YES-SKIP
2872 027112 104443          .ERRHRD  #02,ERR3        ;SET NAME MESSAGE PTR
(5) 027114 001276          .TRAP   TSERCODE
(5) 027116 013154          .WORD   ERR3
2873 027120          .WORD   ERR3
(3) 027122 000156          .EXIT   CSXIT
(3) 027124 005737          .EXIT   CSXIT
2874 027126 000156          .WORD   L10031--
2875 027130 100034          .TST   #ISWIM           ;TEST IF MANUAL INTERVENTION RUN
2876 027134 005937 003062          .BPL   #ASNUM           ;NO-SKIP
2877 027136 001031          .TST   #ASNUM           ;CHECK IF FIRST PASS
2878 027140 032701 000100          .BNE    #HSTAT,R1       ;NO-SKIP
2879 027144 001411          .BIT    #HSTAT,R1        ;ELSE CHECK HD 0 SELECTED
2880 027146 012703 011044          .REQ   #HSTA,R3         ;YES-SKIP
2881 027152 012704 011763          .MOV    #CCVLP,R4        ;SET NAME MESSAGE PTR
2882 027156          .ERRHRD  #03,ERR4        ;SET CONDITION POINTER
(5) 027160 001277          .TRAP   TSERCODE
(5) 027162 013222          .WORD   ERR4
2883 027164          .EXIT   CSXIT
(3) 027166 104032          .EMT    TSERCODE
(3) 027168 000156          .EXIT   CSXIT
2884 027170 032701 001000          .BIT    #VCSTAT,R1       ;CHECK VOL CHECK SET
    
```

```

2888 027174 001003 BNE 155
2889 027175 002703 MOV #VOLCK,R3
2890 027176 000743 BR 95
2891 027177 032737 040000 002466 15S: BIT #DRVERR,T.CS
2892 027178 001003 BNE 165
2893 027179 002703 MOV #DRERR,R3
2894 027180 000734 BR 95
2895 027181 032707 020000 16S: BIT #WLSTAT,R1
2896 027182 001405 BEQ 177
2897 027183 012703 MOV #WLSTA,R3
2898 027184 104443 ERRHRD 703,ERR2
2899 027185 001300 TRAP T$ERRCODE
2900 027186 013106 .WORD 104
2901 027187 013106 .WORD 104
2902 027188 005781 BEQ 177
2903 027189 001405 BEQ 177
2904 027190 104443 ERRHRD 703,ERR6
2905 027191 001300 TRAP T$ERRCODE
2906 027192 013342 .WORD 104
2907 027193 013342 .WORD 104
2908 027194 104032 EMT CSETST
2909 027195 000016 EMT CSETST
2910 027196 013701 MOV T,CS,R1
2911 027197 027011 BIC #141777,R1
2912 027198 005781 TST R1
2913 027199 001365 BNE 185
2914 027200 25S:
2915 027201 65S:
2916 027202 2907:
2917 027203 2907:
2918 027204 2907:
2919 027205 2907:
2920 027206 2907:
2921 027207 2907:
2922 027208 2907:
2923 027209 2907:
2924 027210 2907:
2925 027211 2907:
2926 027212 2907:
2927 027213 2907:
2928 027214 2907:
2929 027215 2907:
2930 027216 2907:
    
```

```

2909 027302 :SRTTL *TEST 8 INITIAL RESET STATE
2910 027303 :BGNTST *TEST 8
2911 027304 012737 006347 002434 MOV #INIT$T,ERHEAD
2912 027305 004737 016446 JSR PC,T$INT
2913 027306 004737 JSR PC,G$STATR
2914 027307 004737 JSR PC,T$INT
2915 027308 004737 JSR PC,T$INT
2916 027309 004737 JSR PC,T$INT
2917 027310 004737 JSR PC,T$INT
2918 027311 004737 JSR PC,T$INT
2919 027312 004737 JSR PC,T$INT
2920 027313 004737 JSR PC,T$INT
2921 027314 004737 JSR PC,T$INT
2922 027315 004737 JSR PC,T$INT
2923 027316 004737 JSR PC,T$INT
2924 027317 004737 JSR PC,T$INT
2925 027318 004737 JSR PC,T$INT
2926 027319 004737 JSR PC,T$INT
2927 027320 004737 JSR PC,T$INT
2928 027321 004737 JSR PC,T$INT
2929 027322 004737 JSR PC,T$INT
2930 027323 004737 JSR PC,T$INT
    
```

```

2932
2933
2934
2935 0273666          .SBTTL *TEST 9          DRIVE READY
2936 0273666          RGNTST *TEST 9          ;TEST 9
2937 0273774 012737 006375 002434 MOV #TOERR,ERHEAD ;SET ERROR HEADER
2938 0274000 005021 002524 CLR #NEWCVL,R1 ;GET POINTER TO DESIRED LOC
2939 0274002 005021 CLR (R1)+ ;CLEAR NEW CVL
2940 0274004 005021 CLR (R1)+ ;CLEAR CURRENT CVL
2941 0274006 005011 CLR (R1) ; DIFFERENCE
2942 0274110 004737 016446 JSR PC,ISTINT ; SIGN
2943 0274114 004737 016464 JSR PC,GSTATR ; INITIALIZE TEST
2944 0274200 027664 JSR PC,GSTATR ;GET STATUS WITH RESET
2945 0274222 004737 JSR PC,POSHSR ;POSITION HEAD SELECTED BIT
2946 0274266 012537 002534 MOV RS,DESHD ;STORE AS DESIRED HEAD
2947 0274332 004737 017346 JSR PC,SIMSEK ;EXECUTE SIMPLE SEEK
2948 0274400 012703 010711 MOV #WDRDV,R3 ;SET NAME MESSAGE PTR
2949 0274404 012704 011722 MOV #CDRDV,R4 ;SET CONDITION POINTER
2950 0274454 027664 JSR PC,GSTAT ;GET STATUS
2951 0274540 004737 000001 002466 BIT #DRDVMASK,T.CS ;TEST READY SET
2952 0274566 001405 BEQ ERHRD 901 ;NO-SKIP
2953 0274666 104443 TRAP TSERCODE ;REPORT READY ERROR
2954 0274700 001605 .WORD 901
2955 0274722 013222 .WORD ERR4
2956 0274774 104032 EMT CSEXT ;EXIT
2957 0274776 000166 .WORD L10033- ;EXIT
2958 0275000 012764 000171 4S: MOV #R1,ERR ;SET WAIT COUNT
2959 0275110 027664 JSR PC,GSTAT ;GET STATUS
2960 0275112 012703 000005 MOV #R3 ;SET EXPECTED STATE VALUE
2961 0275116 023703 002502 CMP #STAT,R3 ;CHECK STATE IS 5
2962 0275224 001405 BEQ ERHRD 902 ;YES-SKIP
2963 0275244 001606 ERHRD 902 ;ELSE REPORT
2964 0275266 104443 TRAP TSERCODE
2965 0275332 014220 .WORD 902
2966 0275334 104032 EMT CSEXT ;EXIT
2967 0275336 000170 .WORD L10033- ;EXIT
2968 0275422 032737 000001 002466 7S: MOV #WDRDV,R3 ;CHECK READY SET
2969 0275500 001013 BIT #DRDVMASK,T.CS ;YES-SKIP
2970 0275522 005301 RNE ICS ;ELSE DEC WAIT COUNT
2971 0275566 001404 DEC R4 ;SKIP IF 0
2972 0275566 000001 WAITUS #1,R0
2973 0275622 104027 MOV #R0 ;NO-SKIP
2974 0275624 000747 EMT CSETU
2975 0275666 104443 ERHRD 903 ;REPORT READY ERROR
2976 0275700 001607 TRAP TSERCODE
2977 .WORD 903
    
```

```

2973 0275722 013272 .WORD ERR5
2974 0275774 104032 EMT TST
2975 0275776 000066 EMT CSEXT
2976 0275776 000066 .WORD L10033-
2977 0276000 005737 002466 12S: TST T.CS ;TEST IF ANY ERROR
2978 0276002 100005 BEQ ERHRD 904 ;NO-SKIP
2979 0276006 104443 TRAP TSERCODE
2980 0276110 001610 .WORD 904
2981 0276112 013342 ERHRD 904 ;ELSE REPORT
2982 0276114 000001 EMT CSEXT ;EXIT
2983 0276116 104032 .WORD L10033- ;EXIT
2984 0276166 000546 MOV #WDRDV,R3 ;SET NAME MESSAGE PTR
2985 0276224 004737 020530 JSR PC,POSHSR ;POSITION HEAD SELECT BIT FOR TEST
2986 0276300 020537 002534 CMP RS,DESHD ;CHECK IF CORRECT HEAD SELECTED
2987 0276334 001433 BEQ ERHRD 905 ;YES-SKIP
2988 0276336 001405 TRAP TSERCODE ;ELSE TEST IF 1 DESIRED
2989 0276422 001405 BEQ ERHRD 905 ;NO-REPORT SB 1
2990 0276444 104443 TRAP TSERCODE ;ELSE REPORT SB 1
2991 0276446 001611 .WORD 905
2992 0276500 013154 .WORD ERR3
2993 0276522 000010 EMT CSEXT ;EXIT
2994 0276524 104032 .WORD L10033- ;EXIT
2995 0276566 000010 ERHRD 906 ;EXIT
2996 0276566 104443 TRAP TSERCODE
2997 0276600 001612 .WORD 906
2998 0276664 013106 .WORD ERR2
2999 0276664 20S:
3000 0276664 6S:
3001 0276664 ENDTST
3002 0276664 L10033: EMT CSETST
3003 0276664 104001
    
```



```

2992      027666
2993      027666
2994      027666
2995      027674 012737 006405 002434      MOV     #T1ERR,ERHEAD ;SET ERROR HEADER      T10.1:
2996      027700 005021      MOV     #NEWV,LR1      ;CLEAR NEW CYL
2997      027702 005021      CLR    (R1)+           ;CLEAR CURRENT CYLINDER
2998      027704 005021      CLR    (R1)+           ;CLEAR DIFFERENCE
2999      027706 052731      RTS    #R10,(R1)+     ;SET FOR SIGN OF 1
3000      027712 004737 020530      JSR    PC,POSRSB      ;GET SELECTED HEAD
3001      027716 010521      MOV     R5,(R1)+      ;SET AS DESTRED HEAD
3002      027720
3003      027720      (3)
3004      027720      104002
3005      027722 004737 016446      EMT     C$RSUB        ;INITIALIZE TEST      T10.1:
3006      027724 004737 018464      JSR     PC,$STATR     ;GET STATUS
3007      027726 030166      60S
3008      027728 004737      JSR     PC,$IMSEK     ;DO SEEK
3009      027730 030166      60S
3010      027732 004737      JSR     #MRDRY,R3     ;SET NAME MESSAGE PTR
3011      027734 004737      JSR     #CDRDY,R4     ;SET CONDITION MESSAGE PTR
3012      027736 004737      JSR     PC,$STAT      ;GET STATUS
3013      027740 030166      BIT     #DRDVM$K,T.CS ;CHECK READY RESET
3014      027742 001405      BEQ    45              ;YES-SKIP
3015      027744 104443      ERHRD  1001,ERR4      ;REPORT READY ERROR
3016      027746 001751      TRAP   T$ERRCODE
3017      027748 013222      .WORD  ERR4
3018      027750      EXIT  SUB             ;EXIT SUBTEST
3019      027752      EMT     L10035-.
3020      030002 012701 000121      4$:    MOV     #R1,R1      ;SET WAIT COUNT
3021      030004 004737 016514      JSR     PC,$STAT      ;GET STATUS
3022      030006 030166      60S
3023      030008 012703      MOV     #R3,R3        ;SET EXPECTED STATE
3024      030010 001405      CMP    R3,T$STAT      ;CHECK STATE IS 5
3025      030012 001405      BND    1002,ERR7      ;YES-SKIP
3026      030014 104443      ERHRD  1002,ERR7      ;REPORT STATE ERROR
3027      030016 014220      TRAP   T$ERRCODE
3028      030018 1002      .WORD  1002
3029      030020      EXIT  SUB             ;EXIT
3030      030022      EMT     C$EXIT
3031      030024      .WORD  L10035-.
3032      030026 001405      BPL    T$CS          ;SET NAME MESSAGE PTR
3033      030028 032737 000001 002466 7$:    BIT     #DRDVM$K,T.CS ;CHECK READY SET
3034      030030 001013      BNE    12S           ;YES-SKIP
3035      030032 001404      DEC    R1            ;DO WAIT COUNT
3036      030034 001404      BEG    #1            ;SKIP IF 0
3037      030036 012700 000001      MOV     #R0,R0
3038      030038 104027      EMT     C$WTU
    
```

```

3033      030066 000747      BR     5$
3034      030070
3035      030072 104443      9$:    ERHRD  1003,ERR5      ;REPORT READY ERROR
3036      030074 013272      TRAP   T$ERRCODE
3037      030076 010432      .WORD  1003
3038      030078 005737 002466      12$:   ERHRD  1004,ERR6      ;TEST IF ANY OTHER ERROR
3039      030080 100005      TRAP   T$ERRCODE
3040      030082 104443      ERHRD  1004,ERR6      ;REPORT ALL ERRORS
3041      030084 001754      TRAP   T$ERRCODE
3042      030086 013342      .WORD  1004
3043      030088 104032      EXIT  SUB             ;EXIT
3044      030090 000046      EMT     C$EXIT
3045      030092      .WORD  L10035-.
3046      030122 012703 011044      15$:   MOV     #M$STA,R3     ;SET NAME MESSAGE PTR
3047      030124 004737 020530      JSR     PC,POSRSB     ;GET SELECTED HEAD BIT
3048      030126 001405      BEQ    20S           ;CHECK IF CORRECT
3049      030128 005737 002534      TST    DE$SHD        ;WAS IT SET
3050      030130 001405      BEG    17S           ;NO-SKIP
3051      030132 104443      ERHRD  1005,ERR3      ;REPORT SR 1
3052      030134 001755      TRAP   T$ERRCODE
3053      030136 013154      .WORD  1005
3054      030138 104032      EXIT  SUB             ;EXIT
3055      030140 000010      .WORD  L10035-.
3056      030142 104443      17$:   ERHRD  1006,ERR2      ;REPORT SR 0
3057      030144 001756      TRAP   T$ERRCODE
3058      030146 013166      .WORD  1006
3059      030148      EXIT  SUB             ;EXIT
3060      030150      EMT     C$SETST
3061      030152      .WORD  L10034:
3062      030154 104003      EMT     C$ESGN
3063      030156 005737 002532      BEG    25S           ;CHECK IF BOTH SIGN USED
3064      030158 001404      CLP    DE$SGN        ;YES-SKIP
3065      030160 000137 027720      CLP    DE$SGN        ;SET FOR SIGN OF 0
3066      030162 000137 027720      JMP    T104$         ;DO TEST AGAIN
3067      030164 002066      25$:   ENDTST
3068      030166 104001      L10034: EMT     C$SETST
    
```

```

3062
3063
3064
3065      030210      BRTTL *TEST 11      HEAD ALIGNMENT SUPPORT
3066      030210      BGNSTST *TEST 11
3067      030210      BIT #HDALGN,MISWIW ;CHECK IF RUN HEAD ALIGNMENT T11.1:
3068      030210      BFC 1S ;NO-EXIT
3069      030210      BNE 1S ;TEST IF PASS 0
3070      030226      CMP RLDV,HADONE ;TEST IF HEAD ALIGN DONE THIS DRIVE
3071      030238      JMP 2115S ;NO-SKIP
3072      030242      EXIT TST ;GO CHECK WRITE LOCK
3073      030242      EMT CSEXT ;
3074      030246      MOV RLDV,HADONE ;SET HEAD ALIGN DONE FLAG
3075      030254      PRINTF #FMT5,#BASADD,RLBAS,#DRVNAM,<8,RLDRV+1>
3076      030254      CLR -(SP)
3077      030254      BTR #DRV+1,(SP)
3078      030254      MOV #DRVNAM,-(SP)
3079      030254      MOV RLBAS,-(SP)
3080      030254      MOV #BASADD,-(SP)
3081      030254      MOV #5,-(SP)
3082      030254      MOV SP,RO
3083      030254      EMT C$PNTF
3084      030254      END SP
3085      030254      PRINTF #FMT9,#HAMES1 ;TYPE INSTRUCTIONS
3086      030254      MOV #HAMES1,-(SP)
3087      030254      MOV #FMT9,-(SP)
3088      030254      MOV #2,-(SP)
3089      030254      MOV SP,RO
3090      030254      EMT C$PNTF
3091      030254      ADD #6,SP
3092      030254      PRINTF #FMT9,#HAMES2
3093      030254      MOV #HAMES2,-(SP)
3094      030254      MOV #FMT9,-(SP)
3095      030254      MOV #2,-(SP)
3096      030254      MOV SP,RO
3097      030254      EMT C$PNTF
3098      030254      ADD #6,SP
3099      030362      BGN$UR 000014
3100      030362      EMT C$BSUR T11.2:
3101      030362      JSR PC,T$TINT ;INITIALIZE TEST
3102      030362      CLR DONE ;CLEAR DONE
3103      030362      MOV RLDV,L,CS ;SET UP FOR GET STATUS
3104      030362      BTR #GETSTAT,L,CS
3105      030362      MOV #GETSTAT,DRSET,L,DA
3106      030362      MOV L,DA,RLDA(R2) ;DO GET STATUS
3107      030362      MOV #0,RLCSR(R2) ;WAIT FOR INTERRUPT
3108      030362      WAITMS #50,RO
3109      030362      MOV #50,RO
3110      030362      EMT C$PNTF
3111      030362      TST DONE ;CHECK IF DONE
3112      030362      BEQ 3S ;NO-GO CLR CONTROLLER
    
```

```

3089
3090
3091      030457      012737      000021      002462      MOV #H$SELI,MBSETC,L,DA ;LOAD FOR HEAD 1
3092      030460      020000      020000      002474      BIT #W$STAT,T,MP ;CHECK IF WRITE LOCK SET
3093      030466      001003      000000      000000      BFC 1S ;YES-SKIP
3094      030470      042737      000020      002462      BTR #H$SEL,L,DA ;ELSE CLEAR TO HEAD 0
3095      030476      013737      002454      002456      MOV RLDV,L,CS ;LOAD IN DRIVE NUMBER
3096      030504      052737      000106      002456      DIS #SEEK,L,CS ;SET FOR SEEK
3097      030520      013762      002452      000004      MOV L,CS,RLCSR(R2) ;LOAD & EXECUTE SEEK
3098      030520      013762      002456      000000      MOV L,CS,RLCSR(R2)
3099      030526      012700      000036      WAITMS #30,RO ;WAIT FOR INTERRUPT
3100      030526      042000      000000      MOV #30,RO
3101      030534      000715      EMT C$PNTF
3102      030534      BR 3S ;LOOP
3103      030536      104003      59$:
3104      030536      104003      ENDSUB L10037:
3105      030536      104003      EMT C$ESUR
3106      030540      104002      T11.2:
3107      030540      004737      016446      EMT C$BSUR
3108      030540      004737      016464      JSR PC,T$TINT ;INITIALIZE TEST
3109      030540      004737      016464      JSR PC,G$STATR ;CLEAR DRIVE
3110      030544      032938      020000      002474      BTR #W$STAT,T,MP ;CHECK WRITE LOCK RESET
3111      030544      001425      000000      000000      BFC 1S ;YES-SKIP
3112      030544      012746      010377      19$: PRINTF #FMT9,#OPR12 ;REQUEST WRITE LOCK RESET
3113      030544      012746      012327      MOV #OPR12,-(SP)
3114      030544      012746      000002      MOV #FMT9,-(SP)
3115      030544      016600      MOV #2,-(SP)
3116      030544      104017      MOV SP,RO
3117      030544      062706      ADD #6,SP
3118      030544      005037      CLR DBUFF ;CLEAR FOR RESPONSE
3119      030544      104043      G$ANIL OPR002,DBUFF,1,NO ;GET RESPONSE
3120      030544      000404      EMT C$CMAN
3121      030544      004066      BR 10000$
3122      030544      000120      .WORD DBUFF
3123      030544      000120      .WORD T$CODE
3124      030544      000001      .WORD OPR002
3125      030544      000001      .WORD 1
3126      030544      005737      004066      10000$: TST DBUFF ;WAS ANSWER YES
3127      030544      001753      001753      REQ 19$ ;NO-REPEAT REQUEST
3128      030544      104003      19$:
3129      030544      104003      ENDSUB L10040:
3130      030544      104003      EMT C$ESUB
3131      030544      104001      20$:
3132      030544      104001      FNDST L10036:
3133      030544      104001      EMT C$ETST
    
```

```

3121
3122
3123
3124 030642          BRTTL *TEST 12          HEAD SWITCHING
3125 030642          BGNTST          *TEST 12
3126 030642          012737 006425 002434 MOV #T12ERR,ERHEAD ;SET ERROR HEADER T12:
3127 030650          012701 002524 MOV #HEVCYL,R1 ;GET POINTER TO DESIRED LOCATION
3128 030654          005021 CLR (R1)+ ;CLEAR NEW CYLINDER
3129 030656          005021 CLR (R1)+ ;CLEAR CURRENT CYL.
3130 030662          005021 CLR (R1)+ ;CLEAR DIFFERENCE
3131 030664          012721 000001 MOV #1,(R1)+ ;CLEAR SIGN
3132 030670          T1243: ;SET FOR HEAD 1
3133 030670          BGNSUR
3134 030670          104002 EMT CSBSUR ;INITIALIZE TEST T12.1:
3135 030672          004737 JSR PC,TSTINT ;GET STATUS WITH RESET
3136 030702          031136 JSR PC,GSTATR ;DO SEEK
3137 030704          004937 JSR PC,SIMSEK ;DO SEEK
3138 030710          031136 MOV #DRDY,R3 ;SET NAME MESSAGE PTR
3139 030716          012704 MOV #DRDY,R4 ;SET CONDITION POINTER
3140 030722          004737 JSR PC,GSTAT ;GET STATUS
3141 030726          031136 BIT #DRDYMASK,T.CS ;CHECK IF READY
3142 030736          001405 BEQ ERRHRD ;NO-SKIP
3143 030740          104443 TRAP TSEPCODE ;REPORT READY ERROR
3144 030744          013222 .WORD L201 ;ERR4
3145 030746          031136 .WORD L201 ;ERR4
3146 030746          031136 EXIT SUB ;EXIT
3147 030746          031136 EMT CSEEXIT ;EXIT
3148 030750          000186 .WORD L10042-.
3149
3150 030752          012701 000121 5S: MOV #R1,R1 ;SET WAIT COUNT
3151 030756          004737 6S: JSR PC,GSTAT ;GET STATUS
3152 030764          012704 COS ;SET EXPECTED STATE VALUE
3153 030770          020337 CMP R3,#.STAT ;CHECK IF STATE IS 5
3154 030774          001405 BEQ ERRHRD ;YES-SKIP
3155 030776          104443 TRAP TSEPCODE ;REPORT STATE ERROR
3156 031000          002262 .WORD L202 ;ERR7
3157 031002          014220 .WORD L202 ;ERR7
3158 031004          031136 EXIT SUB ;EXIT
3159 031006          031136 EMT CSEEXIT ;EXIT
3160          031006 .WORD L10042-.
3161
3162 031010          012703 010711 7S: MOV #DRDY,R3 ;SET NAME MESSAGE PTR
3163 031014          032737 BNE ERDYMASK,T.CS ;CHECK DRIVE READY
3164 031018          000001 002466 JSR PC,GSTAT ;GET STATUS
3165 031022          005301 DEC R5 ;DEC WAIT COUNT
3166 031024          001404 BEQ WITUS ;SKIP IF 0
3167 031030          012700 000001 MOV #1,R0
3168 031030          012700
    
```

```

3163 031034          104027 EMT CSWTU
3164 031036          000747 BR 6S
3165
3166 031040          9S: ERRHRD L203 ;ERR5 ;REPORT READY ERROR
3167 031042          002262 TRAP TSEPCODE
3168 031044          013272 .WORD L203
3169 031046          031136 EXIT SUB ;EXIT
3170 031050          104032 EMT CSEEXIT ;EXIT
3171 031050          000086 .WORD L10042-.
3172
3173 031052          005737 12S: TST T.CS ;TEST IF ANY ERROR
3174 031060          100005 BR 12S ;NO-SKIP
3175 031060          104443 ERRHRD L204 ;ERR6 ;REPORT ALL ERRORS
3176 031062          002262 TRAP TSEPCODE
3177 031064          013342 .WORD L204
3178 031066          031136 EXIT SUB ;EXIT
3179 031070          104032 EMT CSEEXIT ;EXIT
3180 031070          000386 .WORD L10042-.
3181 031076          004737 15S: MOV #MHSTA,R3 ;SET NAME MESSAGE PTR
3182 031076          020530 JSR PC,POSHS ;POSITION HEAD SELECT BIT
3183 031082          023705 CMP DESHD,R5 ;CHECK IF CORRECT HEAD SELECTED
3184 031084          005737 BEQ 20S ;YES-SKIP
3185 031086          001405 TST DESHD ;WAS HEAD 0 SELECTED
3186 031086          001405 BEQ ERRHRD ;YES-SKIP
3187 031086          104443 TRAP TSEPCODE ;REPORT HEAD SB 1
3188 031088          002262 .WORD L205 ;ERR3
3189 031090          013154 .WORD L205 ;ERR3
3190 031092          031136 EXIT SUB ;EXIT
3191 031094          104032 EMT CSEEXIT ;EXIT
3192 031094          000010 .WORD L10042-.
3193 031096          104443 17S: ERRHRD L206 ;ERR2 ;ELSE REPORT HEAD SB 0
3194 031100          002262 TRAP TSEPCODE
3195 031102          013106 .WORD L206 ;ERR2
3196 031104          013106 .WORD ERR2
3197
3198 031136          20S: ENDSUB L10042:
3199 031136          104003 EMT CSBSUR
3200 031140          005737 TST DESHD ;CHECK IF HD 0 WAS DONE
3201 031144          001404 BEQ 25S ;YES-SKIP
3202 031146          005737 JMP DESHD ;ELSE SET TO HEAD 0
3203 031148          000137 030870 JMC T1243 ;REDO TEST
3204 031150          25S: ENDTST L10041:
3205 031156          104001 EMT CSETST
    
```

```

3192
3193
3194
3195      031160          SRTTL      *TEST 13      READ HEADER (PART 1)
3196      031160          RCNTST      ;TEST 13
3197      031160          MOV          #T13ERR,ERHEAD      ;SET ERROR HEADER      T13::
3198      031160          MOV          #WECYL,R1      ;GET ADDRESS OF DESIRED LOCATIONS
3199      031172          CLR          (R1)+      ;CLEAR NEW CYL
3200      031174          CLR          (R1)+      ;CLEAR CURRENT CYL
3201      031176          CLR          (R1)+      ;CLEAR DIFF
3202      031178          CLR          (R1)+      ;CLEAR SIGN
3203      031180          CLR          (R1)+      ;CLEAR HEAD
3204      031204          T134$:
3205      031204          RGNSUB:
3206      031204          (3)          104002          EMT          CSBSUR      ;INITIALIZE TEST      T13.1:
3207      031206          004737          JSR          PC,TSTINT      ;GET STATUS W/RESET
3208      031210          004737          JSR          PC,GSTATR      ;DO SEEK
3209      031220          004737          JSR          PC,SIMSEK
3210      031224          031304          GOS          20$
3211      031226          012701          MOV          #R1,R1      ;SET WAIT COUNT
3212      031228          004737          JSR          PC,RDWAIT      ;WAIT FOR READY
3213      031230          031304          GOS          10$
3214      031240          004737          JSR          PC,XRDHDC      ;DO READ HEADER
3215      031242          012703          MOV          #MHSTA,R3      ;SET NAME MESSAGE PTR
3216      031244          004737          JSR          PC,POSHW1      ;POSITION HS BIT IN HD WRD 1
3217      031246          020522          CMP          #5,DESHD      ;CHECK IF HEAD CORRECT
3218      031248          011410          BNE          15$
3219      031250          011410          ERHRD      1301,ERR3      ;REPORT SR 1
3220      031264          104443          TRAP        TSERCODE
3221      031266          004225          .WORD      1301
3222      031268          013194          .WORD      ER3
3223      031270          104032          EXIT
3224      031272          000016          EMT          CSEXIT
3225      031274          000016          .WORD      1302,ERR2      ;REPORT SR 0
3226      031276          104443          ERHRD      1302,ERR2
3227      031300          002426          TRAP        TSERCODE
3228      031302          013106          .WORD      1302
3229      031302          .WORD      ERR2
3230      031304          15$:
3231      031304          GOS          15$
3232      031304          ENDSUB:
3233      031304          L10044:
3234      031304          104003          EMT          CSESUB
3235      031306          005737          TST          DESHD      ;TEST IF HEAD 1 DONE
3236      031314          011957          BNE          20$          ;YES-SKIP
3237      031322          012737          MOV          #1,DESHD      ;ELSE SET TO HEAD 1
3238      031324          002474          HDWRD1,TEMPO      ;STORE HDR WORD 1
3239      031326          000725          BR          T134$      ;DO TEST AGAIN
3240      031330          100177          BIC          #CHDCYL,TEMPO      ;CLEAR ALL BUT CYLINDER IN 1ST HEADER
3241      031332          100177          BIC          #CHDCYL,HDWRD1      ;CLEAR ALL BY CYL IN 2ND HEADER
3242      031334          042737          CMP          TEMPO,HDWRD1      ;COMPARE IF EQUAL
3243      031346          002540          002474
    
```

```

3235      031354          001405          RFO          22$
3236      031356          012703          MOV          #CYLPER,R3      ;YES-SKIP
3237      031362          104443          ERHRD      1306,ERR1      ;SET NAME MESSAGE PTR
3238      031364          012703          TRAP        TSERCODE      ;REPORT HEAD ALIGNMENT PROBLEM
3239      031366          013040          .WORD      1306
3240      031368          013040          .WORD      ERR1
3241      031370          22$:
3242      031370          ENDSUB:
3243      031370          L10043:
3244      031370          104001          EMT          CSETST
    
```



```

3306 031646 BGNMOD HRDPPM
3307 031646 BGNHRD
(3) 031646 000025
3308 031650 GPRML .WORD L10047-LSHARD/2
(4) 031650 004130 .WORD CNTYPE,CNT,YES
(4) 031652 031764 .WORD TSCODE
(4) 031652 000001 .WORD CNTYPE
3309 031653 GPRMA .WORD CSRMMSG,CSR,0,160000,177776,YES
(4) 031656 000031 .WORD TSCODE
(4) 031660 031722 .WORD CSRMMSG
(4) 031663 160000 .WORD TSCODE
(4) 031663 177776 .WORD TSHILIM
3310 031666 GPRMA .WORD VECMSG,VECT,0,0,776,YES
(4) 031666 001031 .WORD TSCODE
(4) 031670 031736 .WORD VECMSG
(4) 031672 000000 .WORD TSCODE
(4) 031674 000776 .WORD TSHILIM
3311 031676 GPRMD .WORD BRMSG,PRIOR,0,340,0,7,YES
(4) 031676 032032 .WORD TSCODE
(4) 031678 031736 .WORD BRMSG
(4) 031702 000340 .WORD TSCODE
(4) 031704 000000 .WORD TSHILIM
(4) 031706 000007 .WORD TSHILIM
3312 031710 GPRMD .WORD DRMSG,DRSR,0,340,0,7,YES
(4) 031712 003032 .WORD TSCODE
(4) 031712 031736 .WORD DRMSG
(4) 031714 003400 .WORD TSCODE
(4) 031720 000007 .WORD TSHILIM
3313
3314 031722 ENDHRD
(3) 031722 L10047: .EVEN
3315
3316 031722 052502 020123 042101 CSRMMSG: .ASCIZ /BUS ADDRESS/
(4) 031722 022504 061502 030123
3317 031736 042526 052103 051117 VECMSG: .ASCIZ /VECTOR/
(4) 031744 000
3318 031745 042512 020122 042514 BRMSG: .ASCIZ /RR LEVEL/
(4) 031756 051104 053111 000105 DRMSG: .ASCIZ /DRIVE/
3319 031764 046122 030461 000 CNTYPE: .ASCIZ /RL11/
3320 031771 ENDMOD
(4) 031771 031772 .EVEN
3321
3322
3323 031772 BGNMOD SFTPRM
3324 031772 BGNMMSG .WORD L10050-LSSOFT/2
3325 031772
3326
3327 031774 GPRML .WORD SELQ,WISWI,4,YES
(4) 031774 000130 .WORD TSCODE
(4) 032000 000004 .WORD SELQ
3333 032002 GPRML .WORD ALGNQ,WISWI,10,YES
(4) 032002 000130 .WORD TSCODE
(4) 032004 032071 .WORD ALGNQ
    
```

```

(4) 032006 000010
3335 032010 GPRML .WORD 10
(4) 032010 000130 .WORD MANG,WISWI,100000,YES
(4) 032012 032130 .WORD TSCODE
(4) 032014 100000 .WORD MANG
3336 032016 3S: .WORD AUTOQ,MISWI,20,YES
(4) 032016 004052 .WORD TSCODE
(4) 032022 006177 .WORD ERLIMQ
(4) 032024 000000 .WORD 377
(4) 032026 000377 .WORD TSHILIM
3352 032030 GPRML .WORD AUTOQ,MISWI,20,YES
(4) 032030 000130 .WORD TSCODE
(4) 032032 032216 .WORD AUTOQ
(4) 032034 000020 .WORD 20
3353 032036 ENDSFT
(3) 032036 L10050: .EVEN
3354
3356 032036 054105 041505 052125 SELQ: .ASCIZ /EXECUTE DRIVE SELECT TESTS/
(4) 032036 050105 042503 042511
(4) 032036 052103 052040 051505
3361 032036 051524 0000 .WORD 100000
(4) 032076 042524 044040 052503 ALGNQ: .ASCIZ /EXECUTE HEAD ALIGNMENT SUPPORT/
(4) 032104 020104 046101 043511
(4) 032112 046516 047105 020124
(4) 032126 052523 050120 051117
3363 032130 054105 041505 052125 MANG: .ASCIZ /EXECUTE MANUAL INTERVENTION TESTS/
(4) 032136 020105 040515 052516
(4) 032152 046101 044440 052116
(4) 032160 051105 042526 052116
(4) 032166 047511 020116 042524
3371 032166 052123 000153 043111 ERLIMQ: .ASCIZ /SPECIFY ERROR LIMIT/
(4) 032170 050105 041505 047522
(4) 032206 020122 044514 044515
3372 032218 000124 .WORD 042040
(4) 032224 044522 047526 044440
(4) 032232 020106 047516 051040
(4) 032240 051505 047520 051516
(4) 032246 000105
3376 032250 ENDMOD .EVEN
3377
3378
3379
3380
3381 032514 .=32514
3382
3383 ;AREA RESERVED AS PATCH AREA FOR DIAGNOSTICS.
3384 ;-32514 WAS SELECTED AS "LASTAD" TO PROVIDE APT TO LSI-11 COMPATIBILITY.
3385 ;BIT 7 OF "LASTAD" MUST BE CLEARED TO ACHIEVE A VALID MAILBOX ADDRESS
3386 ;WHEN RUNNING ON THE LSI-11 UNDER APT.
    
```

ASSEMBLY ROUTINES MACV11 30A(1052) 22-NOV-78 16:20 PAGE 3-2
CZRLCB.PT2 22-NOV-78 16:19 *TEST 14 READ HEADER (PART 2)

SEQ 0113

3387
3389
3399
3400 032514 LASTAD
(2) L\$LAST:: -EVEN
(3) 032514

ASSEMBLY ROUTINES MACV11 30A(1052) 22-NOV-78 16:20 PAGE 4
CZRLCB.SUP 23-OCT-78 09:52 DIAGNOSTIC SUPERVISOR -- LOW CORE SET UP

SEQ 0114

3402			.SBTTL	DIAGNOSTIC SUPERVISOR -- LOW CORE SET UP
14273	063310	000900	.WORD	0 ;SPACE FOR USER POOL POINTER
14274	063312	060900	.WORD	0 ;SIZE
14275	063314	000000	.WORD	0 ;CHECKSUM (NOT CURRENTLY USED)
14276	063316	000000	.WORD	0 ;SIZE OF H.W. PTAB. ALLOCATION
14277		063322	END.SUPV=.	+2
14278		000200	.END	200

ASSEMBLY ROUTINES MACV11 30A(1052) 22-NOV-78 16:20 PAGE 5-4
CZRLCB.SUP 23-OCT-78 09:52 SYMBOL TABLE

SEQ 0119

ERRORS DETECTED: 0

DSKZ: CZRLCB, DSKZ: CZRLCB/EO: PART1=CZRLCB/ML, CZRLCB.PT1, CZRLCB.P11, CZRLCB.PT2, CZRLCB.SUP
RUN-TIME: 50.491 SECONDS
RUN-TIME RATIO: 177/101=1.7
CORE USED: 16K (31 PAGES)