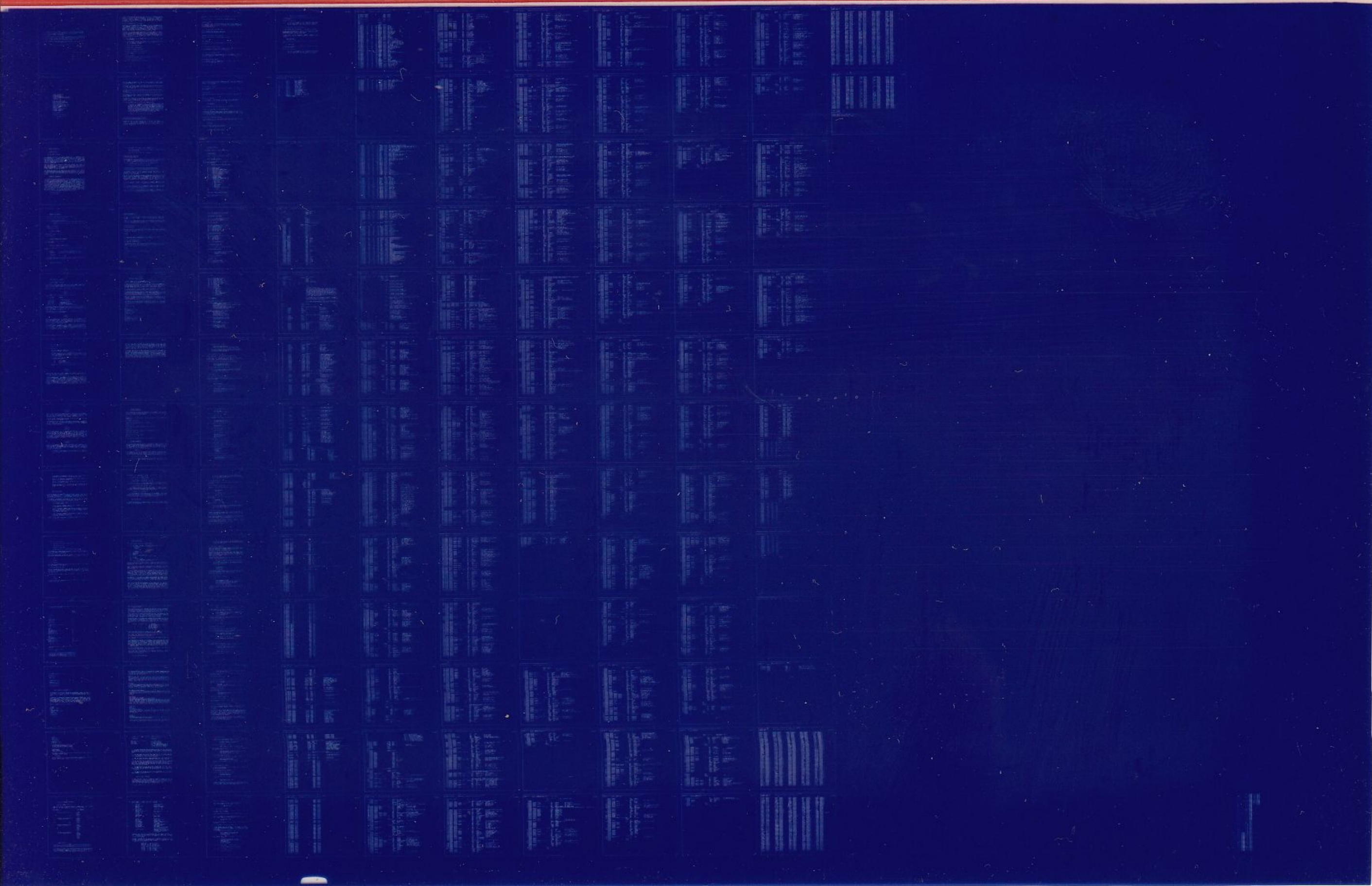


RL11,RLV11

RL01,02 DR TST 1
CZRLIA0

AH-F118A-MC
COPYRIGHT 1979
FICHE 1 OF 1

MAY 1979
digital
MADE IN USA



IDENTIFICATION

0 1

SEQ 0001

PRODUCT CODE: AC-#119A-MC
PRODUCT NAME: CZRLIAO RLO1/02 DRIVE TEST PART 1
DATE CREATED: 5-JAN-79
MAINTAINER: DIAGNOSTIC ENGINEERING
AUTHOR: D. DEKNIS, C. CAMPBELL

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) 1979, DIGITAL EQUIPMENT CORPORATION

TABLE OF CONTENTS

1.0	GENERAL INFORMATION
1.1	PROGRAM ABSTRACT
1.2	SYSTEM REQUIREMENTS
1.3	RELATED DOCUMENTS AND STANDARDS
1.4	DIAGNOSTIC HIERARCHY PREREQUISITES
1.5	ASSUMPTIONS
2.0	OPERATING INSTRUCTIONS
2.1	HOW TO RUN THIS DIAGNOSTIC
2.1.1	THE SIX STEPS OF EXECUTION
2.1.2	SAMPLE RUN-THROUGH
2.2	HOW TO CREATE A CHAINABLE FILE
2.3	DETAILS OF COMMANDS AND SYNTAX
2.3.1	TABLE OF COMMAND VALIDITY
2.3.2	COMMAND SYNTAX
2.4	EXTENDED P-TABLE DIALOGUE
2.5	HARDWARE PARAMETERS
2.6	SOFTWARE PARAMETERS
3.0	ERROR INFORMATION
4.0	PERFORMANCE AND PROGRESS REPORTS
5.0	DEVICE INFORMATION TABLES
6.0	TEST SUMMARIES

1.0 GENERAL INFORMATION
-----1.1 PROGRAM ABSTRACT
-----1.1.1 STRUCTURE OF PROGRAM

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

WHEN THIS DIAGNOSTIC IS STARTED AT ADDRESS 200, CONTROL GOES FIRST TO THE SUPERVISOR PORTION, WHICH WILL ASK CERTAIN "HARD CORE" QUESTIONS ABOUT THE ENVIRONMENT. THEN IT WILL ENTER COMMAND MODE, INDICATED BY A PROMPT CHARACTER (DS B>). AT COMMAND MODE THE OPERATOR MAY ENTER ANY OF SEVERAL COMMANDS AS DESCRIBED BELOW.

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

1.1.2 DIAGNOSTIC INFORMATION

THIS PROGRAM TESTS AND EXERCISES RL01/02 DISK DRIVES RL11/RLV11 CONTROLLERS (4 DRIVES PER CONTROLLER). THE ENTIRE PROGRAM IS RUN ON THE FIRST DRIVE BEFORE STARTING ON THE SECOND. THE PROGRAM STARTS BY TESTING THE SIMPLEST FUNCTIONS FIRST USING THE LOGIC TESTED IN EARLIER TESTS TO TEST MORE COMPLEX FUNCTIONS. THIS PROGRAM TESTS THE RL01/02 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.)
- 1 OR 2 RL11/RLV11 CONTROLLER(S) WITH:
 - 1 - 8 RLO1 DRIVES WITH RLO1K CARTRIDGES CONTAINING A 'BAD SECTOR FILE'
 - 1 - 8 RLO2 DRIVES WITH RLO2K CARTRIDGES CONTAINING A 'BAD SECTOR FILE'
- KW11P, KW11L (OPTIONAL)
- LINE PRINTER (OPTIONAL)

1.2.2 SOFTWARE REQUIREMENTS

CZRLIAO RLO1/02 DRIVE TEST PART 1
(FORMERLY CZRLCB)

1.3 RELATED DOCUMENTS AND STANDARDS

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

1.4 DIAGNOSTIC HIERARCHY PREREQUISITES

THE RLO1/02 SUBSYSTEM SHOULD HAVE SUCCESSFULLY RUN THE FOLLOWING PROGRAMS:

CVRLAAO	RLV11 RLO1 DISKLESS TEST (RLV11 ONLY)
CZRLGAO	RL11/RLV11 RLO1/02 CONTROLLER TEST (PART 1)
CZRLHAO	RL11/RLV11 RLO1/02 CONTROLLER TEST (PART 2)

1.5 ASSUMPTIONS

THE HARDWARE OTHER THAN THE RLO1/02 SUBSYSTEM IS ASSUMED TO WORK PROPERLY. FALSE ERRORS MAY BE REPORTED IF THE PROCESSOR, ETC., DO NOT FUNCTION PROPERLY.

2.0 OPERATING INSTRUCTIONS

2.1 HOW TO RUN THIS DIAGNOSTIC

2.1.1 THE SIX STEPS OF EXECUTION

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

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

* STEP 1 *

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

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

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

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

* STEP 2 *

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

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

ALL YOU NEED TO SAY IS SOMETHING LIKE THIS:

STA/PASS:1/FLAGS:HOE

THINGS TO NOTE HERE:

1. ONLY THE FIRST THREE CHARACTERS OF THIS OR ANY COMMAND AT THE "DS-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 ON ERROR
HOE	HALT ON ERROR
IER	INHIBIT ERROR PRINTOUT

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

* STEP 3 *

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

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

* STEP 4 *

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

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

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

* STEP 5 *

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

* STEP 6 *

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

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

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

HOE SET: THE ERROR WILL BE REPORTED ON THE CONSOLE AND THE DIAGNOSTIC WILL RETURN TO COMMAND MODE.

LOE SET: THE DIAGNOSTIC WILL LOOP ENDLESSLY ON THE BLOCK OF CODE THAT DETECTED THE ERROR.

NEITHER HOE NOR LOE SET: THE ERROR WILL BE REPORTED ON THE CONSOLE AND NORMAL EXECUTION WILL RESUME AS IF NO ERROR HAD OCCURED.

2.1.2 SAMPLE RUN-THROUGH

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

IF AN ERROR IS ENCOUNTERED, THE ERROR WILL BE REPORTED AND THE PROMPT WILL BE REISSUED (BECAUSE THE HOE FLAG IS SET). AT THIS POINT THERE ARE FOUR DIFFERENT WAYS YOU CAN GET THE PROGRAM GOING AGAIN:

1. ISSUE ANOTHER "START" COMMAND (THUS GOING THRU ALL OF STEPS 2, 3, 4, 5, AND 6 AGAIN)
2. ISSUE A "RESTART" COMMAND (SAME AS START COMMAND EXCEPT THAT THE HARDWARE QUESTIONS ARE NOT ASKED)
3. ISSUE A "CONTINUE" COMMAND (EXECUTION WILL RESUME AT THE BEGINNING OF THE PARTICULAR HARDWARE TEST (MOST DIAGNOSTICS CONSIST OF A NUMBER OF THESE) THAT IT WAS IN WHEN THE ERROR HALT OCCURED. NO QUESTIONS ASKED.
4. ISSUE A "PROCEED" COMMAND: EXECUTION WILL RESUME AT THE INSTRUCTION FOLLOWING THE ERROR REPORT (THIS IS A SPECIAL COMMAND AND CAN BE ISSUED ONLY AT A HALT)

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

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

THIS WILL DO THE FOLLOWING:

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

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

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

1. START
2. RESTART
3. CONTINUE

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

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

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

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

1. START
2. RESTART

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

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

	BY WHOM ENTERED: -----
.R CZRLIA	O
CZRLI	D
L-CLK (L) N ? Y	D,O
SOMZ (L) N ?	D
LSI (L) N ?	D
LPT (L) N ?	D
MEM (K) (D) 16 ?	D
DS-B>STA/PASS:1/FLAGS:HOE	D,O
# UNITS (D) ? 2	D,O
UNIT 1	D
RL11 (L) Y ?	D,O
BUS ADDRESS (O) 174400 ?	D,O
VECTOR (O) 160 ?	D,O
DRIVE (O) 0 ?	D,O
DRIVE TYPE = RL01 (L) Y ?	D,O
BR LEVEL (O) 5 ?	D,O
UNIT 2	D
RL11 (L) Y ?	D,O
BUS ADDRESS (O) 174400 ?	D,O
VECTOR (O) 160 ?	D,O
DRIVE (O) 0 ? 1	D,O
DRIVE TYPE = RL01 (L) ? N	D,O (N=RL02)
BR LEVEL (O) 5 ?	D,O
CHANGE SW (L) ? N	D,O
CZRLI HRD ERR 00004 TST 003 SUB 002 PC:004130 ERR HLT	
DS-B>PRO/FLAGS:IER:LOE:HOE=0	D,O

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

```

^C                                0
DS-B>CON/FLAGS:MOE:IER:LOE=0    D,0
CHANGE SW (L) ? N                D,0
CZRL1 EOP 1                       D
^C
DS-B>RESTART/PASS:1              D,0
CHANGE SW (L) ? N                D,0
-----
-----
-----
-----

```

2.2 HOW TO CREATE A CHAINABLE FILE

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

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

HERE IS A SAMPLE DIALOGUE TO ACCOMPLISH THIS:

```

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

```

DS-B>CCI
UNITS (D) ? 4

CHANGE SW (L) ? N
PTAB END: 60632

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

*MICORE 60632
CORE: 0,60632
*DUMP DK0: DIAG.BIC

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

.R DIAG.BIC

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

2.3 DETAILS OF COMMANDS AND SYNTAX

2.3.1 TABLE OF COMMAND VALIDITY

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

HOW ENTERED	LEGAL COMMANDS
-----	-----

- | | |
|---|--|
| 1. OPERATOR ENTERED 'RUN DIAG' | START
PRINT
DISPLAY
FLAGS
ZFLAGS |
| 2. DIAGNOSTIC HAS FINISHED ALL ITS REQUESTED PASSES | START
RESTART
PRINT
DISPLAY
FLAGS
ZFLAGS |
| 3. OPERATOR INTERRUPTED THE DIAGNOSTIC WITH CTRL/C | START
RESTART
CONTINUE
PRINT
DISPLAY
FLAGS
ZFLAGS |
| 4. AN ERROR WAS ENCOUNTERED WITH THE MOE FLAG SET SET | START
RESTART
CONTINUE
PROCEED
PRINT
DISPLAY
FLAGS
ZFLAGS |

2.3.2 COMMAND SYNTAX

```

*****
STA(RT)/TESTS:TEST-LIST/PASS:PASS-CNT/FLAGS:FLAG-LIST/EOP:EOP-INCR
*****

```

THE DIAGNOSTIC IN CORE IS EXECUTED IN ACCORDANCE WITH THE SWITCHES SPECIFIED. THE MESSAGE '# UNITS?' IS PRINTED. THE START COMMAND MAY BE ISSUED WHEN DIAGNOSTIC COMMAND MODE HAS BEEN ENTERED VIA ONE OF THE FOLLOWING: A) OPERATOR TYPED 'RUN DIAGNOSTIC' B) DIAGNOSTIC FINISHED EXE-

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

THE SWITCH ARGUMENTS ARE AS FOLLOWS:

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

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

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

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

IER INHIBIT ERROR REPORTING

IBE INHIBIT BASIC ERROR REPORTS

IXE INHIBIT EXTENDED ERROR REPORTS

PRI DIRECT ALL MESSAGES TO A LINE PRINTER

PNT PRINT NUMBER OF TEST BEING EXECUTED

BOE BELL ON ERROR

UAM RUN IN UNATTENDED MODE, BYPASSING MANUAL INTERVENTION TESTS

ISR INHIBIT STATISTICAL REPORTS

IDU INHIBIT DROPPING OF UNITS BY DIAGNOSTIC

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

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

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

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

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

THE SWITCH ARGUMENTS ARE AS IN THE START COMMAND EXCEPT:

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

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

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

THE SWITCH ARGUMENTS ARE AS IN THE START COMMAND EXCEPT:

1. DEFALT FOR PASS-CNT IS THE UNSATISFIED PASS-CNT FROM THE PREVIOUS START OR RESTART
2. UNSPECIFIED FLAG SETTINGS ARE UNCHANGED

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

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

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

1. UNSPECIFIED FLAG SETTINGS ARE UNCHANGED

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

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

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

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

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

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

DRO(P)/UNITS:UNIT-LIST

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

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

ADD/UNITS:UNIT-LIST

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

PRI(NT)

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

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

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

FLA(GS)

THE CURRENT SETTINGS OF ALL FLAGS ARE PRINTED.

ZFL(AGS)

ALL FLAGS ARE CLEARED.

2.4 EXTENDED P-TABLE DIALOGUE

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

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

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

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

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

FOLLOWING IS THE DIALOGUE FOR THIS 8 RLOX DRIVE SYSTEM. THIS SYSTEM HAS TWO (2) RL11 TYPE CONTROLLERS ALL TO BE SET AT "BR LEVEL" 5. THE FIRST 4 DRIVES ARE RLO1'S AND THE LAST 4 DRIVES ARE RLO2'S (ON THE SECOND CONTROLLER):

UNITS (D) ? 8

UNIT 1

RL11 (L) Y ?

BUS ADDRESS (O) 174400 ?

VECTOR (O) 160 ?

DRIVE (O) 0 ? 0-3

DRIVE TYPE - RLO1 (L) Y ?

BR LEVEL (O) 5 ?

UNIT 5

RL11 (L) Y ?

BUS ADDRESS (O) 174400 ? 175400

VECTOR (O) 160 ? 164

DRIVE (O) 0 ? 0-3

DRIVE TYPE = RLO1 (L) Y ? N

BR LEVEL (O) 5 ?

THE FIRST TIME THRU THE P-TABLE QUESTIONS THE DEFAULT VALUES ARE USED FOR THE CONTROLLER TYPE (QUESTION #1), CSR ADDRESS OF THE CONTROLLER (QUESTION #2), THE CONTROLLER VECTOR ASSIGNMENT (QUESTION #3), THE DRIVE TYPE (QUESTION #5), AND THE "BR LEVEL" (QUESTION #6). THE ACTUAL UNIT NUMBERS OF THE RLO1'S FOR QUESTION #4 WAS ASSIGNED 0 THRU 3 FOR THE FIRST 4 P-TABLE SLOTS.

THE SECOND TIME THRU THE P-TABLE QUESTIONS (FOR THE RLO2 ASSIGNMENT ON THE SECOND CONTROLLER), THE FIRST QUESTION DEFAULTED TO "RL11" TYPE CONTROLLER. THE SECOND QUESTION WAS ANSWERED TO REFLECT THE CHANGE IN CSR ADDRESS FOR THE RLO2 CONTROLLER (175400). THE SECOND CONTROLLER'S VECTOR WAS ALSO CHANGED TO 164 IN QUESTION #3. THE RLO2 TEST UNIT NUMBERS WERE ASSIGNED VALUES 0 TO 3 IN QUESTION #4 AND THE DRIVE TYPE WAS SET FOR RLO2'S FOR THE REMAINING 4 UNITS IN QUESTION #5. THE LAST QUESTION WAS DEFAULTED USING THE "BR LEVEL" FROM THE FIRST PASS.

2.5 HARDWARE PAPAMETERS

THE FOLLOWING QUESTIONS WILL BE ASKED ON A START COMMAND. THE VALUE LOCATED TO THE LEFT OF THE QUESTION MARK IS THE DEFAULT VALUE THAT WILL BE TAKEN ON A CARRIAGE RETURN RESPONCE.

RL11 (L) Y?

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

BUS ADDRESS (O) 174400?

ANSWER WITH THE BUS ADDRESS OF THE CONTROLLER.

VECTOR (O) 160?

ANSWER WITH THE INTERRUPT VECTOR OF THE CONTROLLER.

DRIVE (O) 0?

ANSWER WITH THE DRIVE(S) CONNECTED TO THE CONTROLLER

DRIVE TYPE = RL01 (L) ?

ANSWER NO (N) IF DRIVE IS AN RL02

BR LEVEL (O) 5?

ANSWER WITH THE INTERRUPT PRIORITY OF THE CONTROLLER.

2.6 SOFTWARE PARAMETERS

THE FOLLOWING QUESTIONS ARE ASKED IF REQUESTED ON A START, RESTART, OR CONTINUE. THEY ALLOW FLEXABILITY IN THE WAY THE PROGRAM BEHAVES. THE SOFTWARE PARAMETERS GIVE THE PROGRAM FLEXIBILITY IN THE WAY IT RUNS. THE PARAMETERS CAN BE MODIFIED ON A START, RESTART, OR CONTINUE BY ANSWERING (Y)ES TO THE FOLLOWING QUESTION:

CHANGE S.W. ?

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

EXECUTE DRIVE SELECT TESTS (N)?

IF "YES" TESTS 5 AND 6 ARE EXECUTED IN THE FIRST PASS OF THE PROGRAM. THESE TESTS REQUIRE MANUAL INTERVENTION TO CHANGE ADDRESS PLUGS

AND REQUIRE A FULL COMPLEMENT OF ADDRESS PLUGS (0 - 3).

EXECUTE HEAD ALIGNMENT SUPPORT (N)?

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

EXECUTE MANUAL INTERVENTION TESTS (N)?

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

SPECIFY ERROR LIMIT (DECIMAL) (20)?

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

DROP DRIVE IF NO RESPONSE (N)?

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

3.0 ERROR INFORMATION

ALL ERRORS ARE PRINTED VIA CONSOLE DEVICE. THE ERROR INCLUDES ERROR NUMBER, TYPE AND PROGRAM LOCATION. ERRORS INCLUDE REGISTERS BEFORE AND AT ERROR WITH RELEVANT DATA.

3.1 ERROR REPORTING

MOST ERROR REPORTS HAVE THE FOLLOWING FORMAT.

- (1) PRGG NAME ERR NUM TEST NUM SUBTEST NUM ERR PC
- (2) ROUTINE TRACE SEQ (IN SEQ CALLED)
- (ASSIGNMENT)
- (ASSIGNMENT)
- (ASSIGNMENT)
- (3) TEST DESCRIPTION
- (4) OPERATION:
- (5) RESULT:
- (6) ADDRESS OF UNIT UNDER TEST
- (7) RLCS RLDA RLBA RLMP CYL HD
- (8) OP INIT
- (9) OP DONE
- (10) DRIVE STATUS
- (11) WORD NUM IS (XXXXXX) SB (YYYYYY)
- (12) TOTAL COMPARE ERRS: (ZZZ) OF (128)

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

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

THE SUBTEST NUMBER IS UNIQUE IN THIS PROGRAM IN THAT IT DOES NOT REFER TO A PHYSICAL SUBTEST WITHIN A GIVEN TEST. RATHER IT REFLECTS THE NUMBER OF TIMES A SUBTEST HAS BEEN EXECUTED WITHIN A TEST. CONSEQUENTLY, ON A TEST THAT TESTS AN INCREMENTAL TYPE OF OPERATION (SUCH 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.

5.1.1 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	(HDR CRC OR DATA CRC ERROR BIT 11 OF CS REGISTER)
HDR NOT FND/DAT LATE	(HDR NOT FOUND OR DATA LATE ERROR BIT 12 OF CS REGISTER)
CYL	(CYLINDER WHEN REPORTING A SEEK ERROR)

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

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

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

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

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

"INTERRUPT TO LATE"

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

"FAIL TO RELOAD HEADS AFTER ERR CLEAR"

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

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

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

"WRITE ABORTED"

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

"COULD NOT RETRIEVE DRIVE STATUS"

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

"OPI SET-NO DRIVE RESPONSE"

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

"NO INTERRUPT ON CMND COMPLETE"

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

"ERR DID NOT CLEAR"

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

"DRV ERR IS NOT CLEARED"

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

"UNEXPECTED ERR"

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

"BAD SEC FILE FMT ERR"

THIS IS REPORTED IF THE CONTENTS OF THE FILES DO NOT CORRESPOND TO THE EXPECTED FORMAT. (REFER TO DEC STANDARD 144 FOR FORMAT SPECIFICS.)

3.1.3 OTHER MESSAGES

OTHER INFORMATION IS REPORTED UNDER VARIOUS CIRCUMSTANCES. THESE ARE:

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

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

"ERROR LIMIT EXCEEDED-UNIT DROPPED"

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

3.2 ERROR HALTS

ERROR HALTS ARE SUPPORTED PER DESCRIBED IN THE PREVIOUS SECTION WITH /FLAG:MOE. THERE ARE NO OTHER HALTS.

4.0 PERFORMANCE AND PROGRESS REPORTS

4.1 PERFORMANCE REPORTS

THIS PROGRAM WILL NOT GIVE ANY PERFORMANCE REPORTS.

4.2 PROGRESS REPORTS

THIS PROGRAM WILL NOT GIVE ANY PROGRESS REPORTS.

5.0 DEVICE INFORMATION TABLES

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

RLCS - CONTROL AND STATUS REGISTER (XXXXX0)

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

BIT 0 - DRIVE READY

RLBA - BUS ADDRESS REGISTER (XXXXX2)

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

RLDA - DISK ADDRESS REGISTER (XXXXX4)

FOR READ/WRITE FUNCTIONS

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

FOR SEEK FUNCTION

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

FOR GET STATUS FUNCTION

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

RLMP - MULTIPURPOSE REGISTER

FOR READ/WRITE FUNCTION

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

FOR READ HEADER FUNCTION

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

FOR GET STATUS FUNCTION

HAS DRIVE 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 - DRIVE TYPE IS RLO2 IF SET
 BIT 6 - SURFACE (0=UPPPER, 1=LOWER)
 BIT 5 - COVER OPEN
 BIT 4 - HEADS HOME
 BIT 3 - BRUSHES HOME
 BIT 2-0 - STATE BITS
 0 - LOAD STATE
 1 - SPIN UP
 2 - BRUSH CYCLE
 3 - LOAD HEADS
 4 - SEEK - TRACK COUNTING
 5 - SEEK - LINEAR MODE
 6 - UNLOAD HEADS
 7 - SPIN DOWN

6.0 TEST SUMMARIES

TEST 1 BASIC INTERFACE TEST (PART 1)

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

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

CHECK WRITE LOCK STATUS BIT SET. IF NOT:
 BAD SWITCH OR WRITE LOCK LOGIC
 DRIVE COMMAND SHIFT REGISTER

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

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

CHECK DRIVE ERROR RESET. IF NOT:

BAD DRIVE ERROR INTERFACE
SOME OTHER ERROR STUCK ON. REPORT WHICH ERROR.

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

TEST 2 BASIC INTERFACE TEST (PART 2)

REQUEST OPERATOR TO CLOSE COVER AND RESET WRITE LOCK.

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

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

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

IF NEITHER CHANGED, REPORT BOTH FAILURES.

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

TEST 3 HEAD LOADING TEST

REQUEST OPERATOR TO PRESS LOAD SWITCH.

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

IF LOAD WAS PRESSED:

BAD STATE ROM
BAD LOAD SWITCH OR LOGIC

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

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

AND CHECK IF SPINUP TIMEOUT ERROR SET. IF NOT:

BAD STATE ROM
BAD TIMEOUT DETECTION LOGIC

CHECK THAT STATE GOES TO 2. IF NOT:

BAD STATE ROM

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

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

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

BAD AC SERVO
BAD SWITCH OR LATCH

CHECK THAT STATE HAS CHANGED TO 3. IF NOT:

BAD STATE ROM

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

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

CHECK VOLUME CHECK IS SET. IF NOT:

BAD VOLUME CHECK LOGIC

CHECK IF DRIVE ERROR IS SET. IF NOT:

BAD DRIVE ERROR LOGIC OR INTERFACE

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

STATE ROM BAD
SEEK ROM
VEL ROM
GUARD BAND DETECTION

WAIT 15 MS FOR STATE TO CHANGE TO 5.

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

INTEGRATOR OR NULL DETECTION FAILURE
READY ONE SHOT BAD
ENABLE TIMEOUT W 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.

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
200, HEAD ALIGNMENT SUPPORT IS REQUESTED, AND IN THE FIRST
PASS ONLY.

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, MS 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.

TEST 15 DIFFERENCE OF 1 SEEK TEST (PART 1)

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

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

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

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

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

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

FAILURE IN READY LATCH OR INTEGRATOR

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

FAILURE IN INTEGRATOR

UNEXPECTED GUARD BAND DETECTED

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

REPEAT TEST USING HEAD 1.

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

TEST 16 DIFFERENCE OF 1 SEEK TEST (PART 2)

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

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

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

COUNT LOGIC BAD
INTERGRATOR FAILED

CHECK THAT HEADS MOVED FORWARD OR REVERSE AS EXPECTED. IF NOT:

SEEK ROM FAILED

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

REPEAT TEST USING HEAD 1.

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

3-	2	*TEST 1	BASIC INTERFACE (PART 1)
4-	1	*TEST 2	BASIC INTERFACE (PART 2)
5-	1	*TEST 3	HEAD LOADING
6-	1	*TEST 4	HEAD UNLOADING
7-	1	*TEST 5	DRIVE SELECT
8-	1	*TEST 6	DRIVE SELECT TEST
9-	1	*TEST 7	INITIAL STATE
10-	3	*TEST 8	INITIAL RESET STATE
11-	3	*TEST 9	DRIVE READY
12-	1	*TEST 10	SEEK SIGN SWITCH
13-	3	*TEST 11	HEAD ALIGNMENT SUPPORT
14-	3	*TEST 12	HEAD SWITCHING
15-	3	*TEST 13	READ HEADER (PART 1)
16-	1	*TEST 14	READ HEADER (PART 2)
17-	1	*TEST 15	DIFFERENCE OF 1 SEEK (PART 1)
18-	1	*TEST 16	DIFFERENCE OF 1 SEEK (PART 2)
19-	1	PARAMETER CODING	
20-	1	DIAGNOSTIC SUPERVISOR -- LOW CORE SET UP	

1

000001

PART1 -1

1			.NLIST	CND,MD,ME
2	000000		.ENABLE	ABS
3			.ENABLE	AMA
4		002000	.=2000	
5			.MCALL	SVC
6				
7	002000		SVC	
8		000001	SVCTST=1	
9		000001	SVC SUB=1	
10		000001	SVCBGL-1	
11		000000	SVCINS=0	
12		000000	SVCTAG=0	
13	002000		POINTER	BGNSW,BGNSFT,BGNDJ
14				
15	002000		BGNMOD	MDHEDR
17	002000		HEADER	CZRLI,A,0,1,1,1,RL01
	002000	103	.ASCII	/C/
	002001	132	.ASCII	/Z/
	002002	122	.ASCII	/R/
	002003	114	.ASCII	/L/
	002004	111	.ASCII	/I/
	002005	000	.BYTE	0
	002006	000	.BYTE	0
	002007	000	.BYTE	0
	002010	101	.ASCII	/A/
	002011	060	.ASCII	/O/
	002012	000000	.WORD	0
	002014	000001	.WORD	1
	002016	032570	.WORD	L\$HARD
	002020	032734	.WORD	L\$SOFT
	002022	013674	.WORD	L\$HW
	002024	013712	.WORD	L\$SW
	002026	033200	.WORD	L\$LAST
	002030	000000	.WORD	0
	002032	000000	.WORD	0
	002034	000000	.WORD	0
	002036	000000	.WORD	0
	002040	013730	.WORD	L\$DISPATCH
	002042	000000	.WORD	0
	002044	000000	.WORD	0
	002046	000000	.WORD	0
	002050	002	.BYTE	C\$REVISION
	002051	002	.BYTE	C\$EDIT
	002052	000001	.WORD	1
	002054	000001	.WORD	1
	002056	000000	.WORD	0
	002060	000000	.WORD	0
	002062	000000	.WORD	0
	002064	002114	.WORD	L\$DVTYP
	002066	000000	.WORD	0
	002070	002112	.WORD	L\$DR
	002072	002112	.WORD	L\$DRST
	002074	000000	.WORD	0
	002076	015256	.WORD	L\$DU
	002100	000014	.WORD	14
	002102	000000	.WORD	0
	002104	013770	.WORD	L\$INI'

```

002106 015136
22 002110
23 002110 000000
002110
24 002114
002114 122 114 060
002117 061 054 122
002122 114 060 062
002125 000

.FNDMOD .WORD L$CLEAN
.DEVREG
.WORD 0
.BLKW
.DEVTYP <RLO1,RLO2>
.ASCIZ /RLO1,RLO2/

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

002126
43 002126
44
45 002126
46
47 000000
48 000002
49 000004
50 000006
51 000010
52 000012
53
54
55 000000
56 000002
57 000004
58 000006
59 000010
60 000012
61
62
63 000001
64 000002
65 000004
66 000010
67 000020
68 010000
69 020000
70 040000

.BGNMOD GLBEQAT

.EQUALS
; OFFSETS FOR HARDWARE P-TABLE
CSR =0 ;BUS ADDRESS
VECT =2 ;VECTOR ADDRESS
PRIOR =4 ;PRIORITY
TYPDR=6
DRSB =10 ;DRIVE SELECT BIT
CNT =12 ;CONTROLLER TYPE

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

; BIT ASSIGNMENT FOR SOFTWARE P-TABLE SWITCHES
ALLCYL =BIT00 ;USE ALL CYLINDERS
ALLSEC =BIT01 ;USE ALL SECTORS
DRSELT =BIT02 ;EXECUTE DRIVE SELECT TEST
HDALIGN =BIT03 ;EXECUTE HEAD ALIGNMENT TEST
AUTOSZ =BIT04 ;AUTO SIZE FOR DRIVE-DROP IF NO RESPONSE
HEADLM =BIT12 ;HEAD LIMIT SPECIFIED FLAG
MICYL =BIT13 ;HI LIMIT SPECIFIED FLAG
LOCYL =BIT14 ;LO LIMIT SPECIFIED

```

```

71      100000      MITEST  =BIT15      ;EXECUTE MANUAL INTERVENTION TESTS
72
73      ;          SUBSYSTEM FUNCTIONS
74      000102      CKDATA  =102      ;WRITE CHECK
75      000104      GTSTAT  =104      ;GET STATUS
76      000106      SEEK    =106      ;SEEK
77      000110      RDHEAD  =110      ;READ HEADER
78      000112      WTDATA  =112      ;WRITE DATA
79      000114      RDDATA  =114      ;READ DATA
80      000116      RDNOHR  =116      ;READ DATA, IGNORE HEADERS
81      000100      NOOP    =100      ;NO OPERATION
82
83      ;          OPERATION FLAGS
84      007777      COMPOP  =7777      ;COMPOSITE OPERATION FLAGS
85      000002      HDRCMP  =BIT01     ;HEADER COMPARE OPERATION
86      000001      DATACMP =BIT00     ;DATA COMPARE OPERATION
87      000004      CYLUP   =BIT02     ;CYCLE UP OPERATION
88      000010      ULOAD   =BIT03     ;UNLOAD OPERATION
89      000020      INOUTS  =BIT04     ;IN-OUT SEEK OPERATION
90      000040      OUTINS  =BIT05     ;OUT-IN SEEK OPERATION
91      000100      FOLWRT  =BIT06     ;FOLLOWING WRITE OPERATION
92      000200      REVSKS  =BIT07     ;REV SEEK SEQ (ADJ INTERFERENCE)
93      000400      FWDSKS  =BIT08     ;FWD SEEK SEQ (ADJ INTERFERENCE)
94      001000      REVSKO  =BIT09     ;REV SEEK SEQ (OVERWRITE)
95      002000      FWDSKO  =BIT10     ;FWD SEEK SEQ (OVERWRITE)
96      004000      BADADD  =BIT11     ;BAD DISK ADDRESS
97      010000      SEEKOP  =BIT12     ;SEEK OPERATION
98      020000      RORWOP  =BIT13     ;READ OR WRITE OPERATION
99      040000      RELDWT  =BIT14     ;RELOAD WAIT
100     100000      HDR40   =BIT15     ;40 HEADER OPERATION
101     003760      MQUALS  =OUTINS!INOUTS!FOLWRT!REVSKS!FWDSKS!REVSKO!FWDSKO
102                                     ;MESSAGE QUALIFIER BITS
103
104      ;          ERROR FLAGS FROM SUBROUTINES
105     000001      TOSLOW  =BIT00     ;OPERATION TOOK TO LONG
106     000002      NOIRPT  =BIT01     ;NO INTERRUPT FROM OPERATION
107     000004      CONHNG  =BIT02     ;CONTROLLER HUNG
108     000010      NOCLR   =BIT03     ;BAD CONTROLLER CLEAR
109
110     000000      RLCS    =0          ;CONTROL AND STATUS REGISTER
111     000002      RLBA    =2          ;BUS ADDRESS REGISTER
112     000004      RLDA    =4          ;DISK ADDRESS REGISTER
113     000006      RLMP    =6          ;MULTI-PURPOSE REGISTER
114
115      ;          REGISTER BIT DEFINITIONS - CONTROL STATUS REGISTER
116     000000      RLCSR   =0          ;CONTROL AND STATUS REGISTER
117     100000      ANYERR  =100000    ;ANY ERROR BIT
118     040000      DRVERR  =40000    ;DRIVE ERROR BIT
119     020000      NXMERR  =20000    ;NON-EXISTANT MEMORY ERROR
120     010000      DLTERR  =10000    ;DATA LATE ERROR
121     010000      HNFERR  =10000    ;HEADER NOT FOUND ERROR
122     004000      DCKERR  =4000     ;DATA CHECK ERROR
123     004000      HCRERR  =4000     ;HEADER CHECK ERROR
124     002000      OPIERR  =2000     ;OPERATION INCOMPLETE ERROR
125     001400      DSMSK   =1400     ;DRIVE SELECT MASK
126     000200      CRDYMSK =200      ;CONTROLLER READY MASK
127     000100      INTEBL  =100      ;INTERRUPT ENABLE MASK

```

```

128      000060      BAMSK   =60           ;BUS ADDRESS UPPER MASK
129      000001      DRDYMSK  -1           ;DRIVE READY MASK
130
131      ;           REGISTER BIT DEFINITIONS - DISK ADDRESS FOR DATA XFER
132      000077      SAMSK    =77           ;SECTOR ADDRESS MASK
133      000100      HSMSK    =100          ;HEAD SELECT MASK
134
135      ;           REGISTER BIT DEFINITIONS - DISK ADDRESS FOR SEEK
136      000001      MBSETO   =1           ;MUST BE SET, BIT 0
137      000004      DIRBIT   =4           ;DIRECTION BIT
138      000020      HDSEL    =20          ;HEAD SELECT BIT
139
140      ;           REGISTER BIT DEFINITIONS - DISK ADDRESS FOR GET STATUS
141      000003      GETSTAT  =3           ;GET STATUS SETUP
142      000010      DRSET    =10          ;DRIVE RESET MASK
143
144      ;           REGISTER BIT DEFINITIONS - MP FOR DATA XFER
145      017777      WCMSK    =17777        ;WORD COUNT MASK
146      160000      WCRNG    =160000       ;WORD COUNT RANGE MASK
147
148      ;           REGISTER BIT DEFINITIONS - MP FOR READ HEADER
149      000077      HDSEC    =77           ;SECTOR MASK
150      000100      HDHSEL   =100          ;HEAD SELECT MASK
151
152      ;           REGISTER BIT DEFINITIONS - MP FOR GET STATUS
153      000007      STAMSK   =7           ;STATE MASK
154      000010      BHSTAT   =10          ;BRUSH HOME STATUS
155      000020      HOSTAT   =20          ;HEADS OUT STATUS
156      000040      COSTAT   =40          ;COVER OPEN STATUS
157      000100      HSSTAT   =100          ;HEAD SELECT STATUS
158      000400      DSESTAT  =400          ;DRIVE SELECT ERROR STATUS
159      001000      VCSTAT   =1000          ;VOLUME CHECK STATUS
160      002000      WGESTAT  =2000          ;WRITE GATE ERROR STATUS
161      004000      SPDSTAT  =4000          ;SPIN ERROR STATUS
162      010000      STOSTAT  =10000          ;SEEK TIMEOUT ERROR STATUS
163      020000      WLSTAT   =20000          ;WRITE LOCK STATUS
164      040000      HCESTAT  =40000          ;HEAD CURRENT ERROR STATUS
165      100000      WDESTAT  =100000          ;WRITE DATA ERROR STATUS
166
167 002126      ENDMOD
168 002126      BGNMOD   GLBDAT
169
170      ;           TABLE OF OPERATION MESSAGES
171 002126 000000      OPMSG: .WORD 0           ;FILLER
172 002130 005163      .WORD MWRCHK          ;MESSAGE FOR WRITE CHECK
173 002132 005207      .WORD MGTSTA          ;GET STATUS
174 002134 005134      .WORD MSEEK          ;SEEK
175 002136 005153      .WORD MREADH         ;READ HEADER
176 002140 005175      .WORD MWRITE         ;WRITE DATA
177 002142 005142      .WORD MREAD          ;READ DATA
178 002144 005272      .WORD MWRSET         ;WITH RESET
179 002146 005221      .WORD MDATCP         ;WITH DATA COMPARE
180 002150 005240      .WORD MHDRCP         ;WITH HEADER COMPARE
181 002152 005337      .WORD MCYLUP         ;LOAD HEADS
182 002154 005326      .WORD MULOAD         ;UNLOAD HEADS
183 002156 005370      .WORD MINOUT         ;IN-OUT SEQ
184 002160 005347      .WORD MOUTIN         ;OUT-IN SEQ

```

```

185 002162 005413      .WORD  MFOLWRT      :
186 002164 005435      .WORD  MREVSK       :
187 002166 005470      .WORD  MFWDISK      :
188 002170 005557      .WORD  MRESKO       :
189 002172 005523      .WORD  MFWSKO       :
190 002174 005613      .WORD  MBADAD       :
191 002176 005256      .WORD  M40HDR       :
192 002200 000000      .WORD  0            :
193 002202 000000      T.DRIVE: .WORD  0
194 002204 000000      JJJ:      .WORD  0
195 002206 000000      HLMTW:   .WORD  0
196 002210 000000      CLRBYT:  .WORD  0
197 002212 000000      NXTHL:   .WORD  0
198 002214 000000      GBND:    .WORD  0
199 002216 000000      CAMSK:   .WORD  0
200 002220 000000      DIRMSK:  .WORD  0
201      HDCYL:  .WORD  0
202      ;
203 002222 010024      ; RESTBL: TABLE OF RESULT NAME MESSAGE ADDRESSES
204 002224 010135      .WORD  MCERR        ;CONTROLLER ERROR
205 002226 010450      .WORD  MDRERR       ;DRIVE ERROR
206 002230 010422      .WORD  MNEERR       ;NON-EXISTANT MEMORY ERROR
207 002232 010405      .WORD  MFLERR       ;HEADER NOT FOUND-DATA LATE
208 002234 010375      .WORD  MHDERR       ;HEADER OR DATA ERROR
209 002236 010466      .WORD  MOPERR       ;OPERATION INCOMPLETE
210 002240 000000      .WORD  MMDRST      ;NO DRIVE STATUS AVAILABLE
211 002242 010360      .WORD  0
212 002244 010342      .WORD  MWDERR       ;WRITE DATA ERROR
213 002246 000000      .WORD  MHCERR       ;HEAD CURRENT ERROR
214 002250 010324      .WORD  0
215 002252 010271      .WORD  MSTERR       ;SEEK TIMEOUT ERROR
216 002254 010307      .WORD  MSPERR       ;SPINDLE ERROR
217 002256 000000      .WORD  MWGERR       ;WRITE GATE ERROR
218 002260 010241      .WORD  MDSERR       ;DRIVE SELECT ERROR
219
220      ;
221 002262 004656      ; PATTBL: PATTERN TABLE
222 002264 004660      .WORD  PAT1
223 002266 004720      .WORD  PAT2
224 002270 004760      .WORD  PAT3
225 002272 005020      .WORD  PAT4
226 002274 005026      .WORD  PAT5
227 002276 005066      .WORD  PAT6
228 002300 005070      .WORD  PAT7
229 002302 005130      .WORD  PAT8
230 002304 005132      .WORD  PAT9
231      .WORD  PAT10
232
233      ;
234 002306 000000      ; SUBSTK: SUBROUTINE CALLING STACK
235 002310 000000      .WORD  0            ;STACK IS 12 WORDS LONG
236 002312 000000      .WORD  0
237 002314 000000      .WORD  0
238 002316 000000      .WORD  0
239 002320 000000      .WORD  0
240 002322 000000      .WORD  0
241 002324 000000      .WORD  0

```



```

242 002326 000000          .WORD 0
243 002330 000000          .WORD 0
244
245          ;RL01 TABLE OF CYLINDERS
246 002332 000002          T25TBL: .WORD 2          ;TABLE OF DIFFERENCES TO BE USED
247 002334 000006          .WORD 6          ;IN TEST 25
248 002336 000011          .WORD 9.
249 002340 000014          .WORD 12.
250 002342 000021          .WORD 17.
251 002344 000026          .WORD 22.
252 002346 000033          .WORD 27.
253 002350 000042          .WORD 34.
254 002352 000051          .WORD 41.
255 002354 000200          .WORD 128.
256 002356 000377          .WORD 255.
257
258          ;RL02 TABLE OF CYLINDERS
259 002360 000004          T25TB2: .WORD 4
260 002362 000014          .WORD 12.
261 002364 000022          .WORD 18.
262 002366 000030          .WORD 24.
263 002370 000042          .WORD 34.
264 002372 000054          .WORD 44.
265 002374 000066          .WORD 54.
266 002376 000104          .WORD 68.
267 002400 000122          .WORD 82.
268 002402 000400          .WORD 256.
269 002404 000777          .WORD 511.
270
271          ;          TABLE TO BE USED IN TEST 33 AND 34 TO BUILD AND STORE THE
272          ;          CYLINDERS TO BE USED IN THE TEST.
273
274 002406          T33TBL: .BLKW 16.
275 002440          TBT: .BLKW 16.
276
277
278 002506 002          CYLTBL: .BYTE 2          ;TABLE OF DEFAULT CYLINDERS
279 002507 007          .BYTE 7.
280 002510 016          .BYTE 14.
281 002511 024          .BYTE 20.
282 002512 033          .BYTE 27.
283 002513 041          .BYTE 33.
284 002514 046          .BYTE 38.
285 002515 055          .BYTE 45.
286 002516 064          .BYTE 52.
287 002517 072          .BYTE 58.
288 002520 101          .BYTE 65.
289 002521 110          .BYTE 72.
290 002522 115          .BYTE 77.
291 002523 124          .BYTE 84.
292 002524 133          .BYTE 91.
293 002525 141          .BYTE 97.
294 002526 146          .BYTE 102.
295 002527 154          .BYTE 108.
296 002530 161          .BYTE 113.
297 002531 170          .BYTE 120.
298 002532 177          .BYTE 127.

```

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

```

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

```

```

413 003032 000000      TEMP5:  .WORD  0      ;TEMPORARY STORAGE
414 003034 000000      TEMP6:  .WORD  0      ;TEMPORARY STORAGE
415 003036 000000      TEMP7:  .WORD  0      ;TEMPORARY STORAGE
416 003040 000000      TEMP8:  .WORD  0      ;TEMPORARY STORAGE
449 003042 000004      ERRVEC: .WORD  4      ;ERROR VECTOR USED WHEN AUTO SIZING
450
451      ; MISCELLANEOUS COUNTERS
452 003044 000000      PASCNT: .WORD  0      ;PASS COUNTER (LOCAL TO A TEST)
453 003046 000000      COUNT:  .WORD  0      ;A COUNTER (LOCAL TO A TEST)
454 003050 000000      ERRPOINT: .WORD  0      ;ERROR POINTER
455 003052      ERRCNT:  .BLKW  64.  ;ERROR COUNTER FOR PROGRAM
456 003252 000000      PASNUM: .WORD  0      ;PASS NUMBER FOR PROGRAM
457 003254 000000      PSETNM: .WORD  0      ;COUNTER FOR PARAMETER SET NUMBER IN USE
458 003256      000      LOCERR:  .BYTE  0      ;LOCAL ERROR COUNTER
459 003257      000      NOERCT: .BYTE  0      ;INHIBIT ERROR COUNTING FLAG
460 003260 000000      TRPFLG: .WORD  0      ;HARDWARE TRAP OCCURANCE
461 003262 000000      PWRFLG: .WORD  0      ;POWER FAILURE OCCURANCE
462
463      ; BAD SECTOR TABLES AND POINTERS
464 003264 000000      BSFVAL: .WORD  0      ;BAD SECTORS FILES VALID FLAG
465
466 003266      SBSFIL: .BLKW  76      ;SOFTWARE BAD SECTOR FILE
467 003462      FBSFIL: .BLKW  76      ;FACTORY BAD SECTOR FILE
468
469 003656      IBUFF:  .BLKW  200      ;INPUT BUFFER
470 004256      OBUFF:  .BLKW  200      ;OUTPUT BUFFER
471
472 004656 000000      PAT1:  .WORD  0      ;PATTERN 1 (ALL ZEROS)
473 004660 177772      PAT2:  .WORD  177772
474 004662 177777      .WORD  177777
475 004664 177777      .WORD  177777
476 004666 052525      .WORD  052525
477 004670 052525      .WORD  052525
478 004672 052525      .WORD  052525
479 004674 177777      .WORD  177777
480 004676 177777      .WORD  177777
481 004700 052525      .WORD  052525
482 004702 052525      .WORD  052525
483 004704 177777      .WORD  177777
484 004706 052525      .WORD  052525
485 004710 177252      .WORD  177252
486 004712 177252      .WORD  177252
487 004714 172765      .WORD  172765
488 004716 172765      .WORD  172765
489
490 004720 000003      PAT3:  .WORD  000003
491 004722 000000      .WORD  000000
492 004724 000000      .WORD  000000
493 004726 177777      .WORD  177777
494 004730 177777      .WORD  177777
495 004732 177777      .WORD  177777
496 004734 000000      .WORD  000000
497 004736 000000      .WORD  000000
498 004740 177777      .WORD  177777
499 004742 177777      .WORD  177777
500 004744 000000      .WORD  000000
501 004746 177777      .WORD  177777

```

502	004750	000000		.WORD	000000
503	004752	177777		.WORD	177777
504	004754	000000		.WORD	000000
505	004756	177777		.WORD	177777
506					
507	004760	025252	PAT4:	.WORD	025252
508	004762	052525		.WORD	052525
509	004764	052525		.WORD	052525
510	004766	125252		.WORD	125252
511	004770	125252		.WORD	125252
512	004772	125252		.WORD	125252
513	004774	052525		.WORD	052525
514	004776	052525		.WORD	052525
515	005000	125252		.WORD	125252
516	005002	125252		.WORD	125252
517	005004	052525		.WORD	052525
518	005006	125252		.WORD	125252
519	005010	052525		.WORD	052525
520	005012	125252		.WORD	125252
521	005014	052525		.WORD	052525
522	005016	125252		.WORD	125252
523					
524	005020	155555	PAT5:	.WORD	155555
525	005022	133333		.WORD	133333
526	005024	066666		.WORD	066666
527					
528	005026	121105	PAT6:	.WORD	121105
529	005030	150442		.WORD	150442
530	005032	064221		.WORD	064221
531	005034	132110		.WORD	132110
532	005036	055044		.WORD	055044
533	005040	026442		.WORD	026442
534	005042	013211		.WORD	013211
535	005044	105504		.WORD	105504
536	005046	042642		.WORD	042642
537	005050	021321		.WORD	021321
538	005052	110550		.WORD	110550
539	005054	044264		.WORD	044264
540	005056	022132		.WORD	022132
541	005060	011055		.WORD	011055
542	005062	104426		.WORD	104426
543	005064	042213		.WORD	042213
544					
545	005066	177777	PAT7:	.WORD	177777
546					
547	005070	045513	PAT8:	.WORD	045513
548	005072	122645		.WORD	122645
549	005074	151322		.WORD	151322
550	005076	064551		.WORD	064551
551	005100	132264		.WORD	132264
552	005102	055132		.WORD	055132
553	005104	026455		.WORD	026455
554	005106	113226		.WORD	113226
555	005110	045513		.WORD	045513
556	005112	122645		.WORD	122645
557	005114	151322		.WORD	151322
558	005116	064551		.WORD	064551

```

559 005120 132264          .WORD 132264
560 005122 055132          .WORD 055132
561 005124 026455          .WORD 026455
562 005126 113226          .WORD 113226
563
564 005130 125252          PAT9: .WORD 125252
565
566 005132 155555          PAT10: .WORD 155555
567
568 005134                ENDMOD
569
573 005134                BGNMOD GLBTXT
574 005134      123      105      105 MSEEK: .ASCIZ /SEEK /
575 005142      122      104      040 MREAD: .ASCIZ /RD DATA /
576 005153      122      104      040 MREADH: .ASCIZ /RD HDR /
577 005163      127      122      124 MWRCHK: .ASCIZ /WRT CHECK/
578 005175      127      122      124 MWRITE: .ASCIZ /WRT DATA /
579 005207      107      105      124 MGTSTA: .ASCIZ /GET STAT /
580 005221      127      111      124 MDATCP: .ASCIZ /WITH DATA CMP /
581 005240      127      111      124 MHDRCP: .ASCIZ /WITH HDR CMP /
582 005256      106      117      122 M4OHDR: .ASCIZ /FOR 40 HDRS/
583 005272      127      111      124 MWRSET: .ASCIZ /WITH RESET /
584 005306      117      120      105 MOPER: .ASCIZ /OPER: /
585 005315      122      105      123 MRSLT: .ASCIZ /RESULT: /
586 005326      125      116      114 MLOAD: .ASCIZ /UNLD DRV/
587 005337      114      104      040 MCYLUP: .ASCIZ /LD DRV /
588 005347      106      117      114 MOUTIN: .ASCIZ /FOL 0 TO CC SEEK/
589 005370      106      117      114 MINOUT: .ASCIZ /FOL 255 TO CC SEEK/
590 005413      106      117      114 MFOLWRT: .ASCIZ /FOL WRT (NO SEEK)/
591 005435      101      104      112 MREVSK: .ASCIZ /ADJ CYL WRITN AFTER REV SK/
592 005470      101      104      112 MFWDSK: .ASCIZ /ADJ CYL WRITN AFTER FWD SK/
593 005523      123      113      040 MFWSKO: .ASCIZ /SK FWD,WRT - SK REV,OVERWRT/
594 005557      123      113      040 MRESKO: .ASCIZ /SK REV,WRT - SK FWD,OVERWRT/
595 005613      117      116      040 MBADAD: .ASCIZ /ON BAD SEC FILES/
596 005634      103      101      116 MBADSF: .ASCIZ /CAN'T GET BAD SEC FILES/
597 005664      102      101      104 MFMTERR: .ASCIZ /BAD SEC FILE FMT ERR/
598 005711      124      117      040 MTMBS: .ASCIZ /TO MANY BAD SEC /
599 005732      102      125      123 BASADD: .ASCIZ /BUS ADD=/
600 005743      104      122      126 DRVNAM: .ASCIZ /DRV=/
601 005750      116      117      040 DRVNAV: .ASCIZ /NO DRV FOR TST/
602 005767      104      122      126 NOPWR: .ASCIZ /DRV DID NOT REC'R FROM PWR FAIL/
603 006027      122      114      103 CSNAM: .ASCIZ /RLCS/
604 006034      122      114      102 BANAM: .ASCIZ /RLBA/
605 006041      122      114      104 DANAM: .ASCIZ /RLDA/
606 006046      122      114      115 MPNAM: .ASCIZ /RLMP/
607 006053      117      120      040 LAB1: .ASCIZ /OP INIT = /
608 006066      117      120      040 LAB2: .ASCIZ /OP DONE = /
609 006101      127      117      122 MWORD: .ASCIZ /WORD /
610 006107      111      116      124 MTOSLOW: .ASCIZ /INTRPT TO LATE/
611 006126      116      117      040 MDRRES: .ASCIZ /NO DRV RESPONSE/
612 006146      116      117      040 MNOINT: .ASCIZ /NO INTRPT ON CMND COMPLETE/
613 006201      103      116      124 MCONHNG: .ASCIZ /CNTLR HUNG /
614 006215      105      122      122 MNOCLR: .ASCIZ /ERR DID NOT CLR/
615 006235      126      117      114 VCNRST: .ASCIZ /VOL CHK NOT RSET/
616 006256      125      116      130 UNXERR: .ASCIZ /UNXPCTED ERR/
617 006273      040      124      105 TSTLAB: .ASCIZ / TEST/
619 006301      115      101      116 MISTST: .ASCIZ /MAN INTERVENT STAT/

```

620	006324	123	124	101	NSTACHG:	.ASCIZ	/STATE CHG/
621	006336	123	120	116	SPDERR:	.ASCIZ	/SPNDL TIMEOUT FAILED TO SET/
622	006372	106	101	111	GSTER1:	.ASCIZ	/FAIL FORCING DRV SEL ERR/
623	006423	111	116	111	INITST:	.ASCIZ	/INIT STATE/
624	006436	104	122	126	T05ERR:	.ASCIZ	/DRV SELECT/
625	006451	104	122	126	T09ERR:	.ASCIZ	/DRV RDY/
626	006461	123	105	105	T10ERR:	.ASCIZ	/SEEK SGN SWITCH/
627	006501	110	104	040	T12ERR:	.ASCIZ	/HD SWITCH/
628	006513	122	104	040	T13ERR:	.ASCIZ	/RD HDR (P1)/
629	006527	122	104	040	T14ERR:	.ASCIZ	/RD HDR (P2)/
630	006543	127	122	124	T16ERR:	.ASCIZ	/WRT LCK/
631	006553				P2T01E:		
632	006553	104	111	106	P2T02E:	.ASCIZ	/DIFF OF 1 SEEK/

8	006572	110	104	123	HDMOVF: .ASCIZ	/HDS FAILED TO MOVE IN 10 TRIES/
10	006631	103	131	114	CYLPER: .ASCIZ	/CYL PORTION OF HDRS DIFFER WHEN READ FROM TRK 0 & 1/
11	006715	110	105	101	HAMES1: .ASCIZ	/HEAD ALIGN. RSET WRT LCK TO SEL HD 0, SET FOR HD 1/
12	007000	124	131	120	HAMES2: .ASCIZ	/TYPE "CTL C" & "CONT" TO CONTINUE TESTING/
13	007052	101	102	117	OPR002: .ASCIZ	/ABOVE CONDITIONS MET/
14	007077	127	101	123	OPR003: .ASCIZ	/WAS LOAD DEPRESSED/
15	007122	103	110	113	OPR1: .ASCIZ	/CHK DRV IS UNLDED, COVER OPN, AND WRTE LCKED /
16	007200	103	114	123	OPR2: .ASCIZ	/CLSE COVER & RST WRT LCK /
17	007232	120	122	105	OPR3: .ASCIZ	/PRESS LOAD /
18	007246	120	122	105	OPR5: .ASCIZ	/PRESS LOAD & WAIT FOR LOAD LIGHT /
19	007310	120	122	105	OPR6: .ASCIZ	/PRESS LOAD & WAIT FOR RDY /
20	007343	122	105	115	OPR7: .ASCIZ	/REMOVE ADD PLGS EXCPT /
21	007372	111	116	123	OPR8: .ASCIZ	/INSRT ADD PLG /
22	007411	111	116	040	OPR9: .ASCIZ	/IN ALL DRVS /
23	007426	111	116	123	OPR10: .ASCIZ	/INSUFFICIENT DRVS FOR DRV SEL ERR TST/
24	007474	122	120	114	OPR11: .ASCIZ	/RPLCE ADD PLGS AS BEFORE/
26	007525	122	105	123	OPR12: .ASCIZ	/RESET WRT LCK /
27	007544	117	116	040	OPR1A: .ASCIZ	/ON /
28	007550	117	116	040	OPR1B: .ASCIZ	/ON DRV /
29	007560	125	116	104	UNDTST: .ASCIZ	/UNDER TEST/
30	007573	123	105	124	OPR004: .ASCIZ	/SET WRT LCK /
31	007610	104	111	106	DIFWD: .ASCIZ	/DIFF /
32	007616	123	107	116	SGNWD: .ASCIZ	/SGN /
33	007623	110	104	040	HDWD: .ASCIZ	/HD /
34	007627	123	105	103	SECWD: .ASCIZ	/SEC /
35	007634	103	131	114	CYLWD: .ASCIZ	/CYL /
36	007641	106	122	117	FRMWD: .ASCIZ	/FROM /
37	007647	040	102	131	BYPNM: .ASCIZ	/ BYPASSED /
38	007662	122	117	125	SEQMES: .ASCIZ	/ROUTINE TRACE SEQ:/
39	007705	104	122	126	STAMES: .ASCIZ	/DRV STAT/
40	007716	102	101	104	BSNSTR: .ASCIZ	/BAD SEC FILES NOT STRD. ALL SEC ASSUMED OK./
41	007772	124	117	124	TCERR: .ASCIZ	/TOTAL CMP ERRS: /
42						
43						
44	010013	104	122	126	MDRDY: .ASCIZ	/DRV RDY /
45	010024	103	117	116	MCERR: .ASCIZ	/CONT ERR /
46	010036	110	104	122	MHCRC: .ASCIZ	/HDR CRC/
47	010046	104	101	124	MDCRC: .ASCIZ	/DATA CRC/
48	010057	110	104	122	MHNF: .ASCIZ	/HDR NOT FND/
49	010073	104	101	124	MDLT: .ASCIZ	/DATA LATE/
50	010105	110	104	122	MHFCRC: .ASCIZ	&HDR NOT FND/HDR CRC/OPI&
51	010135	104	122	126	MDRERR: .ASCIZ	/DRV ERR /
53	010146	123	105	114	MHSTA: .ASCIZ	/SEL'D HD /
54	010160	126	117	114	MVOLCK: .ASCIZ	/VOL CHK /
55	010171	103	117	126	MCOSTA: .ASCIZ	/COVER OPN /
56	010204	102	122	125	MBHSTA: .ASCIZ	/BRUSH HME /
57	010217	127	122	124	MWLSTA: .ASCIZ	/WRT LCK /
58	010230	110	104	123	MHOSTA: .ASCIZ	/HDS OUT /
60	010241	104	122	126	MDSERR: .ASCIZ	/DRV SEL ERR /
61	010256	104	122	126	MDRVST: .ASCIZ	/DRV STATE /
62	010271	123	120	111	MSPERR: .ASCIZ	/SPIN TIMEOUT /
63	010307	127	122	124	MWGERR: .ASCIZ	/WRT GAT ERR /
64	010324	123	105	105	MSTERR: .ASCIZ	/SEEK TIMEOUT /
65	010342	110	105	101	MHCERR: .ASCIZ	/HEAD CUR ERR /
66	010360	127	122	124	MWDERR: .ASCIZ	/WRT DAT ERR /
67	010375	117	120	122	MOPERR: .ASCIZ	/OPR-INC/
68	010405	110	104	122	MHDERR: .ASCIZ	&HDR/DAT ERR &


```

69 010422      110      104      122  MFLERR: .ASCIZ  &HDR NOT FND/DAT LATE &
70 010450      116      055      130  MNEERR: .ASCIZ  /N-X-MEM /
71 010461      103      131      114  MCYLOC: .ASCIZ  /CYL /
72 010466      103      101      116  MNRST: .ASCIZ  /CAN'T GET DRV STAT/
73 010511      125      116      113  MUNDEF: .ASCIZ  /UNKN DRV STATE-NO RDY,NO ERR,HDS OUT/
74 010556      106      101      111  MRLFAL: .ASCIZ  /FAIL TO RELD HDS AFTER ERR CLEAR/
75 010617      127      122      124  MWRTAB: .ASCIZ  /WRT ABORTED/
76 010633      040      117      126  MEXERS: .ASCIZ  / OVER ERR LIMIT - UNIT DROPPED /
77 010673      040      105      122  MERRS:  .ASCIZ  / ERROR/
78 010702      207      377      377  BELL:   .ASCIZ  <207><377><377>
79
80
81 010706      111      123      040  RESE3:  .ASCIZ  /IS /
82 010712      040      123      102  RESE4:  .ASCIZ  / SB /
83
84
85 010717      040      111      116  RESE5:  .ASCIZ  / IN /
86 010724      040      117      106  RESE6:  .ASCIZ  / OF /
87 010731      123      124      101  STATE2: .ASCIZ  /STATE 2/
88 010741      123      124      101  STATE3: .ASCIZ  /STATE 3/
89 010751      123      124      101  STATE5: .ASCIZ  /STATE 5/
91 010761      123      105      105  CDRDY:  .ASCIZ  &SEEK W/O MOTION&
93 011001      061      123      124  C10MS:  .ASCIZ  /1ST 3 MS/
94 011012      065      060      060  C500MS: .ASCIZ  /500MS/
95 011020      103      131      103  CCYLUP: .ASCIZ  /CYCLE UP/
96 011031      104      101      124  CAFDT:  .ASCIZ  /DATA XFR/
97 011042      065      040      123  C5SEC:  .ASCIZ  /5 SEC/
98
99 011050      045      116      045  FMTOP1: .ASCIZ  /%N%T%N%T%T%06%$%T%01%N/
100 011077      045      116      045  FMTOP2: .ASCIZ  /%N%T%01%$1%T%01%N/
101 011121      045      116      045  FMTOP3: .ASCIZ  /%N%T%01%$1%T%T%N/
102 011142      045      124      045  FMT1:   .ASCIZ  /%T%T/
103 011147      045      116      045  FMT1.1: .ASCIZ  /%N%T%T/
104 011156      045      124      000  FMT2:   .ASCIZ  /%T/
105 011161      045      116      000  FMT3:   .ASCIZ  /%N/
106 011164      045      116      045  FMT4:   .ASCIZ  /%N%T%T%N/
107 011175      045      116      045  FMT5:   .ASCIZ  /%N%T%06%$1%T%01/
108 011215      045      116      045  FMT6:   .ASCIZ  /%N%$11%T%$4%T%$4%T%$4%T%$4%T%$2%T/
109 011257      045      116      045  FMT7:   .ASCIZ  /%N%T%06%$2%06%$2%06%$2%06%$3%03%$2%01%N/
110 011327      045      116      045  FMT8:   .ASCIZ  /%N%T%06%$2%06%$2%06%$2%06/
111 011361      045      116      045  FMT9:   .ASCIZ  /%N%T/
112 011366      045      124      045  FMT11:  .ASCIZ  /%T%01/
113 011374      045      124      045  FMT12:  .ASCIZ  /%T%03/
114 011402      045      116      045  FMT13:  .ASCIZ  /%N%$11%T%03%$1%T%03%$1%T%01%$1%T%01/
115 011446      045      116      045  FMT14:  .ASCIZ  /%N%T%T%D3%$1%T%06%$1%T%06/
116 011500      045      116      045  FMT15:  .ASCIZ  /%N%$11%T%D3%$1%T%06%$1%T%06/
117 011534      045      116      045  FMT16:  .ASCIZ  /%N%$5%06/
118 011545      045      123      061  FMT17:  .ASCIZ  /%$10%T%N%$11%06%N/
119 011567      045      116      045  FMT18:  .ASCIZ  /%N%$15%T%$5%T%$4%T%$5%T%N/
120 011621      045      124      045  FMT19:  .ASCIZ  /%T%$4%D6%$4%D6%$4%D6%$4%D6%N/
121 011656      045      124      045  FMT20:  .ASCIZ  /%T%$2%D6%$14%D6%$4%D6%N/
122 011706      045      124      045  FMT21:  .ASCIZ  /%T%$12%D6%$14%D6%N/
123 011731      045      116      045  FMT22:  .ASCIZ  /%N%$11%T%03%$1%T%01%$1%T%02/
124 011765      045      124      045  FMT23:  .ASCIZ  /%T%T%T%01%N/
125 012001      045      116      045  FMT24:  .ASCIZ  /%N%T/
126 012006      045      116      045  FMT25:  .ASCIZ  /%N%D2%T/
127 012016      045      116      045  FMT26:  .ASCIZ  /%N%$1%T%D4%T%'D3%N/

```

```

128 012042      045      116      045  FMT27:  .ASCIZ  /%N%T%D3%T%D3%N/
129 012061      045      116      045  FMT28:  .ASCIZ  /%N%T%T%T/
130 012072
135
136 012072      BGNMOD  GLBERR
137          :      ERR1    R3 POINTS TO RESULT MESSAGE
138          :          RESULT: (R3)
139
140          :      ERR2    R3 POINTS TO RESULT NAME
141          :          RESULT: (R3) IS 1 SB 0
142
143          :      ERR3    R3 POINTS TO RESULT NAME
144          :          RESULT: (R3) IS 0 SB 1
145
146          :      ERR4    R3 POINTS TO RESULT NAME
147          :          R4 POINTS TO RESULT CONDITIONS
148          :          RESULT: (R3) IS 1 SB 0 (R4)
149
150          :      ERR5    R3 POINTS TO RESULT NAME
151          :          R4 POINTS TO RESULT CONDITIONS
152          :          RESULT: (R3) IS 0 SB 1 (R4)
153
154          :      ERR6    RESULT ROUTINE DETERMINES WHICH ERROR(S) ARE SET AND
155          :          REPORTS ALL
156          :          RESULT: "ERROR" IS 1 SB 0
157
158          :      ERR7    DRIVE STATE ERROR REPORT
159          :          R3 CONTAINS EXPECTED STATE
160          :          T.STAT CONTAINS BAD STATE
161          :          RESULT: DRIVE STATE IS (T.STAT) SB (R3)
162
163          :      ERR8    HEAD POSITIONING ERROR REPORT
164          :          NEWCYL CONTAINS EXPECTED CYLINDER
165          :          HDWRD1 CONTAINS BAD CYLINDER
166          :          RESULT: CYLINDER IS (HDWRD1) SB (NEWCYL)
167
168          :      ERR9    UTILITY RESULT REPORT
169          :          R3 POINTS TO RESULT NAME
170          :          R4 POINTS TO VALUE 1
171          :          R5 POINTS TO VALUE 2
172          :          RESULT: (R3-NAME) IS (R4-VALUE 1) SB (R5-VALUE 2)
173
174          :      EPR10.   COMPARE ERROR REPORT
175          :          R3 CONTAINS THE BAD WORD NUMBER
176          :          R4 POINTS TO BAD WORD
177          :          R5 POINTS TO GOOD WORD
178          :          RESULT: WORD (R3) IS (R4) SB (R5)
179
180
181 012072      BGNMSG  ERR1
182 012072      105737  003257  TSTB  NOERCT          ;TEST IF ERROR COUNTING INHIBITED
183 012076      001002          BNE    1$          ;YES - SKIP
184 012100      005277  170744  INC   @ERRPOINT    ;ELSE BUMP ERROR COUNT
185 012104      010146          1$:  MOV   R1,-(SP)    ;STORE R1
186 012106      004737  021604  JSR   PC,RPTOP    ;REPORT OPERATION
187 012112      012721  000001  MOV   #1,(R1)+    ;SET PARAM NUMBER
188 012116      010321          MOV   R3,(R1)+    ;INSERT MESSAGE ADDRESS POINTER

```

```

189 012120 004737 022372      JSR    PC,RPTRES      ;REPORT RESULTS
190 012124 004737 022600      JSR    PC,RPTREM      ;REPORT REMAINDER
191 012130 012601              MOV    (SP)+,R1       ;RESTORE R1
192 012132 004737 015326      JSR    PC,CKERLM      ;GO CHECK IF ERROR COUNT EXCEEDED
193 012136              ENDMSG
    012136              L10000:
    012136 104023          EMT    C$MSG
194
195 012140              BGNMSG  ERR2
196 012140 005277 170704      INC    @ERRPOINT      ;BUMP ERROR COUNT
197 012144 010146              MOV    R1,-(SP)       ;STORE R1
198 012146 004737 021604      JSR    PC,RPTOP       ;REPORT OPERATION
199 012152 012721 000003      MOV    #3,(R1)+       ;SET PARAM NUMBER
200 012156 010321              MOV    R3,(R1)+       ;INSERT NAME ADD POINTER
201 012160 012721 000001      MOV    #1,(R1)+       ;SET IS VALUE
202 012164 005021              CLR    (R1)+          ;SET SB VALUE
203 012166 004737 022372      JSR    PC,RPTRES      ;REPORT RESULTS
204 012172 004737 022600      JSR    PC,RPTREM      ;REPORT REMAINDER
205 012176 012601              MOV    (SP)+,R1       ;RESTORE R1
206 012200 004737 015326      JSR    PC,CKERLM      ;GO CHECK IF ERROR COUNT EXCEEDED
207 012204              ENDMSG
    012204              L10001:
    012204 104023          EMT    C$MSG
208
209 012206              BGNMSG  ERR3
210 012206 005277 170636      INC    @ERRPOINT      ;BUMP ERROR COUNT
211 012212 010146              MOV    R1,-(SP)       ;STORE R1
212 012214 004737 021604      JSR    PC,RPTOP       ;REPORT OPERATION
213 012220 012721 000003      MOV    #3,(R1)+       ;SET PARAM NUMBER
214 012224 010321              MOV    R3,(R1)+       ;INSEPT NAME ADD POINTER
215 012226 005021              CLR    (R1)+          ;SET IS VALUE
216 012230 012721 000001      MOV    #1,(R1)+       ;SET SB VALUE
217 012234 004737 022372      JSR    PC,RPTRES      ;REPORT RESULTS
218 012240 004737 022600      JSR    PC,RPTREM      ;REPORT REMAINDER
219 012244 012601              MOV    (SP)+,R1       ;RESTORE R1
220 012246 004737 015326      JSR    PC,CKERLM      ;GO CHECK IF ERROR COUNT EXCEEDED
221 012252              ENDMSG
    012252              L10002:
    012252 104023          EMT    C$MSG
222
223 012254              BGNMSG  ERR4
224 012254 005277 170570      INC    @ERRPOINT      ;BUMP ERROR COUNT
225 012260 010146              MOV    R1,-(SP)       ;STORE R1
226 012262 004737 021604      JSR    PC,RPTOP       ;REPORT OPERATION
227 012266 012721 000004      MOV    #4,(R1)+       ;SET PARAM NUMBER
228 012272 010321              MOV    R3,(R1)+       ;INSERT NAME ADD POINTER
229 012274 012721 000001      MOV    #1,(R1)+       ;SET IS VALUE
230 012300 005021              CLR    (R1)+          ;SET SB VALUE
231 012302 010411              MOV    R4,(R1)        ;INSERT ADD OF CONDITION POINTER
232 012304 004737 022372      JSR    PC,RPTRES      ;REPORT RESULTS
233 012310 004737 022600      JSR    PC,RPTREM      ;REPORT REMAINDER
234 012314 012601              MOV    (SP)+,R1       ;RESTORE R1
235 012316 004737 015326      JSR    PC,CKERLM      ;GO CHECK IF ERROR COUNT EXCEEDED
236 012322              ENDMSG
    012322              L10003:
    012322 104023          EMT    C$MSG
237

```

```

238 012324          BGNMSG  ERR5
239 012324 005277 170520      INC      @ERRPOINT      ;BUMP ERROR COUNT
240 012330 010146          MOV      R1,-(SP)      ;STORE R1
241 012332 004737 021604      JSR      PC,RPTOP      ;REPORT OPERATION
242 012336 012721 000004      MOV      #4,(R1)+      ;SET PARAM NUMBER
243 012342 010321          MOV      R3,(R1)+      ;INSERT NAME ADD POINTER
244 012344 005021          CLR      (R1)+          ;SET IS VALUE
245 012346 012721 000001      MOV      #1,(R1)+      ;SET SB VALUE
246 012352 010411          MOV      R4,(R1)       ;INSERT ADD OF CONDITION POINTER
247 012354 004737 022372      JSR      PC,RPTRES     ;REPORT RESULTS
248 012360 004737 022600      JSR      PC,RPTREM     ;REPORT REMAINDER
249 012364 012601          MOV      (SP)+,R1      ;RESTORE R1
250 012366 004737 015326      JSR      PC,CKERLM     ;GO CHECK IF ERROR COUNT EXCEEDED
251 012372          ENDMSG
      012372          L10004:
      012372 104023      EMT      C$MSG

252
253 012374          BGNMSG  ERR6
254 012374 105737 003257      TSTB     NOERCT        ;TEST IF ERROR COUNTING INHIBITED
255 012400 001002          BNE      17$          ;YES - SKIP
256 012402 005277 170442      INC      @ERRPOINT     ;ELSE BUMP ERROR COUNT
257 012406 010146          17$:  MOV      R1,-(SP)      ;STORE R1
258 012410 010346          MOV      R3,-(SP)      ;STORE R3
259 012412 010446          MOV      R4,-(SP)      ;STORE R4
260 012414 010546          MOV      R5,-(SP)      ;STORE R5
261 012416 004737 021604      JSR      PC,RPTOP      ;REPORT OPERATION
262 012422 012721 000003      MOV      #3,(R1)+      ;SET PARAM NUMBER
263 012426 012761 000001 000002 MOV      #1,2(R1)      ;INSERT IS VALUE
264 012434 005037 003026          CLR      TEMP3        ;CLEAR FOR STATUS STORAGE
265 012440 013703 002746          MOV      T.CS,R3      ;GET T.CS
266 012444 042703 177761          BIC      #177761,R3    ;AND CLEAR ALL BUT FUNCTION
267 012450 022703 000004          CMP      #4,R3        ;CHECK IF IT WAS GET STATUS
268 012454 001432          BEQ      1$          ;YES - STATUS IS IN T.MP, SKIP
269 012456 012762 000003 000004 MOV      #GETSTAT,RLDA(R2) ;ELSE DO GET STATUS
270 012464 012703 000004          MOV      #4,R3
271 012470 053703 002734          BIS      RLDRV,R3
272 012474 010362 000000          MOV      R3,RLCS(R2)
273 012500          WAITUS #10.          ;WAIT FOR CONTROLLER READY
      012500 012700 000012          MOV      #10.,R0
      012504 104027          EMT      C$WTU
274 012506 032762 000200 000000 BIT      #CRDYMSK,RLCS(R2) ;TEST IF READY
275 012514 001003          BNE      10$          ;YES - SKIP
276 012516 012703 001000          9$:  MOV      #BIT9,R3    ;ELSE SET NO DRIVE STATUS BIT
277 012522 000413          BR      2$          ;IN MESSAGE WORD AND SKIP
278 012524 016203 000006          10$: MOV      RLMP(R2),R3   ;STORE STATUS FOR REPORT
279 012530 010337 003026          MOV      R3,TEMP3
280 012534 113703 003027          MOV      TEMP3+1,R3   ;GET ERROR BITS IN PROPER POSITION
281 012540 000402          BR      13$
282 012542 113703 002755          1$:  MOV      T.MP+1,R3    ;GET ERROR BITS FROM MP REG
283 012546 042703 177442          13$: BIC      #177442,R3    ;CLEAR UNUSED BITS
284 012552 013704 002746          2$:  MOV      T.CS,R4      ;GET ERROR BITS FROM CS REG
285 012556 042704 001777          BIC      #1777,R4     ;CLEAR UNUSED BITS
286 012562 050403          BIS      R4,R3        ;MAKE ONE WORD OF POSSIBLE ERRORS
287 012564 032703 002000          BIT      #OPIERR,R3   ;TEST IF OPI SET
288 012570 001442          BEQ      11$          ;NO - SKIP
289 012572 032703 010000          BIT      #HNFERR,R3   ;TEST IF HDR NOT FOUND ERROR
290 012576 001026          BNE      107$         ;YES - SKIP

```

```

291 012600 032703 004000      BIT      #HRCERR,R3      ;TEST IF HDR CRC ERR
292 012604 001020      BNE      105$           ;YES - SKIP
293 012606 012704 010375      MOV      #MOPERR,R4     ;SET OPI ALONE MESSAGE
294 012612      100$: PRINTB    #FMT28,#MRSLT,R4,#MERRS ;REPORT ERROR
      012612 012746 010673      MOV      #MERRS,-(SP)
      012616 010446      MOV      R4,-(SP)
      012620 012746 005315      MOV      #MRSLT,-(SP)
      012624 012746 012061      MOV      #FMT28,-(SP)
      012630 012746 000004      MOV      #4,-(SP)
      012634 010600      MOV      SP,R0
      012636 104014      EMT      C$PNTB
      012640 062706 000012      ADD      #12,SP
295 012644 000430      BR       120$           ;SKIP
296 012646 012704 010036      105$: MOV      #MHCRC,R4   ;HDR CRC MESSAGE
297 012652 000757      BR       100$
298 012654 032703 004000      107$: BIT      #HRCERR,R3   ;TEST IF HCRC WITH HDR NOT FND
299 012660 001003      BNE      109$           ;YES - SKIP
300 012662 012704 010057      MOV      #MHNH,R4       ;MESSAGE HEADER NOT FOUND
301 012666 000751      BR       100$
302 012670 012704 010105      109$: MOV      #MHHCRC,R4   ;HNF AND HCRC MESSAGE
303 012674 000746      BR       100$           ;SKIP
304 012676 032703 004000      115$: BIT      #DCKERR,R3   ;TEST IF DATA CHECK SET, NOT OPI
305 012702 001403      BEQ      118$           ;NO - SKIP
306 012704 012704 010046      MOV      #MDCRC,R4     ;SET MESSAGE DATA CHECK
307 012710 000740      BR       100$           ;SKIP
308 012712 032703 010000      118$: BIT      #DLTERR,R3   ;TEST IF DATA LATE ERROR
309 012716 001403      BEQ      120$           ;NO - SKIP
310 012720 012704 010073      MOV      #MDLT,R4       ;SET MESSAGE DATA LATE
311 012724 000732      BR       100$           ;SKIP
312 012726 012705 100000      120$: MOV      #BIT15,R5   ;SET BIT POINTER FOR TEST
313 012732 005004      CLR      R4             ;CLEAR R4 FOR TABLE COUNT
314 012734 030503      3$:  BIT      R5,R3       ;TEST IF BIT IS SET
315 012736 001005      BNE      6$            ;YES - SKIP TO REPORT
316 012740 005724      4$:  TST      (R4)+       ;ELSE BUMP TABLE POINTER
317 012742 000241      CLC                          ;CLEAR CARRY
318 012744 006005      ROR      R5             ;SHIFT BIT POINTER TO NEXT BIT
319 012746 001377      BNE      3$            ;LOOP IF NOT 0
320 012750 000405      BR       7$            ;ELSE REPORT REMAINDER
321 012752 016411 002222      6$:  MOV      REXBL(R4),(R1) ;INSERT NAME ADDRESS
322 012756 004737 022372      JSR      PC,RPTRES      ;REPORT RESULTS
323 012762 000766      BR       4$            ;GET NEXT BIT
324 012764 004737 022600      7$:  JSR      PC,RPTREM      ;REPORT REMAINDER
325 012770 005737 003026      TST      TEMP3         ;TEST IF ANY NEW STATUS
326 012774 001414      BEQ      15$           ;NO - SKIP
327 012776      PRINTB    #FMT17,#STAMES,TEMP3
      012776 013746 003026      MOV      TEMP3,-(SP)
      013002 012746 007705      MOV      #STAMES,-(SP)
      013006 012746 011545      MOV      #FMT17,-(SP)
      013012 012746 000003      MOV      #3,-(SP)
      013016 010600      MOV      SP,R0
      013020 104014      EMT      C$PNTB
      013022 062706 000010      ADD      #10,SP
328 013026 032737 004000 002746 15$: BIT      #DCKERR,T.CS   ;TEST IF DATA CHECK ERROR
329 013034 001453      BEQ      25$           ;NO - SKIP
330 013036 032737 002000 002746 BIT      #OPIERR,T.CS   ;TEST IF OPI SET
331 013044 001047      BNE      25$           ;YES - SKIP
332 013046 005037 002716      CLR      MORECE        ;CLEAR COMPARE ERROR COUNT

```

```

333 013052 012701 000200      MOV      #128.,R1      ;SET COMPARE LENGTH
334 013056 012703 000001      MOV      #1,R3        ;SET WORD COUNT
335 013062 012705 004256      MOV      #OBUFF,R5    ;SET GOOD WORD POINTER
336 013066 012704 003656      MOV      #IBUFF,R4    ;SET TEST WORD POINTER
337 013072 021514              18$:    CMP      (R5),(R4)    ;CHECK WORD
338 013074 001427              BEQ      19$          ;GOOD - SKIP
339 013076 023727 002716 000012  CMP      MORECE,#10.  ;TEST IF COMPARE LIMIT REACHED
340 013104 003021              BGT      20$          ;YES - SKIP
341 013106              PRINTB  #FMT15,#MWORD,R3,#RESE3,(R4),#RESE4,(R5)
      013106 011546              MOV      (R5),-(SP)
      013110 012746 010712              MOV      #RESE4,-(SP)
      013114 011446              MOV      (R4),-(SP)
      013116 012746 010706              MOV      #RESE3,-(SP)
      013122 010346              MOV      R3,-(SP)
      013124 012746 006101              MOV      #MWORD,-(SP)
      013130 012746 011500              MOV      #FMT15,-(SP)
      013134 012746 000007              MOV      #7,-(SP)
      013140 010600              MOV      SP,R0
      013142 104014              EMT      C$PNTB
      013144 062706 000020              ADD      #20,SP
342 013150 005237 002716              20$:    INC      MORECE      ;BUMP ERROR COUNTER
343 013154 022524              19$:    CMP      (R5)+,(R4)+ ;BUMP POINTERS
344 013156 005203              INC      R3          ;BUMP COUNTER
345 013160 005301              DEC      R1          ;DEC LENGTH COUNT
346 013162 001343              BNE      18$         ;LOOP IF NOT DONE
347 013164 005737 002716              25$:    TST      MORECE    ;TEST IF ANY COMPARE ERRORS
348 013170 001421              BEQ      27$         ;NO - SKIP
349 013172 012701 000200      MOV      #128.,R1    ;SET COMPARE LENGTH
350 013176              PRINTB  #FMT27,#TCERR,MORECE,#RESE6,R1
      013176 010146              MOV      R1,-(SP)
      013200 012746 010724              MOV      #RESE6,-(SP)
      013204 013746 002716              MOV      MORECE,-(SP)
      013210 012746 007772              MOV      #TCERR,-(SP)
      013214 012746 012042              MOV      #FMT27,-(SP)
      013220 012746 000005              MOV      #5,-(SP)
      013224 010600              MOV      SP,R0
      013226 104014              EMT      C$PNTB
      013230 062706 000014              ADD      #14,SP
351 013234 012605              27$:    MOV      (SP)+,R5    ;RESTORE R5, 4, 3, 1
352 013236 012604              MOV      (SP)+,R4
353 013240 012603              MOV      (SP)+,R3
354 013242 012601              MOV      (SP)+,R1
355 013244 004737 015326              JSR      PC,CKERLM   ;GO CHECK IF ERROR COUNT EXCEEDED
356 013250              ENDMSG
      013250              L10005:
      013250 104023              EMT      C$MSG
357
358 013252              BGNMSG  ERR7
359 013252 005277 167572              INC      @ERRPOINT   ;BUMP ERROR COUNT
360 013256 010146              MOV      R1,-(SP)    ;STORE R1
361 013260 004737 021604              JSR      PC,RPTOP    ;REPORT OPERATION
362 013264 012721 000003              MOV      #3,(R1)+   ;SET PARAM NUMBER
363 013270 012721 010256              MOV      #MDRVST,(R1)+ ;INSERT NAME ADD POINTER
364 013274 013721 002762              MOV      T,STAT,(R1)+ ;INSERT IS VALUE
365 013300 010311              MOV      R3,(R1) ;INSERT SB VALUE
366 013302 004737 022372              JSR      PC,RPTRES   ;REPORT RESULTS
367 013306 004737 022600              JSR      PC,RPIREM   ;REPORT REMAINDER

```

```

368 013312 012601          MOV      (SP)+,R1      ;RESTORE R1
369 013314 004737 015326  JSR      PC,CKERLM    ;GO CHECK IF ERROR COUNT EXCEEDED
370 013320          ENDMMSG
    013320          L10006:
    013320 104023          EMT      C$MSG
371
372 013322          BGNMSG  ERR8
373 013322 005277 167522  INC      @ERRPOINT    ;BUMP ERROR COUNT
374 013326 010146          MOV      R1,-(SP)     ;STORE R1
375 013330 010346          MOV      R3,-(SP)     ;STORE R3
376 013332 004737 021604  JSR      PC,RPTOP     ;REPORT OPERATION
377 013336 012721 000003  MOV      #3,(R1)+     ;SET PARAM NUMBER
378 013342 012721 010461  MOV      #MCYLOC,(R1)+ ;INSERT NAME ADD POINTER
379 013346 013711 002754  MOV      HDWRD1,(R1)  ;GET HEADER WORD
380 013352 012703 000007  MOV      #7,R3       ;SET SHIFT COUNT
381 013356 000241          3$:      CLC
382 013360 006011          ROR      (R1)         ;ALIGN CHAR FOR PRINTING
383 013362 005303          DEC      R3          ; AS IS VALUE ..
384 013364 001374          BNE      3$
385 013366 005721          TST      (R1)+       ;BUMP PARAM POINTER
386 013370 013711 003004  MOV      NEWCYL,(R1)  ;INSERT SB VALUE
387 013374 004737 022372  JSR      PC,RPTRES   ;REPORT RESULTS
388 013400 004737 022600  JSR      PC,RPTREM   ;REPORT REMAINDER
389 013404 012603          MOV      (SP)+,R3    ;RESTORE R3
390 013406 012601          MOV      (SP)+,R1    ;RESTORE R1
391 013410 004737 015326  JSR      PC,CKERLM    ;GO CHECK IF ERROR COUNT EXCEEDED
392 013414          ENDMMSG
    013414          L10007:
    013414 104023          EMT      C$MSG
393
394 013416          BGNMSG  ERR9
395 013416 005277 167426  INC      @ERRPOINT    ;BUMP ERROR COUNT
396 013422 010146          MOV      R1,-(SP)     ;STORE R1
397 013424 004737 021604  JSR      PC,RPTOP     ;REPORT OPERATION
398 013430 012721 000003  MOV      #3,(R1)+     ;SET PARAM NUMBER
399 013434 010321          MOV      R3,(R1)+     ;INSERT NAME ADD POINTER
400 013436 010421          MCV      R4,(R1)+     ;SET IS VALUE
401 013440 010521          MOV      R5,(R1)+     ;SET SB VALUE
402 013442 004737 022372  JSR      PC,RPTRES   ;REPORT RESULTS
403 013446 004737 022600  JSR      PC,RPTREM   ;REPORT REMAINDER
404 013452 012601          MOV      (SP)+,R1    ;RESTORE R1
405 013454 004737 015326  JSR      PC,CKERLM    ;GO CHECK IF ERROR COUNT EXCEEDED
406 013460          ENDMMSG
    013460          L10010:
    013460 104023          EMT      C$MSG
407 013462          BGNMSG  ERR10
408 013462 010146          MOV      R1,-(SP)     ;STORE R1
409 013464 005737 002716  TST      MORECE       ;TEST IF 2ND BAD LINE
410 013470 001051          BNE      3$          ;YES - SKIP
411 013472 005277 167352  INC      @ERRPOINT    ;BUMP ERROR COUNT
412 013476 004737 021604  JSR      PC,RPTOP     ;REPORT OPERATION
413 013502          PRINTB  #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1> ;REPORT ID
    013502 005046          CLR      -(SP)
    013504 153716 002735  BISB    RLDRV+1,(SP)
    013510 012746 005743  MOV      #DRVNAM,-(SP)
    013514 013746 002730  MOV      RLBAS,-(SP)
    013520 012746 005732  MOV      #BASADD,-(SP)

```

```

013524 012746 011175      MOV      #FMT5,-(SP)
013530 012746 000005      MOV      #5,-(SP)
013534 010600      MOV      SP,R0
013536 104014      EMT      C$PNTB
414 013540 062706 000014      ADD      #14,SP
013544 011546      PRINTB  #FMT14,#MRSLT,#MWORD,R3,#RESE3,(R4),#RESE4,(R5)
013546 012746 010712      MOV      (R5),-(SP)
013552 011446      MOV      #RESE4,-(SP)
013554 012746 010706      MOV      (R4),-(SP)
013560 010346      MOV      #RESE3,-(SP)
013562 012746 006101      MOV      R3,-(SP)
013566 012746 005315      MOV      #MWORD,-(SP)
013572 012746 011446      MOV      #MRSLT,-(SP)
013576 012746 000010      MOV      #FMT14,-(SP)
013602 010600      MOV      #10,-(SP)
013604 104014      MOV      SP,R0
013606 062706 000022      EMT      C$PNTB
415 013612 000421      ADD      #22,SP
416 013614 000421      BR       4$
013614 011546      3$: PRINTB #FMT15,#MWORD,R3,#RESE3,(R4),#RESE4,(R5) ;REPORT DATA
013616 012746 010712      MOV      (R5),-(SP)
013622 011446      MOV      #RESE4,-(SP)
013624 012746 010706      MOV      (R4),-(SP)
013630 010346      MOV      #RESE3,-(SP)
013632 012746 006101      MOV      R3,-(SP)
013636 012746 011500      MOV      #MWORD,-(SP)
013642 012746 011500      MOV      #FMT15,-(SP)
013646 010600      MOV      #7,-(SP)
013650 104014      MOV      SP,R0
013652 062706 000020      EMT      C$PNTB
417 013656 005237 002716      ADD      #20,SP
418 013662 012601      4$: INC     MORECE           ;INC COMPARE ERROR COUNT
419 013664 004737 015326      MOV      (SP)+,R1         ;RESTORE R1
420 013670 004737 015326      JSR      PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
013670 104023      ENDMSG  L10011:
421 013672      EMT      C$MSG
422      ENDMOD
423      .EVEN
424 013672      BGNMOD  HPTCODE
425 013672      BGNHW
      .WORD  L10012-L$HW/2
426 013674 174400      .WORD  174400           ;CSR BASE ADDRESS DEFAULT
427 013676 000160      .WORD  160              ;VECTOR DEFAULT
428 013700 000240      .WORD  240              ;PRIORITY DEFAULT
429 013702 000001      .WORD  1                 ;TYPE OF DRIVE
430 013704 000000      .WORD  0                 ;DRIVE NUMBER DEFAULT
431 013706 000001      .WORD  1                 ;RL11 CONTROLLER
432 013710      ENDDW  L10012:
013710      ENDMOD
433 013710      BGNMOD  SPTCODE
434      BGNSW
435 013710      .WORD  L10013-L$SW/2
436 013710      .WORD  0                 ;BIT 0 = USE ALL CYLINDERS
437 013712 000006      MISWIW: .WORD  0
013712 000000

```


438
439
440
441
442
443
444
445

;BIT 1 = USE ALL SECTORS
;BIT 2 = EXECUTE DRIVE SELECT TEST
;BIT 3 = EXECUTE HEAD ALIGNMENT
;BIT 4 = DROP DRIVE IF NO RESPONSE
;BIT 12 = HEAD SELECT SUPPLIED FLAG
;BIT 13 = HILIMIT SPECIFIED FLAG
;BIT 14 = LO LIMIT SPECIFIED FLAG
;BIT 15 = DO MANUAL INTERVENTION

446 013714 000000
447 013716 000377
448 013720 000000
449 013722 000024
450 013724 000012
451 013726
452 013726

LOLIMW: .WORD 0
HILIMW: .WORD 255.
HEADW: .WORD 0
ERLIMW: .WORD 20.
DCLIMW: .WORD 10.
ENDSW
L10013:
ENDMOD

;ERROR LIMIT
;COMPARE ERROR LIMIT

453

454 013726
456 013726 000020
013730 023064
013732 023334
013734 023536
013736 024772
013740 025600
013742 026204
013744 027056
013746 027410
013750 027474
013752 027774
013754 030316
013756 030750
013760 031266
013762 031500
013764 031754
013766 032276

BGNMOD DSPCODE
DISPATCH 16
.WORD 16
.WORD T1
.WORD T2
.WORD T3
.WORD T4
.WORD T5
.WORD T6
.WORD T7
.WORD T8
.WORD T9
.WORD T10
.WORD T11
.WORD T12
.WORD T13
.WORD T14
.WORD T15
.WORD T16

461

013770
462
463 013770
464 013770
465 013770 012700 000340
013774 104041

ENDMOD
BGNMOD INITCODE
BGNINIT
SETPRI #340
MOV #340,RO
EMT C\$SPRI

466 013776 104051
467 014000 103403
014000 042737 100014 013712
468 014002 005037 002704
469 014014 012700 000034
014020 104050

MANUAL ;CHECK IF MANUAL INTERVENTION ALLOWED
EMT C\$MANI
BCOMplete 1\$;YES - SKIP
BCS 1\$
BIC #MITEST!DRSELT!HDALIGN,MISWIW ;CLEAR ALL MANUAL
; INTERVENTION FLAGS
1\$: CLR SSINDX ;CLEAR SUBROUTINE STACK INDEX
READEf #EF.PWR ;POWER FAILURE
MOV #EF.PWR,RO
EMT C\$REFG
BNCOMplete 4\$;NO, GO CHECK NEW PASS
BCC 4\$

472 014022 103005
473 014024 013737 002012 003262
474 014032 000137 014444

MOV L\$UNIT,PWRFLG ;SET POWER FAIL FLAG
JMP PWCON ;GO SERVICE POWER FAIL

```

475 014036          4$:  READEF #EF.START      ;CHECK IF START
    014036 012700 000040  MOV #EF.START,R0
    014042 104050  EMT C$REFG
476 014044          BNCOMPLETE RESTART ;NO - SKIP
    014044 103034  BCC RESTART
477          : ON START INITIALIZE TO START AT FIRST DRIVE, CLEAR INTERNAL
478          : PASS COUNT, AND ERROR COUNT.
479 014046 013737 002012 002776  MOV L$UNIT,DRVCNT ;SET UP UNIT COUNT
480 014054 005037 003252  RSTRT: CLR PASNUM ;CLEAR PASS NUMBER
481 014060 012700 003052  MOV #ERRCNT,R0
482 014064 012701 000100  MOV #64.,R1 ;GET A COUNT
483 014070 005020 1$: CLR (R0)+ ;CLEAR A ERROR COUNTER STORAGE AREA
484 014072 005301  DEC R1
485 014074 001375  BNE 1$ ;LOOP TILL ALL CLEARED
486 014076 012737 003050 003050  MOV #ERRCNT-2,ERRPOINT ;INIT ERROR POINTER
487 014104 012737 177777 003254  MOV #-1,PSETNM ;SET PARAM SELECT TO INITIAL VALUE
488 014112 012737 177777 002712  MOV #-1,HADONE ;PRESET HEAD ALIGN DONE FLAG
489 014120 032737 040000 013712  LAB: BIT #LOCYL,MISWIW ;TEST IF LO LIMIT SET
490 014126 001002  BNE 5$ ;YES - SKIP
491 014130 005037 013714  CLR LOLIMW ;ELSE CLEAR LO LIMIT
492 014134 000432 5$: BR SETDON
493 014136  RESTART:
494 014136          READEF #EF.RESTART ;CHECK IF RESTART
    014136 012700 000037  MOV #EF.RESTART,R0
    014142 104050  EMT C$REFG
495 014144          BCOMPLETE RSTRT ;NO - SKIP
    014144 103743  BCS RSTRT
496 014146  CONTINUE:
497 014146          READEF #EF.CONTINUE ;TEST IF CONTINUE
    014146 012700 000036  MOV #EF.CONTINUE,R0
    014152 104050  EMT C$REFG
498 014154          BCOMPLETE PWCON
    014154 103533  BCS PWCON
499          : ON CONTINUE PICK UP UNIT LAST UNDER TEST
500 014156          READEF #EF.NEW ;CHECK IF STARTING NEW PASS
    014156 012700 000035  MOV #EF.NEW,R0
    014162 104050  EMT C$REFG
501 014164          BCOMPLETE PASNEW
    014164 103403  BCS PASNEW
502 014166  NXTPAS:
503 014166 005737 002776  TST DRVCNT ;TEST IF ALL UNITS CHECKED
504 014172 001013  BNE SETDON ;NO - SKIP
505 014174 005237 003252  PASNEW: INC PASNUM ;ELSE BUMP PASS COUNT
506 014200 012737 003050 003050  MOV #ERRCNT-2,ERRPOINT ;INIT ERROR POINTER
507 014206 013737 002012 002776  MOV L$UNIT,DRVCNT ;GET ALL DRIVES
508 014214 012737 177777 003254  MOV #-1,PSETNM ;SET PARAM SELECT TO INITIAL
509 014222 005237 003254  SETDON: INC PSETNM ;NEXT SET OF PARAMETERS
510 014226 005337 002776  DEC DRVCNT ;DOWN COUNT DRIVE TOTAL
511 014232 062737 000002 003050  ADD #2,ERRPOINT ;UPDATE THE ERROR POINTER
512 014240 013700 003254  MOV PSETNM,R0 ;SET UP TO GET PARAMETERS
513 014244 012702 002730  MOV #RLBAS,R2
514 014250  GPHARD R0,R1
    014250 104042  EMT C$GPHRD
    014252 010001  MOV R0,R1
515 014254          BCOMPLETE 7$ ;SKIP IF GOOD PARAM
    014254 103406  BCS 7$
516 014256 005737 003262  TST PWRFLG ;RECENT POWER FAILURE

```

```

517 014262 001741          BEQ     NXTPAS ;NO
518 014264 005337 003262  DEC     PWRFLG          ;ACCOUNT FOR DRIVE
519 014270 000736          BR      NXTPAS
520 014272 012122          7$:   MOV     (R1)+,(R2)+ ;STORE PARAMETERS CSR
521 014274 012122          MOV     (R1)+,(R2)+ ;VECTOR
522 014276 005721          TST     (R1)+          ;BUMP PAST PRIORITY
523 014300 012137 002200  MOV     (R1)+,T.DRIVE
524 014304 012122          MOV     (R1)+,(R2)+
525 014306 022737 000001 002200  CMP     #1,T.DRIVE
526 014314 001426          BEQ     65$
527 014316 012737 000776 002210  MOV     #510.,NXTHL
528 014324 012737 000777 002204  MOV     #511.,HLMTW
529 014332 012737 001000 002212  MOV     #512.,GBND
530 014340 012737 177600 002214  MOV     #177600,CAMSK
531 014346 012737 177600 002216  MOV     #177600,DIRMSK
532 014354 012737 177600 002220  MOV     #177600,HDCYL
533 014362 012737 177000 002206  MOV     #177000,CLRBYT
534 014370 000425          BR      PWCON
535
536 014372 012737 000377 002204 65$:  MOV     #255.,HLMTW
537 014400 012737 000400 002212  MOV     #256.,GBND
538 014406 012737 077600 002214  MOV     #77600,CAMSK
539 014414 012737 077600 002216  MOV     #77600,DIRMSK
540 014422 012737 077600 002220  MOV     #77600,HDCYL
541 014430 012737 000376 002210  MOV     #254.,NXTHL
542 014436 012737 177400 002206  MOV     #177400,CLRBYT
543
544 014444 032737 020000 013712 PWCON: BIT     #HICYL,MISWIW
545 014452 001003          BNE     1$
546 014454 013737 002204 013716  MOV     HLMTW,HILIMW
547 014462          1$:   SETVEC  RLVEC,#INTHLR,#340 ;SET UP VECTOR
      014462 012746 000340  MOV     #340,-(SP)
      014466 012746 015270  MOV     #INTHLR,-(SP)
      014472 013746 002732  MOV     RLVEC,-(SP)
      014476 012746 000003  MOV     #3,-(SP)
      014502 104037          EMT     C$SVEC
      014504 062706 000010  ADD     #10,SP
548 014510          SETPRI  #0 ;SET PRIORITY
      014510 012700 000000  MOV     #0,R0
      014514 104041          EMT     C$SPRI
549 014516 013702 002730  MOV     RLBAS,R2 ;SET RL BASE ADDRESS POINTER
550
551
552
553 ; CHECK IF DOING AUTO SIZE AND DROP DRIVE IF NOT READY AND
554 ; ERROR SETS ON GET STATUS.
555 014522 005737 003252  TST     PASNUM ;TEST IF PASS 0
556 014526 001135          BNE     22$ ;NO - SKIP
557 014530 032737 000020 013712  BIT     #AUTOSZ,MISWIW ;TEST IF DOING AUTO SIZE
558 014536 001531          BEQ     22$ ;NO - SKIP
559 ;CHECK IF UNIBUS ADDRESS IS THERE BEFORE WE CHECK DRIVE READY
560 014540 005037 003260  CLR     TRPFLG ;TRAP OCCURANCE
561 014544          SETVEC  ERRVEC,#TRPHAN,#340 ;SET TRAP VECTOR
      014544 012746 000340  MOV     #340,-(SP)
      014550 012746 015262  MOV     #TRPHAN,-(SP)
      014554 013746 003042  MOV     ERRVEC,-(SP)
      014560 012746 000003  MOV     #3,-(SP)

```

```

014564 104037      EMT      C$SVEC
014566 062706 000010  ADD      #10,SP
562 014572 005762 000000  TST      RLCS(R2)      ;ACCESS BUS
563 014576 005737 003260  TST      TRPFLG        ;TRAP OCCUR??
564 014602 001032      BNE      5$           ;YES, DON'T INVESTIGATE FURTHER
565 014604 013705 002734  MOV      RLDRV,R5      ;GET DRIVE NUMBER
566 014610 052705 000200  BIS      #CRDYMSK,R5   ;INSERT CONT READY
567 014614 010562 000000  MOV      R5,RLCS(R2)   ;LOAD IN DRIVE NUMBER
568 014620 032762 000001 000000  BIT      #DRDYMSK,RLCS(R2) ;CHECK IF DRIVE IS READY
569 014626 001072      BNE      20$         ;YES - GO DO TEST
570 014630 012762 000003 000004  MOV      #GETSTAT,RLDA(R2) ;ELSE INSERT GET STATUS
571 014636 052705 000004      BIS      #4,R5        ;LOAD R5 WITH GET STATUS FUNCTION
572 014642 042705 000200      BIC      #CRDYMSK,R5   ;CLEAR CONTROLLER READY
573 014646 010562 000000      MOV      R5,RLCS(R2)   ;LOAD CS REG
574 014652      WAITMS #4           ;WAIT 4 MS
014652 012700 000004      MOV      #4,R0
014656 104026      EMT      C$WTM
575 014660 032762 002000 000000  BIT      #OPIERR,RLCS(R2);TEST IF OPI SET
576 014666 001452      BEQ      20$         ;NO - SKIP
577 014670      5$: CLRVEC  ERRVEC
014670 013700 003042      MOV      ERRVEC,R0
014674 104036      EMT      C$CVEC
578 014676      PRINTF #FMT24,#DRVNAV
014676 012746 005750      MOV      #DRVNAV,-(SP)
014702 012746 012001      MOV      #FMT24,-(SP)
014706 012746 000002      MOV      #2,-(SP)
014712 010600      MOV      SP,R0
014714 104017      EMT      C$PNTF
014716 062706 000006      ADD      #6,SP
579 014722      10$: PRINTF #FMT5,#BASADD,RLBAS,#DRVNAV,<B,RLDRV+1>
014722 005046      CLR      -(SP)
014724 153716 002735      BISB    RLDRV+1,(SP)
014730 012746 005743      MOV      #DRVNAV,-(SP)
014734 013746 002730      MOV      RLBAS,-(SP)
014740 012746 005732      MOV      #BASADD,-(SP)
014744 012746 011175      MOV      #FMT5,-(SP)
014750 012746 000005      MOV      #5,-(SP)
014754 010600      MOV      SP,R0
014756 104017      EMT      C$PNTF
014760 062706 000014      ADD      #14,SP
580 014764      PRINTF #FMT3
014764 012746 011161      MOV      #FMT3,-(SP)
014770 012746 000001      MOV      #1,-(SP)
014774 010600      MOV      SP,R0
014776 104017      EMT      C$PNTF
015000 062706 000004      ADD      #4,SP
581 015004      DODU    PSETNM        ;DROP DRIVE
015004 013700 003254      MOV      PSETNM,R0
015010 104053      EMT      C$DODU
582 015012      DOCLN
015012 104044      EMT      C$DCLN
583 015014      20$: CLRVEC  ERRVEC
015014 013700 003042      MOV      ERRVEC,R0
015020 104036      EMT      C$CVEC
584 015022      22$:
586
587 015022      MANUAL                ;MANUAL INTERVENTION ALLOWED

```

```

588 015022 104051          EMT      C$MANI
      015024          BNCOMPLETE 4$      ;NO
      015024 103004          BCC      4$
589
590 015026 005737 003252  TST      PASNUM      ;YES, CHECK PASS NUMBER
591 015032 001001          BNE      4$          ;NOT FIRST PASS, NEED DRIVE UP
592 015034 000437          BR       8$          ;FIRST PASS, PROGRAM WILL INSTRUCT USER
593
595          ;CHECK IF POWER FAILURE WAIT IS NEEDED
596
597 015036 005737 003262  4$:      TST      PWRFLG      ;NEEDED???
598 015042 001434          BEQ      8$          ;NO, SKIP
599
600 015044 013705 002734  MOV      RLDRV,R5      ;DRIVE SELECT
601 015050 052705 000200  BIS      #CRDYMSK,R5   ;SET CRDY
602 015054 010562 000000  MOV      R5,RLCS(R2)   ;SELECT DRIVE
603 015060 012701 000074  MOV      #60.,R1      ;SIXTY SECOND TIMER
604 015064 032762 000001 000000 9$:      BIT      #DRDYMSK,RLCS(R2) ;DRIVE UP YET
605 015072 001020          BNE      8$          ;YES START TEST
606
607 015074          WAITMS  #10.          ;WAIT A SECOND
      015074 012700 000012  MOV      #10.,R0
608 015102 005301          EMT      C$WTM
609 015104 001367          DEC      R1          ;SIXTY GONE BY'
610 015106          BNE      9$          ;NO
      015106 012746 005767  PRINTF  #FMT24,#NOPWR
      015112 012746 012001  MOV      #NOPWR,-(SP)
      015116 012746 000002  MOV      #FMT24,-(SP)
      015122 010600          MOV      #2,-(SP)
      015124 104017          MOV      SP,R0
      015126 062706 000006  EMT      C$PNTF
611 015132 000673          ADD      #6,SP
612
613 015134          BR       10$
614
615 015134          ENDINIT
      015134          L10014:
      015134 104011          EMT      C$INIT
616 015136          ENDMOD
617
618 015136          BGNMOD  CLNCODE
619 015136          BGNCLN
620
621 015136          SETVEC  ERRVEC,#TRPHAN,#340
      015136 012746 000340  MOV      #340,-(SP)
      015142 012746 015262  MOV      #TRPHAN,-(SP)
      015146 013746 003042  MOV      ERRVEC,-(SP)
      015152 012746 000003  MOV      #3,-(SP)
      015156 104037          EMT      C$SVEC
      015160 062706 000010  ADD      #10,SP
622
623 015164          SETPRI  #7          ;SET PRORITY TO 7
      015164 012700 000007  MOV      #7,R0
      015170 104041          EMT      C$SPRI
624 015172 032762 000200 000000 2$:      BIT      #CRDYMSK,RLCS(R2) ;TEST IF CONTROLLER READY
625 015200 001407          BEQ      3$          ;NO LOOP UNTIL READY

```

```

626 015202 053762 002734 000000      BIS      RLDRV,RLCS(R2) ;SET DRIVE NUMBER
627 015210 032762 000001 000000      BIT      #DRDYMSK,RLCS(R2) ;TEST IF DRIVE BUSY
628 015216 001003                      BNE      5$ ;NO - SKIP
629 015220                      3$:      WAITMS #3 ;WAIT 300 MS
        015220 012700 000003      MOV      #3,RO
        015224 104026      EMT      C$WTM
630 015226                      5$:      CLRVEC  RLVEC ;RELEASE VEC
        015226 013700 002732      MOV      RLVEC,RO
        015232 104036      EMT      C$CVEC
631 015234 005737 003262      TST      PWRFLG ;PWR FAIL SET
632 015240 001402                      BEQ      7$ ;NO
633 015242 005337 003262      DEC      PWRFLG
634 015246                      7$:      CLRVEC  ERRVEC
        015246 013700 003042      MOV      ERRVEC,RO
        015252 104036      EMT      C$CVEC
635 015254                      ENDCLN
        015254 L10015:      EMT      C$CLEAN
        015254 104012
636
637 015256                      BGNDU
638 015256 000240                      NOP
639 015260                      ENDDU
        015260 L10016:      EMT      C$DU
        015260 104055
640
641 015262                      ENDMOD
642 015262                      BGNMOD  GLBSUB
643
644 015262 005237 003260      TRPHAN: INC      TRPFLG
645 015266 000002                      RTI
646
647 015270                      BGNSRV  INTHLR
648 ;                                INTERRUPT HANDLER. ABORTS WAIT TIMER AND STORES ALL RL11 REGS
649 015270                      ;                                ABORTWAIT
        015270 104021      EMT      C$ABRT
650 015272 012237 002746      MOV      (R2)+,T.CS ;STORE RL REGISTERS
651 015276 012237 002750      MOV      (R2)+,T.BA
652 015302 012237 002752      MOV      (R2)+,T.DA
653 015306 011237 002754      MOV      (R2),T.MP
654 015312 012737 177777 002710      MOV      #-1,DONE ;SET DONE FLAG
655 015320 013702 002730      MOV      RLBAS,R2 ;RESTORE R2
656 015324                      ENDSRV
        015324 L10017:      RTI
        015324 000002
657
658 ;                                ERROR LIMIT CHECKING ROUTINE
659 ;                                DROPS DRIVE IF ERROR LIMIT EXCEEDED
660 015326 027737 165516 013722  CKERLM: CMP      @ERRPOINT,ERLIMW ;TEST IF ERROR LIMIT EXCEEDED
661 015334 002453                      BLT      1$ ;NO - SKIP
662 015336                      INLOOP  ;CHECK IF IN ERROR LOOP
        015336 04020      EMT      C$INLP
663 015340                      BCOMPLETE 1$ ;YES - SKIP
        015340 103451      BCS      1$
664 015342                      PRINTF  #FMT25,ERLIMW,#MEXERS
        015342 012746 010633      MOV      #MEXERS,-(SP)
        015346 013746 013722      MOV      ERLIMW,-(SP)
        015352 012746 012006      MOV      #FMT25,-(SP)

```

```

015356 012746 000003      MOV      #3,-(SP)
015362 010600      MOV      SP,RO
015364 104017      EMT      C$PNTF
665 015366 062706 000010      ADD      #10,SP
015372      PRINTF  #FMT5,#BASADD,RI BAS,#DRVNAM,<B,RLDRV+1>
015372 005046      CLR      -(SP)
015374 153716 002735      BISB    RLDRV+1,(SP)
015400 012746 005743      MOV      #DRVNAM,-(SP)
015404 013746 002730      MOV      RLBA,-(SP)
015410 012746 005732      MOV      #BASADD,-(SP)
015414 012746 011175      MOV      #FMT5,-(SP)
015420 012746 000005      MOV      #5,-(SP)
015424 010600      MOV      SP,RO
015426 104017      EMT      C$PNTF
666 015430 062706 000014      ADD      #14,SP
015434      PRINTF  #FMT3
015434 012746 011161      MOV      #FMT3,-(SP)
015440 012746 000001      MOV      #1,-(SP)
015444 010600      MOV      SP,RO
015446 104017      EMT      C$PNTF
667 015450 062706 000004      ADD      #4,SP
015454      DODU    PSETNM          ;DROP DRIVE
015454 013700 003254      MOV      PSETNM,RO
015460 104053      EMT      C$DODU
668 015462      DOCLN   ;GO TO CLEAN UP
015462 104044      EMT      C$DCLN
669 015464 000207      1$:    RTS      PC
670
671      ; READ AND STORE ALL RL11 REGISTERS
672 015466 016237 000000 002746  ;READRL: MOV      RLCSR(R2),T.CS ;GET CS REG
673 015474 016237 000002 002750      MOV      RLBA(R2),T.BA ;GET BUS ADDRESS REG
674 015502 016237 000004 002752      MOV      RLDA(R2),T.DA ;GET DISK ADDRESS
675 015510 016237 000006 002754      MOV      RLMP(R2),T.MP ;GET MULTI-PURPOSE REG
676 015516 000207      RTS      PC ;RETURN
677
678      ; WAIT FOR CONTROLLER TIMEOUT TO FORCE INTERRUPT ROUTINE
679 015520 011646      ;WAITIN: MOV      (SP),-(SP) ;MAKE ROOM FOR ERROR POINTER
680 015522 005066 000002      CLR      2(SP) ;CLEAR FOR POINTER
681 015526 032762 000200 000000      BIT      #CRDYMSK,RLCSR(R2) ;TEST IF CONTROLLER READY
682 015534 001420      BEQ      4$ ;NO - SKIP TO WAIT
683 015536 004737 015466      JSR      PC,READRL ;READ ALL RL REGS
684 015542 005737 002710      TST     DONE ;TEST IF INTERRUPT OCCURRED
685 015546 001433      BEQ      5$ ;NO - GO SET NO INTERRUPT ERR FLAG
686 015550 012766 006107 000002 1$:    MOV      #MTOSLOW,2(SP) ;ELSE SET TO SLOW ERROR POINTER
687 015556 032737 002000 002746      BIT      #OPIERR,T.CS ;TEST IF OPI SET
688 015564 001403      BEQ      2$ ;NO - SKIP
689 015566 012766 006126 000002      MOV      #MDRRES,2(SP) ;SET MESSAGE FOR NO DRIVE RESPONSE
690 015574 000207      2$:    RTS      PC ;RETURN
691 015576      4$:    WAITMS  #3 ;WAIT 300 MS FOR TIMEOUT
015576 012700 000003      MOV      #3,RO
015602 104026      EMT      C$WTM
692 015604 032762 000200 000000      BIT      #CRDYMSK,RLCS(R2) ;TEST IF READY NOW SET
693 015612 001006      BNE      3$ ;YES - SKIP
694 015614 004737 015466      JSR      PC,READRL ;READ RL REGS
695 015620 012766 006201 000002      MOV      #MCONHNG,2(SP) ;SET MESSAGE FOR CONTROLLER HUNG
696 015626 000762      BR       2$ ;SKIP
697 015630 005737 002710      3$:    TST     DONE ;ELSE CHECK IF INTERRUPT OCCURRED

```

```

698 015634 001345          BNE      1$           ;YES - SKIP TO SET TO SLOW
699 015636 004737 015466 5$: JSR      PC,READRL    ;READ RL REGS
700 015642 012766 006146 000002 MOV     #MNOINT,2(SP) ;ELSE SET NO INTERRUPT FLAG
701 015650 000751          BR       2$           ;GO TO RETURN
702
703          ; OPERATION AND TEST INITIALIZE ROUTINE
704 015652 005037 002706 1$TINT: CLR     OPFLAG      ;CLEAR OPERATION FLAGS
705 015656 105037 003257     CLR    NOERCT      ;RESET INHIBIT ERROR COUNTING
706 015662 005037 002716     CLR    MORECE      ;RESET MORE COMPARE ERRORS
707 015666 000207          RTS      PC
708
709          ; GET STATUS AND GET STATUS WITH RESET ROUTINE
710 015670 013746 003030  GSTATR: MOV     TEMP4,-(SP) ;STORE TEMP4
711 015674 012737 000013 003030 MOV     #GETSTAT!DRSET,TEMP4 ;SET FOR RESET
712 015702 000412          BR       GSTATG
713 015704 013746 003030  GSTATC: MOV    TEMP4,-(SP) ;STORE TEMP4
714 015710 012737 000003 003030 MOV     #GETSTAT,TEMP4 ;SET FOR NO RESET
715 015716 000404          BR       GSTATG
716 015720 013746 003030  GSTAT:  MOV    TEMP4,-(SP) ;STORE TEMP4
717 015724 005037 003030     CLR    TEMP4      ;SET FOR SAVE L. AND T. REGS
718 015730 010346  GSTATG: MCV     R3,-(SP) ;STORE R3
719 015732 013703 002704     MOV    SSINDX,R3 ;GET SUBROUTINE INDEX
720 015736 005723     TST    (R3)+      ;BUMP IT FOR NEXT ENTRY
721 015740 016663 000004 002306 MOV     4(SP),SUBSTK(R3) ;INSERT THIS CALL
722 015746 162763 000004 002306 SUB     #4,SUBSTK(R3) ;ADJUST IT TO CALLING LOCATION
723 015754 010337 002704     MOV    R3,SSINDX ;STORE IT BACK
724 015760 010046     MOV    R0,-(SP) ;STORE R0
725 015762 010146     MOV    R1,-(SP) ;STORE R1
726 015764 012737 000002 002720 MOV     #2,ERRSWI ;SET FOR NO ERROR RETURN
727 015772 032737 000010 003030 BIT     #DRSET,TEMP4 ;TEST IF DRIVE RESET
728 016000 001453     BEQ    11$         ;NO - SKIP
729 016002 032762 040000 000000 BIT     #DRVERR,RLCS(R2) ;TEST IF DRIVE ERROR SET
730 016010 001403     BEQ    49$         ;NO - SKIP
731 016012          WAITMS #3 ;WAIT FOR 300 MS FOR DRIVE TO SETTLE
      016012 012700 000003     MOV    #3,R0
      016016 104026     EMT    C$WTM
732 016020 012701 000062          49$: MOV     #50,R1 ;SET WAIT FOR 5 SEC
733 016024 004737 015720          50$: JSR     PC,GSTAT ;GET DRIVE STATUS
734 016030 016464          3$:
735 016032 032737 000001 002746 BIT     #DRDYMSK,T.CS ;TEST IF DRIVE READY
736 016040 001051     BNE    5$         ;YES - GO DO CLEAR
737 016042 032737 000020 002754 BIT     #HOSTAT,T.MP ;ELSE TEST IF HEADS OUT
738 016050 001010     BNE    51$        ;YES - BYPASS RELOAD WAIT FLAG SETTING
739 016052 032737 144000 002754 BIT     #SPDSTAT!HCESTAT!WDESTAT,T.MP ;TEST IF DRIVE HAS ERROR
740          ;THAT CAUSED HEADS TO
741          ;UNLOAD
742 016060 001441     BEQ    5$         ;NO - SKIP
743 016062 052737 040000 002706 BIS     #RELDWT,OPFLAG ;ELSE SET WAIT FLAG
744 016070 000435     BR     5$         ;SKIP TO CLEAR
745 016072 032737 040000 002746 51$: BIT     #DRVERR,T.CS ;TEST IF DRIVE ERROR NOW
746 016100 001031     BNE    5$         ;YES - SKIP TO CLEAR
747 016102          WAITMS #1 ;WAIT FOR DRIVE TO GET ERROR, RDY, OR HD
      016102 012700 000001     MOV    #1,R0
      016106 104026     EMT    C$WTM
748 016110 005301     DEC    R1 ;DEC WAIT COUNTER
749 016112 001344     BNE    50$        ;IF NOT DONE, LOOP
750 016114 012703 010511     MOV    #MUNDEF,R3 ;MESSAGE FOR UNDEFINED STATE

```



```

751 016120          ERRHRD 10001,,,ERR1
      016120 104443    TRAP   T$ERCODE
      016122 023421    .WORD  10001
      016124 012072    .WORD  ERR1
752 016126 000554    BR     14$          ;EXIT
753 016130 005737 003030 11$:  TST   TEMP4        ;TEST IF SAVE REGISTERS
754 016134 001013    BNE   5$          ;NO SKIP
755 016136 012701 000004    MOV   #4,R1        ;SET SAVE COUNT
756 016142 012703 002746    MOV   #L.MP+2,R3   ;SET ADDRESS OF FIRST SAVE
757 016146 014346    8$:  MOV   -(R3),-(SP) ;PUT REG ON STACK
758 016150 005301    DEC   R1          ;DEC COUNT
759 016152 001375    BNE   8$          ;LOOP UNTIL ALL SAVED
760 016154 012737 000003 002742    MOV   #GETSTAT,L.DA ;SET FOR GET STATUS
761 016162 000403    BR    6$          ;SKIP
762 016164 013737 003030 002742 5$:  MOV   TEMP4,L.DA   ;INSERT PRESET FOR STATUS
763 016172          6$:
764 016172 005037 002710    CLR   DONE        ;CLEAR INTERRUPT FLAG
765 016176 013737 002734 002736    MOV   RLDRV,L.CS  ;SET UP TO GET STATUS
766 016204 042737 002000 002736    BIC   #BIT10,L.CS ;CLEAR FOR DRIVE 4 - 7 SPEC'D
767 016212 052737 000104 002736    BIS   #GTSTAT,L.CS
768 016220 013762 002742 000004    MOV   L.DA,RLDA(R2) ;LOAD RL REGS
769 016226 013762 002736 000000    MOV   L.CS,RLCSR(R2) ;LOAD CS REG
770 016234          WAITUS #1 ;WAIT 100 US FOR INTERRUPT
      016234 012700 000001    MOV   #1,R0
      016240 104027    EMT   C$WTU
771 016242 005737 002710    TST   DONE        ;CHECK IF INTERRUPT OCCURRED
772 016246 001476    BEQ   1$          ;NO - SKIP
773 016250 013737 002754 002762 4$:  MOV   T.MP,T.STAT ;STORE MP REGISTER
774 016256 042737 177770 002762    BIC   #^C<STAMSK>,T.STAT ;CLEAR ALL BUT STATE
775 016264 032737 000010 002742    BIT   #DRSET,L.DA ;TEST IF RESET WAS SPECIFIED
776 016272 001474    BEQ   3$          ;NO - SKIP TO EXIT
777 016274 032737 040000 002706    BIT   #RELDWT,OPFLAG ;TEST IF RELOAD WAIT FLAG SET
778 016302 001424    BEQ   12$         ;NO - SKIP
779 016304 012701 001130    MOV   #600,,R1    ;SET WAIT COUNT FOR 60 SECONDS
780 016310 032762 000001 000000 13$:  BIT   #DRDYMSK,RLCS(R2) ;TEST IF DRIVE NOW READY
781 016316 001016    BNE   12$         ;YES - SKIP
782 016320          WAITMS #1 ;CALL WAIT
      016320 012700 000001    MOV   #1,R0
      016324 104026    EMT   C$WTM
783 016326 005301    DEC   R1          ;DEC COUNT
784 016330 001367    BNE   13$         ;LOOP IF NOT 0
785 016332 004737 015720    JSR   PC,GSTAT    ;GET DRIVE STATUS
786 016336 016464    3$          ;ERROR RETURN
787 016340 012703 010556    MOV   #MRLFAL,R3  ;SET RESULT MESSAGE POINTER
788 016344          ERRHRD 10003,,,ERR1
      016344 104443    TRAP   T$ERCODE
      016346 023423    .WORD  10003
      016350 012072    .WORD  ERR1
789 016352 000442    BR    14$         ;GO TO EXIT
790 016354          12$:  WAITUS #10.       ;WAIT FOR 1MS
      016354 012700 000012    MOV   #10,,R0
      016360 104027    EMT   C$WTU
791 016362 004737 015720    JSR   PC,GSTAT    ;GET DRIVE STATUS
792 016366 016464    3$
793 016370 032737 100000 002746    BIT   #ANYERR,T.CS ;TEST IF ANY ERROR
794 016376 001432    BEQ   3$          ;NO - SKIP
795 016400 032737 001000 002754    BIT   #VCSTAT,T.MP ;CHECK IF VOLUME CHECK RESET

```

```

796 016406 001403          BEQ      7$          ;YES SKIP
797 016410 012703 006235  MOV      #VCMRST,R3 ;SET REASON POINTER
798 016414 000416          BR       2$          ;EXIT
799 016416 032737 040000 002746 7$:  BIT      #DRVERR,1.CS ;CHECK IF DRIVE ERROR
800 016424 001404          BEQ      9$          ;NO - SKIP
801 016426          ERRHRD 10004,,,ERR6
      016426 104443      TRAP     T$ERCODE
      016430 023424      .WORD   10004
      016432 012374      .WORD   ERR6
802 016434 000411          BR       14$         ;EXIT
803 016436 012703 006256          9$:  MOV      #UNXERR,R3 ;SET REASON POINTER
804 016442 000403          BR       2$          ;EXIT
805 016444 004737 015520          1$:  JSR      PC,WAITIN ;WAIT FOR INTERRUPT
806 016450 012603          MOV      (SP)+,R3    ;STORE REASON POINTER FOR RETURN
807 016452          ERRHRD 10002,,,ERR1
      016452 104443      TRAP     T$ERCODE
      016454 023422      .WORD   10002
      016456 012072      .WORD   ERR1
808 016460 005037 002720          14$: CLR      ERRSWI    ;CLEAR FOR ERROR RETURN
809 016464 005737 003030          3$:  TST      TEMP4    ;TEST IF REGISTERS WERE SAVED
810 016470 001007          BNE     22$         ;NO - SKIP
811 016472 012703 002736          MOV      #L.CS,R3  ;SET POINTER TO RESTORE
812 016476 012701 000004          MOV      #4,R1    ;SET REGISTER COUNT
813 016502 012623          MOV      (SP)+,(R3)+ ;RESTORE REG
814 016504 005301          DEC     R1        ;DEC COUNT
815 016506 001375          BNE     20$         ;LOOP UNTIL ALL ARE RESTORED
816 016510 162737 000002 002704 22$: SUB      #2,SSINDX ;REMOVE ENTRY FROM SUBROUT STACK
817 016516 012601          MOV      (SP)+,R1  ;RESTORE R1
818 016520 012600          MOV      (SP)+,R0
819 016522 012603          MOV      (SP)+,R3  ;RESTORE R3
820 016524 012637 003030          MOV      (SP)+,TEMP4 ;RESTORE TEMP4
821 016530 005737 002720          TST      ERRSWI    ;TEST IF ERROR RETURN
822 016534 001403          BEQ      99$        ;YES - SKIP
823 016536 063716 002720          ADD     ERRSWI,(SP) ;ADD IN ERROR RETURN
824 016542 000207          RTS     PC
825 016544 017616 000000          99$: MOV      @ (SP),(SP) ;SET ERROR RETURN ADDRESS
826 016550 000207          RTS     PC
827
828
829          ;
830 016552 012737 177777 003022 XSEEK: MOV      #-1,TEMP1 ;SET SPECIAL TIMING SEEK FLAG
831 016560 000402          BR       XSEEK1
832 016562 005037 003022 XSEEK: CLR      TEMP1 ;CLEAR SPECIAL SEEK FOR TIMING FLAG
833 016566 010346 XSEEK1: MOV      R3,-(SP) ;STORE R3
834 016570 013703 002704          MOV      SSINDX,R3 ;GET SUBROUTINE INDEX
835 016574 005723          TST      (R3)+    ;BUMP IT FOR NEXT ENTRY
836 016576 016663 000002 002306          MOV      2(SP),SUBSTK(R3) ;INSERT THIS CALL
837 016604 162763 000004 002306          SUB      #4,SUBSTK(R3) ;ADJUST IT TO CALLING LOCATION
838 016612 010337 002704          MOV      R3,SSINDX ;STORE IT BACK
839 016616 010046          MOV      R0,-(SP)
840 016620 010146          MOV      R1,-(SP)
841 016622 010546          MOV      R5,-(SP) ;STORE REG
842 016624 012737 000002 002720          MOV      #2,ERRSWI ;SET FOR NO ERROR RETURN
843 016632 005037 003000          CLR      DIFAUG   ;CLEAR DIFFERENCE AUGMENT (FOR SEEKING
844          ; PAST GUARD BAND)
845 016636 004737 021114          JSR      PC,GETPOS ;GET PRESENT POSITION
846 016642 017264          65$

```

```

847 016644 013737 003006 003002      MOV      CURCYL,OLDCYL      ;MOVE CURRENT TO OLD CYLINDER
848 016652 023737 003004 002204      CMP      NEWCYL,HLMTW      ;TEST IF NEW IS GREATER THAN 255
849 016660 003427                    BLE      3$                ;NO - SKIP
850 016662 163737 002204 003004      SUB      HLMTW,NEWCYL      ;ELSE SUBTRACT 255.
851 016670 013737 003004 003000      MOV      NEWCYL,DIFAUG     ;STORE DIFFERENCE AS AUGMENT
852 016676 013737 002204 003004      MOV      HLMTW,NEWCYL     ;SET NEWCYL AS 255.
853 016704 022737 000001 002200      CMP      #1,T.DRIVE
854 016712 001424                    BEQ      6$
855 016714 162737 000001 003004      SUB      #1,NEWCYL
856 016722 012737 000001 003012      MOV      #1,DESSGN
857 016730 012737 000001 003010      MOV      #1,DESDIF
858 016736 000451                    BR       18$
859 016740 005737 003004          3$:    TST      NEWCYL          ;TEST IF NEWCYL HAS NEGATIVE VALUE
860 016744 100007                    BPL      6$                ;NO - SKIP
861 016746 005437 003004          NEG      NEWCYL          ;ELSE MAKE IT POSITIVE
862 016752 013737 003004 003000      MOV      NEWCYL,DIFAUG     ;AND STORE IT AS AUGMENT
863 016760 005037 003004          CLR      NEWCYL          ;AND SET NEWCYL TO 0
864 016764 013705 003006          6$:    MOV      CURCYL,R5        ;COMPUTE DIFFERENCE AND NEW CYLINDER
865 016770 163705 003004          SUB      NEWCYL,R5        ;SUB NEWCYL FROM CURCYL
866 016774 100005                    BPL      13$              ;IF DIFF IS POSITIVE - SKIP(REV SEEK)
867 016776 012737 000001 003012      MOV      #1,DESSGN        ;ELSE SET SIGN FOR FORWARD
868 017004 005405                    NEG      R5                ;MAKE DIFFERENCE POSITIVE
869 017006 000402                    BR       14$              ;SKIP
870 017010 005037 003012          13$:   CLR      DESSGN          ;SET SIGN FOR REVERSE
871 017014 010537 003010          14$:   MOV      R5,DESDIF     ;STORE DIFFERENCE
872 017020 005737 003000          TST      DIFAUG          ;IS THERE A DIFFERENCE AUGMENT
873 017024 001416                    BEQ      18$              ;NO - SKIP
874 017026 023737 003004 002204      CMP      NEWCYL,HLMTW     ;CHECK IF NEW CYL IS 255.
875 017034 001007                    BNE      17$              ;NO - SKIP
876 017036 012737 000001 003012      MOV      #1,DESSGN        ;ELSE FORCE SIGN FOR FORWARD
877                                ;(INNER GUARD BAND)
878 017044 022737 000001 002200      CMP      #1,T.DRIVE
879 017052 001003                    BNE      18$
880 017054 063737 003000 003010          17$:   ADD      DIFAUG,DESDIF
881 017062                                18$:
882 017062 012705 002736          MOV      #L.CS,R5        ;GET L REG ADDRESS
883 017066 012715 000106          MOV      #SEEK,(R5)      ;SET FOR SEEK
884 017072 053715 002734          BIS      RLDRV,(R5)      ;INSERT DRIVE NUMBER
885 017076 042725 002000          BIC      #BIT10,(R5)+    ;CLEAR IF DRIVE 4 - 7 SPEC'D
886 017102 005025                    CLR      (R5)+           ;CLEAR BUS ADDRESS
887 017104 013715 003010          MOV      DESDIF,(R5)     ;LOAD DIFFERENCE
888 017110 012700 000007          MOV      #7,R0          ;SET TO SHIFT DIFFERENCE
889 017114 006315          21$:   ASL      (R5)
890 017116 005300                    DEC      R0
891 017120 001375                    BNE      21$              ;LOOP UNTIL ALIGNED
892 017122 005737 003012          TST      DESSGN          ;TEST SIGN
893 017126 001402                    BEQ      23$              ;SKIP IF 0
894 017130 052715 000004          BIS      #DIRBIT,(R5)    ;ELSE INSERT SIGN
895 017134 005737 003014          23$:   TST      DESHD          ;TEST IF HEAD 0
896 017140 001402                    BEQ      25$              ;YES - SKIP
897 017142 052715 000020          BIS      #HDSSEL,(R5)    ;ELSE SET HEAD BIT
898 017146 052725 000001          25$:   BIS      #MBSET0,(R5)+  ;INSERT MARKER BIT
899 017152 004737 017610          JSR      PC,RDYCHK       ;CHECK IF DRIVE READY
900 017156 017264                    65$:
901 017160 005037 002710          CLR      DONE            ;CLEAR INTERRUPT FLAG -
902 017164 005737 003022          TST      TEMP1          ;CHECK IF SPECIAL SEEK FLAG SET
903 017170 001035                    BNF      65$              ;YES - SKIP, DO NOT START SEEK

```

```

904 017172 014562 000004      MOV      -(R5),RLDA(R2)  ;LOAD RL REGISTERS
905 017176 014562 000002      MOV      -(R5),RLBA(R2)
906 017202 014562 000000      MOV      -(R5),RLCS(R2)
907 017206          30$: WAITUS #10.
      017206 012700 000012      MOV      #10.,RO
      017212 104027          EMI      C$WTU
908 017214 005737 002710      TST      DONE          ;TEST IF INTERRUPT DONE
909 017220 001011          BNE      32$          ;YES - SKIP
910 017222 004737 015520      JSR      PC,WAITIN     ;GO WAIT FOR INTERRUPT
911 017226 012603          MOV      (SP)+,R3      ;GET RESULT MESSAGE POINTER
912 017230          ERRHRD 10005...,ERR1
      017230 104443          TRAP     T$ERCODE
      017232 023425          .WORD   10005
      017234 012072          .WORD   ERR1
913 017236 005037 002720      CLR      ERRSWI        ;CLEAR FOR ERROR ERROR RETURN
914 017242 000410          BR       65$
915 017244 005737 002746          32$: TST      T.CS          ;TEST IF ANY ERROR
916 017250 100005          BPL      65$          ;NO - SKIP
917 017252          ERRHRD 10006...,ERR6
      017252 104443          TRAP     T$ERCODE
      017254 023426          .WORD   10006
      017256 012374          .WORD   ERR6
918 017260 005037 002720      CLR      ERRSWI        ;CLEAR FOR ERROR ERROR RETURN
919 017264 162737 000002 002704 65$: SUB      #2,SSINDEX     ;REMOVE ENTRY FROM SUBROUT STACK
920 017272 012605          MOV      (SP)+,R5      ;RESTORE REGISTER
921 017274 012601          MOV      (SP)+,R1
922 017276 012600          MOV      (SP)+,R0
923 017300 012603          MOV      (SP)+,R3      ;RESTORE R3
924 017302 005737 002720      TST      ERRSWI        ;TEST IF ERROR RETURN
925 017306 001403          BEQ     99$          ;YES - SKIP
926 017310 063716 002720      ADD     ERRSWI,(SP)    ;ADD IN ERROR RETURN
927 017314 000207          RTS     PC
928 017316 017616 000000          99$: MOV     @ (SP),(SP) ;SET ERROR RETURN ADDRESS
929 017322 000207          RTS     PC
930
932 017324 010346          SIMSEK: MOV     R3,-(SP) ;STORE REGISTERS
933 017326 013703 002704      MOV     SSINDEX,R3    ;GET SUBROUTINE INDEX
934 017332 005723          TST     (R3)+        ;BUMP IT FOR NEXT ENTRY
935 017334 016663 000002 002306      MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
936 017342 162763 000004 002306      SUB     #4,SUBSTK(R3) ;ADJUST IT TO CALLING LOCATION
937 017350 010337 002704      MOV     R3,SSINDEX    ;STORE IT BACK
938 017354 010046          MOV     R0,-(SP)
939 017356 010446          MOV     R4,-(SP)
940 017360 012737 000002 002720      MOV     #2,ERRSWI     ;SET FOR NO ERROR RETURN
941 017366 004737 017610      JSR     PC,RDYCHK     ;CHECK IF DRIVE READY
942 017372 017552          65$
943 017374 012704 002736      MOV     #L.CS,R4      ;GET POINTER TO L REGS
944 017400 012714 000106      MOV     #SEEK,(R4)    ;SET FOR SEEK
945 017404 053714 002734      BIS     RLDRV,(R4)    ;INSERT DRIVE NUMBER
946 017410 042724 002000      BIC     #BIT10,(R4)+  ;CLEAR FOR DRIVE 4 - 7 SPEC'D
947 017414 005024          CLR     (R4)+        ;CLEAR BUS ADDRESS
948 017416 013714 003010      MOV     DESDIF,(R4)   ;LOAD DIFFERENCE
949 017422 012703 000007          MOV     #7,R3        ;SET COUNT FOR SHIFT TO ALIGN
950 017426 006314          3$: ASL     (R4)        ;ALIGN DIFFERENCE IN DA
951 017430 005303          DEC     R3
952 017432 001375          BNE     3$
953 017434 005737 003012          TST     DESSGN        ;TEST IF SIGN SET

```

```

954 017440 001402          BEQ      5$          ;NO - SKIP
955 017442 052714 000004   BIS      #DIRBIT,(R4) ;INSERT SIGN
956 017446 005737 003014   5$: TST     DESHD      ;TEST IF HEAD 0
957 017452 001402          BEQ      7$          ;YES - SKIP
958 017454 052714 000020   BIS      #HDSSEL,(R4) ;INSERT HEAD BIT
959 017460 052724 000001   7$: BIS      #MBSET0,(R4)+ ;INSERT MARKER BIT
960 017464 005037 002710   CLR      DONE        ;CLEAR INTERRUPT FLAG
961 017470 012701 000012   MOV      #10.,R1     ;SET WAIT COUNT FOR 800US
962 017474 014462 000004   MOV      -(R4),RLDA(R2) ;LOAD RL REGISTERS
963 017500 014462 000002   MOV      -(R4),RLBA(R2)
964 017504 014462 000000   MOV      -(R4),RLCS(R2)
965 017510 005737 002710   10$: TST     DONE        ;CHECK IF INTERRUPTED
966 017514 001016          BNE      65$        ;YES - SKIP
967 017516 005301          DEC      R1          ;DEC WAIT COUNT
968 017520 001404          BEQ      13$        ;IF 0 - SKIP
969 017522          WAITUS #1
      017522 012700 000001   MOV      #1,R0
      017526 104027          EMT      C$WTU
970 017530 000767          BR       10$        ;GO CHECK DONE
971 017532 004737 015520   13$: JSR     PC,WAITIN ;GO WAIT FOR TIMEOUT
972 017536 012603          MOV      (SP)+,R3    ;GET RESULT MESSAGE POINTER
973 017540          ERRHRD 10011,,ERR1
      017540 104443          TRAP    T$ERCODE
      017542 023433          .WORD  10011
      017544 012072          .WORD  ERR1
974 017546 005037 002720   CLR      ERRSWI      ;CLEAR FOR ERROR ERROR RETURN
975 017552          14$:
976 017552 162737 000002 002704 65$: SUB      #2,SSINDX    ;REMOVE ENTRY FROM SUBROUT STACK
977 017560 012604          MOV      (SP)+,R4    ;RESTORE REGS
978 017562 012600          MOV      (SP)+,R0
979 017564 012603          MOV      (SP)+,R3
980 017566 005737 002720   TST     ERRSWI      ;TEST IF ERROR RETURN
981 017572 001403          BEQ      99$        ;YES - SKIP
982 017574 063716 002720   ADD     ERRSWI,(SP)  ;ADD IN ERROR RETURN
983 017600 000207          RTS      PC
984 017602 017616 000000   99$: MOV      @ (SP),(SP) ;SET ERROR RETURN ADDRESS
985 017606 000207          RTS      PC
987
1063          ; DRIVE READY TEST ROUTINE. CHECKS DEIVE IS READY. IF NOT, WAIT
1064          ; 500MS FOR READY TO SET.
RDYCHK: MOV      R3,-(SP) ;STORE REGS
      MOV      SSINDX,R3 ;GET SUBROUTINE INDEX
      TST     (R3)+ ;BUMP IT FOR NEXT ENTRY
      MOV      2(SP),SUBSTK(R3) ;INSERT THIS CALL
      SUB     #4,SUBSTK(R3) ;ADJUST IT TO CALLING LOCATION
      MOV      R3,SSINDX ;STORE IT BACK
      MOV      R0,-(SP)
      MOV      R1,-(SP)
      MOV      R4,-(SP)
1074 017646 012737 000002 002720   MOV      #2,ERRSWI ;SET FOR NO ERROR RETURN
1075 017654 012701 011610   MOV      #5000.,R1 ;SET WAIT COUNT
1076 017660 004737 015720   1$: JSR     PC,GSTAT ;GET DRIVE STATUS
1077 017664 020004          4$
1078 017666 032737 000001 002746   BIT      #DRDYMSK,T.CS ;TEST IF DRIVE READY
1079 017674 001045          BNE      5$          ;YES - EXIT
1080 017676          WAITUS #1
      017676 012700 000001   MOV      #1,R0

```

```

1081 017702 104027      EMT      CSWTU
1081 017704 005301      DEC      R1          ;DEC WAIT COUNT
1082 017706 001364      BNE      1$          ;LOOP IF NOT 0
1083 017710 012703 010013  MOV      #MDRDY,R3   ;SET RESULT MESSAGE POINTER
1084 017714 012704 011012  MOV      #C500MS,R4  ;SET CONDITION MESSAGE POINTER
1085 017720      ERRHRD 10010,,,ERR5
      017720 104443      TRAP    T$ERCODE
      017722 023432      .WORD  10010
      017724 012324      .WORD  ERR5
1086 017726 012701 000062  MOV      #50,,R1     ;SET WAIT COUNT FOR 5 SECONDS
1087 017732 004737 015720 2$:      JSR     PC,GSTAT    ;GET DRIVE STATUS
1088 017736 020004      4$
1089 017740 032737 000001 002746  BIT      #DRDYMSK,T.CS ;TEST IF DRIVE READY
1090 017746 001005      BNE      3$          ;YES - SKIP
1091 017750      WAITMS #1          ;WAIT FOR 100MS
      017750 012700 000001  MOV      #1,R0
1092 017756 005301      EMT      CSWTM
1093 017760 001364      DEC      R1          ;DEC WAIT COUNTER
1094 017762 032737 100000 002746 3$:      BNE      2$          ;LOOP UNTIL TIME DONE
1095 017770 001405      BIT      #ANYERR,T.CS ;TEST IF ANYERR SET
1096 017772      BEQ     4$          ;NO - SKIP
      017772 104443      ERRHRD 10011,,,ERR6 ;REPORT ALL ERRORS
      017774 023433      TRAP    T$ERCODE
      017776 012374      .WORD  10011
      017776 012374      .WORD  ERR6
1097 020000 005337 003052      DEC      ERRCNT     ;REDUCE ERROR COUNT FOR DUAL ERRORS
1098 020004 005037 002720 4$:      CLR     ERRSWI     ;CLEAR FOR ERROR RETURN
1099 020010 162737 000002 002704 5$:      SUB     #2,SSINDX   ;REMOVE ENTRY FROM SUBROUT STACK
1100 020016 012604      MOV     (SP)+,R4    ;RESTORE REGS
1101 020020 012601      MOV     (SP)+,R1
1102 020022 012600      MOV     (SP)+,R0
1103 020024 012603      MOV     (SP)+,R3
1104 020026 005737 002720      TST     ERRSWI     ;TEST IF ERROR RETURN
1105 020032 001403      BEQ     99$        ;YES - SKIP
1106 020034 063716 002720      ADD     ERRSWI,(SP) ;ADD IN ERROR RETURN
1107 020040 000207      RTS     PC
1108 020042 017616 000000 99$:      MOV     @ (SP),(SP) ;SET ERROR RETURN ADDRESS
1109 020046 000207      RTS     PC
1110
1111      ;      CHOSE HEAD ROUTINE. PICKS HEAD 0 UNLESS SPECIFIC HEAD IS
1112      ;      SELECTED BY SOFTWARE PARAMETER.
1113 020050 005037 003014  CHOSHD: CLR     DESHD    ;CLEAR TO HEAD 0
1114 020054 032737 010000 013712  BIT      #HEADLM,MISWIW ;TEST IF HEAD SPECIFIED
1115 020062 001403      BEQ     1$          ;NO - SKIP
1116 020064 013737 013720 003014  MOV     HEADW,DESHD ;INSERT SPECIFIED HEAD
1117 020072 000207      RTS     PC
1118
1119      ;      SWAP HEAD ROUTINE. CHANGES SELECTED HEAD TO HEAD 1
1120      ;      UNLESS HEAD 0 SPECIFICALLY SELECTED BY SOFTWARE PARAMETER.
1121 020074 032737 010000 013712  SWAPHD: BIT      #HEADLM,MISWIW ;TEST IF HEAD SPECIFIED
1122 020102 001011      BNE     2$          ;YES - TAKE ABORT EXIT
1123 020104 005737 003014      TST     DESHD     ;TEST IF HEAD ONE USED
1124 020110 001006      BNE     2$          ;YES - TAKE ABORT EXIT
1125 020112 012737 000001 003014  MOV     #1,DESHD   ;ELSE SET FOR HEAD ONE
1126 020120 062716 000002      ADD     #2,(SP)    ;BUMP PAST ABORT RETURN
1127 020124 000207      RTS     PC
1128 020126 017616 000000 2$:      MOV     @ (SP),(SP) ;GET ABORT DESTINATION

```

```

1129 020132 000207      3$:   RTS      PC
1130
1131
1132 020134 010046      ;      SWAP OLD CYLINDER AND NEW CYLINDER ROUTINE.
1133 020136 013700 003002  ;ONSWAP: MOV      RO,-(SP)      ;STORE RO
1134 020142 013737 003004 003002  ;      MOV      OLDCYL,RO      ;MOVE OLD TO RO
1135 020150 010037 003004      ;      MOV      NEWCYL,OLDCYL  ;MOVE NEW TO OLD
1136 020154 012600      ;      MOV      RO,NEWCYL     ;PUT OLD IN NEW
1137 020156 000207      ;      MOV      (SP)+,RO      ;RESTORE RO
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154 020160 012737 000001 003030  ;      XRDHDC: MOV      #1,TEMP4      ;SET FLAG TO BYPASS REG STORAGE
1155 020166 000402      ;      BR      XRDHDG      ;GO DO IT
1156 020170 005037 003030      ;      XRDHD: CLR      TEMP4      ;SET FLAG TO SAVE T. AND L. REGS
1157 020174 010346      ;      XRDHDG: MOV      R3,-(SP)     ;STORE REGISTERS
1158 020176 013703 002704      ;      MOV      SSINDX,R3      ;GET SUBROUTINE INDEX
1159 020202 005723      ;      TST      (R3)+      ;BUMP IT FOR NEXT ENTRY
1160 020204 016663 000002 002306  ;      MOV      2(SP),SUBSTK(R3) ;INSERT THIS CALL
1161 020212 162763 000004 002306  ;      SUB      #4,SUBSTK(R3) ;ADJUST IT TO CALLING LOCATION
1162 020220 010337 002704      ;      MOV      R3,SSINDX     ;STORE IT BACK
1163 020224 010046      ;      MOV      RO,-(SP)
1164 020226 010146      ;      MOV      R1,-(SP)
1165 020230 010446      ;      MOV      R4,-(SP)
1166 020232 012737 000002 002720  ;      MOV      #2,ERRSWI      ;SET FOR NO ERROR RETURN
1167 020240 005737 003030      ;      TST      TEMP4      ;TEST IF REGISTERS TO BE SAVED
1168 020244 001007      ;      BNE      2$          ;NO - SKIP
1169 020246 012703 002746      ;      MOV      #L.MP+2,R3     ;SET POINTER FOR REGS
1170 020252 012701 000004      ;      MOV      #4,R1          ;SET COUNT
1171 020256 014346      1$:   MOV      -(R3),-(SP)      ;SAVE REGISTER
1172 020260 005301      ;      DEC      R1            ;DEC COUNT
1173 020262 001375      ;      BNE      1$          ;LOOP UNTIL ALL ARE SAVED
1174 020264 004737 017610      2$:   JSR      PC,RDYCHK      ;CHECK DRIVE READY
1175 020270 020524      65$:
1176 020272 005037 002710      CLR      DONE          ;CLEAR INTERRUPT FLAG
1177 020276 012701 002736      MOV      #L.CS,R1      ;GET ADDRESS OF LOAD REGS
1178 020302 013711 002734      MOV      RLDRV,(R1)    ;LOAD DRIVE NUMBER
1179 020306 042711 002000      BIC      #BIT10,(R1)   ;CLEAR FOR DRIVE 4 - 7 SPEC'D
1180 020312 052721 000110      BIS      #RDHEAD,(R1)+ ;INSERT COMMAND
1181 020316 005021      CLR      (R1)+        ;CLEAR BA
1182 020320 005021      CLR      (R1)+        ;CLEAR DA
1183 020322 014162 000004      MOV      -(R1),RLDA(R2) ;LOAD RL11 REGS
1184 020326 014162 000002      MOV      -(R1),RLBA(R2)
1185 020332 014162 000000      MOV      -(R1),RLCSR(R2)
1186 020336      3$:   WAITUS #10.          ;WAIT 1MS FOR INTERRUPT
1187 020336 012700 000012      MOV      #10.,RO
1188 020342 104027      EMT      CSWTU
1189 020344 005737 002710      TST      DONE          ;TEST IN INTERRUPT FLAG SET
1190 020350 001455      BEQ      14$          ;NO - SKIP
1191 020352 032737 000001 002746  5$:   BIT      #DRDYMSK,T.CS ;TEST IF DRIVE READY
1192 020360 001033      BNE      10$          ;YES - SKIP
1193 020362 012703 010013      MOV      #MDRDY,R3     ;SET NO READY MESSAGE
1194 020366 012704 011031      MOV      #CAFDT,R4     ;CONDITION OF AFTER DATA XFER
1195 020372      ERRHRD 10017.,ERR5
1196 020372 104443      TRAP    T$ERRCODE
1197 020374 023441      .WORD  10017
1198 020376 012324      .WORD  ERR5
1199 020400 012701 000062      MOV      #50.,R1      ;SET WAIT COUNT FOR 5 SECONDS

```

```

1195 020404 004737 015720      4$: JSR    PC,GSTAT      ;GET STATUS
1196 020410 020520                60$:
1197 020412 032737 000001 002746 BIT    #DRDYMSK,T.CS  ;TEST IF DRIVE HAS COME READY
1198 020420 001403                11$: BEQ    11$           ;NO - SKIP
1199 020422 005037 002720        CLR    ERRSWI        ;CLEAR ERROR SWITCH
1200 020426 000410                10$: BR     10$           ;SKIP
1201 020430 005301                11$: DEC    R1           ;DEC WAIT COUNT
1202 020432 001364                4$: BNE    4$           ;LOOP UNTIL TIME DONE
1203 020434 012704 011042        MOV    #CSSEC,R4     ;SET CONDITION AFTER 5 SECONDS
1204 020440                ERRHRD 10014,,,ERR5
      020440 104443          TRAP  T$ERCODE
      020442 023436          .WORD 10014
      020444 012324          .WORD ERR5
1205 020446 000424                BR     60$           ;EXIT
1206 020450 005737 002746        10$: TST    T.CS        ;CHECK FOR ANY ERRORS
1207 020454 100004                BPL    12$           ;NO - SKIP
1208 020456                ERRHRD 10016,,,ERR6  ;REPORT ALL ERRORS
      020456 104443          TRAP  T$ERCODE
      020460 023440          .WORD 10016
      020462 012374          .WORD ERR6
1209 020464 000415                BR     60$
1210 020466 012701 002756        12$: MOV    #HDWRD2,R1    ;GET POINTER
1211 020472 016221 000006        MOV    RLMP(R2),(R1)+ ;STORE LAST TWO HEADER WORDS
1212 020476 016221 000006        MOV    RLMP(R2),(R1)+
1213 020502 000410                BR     65$           ;EXIT
1214 020504 004737 015520        14$: JSR    PC,WAITIN   ;WAIT FOR INTERRUPT
1215 020510 012603                MOV    (SP)+,R3      ;GET RESULTS
1216 020512                ERRHRD 10015,,,ERR1  ;REPORT
      020512 104443          TRAP  T$ERCODE
      020514 023437          .WORD 10015
      020516 012072          .WORD  ERR1
1217 020520 005037 002720        60$: CLR    ERRSWI        ;CLEAR FOR ERROR ERROR RETURN
1218 020524 005737 003030        65$: TST    TEMP4       ;TEST IF REGISTERS WERE SAVED
1219 020530 001007                BNE    22$           ;NO - SKIP
1220 020532 012703 002736        MOV    #L.CS,R3     ;SET POINTER TO RESTORE REGS
1221 020536 012701 000004        MOV    #4,R1        ;SET COUNT
1222 020542 012623                20$: MOV    (SP)+,(R3)+   ;RESTORE REGISTER
1223 020544 005301                DEC    R1           ;DEC COUNT
1224 020546 001375                BNE    20$         ;LOOP UNTIL ALL ARE RESTORED
1225 020550 162737 000002 002704 22$: SUB    #2,SSINDX    ;REMOVE ENTRY FROM SUBROUT STACK
1226 020556 012604                MOV    (SP)+,R4     ;RESTORE REGS
1227 020560 012601                MOV    (SP)+,R1
1228 020562 012600                MOV    (SP)+,R0
1229 020564 012603                MOV    (SP)+,R3
1230 020566 005737 002720        TST    ERRSWI        ;TEST IF ERROR RETURN
1231 020572 001403                BEQ    99$           ;YES - SKIP
1232 020574 063716 002720        ADD    ERRSWI,(SP)  ;ADD IN ERROR RETURN
1233 020600 000207                RTS    PC
1234 020602 017616 000000        99$: MOV    @ (SP),(SP) ;SET ERROR RETURN ADDRESS
1235 020606 000207                RTS    PC
1236
1312 ; POSITION HEAD BIT FROM HEADER OR MULTIPURPOSE REGISTER TO LSB.
1313 020610 013705 002754        POSHW1: MOV    HDWRD1,R5 ;START FOR POSITION HD BIT IN WD 1
1314 020614 000402                BR     POSHD0       ;SKIP
1315 020616 013705 002754        POSHSB: MOV    T.MP,R5 ;START FOR POSITION HD BIT IN MP
1316 020622 010146                POSHD0: MOV    R1,-(SP) ;STORE R1
1317 020624 042705 177677        BIC    #^CHSSTAT,R5 ;CLEAR ALL BUT HEAD SEL BIT

```



```

1318 020630 012701 000006          MOV    #6,R1          ;SET SHIFT COUNT
1319 020634 006205          1$:   ASR    R5          ;SHIFT FOR RIGHT JUSTIFY
1320 020636 005301          DEC    R1
1321 020640 001375          BNE   1$
1322 020642 012601          MOV    (SP)+,R1      ;RESTORE R1
1323 020644 000207          RTS   PC             ;RETURN
1324
1325          :           WAIT FOR READY ROUTINE. DURATION OF WAIT PASSED TO THE ROUTINE
1326          :           FROM THE CALLING ROUTINE IN R1.
1327 020646 010346          RDYWAIT:  MOV    R3,-(SP)      ;STORE R3
1328 020650 013703 002704          MOV    SSIDX,R3      ;GET SUBROUTINE INDEX
1329 020654 005723          TST   (R3)+          ;BUMP IT FOR NEXT ENTRY
1330 020656 016663 000002 002306          MOV    2(SP),SUBSTK(R3) ;INSERT THIS CALL
1331 020664 162763 000004 002306          SUB    #4,SUBSTK(R3) ;ADJUST IT TO CALLING LOCATION
1332 020672 010337 002704          MOV    R3,SSIDX      ;STORE IT BACK
1333 020676 010046          MOV    R0,-(SP)
1334 020700 010146          MOV    R1,-(SP)
1335 020702 010446          MOV    R4,-(SP)
1336 020704 012737 000002 002720          MOV    #2,ERRSWI     ;SET FOR NO ERROR RETURN
1337 020712 004737 015720          5$:   JSR    PC,GSTAT     ;GET DRIVE STATUS
1338 020716 021050          10$
1339 020720 032737 000001 002746          BIT    #DRDYMSK,T.CS ;CHECK IF READY
1340 020726 001052          BNE   9$             ;YES - SKIP
1341 020730 005301          DEC    R1             ;DEC WAIT COUNT
1342 020732 001404          BEQ   7$             ;SKIP IF 0
1343 020734          WAITUS #1
1344 020734 012700 000001          MOV    #1,R0
1345 020740 104027          EMT   C$WTU
1346 020742 000763          BR    5$
1347 020744 012703 010013          7$:   MOV    #MDRDY,R3      ;SET NAME MESSAGE PTR
1348 020750          ERRHRD 10020,,,ERR3     ;REPORT READY ERROR
1349 020750 104443          TRAP  T$ERCODE
1350 020752 023444          .WORD 10020
1351 020754 012206          .WORD ERR3
1352 020756 012701 000062          MOV    #50.,R1      ;SET WAIT COUNT FOR 5 SECONDS
1353 020762 004737 015720          6$:   JSR    PC,GSTAT     ;GET DRIVE STATUS
1354 020766 021050          10$
1355 020770 032737 000001 002746          BIT    #DRDYMSK,T.CS ;TEST IF DRIVE READY
1356 020776 001013          BNE   8$             ;YES - SKIP
1357 021000          WAITMS #1           ;WAIT 100 MS
1358 021000 012700 000001          MOV    #1,R0
1359 021004 104026          EMT   C$WTM
1360 021006 005301          DEC    R1             ;DEC WAIT COUNT
1361 021010 001364          BNE   6$             ;LOOP UNTIL TIME DONE
1362 021012 012704 011042          MOV    #C5SEC,R4     ;SET CONDITION AFTER 5 SECDS
1363 021016          ERRHRD 10021,,,ERR5
1364 021016 104443          TRAP  T$ERCODE
1365 021020 023445          .WORD 10021
1366 021022 012324          .WORD ERR5
1367 021024 000407          BR    11$
1368 021026 032737 100000 002746          8$:   BIT    #ANYERR,T.CS ;TEST IF ANY ERROR SET
1369 021034 001405          BEQ   10$            ;NO - SKIP
1370 021036          ERRHRD 10022,,,ERR6 ;REPORT ALL ERRORS
1371 021036 104443          TRAP  T$ERCODE
1372 021040 023446          .WORD 10022
1373 021042 012374          .WORD ERR6
1374 021044 005337 003052          11$:  DEC    ERRCNT        ;DEC FOR DOUBLE ERROR REPORT

```

```

1362 021050 005037 002720 10$: CLR ERRSWI ;CLEAR FOR ERROR ERROR RETURN
1363 021054 162737 000002 002704 9$: SUB #2,SSINDX ;REMOVE ENTRY FROM SUBROUT STACK
1364 021062 012604 MOV (SP)+,R4 ;RESTORE REGISTERS
1365 021064 012601 MOV (SP)+,R1
1366 021066 012600 MOV (SP)+,R0
1367 021070 012603 MOV (SP)+,R3 ;RESTORE R3
1368 021072 005737 002720 TST ERRSWI ;TEST IF ERROR RETURN
1369 021076 001403 BEQ 99$ ;YES - SKIP
1370 021100 063716 002720 ADD ERRSWI,(SP) ;ADD IN ERROR RETURN
1371 021104 000207 RTS PC
1372 021106 017616 000000 99$: MOV @ (SP),(SP) ;SET ERROR RETURN ADDRESS
1373 021112 000207 RTS PC
1374
1375 ; GET POSITION ROUTINE. READS A HEADER FROM CURRENT CYLINDER
1376 ; (WHERE IT IS PRESENTLY POSITIONED) AND STORES CYLINDER
1377 ; NUMBER IN CURCYL.
1378 021114 010346 GETPOS: MOV R3,-(SP) ;STORE REGISTERS
1379 021116 013703 002704 MOV SSINDX,R3 ;GET SUBROUTINE INDEX
1380 021122 005723 TST (R3)+ ;BUMP IT FOR NEXT ENTRY
1381 021124 016663 000002 002306 MOV 2(SP),SUBSTK(R3) ;INSERT THIS CALL
1382 021132 162763 000004 002306 SUB #4,SUBSTK(R3) ;ADJUST IT TO CALLING LOCATION
1383 021140 010337 002704 MOV R3,SSINDX ;STORE IT BACK
1384 021144 010046 MOV R0,-(SP)
1385 021146 010546 MOV R5,-(SP)
1386 021150 004737 020170 JSR PC,XRDHD ;DO READ HEADER
1387 021154 021204 65$
1388 021156 013703 002754 MOV HDWRD1,R3 ;GET HEADER WORD
1389 021162 012705 000007 MOV #7,R5 ;SET SHIFT COUNT
1390 021166 006203 4$: ASR R3 ;SHIFT TO RIGHT JUSTIFY
1391 021170 005305 DEC R5
1392 021172 001375 BNE 4$
1393 021174 042703 177000 BIC #177000,R3
1394 021200 010337 003006 MOV R3,CURCYL ;STORE AS CURRENT CYLINDER
1395 021204 162737 000002 002704 65$: SUB #2,SSINDX ;REMOVE ENTRY FROM SUBROUT STACK
1396 021212 012605 MOV (SP)+,R5 ;RESTORE REGISTERS
1397 021214 012600 MOV (SP)+,R0
1398 021216 012603 MOV (SP)+,R3
1399 021220 005737 002720 TST ERRSWI ;TEST IF ERROR RETURN
1400 021224 001403 BEQ 99$ ;YES - SKIP
1401 021226 063716 002720 ADD ERRSWI,(SP) ;ADD IN ERROR RETURN
1402 021232 000207 RTS PC
1403 021234 017616 000000 99$: MOV @ (SP),(SP) ;SET ERROR RETURN ADDRESS
1404 021240 000207 RTS PC
1405
1434 ; READ ALL HEADERS ROUTINE. 40 HEADERS ARE READ AND STORED
1435 ; IN Ibuff.
1436 021242 010346 RDALHD: MOV R3,-(SP) ;STORE REGISTERS
1437 021244 013703 002704 MOV SSINDX,R3 ;GET SUBROUTINE INDEX
1438 021250 005723 TST (R3)+ ;BUMP IT FOR NEXT ENTRY
1439 021252 016663 000002 002306 MOV 2(SP),SUBSTK(R3) ;INSERT THIS CALL
1440 021260 162763 000004 002306 SUB #4,SUBSTK(R3) ;ADJUST IT TO CALLING LOCATION
1441 021266 010337 002704 MOV R3,SSINDX ;STORE IT BACK
1442 021272 010046 MOV R0,-(SP)
1443 021274 010146 MOV R1,-(SP)
1444 021276 010446 MOV R4,-(SP)
1445 021300 012737 000002 002720 MOV #2,ERRSWI ;SET FOR NO ERROR RETURN
1446 021306 012701 000050 MOV #40.,R1 ;SET HEADER COUNT

```

```

1447 021312 052737 100000 002706      BIS      #HDR40,OPFLAG      ;SET 40 HDR OP FLAG
1448 021320 012703 003656              MOV      #IBUFF,R3        ;SET POINTER TO STORE HDRS
1449 021324 013704 002730              MOV      RLBA,R4          ;GET BASE ADDRESS
1450 021330 062704 000006              ADD      #RLMP,R4         ;MAKE IT POINT TO MP REG
1451 021334 012737 000010 002736      MOV      #10,L.CS        ;LOAD FOR READ HEADER, NO INTERRUPT
1452 021342 053737 002734 002736      BIS      RLDRV,L.CS      ;INSERT DRIVE NUMBER
1453 021350 042737 002000 002736      BIC      #BIT10,L.CS     ;CLEAR FOR DRIVE 4 - 7 SPEC'D
1454 021356 005037 002740              CLR      L.BA             ;CLEAR BA
1455 021362 005037 002742              CLR      L.DA             ;CLEAR DA
1456 021366 005737 003014              TST      DESHD           ;TEST IF HEAD 0
1457 021372 001403              BEQ      3$              ;YES - SKIP
1458 021374 052737 000020 002742      BIS      #HDSEL,L.DA     ;ELSE INSERT HEAD 0
1459 021402 013762 002742 000004 3$:    MOV      L.DA,RLDA(R2)    ;LOAD RLDA REG
1460 021410 013762 002740 000002      MOV      L.BA,RLBA(R2)   ;LOAD RLBA
1461 021416 032762 000200 000000      BIT      #CRDYMSK,RLCS(R2) ;TEST IF CONTROLLER READY
1462 021424 001003              BNE      6$              ;YES - SKIP
1463 021426 004737 017610              JSR      PC,RDYCHK       ;ELSE CHECK READY
1464 021432 021544              65$
1465 021434 013762 002736 000000 6$:    MOV      L.CS,RLCS(R2)   ;LOAD RLCS REG
1466 021442 012700 077777              MOV      #77777,R0       ;SET COUNT FOR WAIT
1467 021446 032762 000200 000000 7$:    BIT      #CRDYMSK,RLCS(R2) ;CHECK THAT OPERATION COMPLETED
1468 021454 001015              BNE      8$              ;YES - SKIP
1469 021456 005300              DEC      R0               ;DEC COUNT
1470 021460 001372              BNE      7$              ;SKIP IF NOT YET 0
1471 021462 004737 015466              JSR      PC,READRL       ;ELSE GET ALL REGISTERS
1472 021466 004737 015520              JSR      PC,WAITIN       ;ELSE WAIT FOR TIMEOUT
1473 021472 012603              MOV      (SP)+,R3        ;GET RESULT MESSAGE POINTER
1474 021474              ERRHRD  10025,,,ERR1
          021474 104443      TRAP    T$ERCODE
          021476 023451      .WORD  10025
          021500 012072      .WORD  ERR1
1475 021502 005037 002720              CLR      ERRSWI          ;CLEAR FOR ERROR RETURN
1476 021506 000416              BR       65$
1477 021510 005737 002746              8$:    TST      T.CS            ;TEST FOR ANY ERRORS
1478 021514 100006              BPL     12$              ;NO - SKIP
1479 021516              ERRHRD  10026,,,ERR6
          021516 104443      TRAP    T$ERCODE
          021520 023452      .WORD  10026
          021522 012374      .WORD  ERR6
1480 021524 005037 002720              CLR      ERRSWI          ;CLEAR FOR ERROR RETURN
1481 021530 000405              BR       65$
1482 021532 011423              12$:   MOV      (R4),(R3)+      ;STORE HEADER WORDS
1483 021534 011423              MOV      (R4),(R3)+
1484 021536 011423              MOV      (R4),(R3)+
1485 021540 005301              DEC      R1               ;DEC HEADER COUNT
1486 021542 001334              BNE      6$
1487 021544 162737 000002 002704 65$:   SUB      #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
1488 021552 012604              MOV      (SP)+,R4        ;RESTORE REGISTERS
1489 021554 012601              MOV      (SP)+,R1
1490 021556 012600              MOV      (SP)+,R0
1491 021560 012603              MOV      (SP)+,R3
1492 021562 005737 002720              TST      ERRSWI          ;TEST IF ERROR RETURN
1493 021566 001403              BEQ     99$              ;YES - SKIP
1494 021570 063716 002720              ADD      ERRSWI,(SP)     ;ADD IN ERROR RETURN
1495 021574 000207              RTS      PC
1496 021576 017616 000000 99$:    MOV      @ (SP),(SP)     ;SET ERROR RETURN ADDRESS
1497 021602 000207              RTS      PC

```

```

1498
1499
1727
1728
1729
1730 021604 010446
1731 021606 005737 002704
1732 021612 001433
1733 021614 012704 000002
1734 021620
      021620 012746 007662
      021624 012746 011361
      021630 012746 000002
      021634 010600
      021636 104014
      021640 062706 000006
1735 021644
      021644 016446 002306
      021650 012746 011534
      021654 012746 000002
      021660 010600
      021662 104014
      021664 062706 000006
1736 021670 062704 000002
1737 021674 020437 002704
1738 021700 003761
1739 021702
      021702 012746 006273
      021706 013746 002714
      021712 012746 011164
      021716 012746 000003
      021722 010600
      021724 104014
      021726 062706 000010
1740 021732 042737 030000 002706
1741 021740 013701 002736
1742 021744 042701 177741
1743 021750 022701 000006
1744 021754 001003
1745 021756 052737 010000 002706
1746 021764 022701 000012
1747 021770 001003
1748 021772 052737 020000 002706
1749 022000 022701 000014
1750 022004 001003
1751 022006 052737 020000 002706
1752 022014
      022014 016146 002126
      022020 012746 005306
      022024 012746 011142
      022030 012746 000003
      022034 010600
      022036 104014
      022040 062706 000010
1753 022044 020127 000004
1754 022050 001007
1755 022052 032737 000010 002742

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

```

```

1756 022060 001403      BEQ      4$      ;NO - SKIP
1757 022062 012701 000016  MOV      #16,R1  ;SET TO PRINT WITH RESET
1758 022066 000436      BR       9$
1759 022070 032737 007777 002706 4$:  BIT      #COMPOP,OPFLAG ;TEST IF ANY OTHER OPERATION
1760 022076 001424      BEQ      8$      ;NO - SKIP
1761 022100 013704 002706  MOV      OPFLAG,R4 ;SET UP TO DETERMINE WHICH ONE
1762 022104 012701 000020  MOV      #20,R1  ;PRESET THE POINTER
1763 022110 032704 000001      5$:  BIT      #BIT00,R4 ;CHECK THE BIT
1764 022114 001003      BNE      6$      ;IF SET - SKIP
1765 022116 005721      TST      (R1)+  ;BUMP POINTER
1766 022120 006204      ASR      R4
1767 022122 000772      BR       5$
1768 022124      6$:  PRINTB   #FMT2,OPMSG$(R1)
      022124 016146 002126  MOV      OPMSG$(R1),-(SP)
      022130 012746 011156  MOV      #FMT2,-(SP)
      022134 012746 000002  MOV      #2,-(SP)
      022140 010600  MOV      SP,R0
      022142 104014  EMT      C$PNTB
      022144 062706 000006  ADD      #6,SP
1769 022150 032737 100000 002706 8$:  BIT      #HDR40,OPFLAG ;TEST IF 40 HEADER OPERATION
1770 022156 001415      BEQ      10$     ;NO - SKIP
1771 022160 012701 000050  MOV      #50,R1  ;ELSE PRINT IT
1772 022164      9$:  PRINTB   #FMT2,OPMSG$(R1)
      022164 016146 002126  MOV      OPMSG$(R1),-(SP)
      022170 012746 011156  MOV      #FMT2,-(SP)
      022174 012746 000002  MOV      #2,-(SP)
      022200 010600  MOV      SP,R0
      022202 104014  EMT      C$PNTB
      022204 062706 000006  ADD      #6,SP
1773 022210 000434      BR       15$     ;SKIP
1774 022212 032737 010000 002706 10$: BIT      #SEEKOP,OPFLAG ;TEST IF SEEK
1775 022220 001430      BEQ      15$     ;NO - SKIP
1776 022222      PRINTB   #FMT13,#FRMWD,OLDCYL,#DIFWD,DESDIF,#SGNWD,DESSGN,#HDWD,DESHD
      022222 013746 003014  MOV      DESHD,-(SP)
      022226 012746 007623  MOV      #HDWD,-(SP)
      022232 013746 003012  MOV      DESSGN,-(SP)
      022236 012746 007616  MOV      #SGNWD,-(SP)
      022242 013746 003010  MOV      DESDIF,-(SP)
      022246 012746 007610  MOV      #DIFWD,-(SP)
      022252 013746 003002  MOV      OLDCYL,-(SP)
      022256 012746 007641  MOV      #FRMWD,-(SP)
      022262 012746 011402  MOV      #FMT13,-(SP)
      022266 012746 000011  MOV      #11,-(SP)
      022272 010600  MOV      SP,R0
      022274 104014  EMT      C$PNTB
      022276 062706 000024  ADD      #24,SP
1777 022302 032737 020000 002706 15$: BIT      #RORWOP,OPFLAG ;TEST IF READ OR WRITE SET
1778 022310 001424      BEQ      17$     ;NO - SKIP
1779 022312      PRINTB   #FMT22,#CYLWD,CURCYL,#HDWD,DESHD,#SECWD,DESSEC
      022312 013746 003016  MOV      DESSEC,-(SP)
      022316 012746 007627  MOV      #SECWD,-(SP)
      022322 013746 003014  MOV      DESHD,-(SP)
      022326 012746 007623  MOV      #HDWD,-(SP)
      022332 013746 003006  MOV      CURCYL,-(SP)
      022336 012746 007634  MOV      #CYLWD,-(SP)
      022342 012746 011731  MOV      #FMT22,-(SP)
      022346 012746 000007  MOV      #7,-(SP)

```

```

022352 010600      MOV      SP,R0
022354 104014      EMT      C$PNTB
022356 062706      ADD      #20,SP
1780 022362 004737 000G20      JSR      PC,CLRPARM      ;CLEAR PARAM TABLE
1781 022366 012604 023034      MOV      (SP)+,R4      ;RESTORE R4
1782 022370 000207      RTS      PC
1783
1784      ;
1785      ;
1786 022372 010146      RPTRES:  REPORT REASON ROUTINE
1787 022374 010346      PRINTS  REASON PORTION FOR ALL ERROR REPORTS.
1788 022376 010446      MOV      R1,-(SP)      ;STORE R1
1789 022400 012701 002764      MOV      R3,-(SP)      ;STORE R3
1790 022404 012103      MOV      R4,-(SP)      ;STORE R4
1791 022406      MOV      #RE3PARM,R1    ;GET START OF PARAM
      MOV      (R1)+,R3    ;GET NUMBER OF PARAM
      PRINTB #FMT1.1,#MRSLT,(R1) ;PRINT NAME
      MOV      (R1),-(SP)
      MOV      #MRSLT,-(SP)
      MOV      #FMT1.1,-(SP)
      MOV      #3,-(SP)
      MOV      SP,R0
      EMT      C$PNTB
      ADD      #10,SP
1792 022434 021127 010466      CMP      (R1),#MNRST    ;TEST IF MESSAGE IS NO DRV STATUS
1793 022440 001453      BEQ      6$            ;YES - SKIP REST OF REPORT
1794 022442 012704 011366      MOV      #FMT11,R4     ;PRISET FOR FORMAT 11
1795 022446 022127 010461      CMP      (R1)+,#MCYLOC ;CHECK IF REPORTING CYLINDER LOC
1796 022452 001002      BNE      3$            ;NO - SKIP
1797 022454 012704 011374      MOV      #FMT12,R4     ;ELSE CHANGE TO FORMAT 12
1798 022460 005303      3$:      DEC      R3            ;DEC PARAM COUNT
1799 022462 001442      BEQ      6$            ;IF 0 - EXIT
1800 022464      PRINTB  R4,#RESE3,(R1)+ ;REPORT IS VALUE
      MOV      (R1)+,-(SP)
      MOV      #RESE3,-(SP)
      MOV      R4,-(SP)
      MOV      #3,-(SP)
      MOV      SP,R0
      EMT      C$PNTB
      ADD      #10,SP
1801 022510      PRINTB  R4,#RESE4,(R1)+ ;REPORT SB VALUE
      MOV      (R1)+,-(SP)
      MOV      #RESE4,-(SP)
      MOV      R4,-(SP)
      MOV      #3,-(SP)
      MOV      SP,R0
      EMT      C$PNTB
      ADD      #10,SP
1802 022534 162703 000010      SUB      #2,R3          ;DEC PARAM COUNT
1803 022540 001413      BEQ      6$            ;IF 0 - EXIT
1804 022542      PRINTB  #FMT1,#RESE5,(R1)+ ;REPORT CONDITION
      MOV      (R1)+,-(SP)
      MOV      #RESE5,-(SP)
      MOV      #FMT1,-(SP)
      MOV      #3,-(SP)
      MOV      SP,R0
      EMT      C$PNTB
      ADD      #10,SP
1805 022570 012604      6$:      MOV      (SP)+,R4      ;RESTORE REGS

```

```

1806 022572 012603      MOV      (SP)+,R3
1807 022574 012601      MOV      (SP)+,R1
1808 022576 000207      RTS      PC              ;RETURN
1809
1810      :      REPORT PHYSICAL ADDRESS OF DEVICE UNDER TEST
1811      :      AND ALL REGISTER CONTENTS.
1812 022600      RPTREM: PRINTB #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
      022600      005046      CLR      -(SP)
      022602      153716      002735      BISB     RLDRV+1,(SP)
      022606      012746      005743      MOV      #DRVNAM,-(SP)
      022612      013746      002730      MOV      RLBAS,-(SP)
      022616      012746      005732      MOV      #BASADD,-(SP)
      022622      012746      011175      MOV      #FMT5,-(SP)
      022626      012746      000005      MOV      #5,-(SP)
      022632      010600      MOV      SP,R0
      022634      104014      EMT      C$PNTB
      022636      062706      000014      ADD      #14,SP
1813      :      REPORT RL11 REGISTERS
1814 022642      PRINTB #FMT6,#CSNAM,#DANAM,#BANAM,#MPNAM,#CYLWD,#HDWD
      022642      012746      007623      MOV      #HDWD,-(SP)
      022646      012746      007634      MOV      #CYLWD,-(SP)
      022652      012746      006046      MOV      #MPNAM,-(SP)
      022656      012746      006034      MOV      #BANAM,-(SP)
      022662      012746      006041      MOV      #DANAM,-(SP)
      022666      012746      006027      MOV      #CSNAM,-(SP)
      022672      012746      011215      MOV      #FMT6,-(SP)
      022676      012746      000007      MOV      #7,-(SP)
      022702      010600      MOV      SP,R0
      022704      104014      EMT      C$PNTB
      022706      062706      000020      ADD      #20,SP
1815 022712      PRINTB #FMT8,#LAB1,L.CS,L.DA,L.BA,L.MP
      022712      013746      002744      MOV      L.MP,-(SP)
      022716      013746      002740      MOV      L.BA,-(SP)
      022722      013746      002742      MOV      L.DA,-(SP)
      022726      013746      002736      MOV      L.CS,-(SP)
      022732      012746      006053      MOV      #LAB1,-(SP)
      022736      012746      011327      MOV      #FMT8,-(SP)
      022742      012746      000006      MOV      #6,-(SP)
      022746      010600      MOV      SP,R0
      022750      104014      EMT      C$PNTB
      022752      062706      000016      ADD      #16,SP
1816 022756      PRINTB #FMT7,#LAB2,T.CS,T.DA,T.BA,T.MP,CURCYL,DESHD
      022756      013746      003014      MOV      DESHD,-(SP)
      022762      013746      003006      MOV      CURCYL,-(SP)
      022766      013746      002754      MOV      T.MP,-(SP)
      022772      013746      002750      MOV      T.BA,-(SP)
      022776      013746      002752      MOV      T.DA,-(SP)
      023002      013746      002746      MOV      T.CS,-(SP)
      023006      012746      006066      MOV      #LAB2,-(SP)
      023012      012746      011257      MOV      #FMT7,-(SP)
      023016      012746      000010      MOV      #10,-(SP)
      023022      010600      MOV      SP,R0
      023024      104014      EMT      C$PNTB
      023026      062706      000022      ADD      #22,SP
1817 023032 000207      RTS      PC
1818
1819      :      CLEAR PARAMETER BLOCK FOR REPORTING

```

```
1820 023034 010546
1821 023036 012701 002764
1822 023042 012705 000005
1823 023046 005021
1824 023050 005305
1825 023052 001375
1826 023054 012701 002764
1827 023060 012605
1828 023062 000207
1829
1830 023064

CLRPARM:      MOV      R5,-(SP)      ;STORE R5
               MOV      #RESPARM,R1 ;GET ADDRESS OF BLOCK
               MOV      #5,R5      ;SET COUNT
2$:            CLR      (R1)+      ;CLEAR WORD
               DEC      R5          ;DEC COUNT
               BNE     2$          ;LOOP UNTIL 0
               MOV      #RESPARM,R1 ;RESET POINTER
               MOV      (SP)+,R5    ;RESTORE R5
               RTS      PC

ENDMOD
```


1

.TITLE CZRLIAO RL01/2 DR TST 1

```

1 023064      BGNMOD  HRDWTST
2              .SBTTL  *TEST 1      BASIC INTERFACE (PART 1)
3
4 023064      BGNSTST              ;TEST01
5 023064      005737  003252      TST      PASNUM          ;CHECK IF FIRST PASS
6 023070      001120      BNE      65$          ;EXIT IF NO
7 023072      005737  013712      TST      MISWIW         ;CHECK IF MANUAL INTERVENTION
8 023076      100115      BPL      65$          ;NO - EXIT TEST
9 023100      012737  006301  002714  MOV      #MISTST,ERHEAD ;LOAD ERR HEADER
10 023106      005046      2$: PRINTF   #FMTOP1,#OPR1,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
    023106      153716  002735      CLR      -(SP)
    023110      012746  005743      BISB    RLDRV+1,(SP)
    023114      013746  002730      MOV     #DRVNAM,-(SP)
    023120      012746  005732      MOV     RLBAS,-(SP)
    023124      012746  007544      MOV     #BASADD,-(SP)
    023130      012746  007122      MOV     #OPR1A,-(SP)
    023134      012746  011050      MOV     #OPR1,-(SP)
    023140      012746  000007      MOV     #FMTOP1,-(SP)
    023144      010600      MOV     #7,-(SP)
    023150      104017      MOV     SP,R0
    023152      062706  000020      EMT     C$PNTF
    023154      005037  004256      ADD     #20,SP
11 023160      005037  004256      CLR     OBUFF          ;CLEAR FOR RESPONSE
12 023164      104043      GMANIL  OPRO02,OBUFF,1,NO
    023166      000404      EMT     C$GMAN
    023170      000120      BR      10000$
    023172      007052      .WORD  OBUFF
    023174      000001      .WORD  T$CODE
    023176      000001      .WORD  OPRO02
    023200      10000$: .WORD  1
13 023200      005737  004256      TST     OBUFF          ;TEST RESPONSE YES
14 023204      001740      BEQ     2$            ;YES - SKIP
15 023206      004737  015652      1$: JSR     PC,TSTINT   ;INITIALIZE TEST
16 023212      004737  015704      JSR     PC,GSTATC     ;GO GET STATUS (NO RESET)
17 023216      032737  000040  002754  65$     BIT      #COSTAT,T.MP   ;CHECK IF COVER OPEN SET
18 023220      001005      BNE     7$            ;YES - SKIP
19 023226      012703  010171      MOV     #MCOSTA,R3    ;SET NAME POINTER
20 023230      012703  010171      ERRHRD  101.,,ERR3
    023234      104443      TRAP   T$ERCODE
    023236      000145      .WORD  101
    023240      012206      .WORD  ERR3
22 023242      032737  000010  002754  7$: BIT      #BHSTAT,T.MP   ;TEST IF BRUSHES HOME
23 023250      001005      BNE     9$            ;YES - SKIP
24 023252      012703  010204      MOV     #MBHSTA,R3    ;SET POINTER FOR BRUSH HOME ERROR
25 023256      012703  010204      ERRHRD  102.,,ERR3
    023256      104443      TRAP   T$ERCODE
    023260      000146      .WORD  102
    023262      012206      .WORD  ERR3
26 023264      032737  020000  002754  9$: BIT      #WLSTAT,T.MP   ;TEST IF WRITE LOCK SET
27 023272      001005      BNE     11$           ;YES - SKIP
28 023274      012703  010217      MOV     #MWLSTA,R3    ;SET NAME POINTER
29 023300      012703  010217      ERRHRD  103.,,ERR3
    023300      104443      TRAP   T$ERCODE
    023302      000147      .WORD  103

```

30	023304	012206			.WORD	ERR3	
	023306	005737	002762	11\$:	TST	T,STAT	:TEST IF STATE ZERO
31	023312	001404			BEQ	15\$:YES - SKIP
32	023314	005003			CLR	R3	:SET STATE EXPECTED
33	023316				ERRHRD	104,,,ERR7	
	023316	104443			TRAP	T\$ERCODE	
	023320	000150			.WORD	104	
	023322	013252			.WORD	ERR7	
34	023324	004737	015670	15\$:	JSR	PC,GSTATR	:DO DRIVE RESET
35	023330	023332			65\$		
36	023332			65\$:			
37	023332			ENDTST			
	023332			L10020:			
38	023332	104001			EMT	C\$ETST	

```
1          .SBTTL *TEST 2          BASIC INTERFACE (PART 2)
2
3 023334    BGNTST          ;TEST 2
023334
4 023334    005737 003252    TST      PASNUM          ;TEST IF PASS 0
5 023340    001075          BNE      65$          ;NO - SKIP
6 023342    005737 013712    TST      MISWIW        ;TEST IF MANUAL INTERVENTION
7 023346    100072          BPL      65$          ;NO - SKIP
8 023350    012737 006301 002714  MOV     #MISTST,ERHEAD ;SET ERROR HEADER
9 023356    005046          2$: PRINTF  #FMTOP1,#OPR2,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1> ;REQUEST CLOSE
023356    153716 002735    CLR     -(SP)
023360    012746 005743    BISB   RLDRV+1,(SP)
023364    013746 002730    MOV     #DRVNAM,-(SP)
023370    012746 005732    MOV     RLBAS,-(SP)
023374    012746 007544    MOV     #BASADD,-(SP)
023400    012746 007200    MOV     #OPR1A,-(SP)
023404    012746 011050    MOV     #OPR2,-(SP)
023410    012746 000007    MOV     #FMTOP1,-(SP)
023414    010600          MOV     #7,-(SP)
023420    104017          MOV     SP,R0
023422    062706 000020    EMT    C$PNTF
023424          ADD     #20,SP
10
11 023430    005037 004256    CLR     OBUFF          ;COVER AND RESET WRITE LOCK
12 023434    104043          GMANIL OPRO02,OBUFF,1,NO ;CLEAR FOR RESPONSE
023434    000404          EMT    C$GMAN
023436    004256          BR     10000$
023440    000120          .WORD OBUFF
023442    007052          .WORD T$CODE
023444    000001          .WORD OPRO02
023446    10000$: .WORD 1
13 023450    005737 004256    TST     OBUFF          ;TEST IF RESPONSE YES
14 023454    001740          BEQ    2$             ;NO - SKIP
15 023456    004737 015652    1$: JSR   PC,TSTINT    ;INITIALIZE TEST
16 023462    004737 015670    JSR   PC,GSTATR      ;GET STATUS WITH RESET
17 023466    023534          65$
18 023470    032737 000040 002754  BIT    #COSTAT,T.MP   ;TEST IF COVER OPEN RESET
19 023476    001405          BEQ    9$             ;YES - SKIP
20 023500    012703 010171    MOV     #MCOSTA,R3    ;SET NAME MESSAGE POINTER
21 023504    104443          ERRHRD 201,,,ERR2
023504    000311          TRAP  T$ERCODE
023510    012140          .WORD 201
22 023512    032737 020000 002754  9$: BIT    #WLSTAT,T.MP ;TEST IF WRITE LOCK RESET
23 023520    001405          BEQ    65$          ;YES - SKIP
24 023522    012703 010217    MOV     #MWLSTA,R3   ;SET NAME MESSAGE POINTER
25 023526    104443          ERRHRD 202,,,ERR2
023526    000312          TRAP  T$ERCODE
023530    012140          .WORD 202
023532    104001          .WORD ERR2
26 023534          65$:
27 023534          FNDTST
023534          L10021:
023534    104001          EMT    C$ETST
28
```

	.SBTTL	*TEST 3	HEAD LOADING
1			
2	023536		BGNTST ;TEST03
3	023536	005737 003252	TST PASNUM ;TEST IF PASS 0
4	023542	001003	BNE 4\$;NO SKIP
5	023544	005737 013712	TST MISWIW ;TEST IF MANUAL INTERVENTION
6	023550	100402	BMI 5\$;YES - SKIP
7	023552		4\$: EXIT TST
	023552	104032	EMT C\$EXIT
	023554	001214	.WORD L10022-
8	023556	004737 015652	5\$: JSR PC,TSTINT ;INITIALIZE TEST
9	023562	004737 015670	JSR PC,GSTATR ;GET STATUS
10	023566	024770	T365\$
11	023570	005737 002762	TST T.STAT ;TEST IF STATE ZERO
12	023574	001440	BEQ 2\$;YES - SKIP
13	023576		1\$: PRINTF #FMTOP1,#OPR5,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1> ;REQUEST DRIVE BE UNL
	023576	005046	CLR -(SP)
	023600	153716 002735	BISB RLDRV+1,(SP)
	023604	012746 005743	MOV #DRVNAM,-(SP)
	023610	013746 002730	MOV RLBAS,-(SP)
	023614	012746 005732	MOV #BASADD,-(SP)
	023620	012746 007544	MOV #OPR1A,-(SP)
	023624	012746 007246	MOV #OPR5,-(SP)
	023630	012746 011050	MOV #FMTOP1,-(SP)
	023634	012746 000007	MOV #7,-(SP)
	023640	010600	MOV SP,R0
	023642	104017	EMT C\$PNTF
	023644	062706 000020	ADD #20,SP
14	023650	005037 004256	CLR OBUFF ;CLEAR FOR RESPONSE
15	023654		GMANIL OPR002,OBUFF,1,NO
	023654	104043	EMT C\$GMAN
	023656	000404	BR 10000\$
	023660	004256	.WORD OBUFF
	023662	000120	.WORD T\$CODE
	023664	007052	.WORD OPR002
	023666	000001	.WORD 1
	023670		10000\$:
16	023670	005737 004256	TST OBUFF ;TEST IF RESPONSE YES
17	023674	001740	BEQ 1\$;NO - SKIP
18	023676		2\$: PRINTF #FMTOP1,#OPR3,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
	023676	005046	CLR -(SP)
	023700	153716 002735	BISB RLDRV+1,(SP)
	023704	012746 005743	MOV #DRVNAM,-(SP)
	023710	013746 002730	MOV RLBAS,-(SP)
	023714	012746 005732	MOV #BASADD,-(SP)
	023720	012746 007544	MOV #OPR1A,-(SP)
	023724	012746 007232	MOV #OPR3,-(SP)
	023730	012746 011050	MOV #FMTOP1,-(SP)
	023734	012746 000007	MOV #7,-(SP)
	023740	010600	MOV SP,R0
	023742	104017	EMT C\$PNTF
	023744	062706 000020	ADD #20,SP
19	023750	012737 000004 002706	MOV #CYLUP,OPFLAG ;SET CYCLE UP FLAG
20	023756	012703 000001	MOV #1,R3 ;SET EXPECTED STATE VALUE
21	023762	012737 006324 002714	MCV #NSTACHG,ERHEAD ;SET ERROR HEADER
22	023770	012701 000454	MOV #300,R1 ;SET WAIT COUNT FOR 30 SECONDS
23	023774	004737 015704	3\$: JSR PC,GSTATC ;GET STATUS

24	024000	024770			T365\$		
25	024002	005737	002762		TST	T,STAT	;TEST IF STATE IS STILL 0
26	024006	001022			BNE	10\$;NO - SKIP
27	024010	005301			DEC	R1	;DEC WAIT COUNT
28	024012	001404			BEQ	6\$;EXIT IF WAIT DONE
29	024014				WAITMS	#1	
	024014	012700	000001		MOV	#1,R0	
	024020	104026			EMT	C\$WTM	
30	024022	000764			BR	3\$	
31	024024	005037	004256	6\$:	CLR	OBUFF	;CLEAR FOR RESPONSE
32	024030				GMANIL	OPRO03,OBUFF,1,NO	
	024030	104043			EMT	C\$GMAN	
	024032	000404			BR	10001\$	
	024034	004256			.WORD	OBUFF	
	024036	000120			.WORD	T\$CODE	
	024040	007077			.WORD	OPRO03	
	024042	000001			.WORD	1	
	024044			10001\$:			
33	024044	005737	004256		TST	OBUFF	;TEST IF RESPONSE YES
34	024050	001004			BNE	11\$;YES - REPORT
35	024052	000651			BR	1\$	
36	024054	020337	002762	10\$:	CMP	R3,T,STAT	;CHECK IF NOW STATE 1
37	024060	001405			BEQ	13\$;YES - SKIP
38	024062			11\$:	ERRHRD	301,,,ERR7	
	024062	104443			TRAP	T\$ERCODE	
	024064	000455			.WORD	301	
	024066	013252			.WORD	ERR7	
39	024070				EXIT	TST	
	024070	104032			EMT	C\$EXIT	
	024072	000676			.WORD	L10022-	
40	024074	012701	000454	13\$:	MOV	#300.,R1	;SET WAIT FOR 30 SECONDS
41	024100	012703	000002		MOV	#2,R3	;SET EXPECTED STATE VALUE
42	024104	004737	015704	14\$:	JSR	PC,GSTATC	;GET STATUS
43	024110	024770			T365\$		
44	024112	020337	002762		CMP	R3,T,STAT	;CHECK IF STATE 2
45	024116	001435			BEQ	20\$;YES - SKIP
46	024120	101005			BHI	17\$;CHECK IF NO CHANGE - YES - SKIP
47	024122				ERRHRD	302,,,ERR7	
	024122	104443			TRAP	T\$ERCODE	
	024124	000456			.WORD	302	
	024126	013252			.WORD	ERR7	
48	024130				EXIT	TST	
	024130	104032			EMT	C\$EXIT	
	024132	000636			.WORD	L10022-	
49	024134	005301		17\$:	DEC	R1	;DEC WAIT COUNT
50	024136	001404			BEQ	18\$;SKIP IF 0
51	024140				WAITMS	#1	
	024140	012700	000001		MOV	#1,R0	
	024144	104026			EMT	C\$WTM	
52	024146	000756			BR	14\$	
53	024150			18\$:	ERRHRD	303,,,ERR7	
	024150	104443			TRAP	T\$ERCODE	
	024152	000457			.WORD	303	
	024154	013252			.WORD	ERR7	
54	024156	032737	004000 002754		BIT	#SPDSTAT,T,MP	;TEST IF SPINDLE TIMEOUT
55	024164	001010			BNE	19\$;YES - SKIP
56	024166	012737	006336 002714		MOV	#SPDERR,ERHEAD	;SET ERROR HEADER

57	024174	012703	010271		MOV	#MSPERR,R3		:SET NAME MESSAGE POINTER
58	024200				ERRHRD	304,,,ERR3		
	024200	104443			TRAP	T\$ERCODE		
	024202	000460			.WORD	304		
	024204	012206			.WORD	ERR3		
59	024206			19\$:	EXIT	TST		
	024206	104032			EMT	C\$EXIT		
	024210	000560			.WORD	L10022-		
60	024212	012737	006301	002714	20\$:	MOV	#MISTST,ERHEAD	:SET ERROR HEADER
61	024220	012704	010731		MOV	#STATE2,R4		:SET CONDITION MESSAGE POINTER
62	024224	012703	010204		MOV	#MBHSTA,R3		:SET NAME MESSAGE POINTER
63	024230	032737	000010	002754	BIT	#BHSTAT,T.MP		:TEST IF BRUSH HOME STILL SET
64	024236	001005			BNE	22\$:YES - SKIP
65	024240				ERRHRD	305,,,ERR5		
	024240	104443			TRAP	T\$ERCODE		
	024242	000461			.WORD	305		
	024244	012324			.WORD	ERR5		
66	024246				EXIT	TST		
	024246	104032			EMT	C\$EXIT		
	024250	000520			.WORD	L10022-		
67	024252	012701	000062		22\$:	MOV	#50.,R1	:SET WAIT COUNT FOR 5 SECONDS
68	024256	004737	015704		23\$:	JSR	PC,G\$STATC	:GET STATUS
69	024262	024770				T365\$		
70	024264	032737	000010	002754	BIT	#BHSTAT,T.MP		:TEST IF BRUSH HOME RESET
71	024272	001413			BEQ	27\$:YES - SKIP
72	024274	005301			DEC	R1		:DEC WAIT COUNT
73	024276	001404			BEQ	26\$:SKIP IF ZERO
74	024300				WAITMS	#1		
	024300	012700	000001		MOV	#1,R0		
	024304	104026			EMT	C\$WTM		
75	024306	000763			BR	23\$:LOOP
76	024310			26\$:	ERRHRD	306,,,ERR4		
	024310	104443			TRAP	T\$ERCODE		
	024312	000462			.WORD	306		
	024314	012254			.WORD	ERR4		
77	024316				EXIT	TST		
	024316	104032			EMT	C\$EXIT		
	024320	000450			.WORD	L10022-		
78	024322	012701	000454		27\$:	MOV	#300.,R1	:SET WAIT COUNT 30 SECONDS
79	024326	004737	015704		28\$:	JSR	PC,G\$STATC	:GET STATUS
80	024332	024770				T365\$		
81	024334	032737	000010	002754	BIT	#BHSTAT,T.MP		:TEST IF BRUSH HOME SET AGAIN
82	024342	001013			BNE	32\$:YES - SKIP
83	024344	005301			DEC	R1		:ELSE DEC WAIT COUNT
84	024346	001404			BEQ	30\$:SKIP IF 0
85	024350				WAITMS	#1		
	024350	012700	000001		MOV	#1,R0		
	024354	104026			EMT	C\$WTM		
86	024356	000763			BR	28\$		
87	024360			30\$:	ERRHRD	307,,,ERR5		
	024360	104443			TRAP	T\$ERCODE		
	024362	000463			.WORD	307		
	024364	012324			.WORD	ERR5		
88	024366				EXIT	TST		
	024366	104032			EMT	C\$EXIT		
	024370	000400			.WORD	L10022-		
89	024372	012737	006324	002714	32\$:	MOV	#NSTACHG,ERHEAD	:SET ERROR HEADER

90	024400	012703	000003			MOV	#3,R3		;SET EXPECTED STATE VALUE
91	024404	004737	015704			JSR	PC,GSTATC		;GET STATUS
92	024410	024770				T365\$			
93	024412	020337	002762			CMP	R3,T.STAT		;CHECK IF STATE 3
94	024416	001405				BEQ	36\$;YES - SKIP
95	024420					ERRHRD	308,,,FRR7		
	024420	104443				TRAP	T\$ERCODE		
	024422	000464				.WORD	308		
	024424	013252				.WORD	ERR7		
96	024426					EXIT	TST		
	024426	104032				EMT	C\$EXIT		
	024430	000340				.WORD	L10022-		
97	024432	012737	006301	002714	36\$:	MOV	#MISTST,ERHEAD		;SET ERROR HEADER
98	024440	012704	010741			MCV	#STATE3,R4		;SET CONDITION MESSAGE POINTER
99	024444	012703	010230			MOV	#MHOSTA,R3		;SET NAME MESSAGE POINTER
100	024450	004737	015704			JSR	PC,GSTATC		;GET STATUS
101	024454	024770				T365\$			
102	024456	032737	00002C	002754		BIT	#HOSTAT,T.MP		;TEST IF HEADS OUT SET
103	024464	001005				BNE	38\$;YES - SKIP
104	024466					ERRHRD	309,,,ERR5		
	024466	104443				TRAP	T\$ERCODE		
	024470	000465				.WORD	309		
	024472	012324				.WORD	ERR5		
105	024474					EXIT	TST		
	024474	104032				EMT	C\$EXIT		
	024476	000272				.WORD	L10022-		
106	024500	032737	001000	002754	38\$:	BIT	#VCSTAT,T.MP		;TEST IF VOLUME CHECK SET
107	024506	001007				BNE	40\$		
108	024510	012703	010160			MOV	#MVOLCK,R3		;SET NAME MESSAGE POINTER
109	024514					ERRHRD	310,,,ERR5		
	024514	104443				TRAP	T\$ERCODE		
	024516	000466				.WORD	310		
	024520	012324				.WORD	ERR5		
110	024522					EXIT	TST		
	024522	104032				EMT	C\$EXIT		
	024524	000244				.WORD	L10022-		
111	024526	032737	040000	002746	40\$:	BIT	#DRVERR,T.CS		;TEST IF DRIVE ERROR SET
112	024534	001007				BNE	42\$;YES - SKIP
113	024536	012703	010135			MOV	#MDRERR,R3		;SET NAME MESSAGE POINTER
114	024542					ERRHRD	311,,,ERR5		
	024542	104443				TRAP	T\$ERCODE		
	024544	000467				.WORD	311		
	024546	012324				.WORD	ERR5		
115	024550					EXIT	TST		
	024550	104032				EMT	C\$EXIT		
	024552	000216				.WORD	L10022-		
116	024554	012701	005670		42\$:	MOV	#3000.,R1		;SET WAIT COUNT FOR 300 MS
117	024560	012737	006324	002714		MOV	#NSTACHG,ERHEAD		;SET ERROR HEADER
118	024566	012703	000004			MOV	#4,R3		;SET EXPECTED STATE VALUE
119	024572	004737	015704		43\$:	JSR	PC,GSTATC		;GET STATUS
120	024576	024770				T365\$			
121	024600	020337	002762			CMP	R3,T.STAT		;CHECK IF STATE 4
122	024604	001413				BEQ	49\$;YES - SKIP
123	024606	005301				DEC	R1		;DEC WAIT COUNT
124	024610	001404				BEQ	47\$;SKIP IF 0
125	024612					WAITUS	#1		
	024612	012700	000001			MOV	#1,R0		

126	024616	104027		EMT	C\$WTU	
	024620	000764		BR	43\$	
127	024622		47\$:	ERRHRD	312,,ERR7	
	024622	104443		TRAP	T\$ERCODE	
	024624	000470		.WORD	312	
	024626	013252		.WORD	ERR7	
128	024630			EXIT	TST	
	024630	104032		EMT	C\$EXIT	
	024632	000136		.WORD	L10022-	
129	024634	012701	000454	49\$:	MOV	#300,,R1 ;SET WAIT COUNT FOR 30 MS
130	024640	012703	000005		MOV	#5,R3 ;SET EXPECTED STATE VALUE
131	024644	004737	015704	50\$:	JSR	PC,G\$STATC ;GET STATUS
132	024650	024770		T365\$		
133	024652	020337	002762	CMP	R3,T\$STAT ;CHECK IF STATE 5	
134	024656	001413		BEQ	55\$;YES - SKIP	
135	024660	005301		DEC	R1 ;DEC WAIT COUNT	
136	024662	001404		BEQ	51\$;ELSE SKIP	
137	024664			WAITUS	#1	
	024664	012700	000001	MOV	#1,R0	
	024670	104027		EMT	C\$WTU	
138	024672	000764		BR	50\$	
139	024674		51\$:	ERRHRD	313,,ERR7	
	024674	104443		TRAP	T\$ERCODE	
	024676	000471		.WORD	313	
	024700	013252		.WORD	ERR7	
140	024702			EXIT	TST	
	024702	104032		EMT	C\$EXIT	
	024704	000064		.WORD	L10022-	
141	024706	012701	000120	55\$:	MOV	#80,,R1 ;SET WAIT FOR 8 MS
142	024712	004737	015704	56\$:	JSR	PC,G\$STATC ;GET STATUS
143	024716	024770		T365\$		
144	024720	032737	000001 002746	BIT	#DRDYMSK,T\$CS ;CHECK IF DRIVE READY	
145	024726	001020		BNE	62\$;YES - SKIP	
146	024730	005301		DEC	R1 ;DEC COUNT	
147	024732	001404		BEQ	60\$;SKIP IF 0	
148	024734			WAITUS	#1	
	024734	012700	000071	MOV	#1,R0	
	024740	104027		EMT	C\$WTU	
149	024742	000763		BR	56\$	
150	024744	012737	006301 002714	60\$:	MOV	#MISTST,ERHEAD ;SET ERROR HEADER
151	024752	012704	010751		MOV	#STAT5,R4 ;SET CONDITION MESSAGE POINTER
152	024756	012703	010013		MOV	#MDRDY,R3 ;SET NAME MESSAGE POINTER
153	024762			ERRHRD	314,,ERR5	
	024762	104443		TRAP	T\$ERCODE	
	024764	000472		.WORD	314	
	024766	012324		.WORD	ERR5	
154						
155	024770		62\$:			
156	024770		T365\$:			
157	024770		ENDTST			
	024770		L10022:			
	024770	104001		EMT	C\$E1\$T	

```

1          .SBTTL *TEST 4          HEAD UNLOADING
2 024772   BGNSTST                  ;TEST04
3 024772   005737 003252            T4::
4 024776   001003                    TST PASNUM ;TEST IF FIRST PASS
5 025000   005737 013712            BNE 8$ ;NO - SKIP
6 025004   100402                    TST MISWIW ;TEST IF MANUAL INTERVENTION
7 025006   104032                    BMI 10$ ;YES - SKIP
8 025012   000566                    EXIT TST
9 025012   BGNSUB                    EMT C$EXIT
10 025014  012737 006324 002714     .WORD L10023-.
11 025022  004737 015652            EMT C$BSUB
12 025026  004737 015670            MOV #NSTACHG,ERHEAD ;SET ERROR HEADER
13 025032  025466                    JSR PC,TSTINI ;INITIALIZE TEST
14 025034  032737 000001 002746     JSR PC,GSTATR ;GET STATUS
15 025042  001040                    T465$
16 025044  005046                    BIT #DRDYMSK,T.CS ;CHECK IF DRIVE READY
17 025046  153716 002735            BNE 3$ ;YES - SKIP
18 025052  012746 005743            PRINTF #FMTOP1,#OPR6,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
19 025056  013746 002730            CLR -(SP)
20 025062  012746 005732            BISB RLDRV+1,(SP)
21 025066  012746 007544            MOV #DRVNAM,-(SP)
22 025072  012746 007310            MOV RLBAS,-(SP)
23 025076  012746 011050            MOV #BASADD,-(SP)
24 025102  012746 000007            MOV #OPR1A,-(SP)
25 025106  010600                    MOV #OPR6,-(SP)
26 025110  104017                    MOV #FMTOP1,-(SP)
27 025112  062706 000020            MOV #7,-(SP)
28 025116  005037 004256            MOV SP,R0
29 025122  104043                    EMT C$PNTF
30 025124  000404                    ADD #20,SP
31 025126  004256                    CLR OBUF ;CLEAR FOR RESPONSE
32 025130  00C120                    GMANIL OPR02,OBUF,1,NO
33 025132  007052                    EMT C$GMAN
34 025134  000001                    BR 10000$
35 025136  005737 004256            .WORD OBUF
36 025142  001740                    .WORD T$CODE
37 025144  052737 000010 002706     .WORD OPR02
38 025146  005046                    .WORD 1
39 025152  005737 004256            TST OBUF ;TST RESPONSE YES
40 025154  012746 005743            BEQ 1$ ;NO - SKIP
41 025160  013746 002730            BIS #UNLOAD,OPFLAG ;SET UNLOAD OPERATION
42 025164  012746 005732            PRINTF #FMTOP1,#OPR3,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
43 025170  012746 007544            CLR -(SP)
44 025174  012746 007232            BISB RLDRV+1,(SP)
45 025176  012746 011050            MOV #DRVNAM,-(SP)
46 025200  012746 000007            MOV RLBAS,-(SP)
47 025204  012746 000007            MOV #BASADD,-(SP)
48 025210  012746 000007            MOV #OPR1A,-(SP)
49 025214  010600                    MOV #OPR3,-(SP)
50 025216  012746 000007            MOV #FMTOP1,-(SP)
51 025218  010600                    MOV #7,-(SP)
52 025220  010600                    MOV SP,R0
    
```

	025216	104017		EMT	CSPNTF	
	025220	062706	000020	ADD	#20,SP	
24	025224	012703	000006	MOV	#6,R3	:SET EXPECTED STATE VALUE
25	025230	012704	000144	MOV	#100.,R4	:SET SECOND LEVEL COUNT
26	025234	012701	001274	MOV	#700.,R1	:SET WAIT COUNT FOR 30 SECONDS
27	025240	004737	015704	JSR	PC,GSTATC	:GET STATUS
28	025244	025466		T4658		
29	025246	020337	002762	CMP	R3,T.STAT	:CHECK IF STATE 6
30	025252	001436		BEQ	118	:YES - SKIP
31	025254	022737	000005 002762	CMP	#5,T.STAT	:TEST IF STATE 5
32	025262	001025		BNE	98	:NO - REPORT WRONG STATE
33	025264	005304		DEC	R4	:DEC 2ND LEVEL COUNT
34	025266	001004		BNE	68	:SKIP IF NOT 0
35	025270	005301		DEC	R1	:ELSE DEC 1ST LEVEL COUNT
36	025272	001406		BEQ	78	:IF 0 - SKIP TO QUESTION
37	025274	012704	000144	MOV	#100.,R4	:ELSE RESET 2ND LEVEL
38	025300			WAITUS	#1	:WAIT 100 US
	025300	012700	000001	MOV	#1,R0	
	025304	104027		EMT	C\$WTU	
39	025306	000754		BR	58	
40	025310	005037	004256	CLR	OBUF	:CLEAR FOR RESPONSE
41	025314			GMANIL	OPR003,OBUF,1,NO	
	025314	104043		EMT	C\$GMAN	
	025316	000404		BR	100018	
	025320	004256		.WORD	OBUF	
	025322	000120		.WORD	T\$CODE	
	025324	007077		.WORD	OPR003	
	025326	000001		.WORD	1	
	025330			100018:		
42	025330	00573	004256	TST	OBUF	:TEST IF RESPONSE YES
43	025334	001706		BEQ	48	:NO - SKIP
44	025336			ERRHRD	401...ERR7	:ELSE REPORT STATE CHANGE WRONG
	025336	104443		TRAP	T\$ERCODE	
	025340	000621		.WORD	401	
	025342	013252		.WORD	ERR7	
45	025344			EXIT	SUB	
	025344	104032		EMT	C\$EXIT	
	025346	000126		.WORD	L10024-	
46	025350	012703	000007	MOV	#7,R3	:SET EXPECTED STATE VALUE
47	025354	012701	005670	MOV	#3000.,R1	:SET COUNT FOR 300MS
48	025360	004737	015704	JSR	PC,GSTATC	:GET STATUS
49	025364	025466		T4658		
50	025366	020337	002762	CMP	R3,T.STAT	:CHECK IF STATE 7
51	025372	001413		BEQ	188	:YES - SKIP
52	025374	005301		DEC	R1	:DEC WAIT COUNT
53	025376	001404		BEQ	168	:SKIP IF 0
54	025400			WAITUS	#1	
	025400	012700	000001	MOV	#1,R0	
	025404	104027		EMT	C\$WTU	
55	025406	000764		BR	128	
56	025410			ERRHRD	402...ERR7	:REPORT WRONG STATE CHANGE
	025410	104443		TRAP	T\$ERCODE	
	025412	000622		.WORD	402	
	025414	013252		.WORD	ERR7	
57	025416			EXIT	SUB	
	025416	104032		EMT	C\$EXIT	
	025420	000054		.WORD	L10024-	

```

58 025422 005003          18$: CLR R3          ;SET EXPECTED STATE VALUE
59 025424 012701 001130   MOV #600,R1      ;SET WAIT COUNT FOR 60 SECONDS
60 025430 004737 015704   20$: JSR PC,GSTATC ;GET STATUS
61 025434 025466          T465$
62 025436 005737 002762   TST T,STAT      ;CHECK IF STATE 0
63 025442 001411          BEQ 24$         ;YES - SKIP
64 025444 005301          DEC R1          ;DEC WAIT COUNT
65 025446 001404          BEQ 22$         ;SKIP IF 0
66 025450          WAITMS #1
   025450 012700 000001   MOV #1,R0
   025454 104026          EMT C$WTH
67 025456 000764          BR 20$
68 025460          22$: ERRHRD 403,,ERR7 ;REPORT WRONG STATE CHANGE
   025460 104443          TRAP T$ERRCODE
   025462 000623          .WORD 403
   025464 013252          .WORD ERR7
69 025466          24$:
70 025466 012737 000002 002720 T465$: MOV #2,ERRSWI ;INIT ERROR SWITCH
71
72 025474          ENDSUB
   025474          L10024:
73 025474 104003          26$: EMT C$ESUB
   025476 005046          PRINTF #FMTOP1,#OPR6,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1> ;REQUEST CYCLE UP
   025500 153716 002735   CLR -(SP)
   025504 012746 005743   BISB RLDRV+1,(SP)
   025510 013746 002730   MOV #DRVNAM,-(SP)
   025514 012746 005732   MOV RLBAS,-(SP)
   025520 012746 007544   MOV #BASADD,-(SP)
   025524 012746 007310   MOV #OPR1A,-(SP)
   025530 012746 011050   MOV #OPR6,-(SP)
   025534 012746 000007   MOV #FMTOP1,-(SP)
   025540 010600          MOV #7,-(SP)
   025542 104017          PUSH SP,R0
   025544 062706 000020   EMT C$PNTF
74 025550 005037 004256   ADD #20,SP
75 025554          CLR OBUFF ;CLEAR FOR RESPONSE
   025554 104043          GMANIL OPR002,OBUFF,1,NC
   025556 000404          EMT C$GMAN
   025560 004256          BR 10000$
   025562 000120          .WORD OBUFF
   025564 007052          .WORD T$CODE
   025566 000001          .WORD OPR002
   025570          .WORD 1
76 025570 005737 004256   10000$: TST OBUFF ;TEST RESPONSE YES
77 025574 001740          BEQ 26$        ;NO - SKIP
78 025576          29$:
79
80 025576          ENDTST
   025576          L10023:
   025576 104001          EMT C$ETST
    
```

1				.SBTTL	*TEST 5	DRIVE SELECT	
2	025600			BGNTST		:TEST05	
	025600						T5::
3	025600	012737	000002	002720	MOV	#2,ERRSWI	;SET FOR NO ERROR RETURN
4	025606	005737	003252		TST	PASNUM	;TEST IF FIRST PASS
5	025612	001173			BNE	EXT05	;NO - SKIP
6	025614	032737	000004	013712	BIT	#DRSELT,MISWIW	;TEST IF SELECT TESTS
7	025622	001567			BEQ	EXT05	;NO - SKIP
8	025624			1\$:	PRINTF	#FMTOP1,#OPR7,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>	;REQUEST REMOVE ALL P
	025624	005046			CLR	-(SP)	
	025626	153716	002735		BISB	RLDRV+1,(SP)	
	025632	012746	005743		MOV	#DRVNAM,-(SP)	
	025636	013746	002730		MOV	RLBAS,-(SP)	
	025642	012746	005732		MOV	#BASADD,-(SP)	
	025646	012746	007544		MOV	#OPR1A,-(SP)	
	025652	012746	007343		MOV	#OPR7,-(SP)	
	025656	012746	011050		MOV	#FMTOP1,-(SP)	
	025662	012746	000007		MOV	#7,-(SP)	
	025666	010600			MOV	SP,RO	
	025670	104017			EMT	C\$PNTF	
	025672	062706	000020		ADD	#20,SP	
9	025676	005037	004256		CLR	OBUFF	;CLEAR FOR RESPONSE
10	025702				GMANIL	OPR002,OBUFF,1,NO	
	025702	104043			EMT	C\$GMAN	
	025704	000404			BR	10000\$	
	025706	004256			.WORD	OBUFF	
	025710	000120			.WORD	T\$CODE	
	025712	007052			.WORD	OPR002	
	025714	000001			.WORD	1	
	025716			10000\$:			
11	025716	005737	004256		TST	OBUFF	;TEST RESPONSE YES
12	025722	001740			BEQ	1\$;NO - SKIP
13	025724	012737	006436	002714	3\$:	MOV	#T05ERR,ERHEAD ;SET ERROR HEADER MESSAGE
14	025732	004737	015652		JSR	PC,TSTINT	;INITIALIZE TEST
15	025736	004737	015704		JSR	PC,GSTATC	;DO SELECT AND GET STATUS
16	025742	026124			T504\$		
17	025744	013737	002734	003020	MOV	RLDRV,TEMPO	;STORE ORIGINAL DRIVE NUMBER
18	025752	013701	002734		MOV	RLDRV,R1	;PUT IT IN R1
19	025756	012704	000004		MOV	#4,R4	;SET COUNT FOR NUMBER OF PLUGS
20	025762	062701	000400		LPT05:	ADD	#400,R1 ;BUMP TO NEXT DRIVE
21	025766	022701	002000		CMP	#2000,R1	;CHECK IF TO LARGE
22	025772	001001			BNE	4\$;NO - SKIP
23	025774	005001			CLR	R1	;ELSE CLEAR TO DRIVE 0
24	025776	010137	002734	4\$:	MOV	R1,RLDRV	;PUT IT BACK IN RLDRV
25	026002			5\$:	PRINTF	#FMTOP3,#OPR8,<B,RLDRV+1>,#OPR1B,#UNDTST	
	026002	012746	007560		MOV	#UNDTST,-(SP)	
	026006	012746	007550		MOV	#OPR1B,-(SP)	
	026012	005046			CLR	-(SP)	
	026014	153716	002735		BISB	RLDRV+1,(SP)	
	026020	012746	007372		MOV	#OPR8,-(SP)	
	026024	012746	011121		MOV	#FMTOP3,-(SP)	
	026030	012746	000005		MOV	#5,-(SP)	
	026034	010600			MOV	SP,RO	
	026036	104017			EMT	C\$PNTF	
	026040	062706	000014		ADD	#14,SP	
26							;INSERT PLUG REQUEST
27	026044	005037	004256		CLR	OBUFF	;CLEAR FOR RESPONSE

```

28 026050          GMANIL  OPR002,OBUFF,1,NO
   026050 104043    EMT      CS$GMAN
   026052 000404    BR        10001$
   026054 004256    .WORD    OBUFF
   026056 000120    .WORD    T$CODE
   026060 007052    .WORD    OPR002
   026062 000001    .WORD    1
   026064          10001$:
29 026064 005737 004256  TST      OBUFF          ;TEST RESPONSE YES
30 026070 001744          BEQ      5$            ;NO - SKIP
31 026072          BGNSUB
   026072          T5.1:
32 026074 004737 015704  EMT      C$BSUB
   026074 004737 015704  JSR      PC,GSTATC      ;GET STATUS - REPORT ANY ERROR
33 026100 026102          60$:
34 026102 012737 000002 002720  MOV      #2,ERRSWI      ;INIT ERROR SWITCH
35
36 026110          ENDSUB
   026110          L10026:
37 026112 005304          EMT      C$ESUB
   026112 005304          DEC      R4              ;DEC COUNT
38 026114 001322          BNE     LPT05          ;LOOP IF NOT ZERO
39 026116 013737 003020 002734  MOV      TEMPO,RLDRV    ;ELSE RESTORE RLDRV
40 026124          T504$:
41 026124          4$:
   026124 012746 007411  MOV      #FMT4,#OPR8,#OPR9
   026130 012746 007372  MOV      #OPR9,-(SP)
   026134 012746 011164  MOV      #OPR8,-(SP)
   026140 012746 000003  MOV      #FMT4,-(SP)
   026144 010600          MOV      #3,-(SP)
   026146 104017          MOV      SP,R0
   026150 062706 000010  EMT      C$PNTF
42 026154 005037 004256  ADD      #10,SP
   026160          CLR      OBUFF          ;CLEAR FOR RESPONSE
43 026160          GMANIL  OPR002,OBUFF,1,NO
   026160 104043    EMT      CS$GMAN
   026162 000404    BR        10000$
   026164 004256    .WORD    OBUFF
   026166 000120    .WORD    T$CODE
   026170 007052    .WORD    OPR002
   026172 000001    .WORD    1
   026174          10000$:
44 026174 005737 004256  TST      OBUFF          ;TEST RESPONSE YES
45 026200 001751          BEQ      4$            ;NO - SKIP
46 026202          EXT05:
47 026202          ENDTST
   026202          L10025:
   026202 104001    EMT      C$TST
48
  
```

```

1          .SBTTL *TEST 6          DRIVE SELECT TEST
2 026204   BGNTST                  ;TEST06
3 026204   005737 003252          TST    PASNUM          ;CHECK IF FIRST PASS
4 026210   001004          BNE    1$          ;NO - SKIP
5 026212   032737 000004 013712  BIT    #DRSELT,MISWIW ;CHECK IF TEST DRIVE SELECT
6 026220   001002          BNE    4$          ;YES - SKIP
7 026222   1$:                   EXIT   TST
8 026222   104032          EMT    C$EXIT
9 026224   000630          .WORD  L10027-
10 026226  012737 006372 002714  4$:   MOV    #GSTER1,ERHEAD ;SET ERROR HEADER
11 026234  004737 015652          JSR    PC,TSTINT    ;INITIALIZE TEST
12 026240  013703 003254          MOV    PSETNM,R3    ;GET PARAM SET NUMBER
13 026244  023727 002012 000001  CMP    LSUNIT,#1    ;TEST IF MORE THAN 1 UNIT
14 026252  101450          BLOS   5$          ;NO - SKIP
15 026254  005203          INC    R3          ;BUMP PARAMETER SET NUMBER
16 026256  020337 002012          CMP    R3,LSUNIT   ;CHECK IF PAST VALID PARAMETER TABLE
17 026262  101401          BLOS   3$          ;NO - SKIP
18 026264  005003          CLR    R3          ;ELSE CLEAR TO POINT TO ENTRY 0
19 026266  3$:                   GPWORD R3,R0
20 026266  010300          MOV    R3,R0
21 026270  104042          EMT    C$GPHRD
22 026272  103370          BNCOMPLETE 2$      ;SKIP IF NOT AVAILABLE
23 026272  010004          BCC    2$
24 026274  021437 002730          MOV    R0,R4      ;PUT POINTER INTO R4
25 026302  001364          CMP    (R4),RLBAS ;CHECK IF SAME CONTROLLER
26 026304  005037 002710          BNE    2$          ;NO - SKIP
27 026310  012737 000104 002736  CLR    DONE        ;CLEAR DONE FLAG
28 026316  056437 000010 002736  MOV    #GTSTAT,L.CS ;LOAD GET STATUS
29 026324  012737 000013 002742  BIS    10(R4),L.CS ;INSERT DRIVE
30 026332  013762 002742 000004  MOV    #GETSTAT!DRSET,L.DA ;SET UP TO CLEAR DRIVE
31 026340  013762 002736 000000  MOV    L.DA,RLDA(R2) ;LOAD DA REG
32 026346  012700 000003          MOV    L.CS,RLCS(R2) ;LOAD CS REG
33 026352  104026          WAITMS #3          ;WAIT 300 MS
34 026354  005737 002710          MOV    #3,R0
35 026360  001735          EMT    C$WTM
36 026362  032737 100000 002746  TST    DONE        ;TEST IF INTERRUPT
37 026370  001415          BEQ    2$          ;NO - SKIP
38 026372  000730          BIT    #ANYERR,T.CS ;TEST IF ANY ERROR SET
39 026374  012746 007426          BEQ    7$          ;NO - GO TEST
40 026400  012746 011361          BR     2$          ;ELSE CHECK NEXT DRIVE
41 026404  012746 000002          PRINTF #FMT9,#OPR10 ;REPORT CAN'T FIND 2ND DRIVE
42 026410  010600          MOV    #OPR10,-(SP)
43 026412  104017          MOV    #FMT9,-(SP)
44 026414  062706 000006          MOV    #2,-(SP)
45 026420  000137 027054          MOV    SP,R0
46 026424  016437 000010 003022  7$:   EMT    C$PNTF
47 026432  013700 002734          ADD    #6,SP
48 026436  013705 003022          JMP    LCLEXT
49 026442  042700 002000          MOV    10(R4),TEMP1 ;STORE NEW ADDRESS
50 026446  042705 002000          ;ASK FOR PLUG CHANGE
51 026452  020527 001400          MOV    RLDRV,R0    ;GET DRIVE UNDER TEST
52 026456  001001          MOV    TEMP1,R5    ;GET NEW ADDRESS
53          BIC    #2000,R0  ;CLEAR FOR ADDRESS 0 TO 3
54          BIC    #2000,R5
55          CMP    R5,#1400 ;TEST IF DRIVE NUMBER 3
56          BNE    21$   ;NO - SKIP
  
```

```

44 026460 005005          CLR      R5          ;ELSE SET TO DRIVE NUMBER 0
45 026462 062705 000400 21$:  ADD      #400,R5      ;BUMP TO NEXT ADDRESS
46 026466 020500          CMP      R5,R0       ;THIS EQUAL TO NEW ADDRESS?
47 026470 001770          BEQ     20$          ;YES - SKIP
48 026472 052705 000200  BIS     #CRDYMSK,R5  ;ELSE SET CONTROLLER READY BIT
49 026476 010562 000000  MOV     R5,RLCS(R2)  ;AND LOAD CS REG
50 026502          PRINTF  #FMTOP2,#OPR8,<B,RLDRV+1>,#OPR1B,<B,TEMP1+1>
    026502 005046          CLR     -(SP)
    026504 153716 003023  BISB   TEMP1+1,(SP)
    026510 012746 007550  MOV     #OPR1B,-(SP)
    026514 005046          CLR     -(SP)
    026516 153716 002735  BISB   RLDRV+1,(SP)
    026522 012746 007372  MOV     #OPR8,-(SP)
    026526 012746 011077  MOV     #FMTOP2,-(SP)
    026532 012746 000005  MOV     #5,-(SP)
    026536 010600          MOV     SP,R0
    026540 104017          EMT    C$PNTF
    026542 062706 000014  ADD     #14,SP
51 026546 005037 004256  CLR     OBUFF        ;CLEAR FOR RESPONSEF
52 026552          GMANIL  OPRO02,OBUFF,1,NO
    026552 104043          EMT    C$GMAN
    026554 000404          BR     10000$
    026556 004256          .WORD  OBUFF
    026560 000120          .WORD  T$CODE
    026562 007052          .WORD  OPRO02
    026564 000001          .WORD  1
    026566          10000$:
53 026566 005737 004256  TST     OBUFF        ;TEST IF RESPONSE YES
54 026572 001717          BEQ     9$           ;NO - SKIP
55 026574 012704 000012  MOV     #10.,R4      ;SET COUNT
56 026600          BGNSUB
    026600          T6.1:
57 026602 013737 002734 002736 JS$:  EMT    C$BSUB
58 026610 013762 002736 000000  MOV     RLDRV,L.CS   ;SET UP TO SELECT MULTIPLE DRIVES
59 026616          MOV     L.CS,RLCSR(R2) ;DO IT
    026616 012700 000012  WAITMS #10.
    026622 104026          MOV     #10.,R0
60 026624 052737 000104 002736  EMT    C$WTM
61 026632 012737 000003 002742  BIS     #GTSTAT,L.CS ;SET GET STATUS
62 026640 013762 002742 000004  MOV     #GETSTAT,L.DA
63 026646 005037 002710          MOV     L.DA,RLDA(R2)
64 026652 013762 002736 000000  CLR     DONE
65 026660          MOV     L.CS,RLCSR(R2) ;DO GET STATUS
    026660 012700 000001  WAITUS #1           ;WAIT FOR INTERRUPT
    026664 104027          MOV     #1,R0
66 026666 005737 002710          EMT    C$WTU
67 026672 001012          TST     DONE        ;CHECK IF INTERRUPTED
68 026674 004737 015520  BNE    12$          ;YES - SKIP
69 026700 012603          JSR    PC,WAITIN    ;WAIT FOR TIMEOUT
70 026702 001406          MOV     (SP)+,R3    ;GET ERROR POINTER
71 026704          BEQ     12$          ;SKIP IF 0
    026704 104463          ERRHRD 601.,GSTER1,ERR1
    026706 001131          TRAP   T$ERCODE
    026710 006372          .WORD  601
    026712 012072          .WORD  GSTER1
72 026714          .WORD  ERR1
          EXIT      SUB
  
```



```

026714 104032      EMT      C$EXIT
026716 000062      .WORD    L10030-
73 026720          12$:  WAITMS   #2          ;WAIT FOR DSE TO SET
026720 012700 000002  MOV      #2,R0
026724 104026      EMT      C$WTM
74 026726 004737 015720  JSR      PC,GSTAT      ;GET STATUS
75 026732 026772      60$:
76 026734 032737 000400 002754  BIT      #DSESTAT,T.MP ;TEST IF DRIVE SELECT ERROR SET
77 026742 001007      BNE      16$          ;YES - SKIP
78 026744 012703 010241  MOV      #MDSERR,R3    ;SET NAME MESSAGE POINTER
79 026750          ERRHRD   602...ERP3
026750 104443      TRAP     T$ERCODE
026752 001132      .WORD    602
026754 012206      .WORD    ERR3
80 026756          EXIT     SUB
026756 104032      EMT      C$EXIT
026760 000020      .WORD    L10030-
81 026762 010562 000000  16$:  MOV      R5,RLCS(R2)   ;LOAD IN DIFFERENT ADDRESS
82 026766 005304      DEC      R4           ;DEC COUNT
83 026770 001304      BNE      8$           ;LOOP IF NOT ZERO
84 026772 012737 000002 002720  60$:  MOV      #2,ERRSWI    ;INIT ERROR SWITCH
85 027000          ENDSUB
027000          L10030:
027000 104003      EMT      C$ESUB
86 027002          15$:  PRINTF   #FMT9,#OPR11 ;REQUEST PLUG CHANGE
027002 012746 007474  MOV      #OPR11,-(SP)
027006 012746 011361  MOV      #FMT9,-(SP)
027012 012746 000002  MOV      #2,-(SP)
027016 010600      MOV      SP,R0
027020 104017      EMT      C$PNTF
027022 062706 000006  ADD      #6,SP
87 027026 005037 004256  CLR      OBUFF        ;CLEAR FOR RESPONSE
88 027032          GMANIL   OPRO02,OBUFF,1,NO
027032 104043      EMT      C$GMAN
027034 000404      BR       10000$
027036 004256      .WORD    OBUFF
027040 000120      .WORD    T$CODE
027042 007052      .WORD    OPRO02
027044 000001      .WORD    1
027046          10000$:
89 027046 005737 004256  TST      OBUFF        ;TEST RESPONSE YES
90 027052 001753      BEQ      15$          ;NO - SKIP
91 027054          LCLEXT:
92 027054          ENDTST
027054          L10027:
027054 104001      EMT      C$ETST
  
```

				.SBTTL	*TEST 7	INITIAL STATE	
1				BGNTST		:TEST 07	
2	027056						T7::
3	027056	012737	006423	002714	MOV	#INITST,ERHEAD	:SET ERROR HEADER
4	027064	004737	015652		JSR	PC,TSTINT	:INITIALIZE TEST
5	027070				WAITUS	#10.	:WAIT 1 MS
	027070	012700	000012		MOV	#10.,R0	
	027074	104027			EMT	C\$WTU	
6	027076	004737	015704		JSR	PC,GSTATC	:GET STATUS
7	027102	027406			65\$		
8	027104	032737	000001	002746	BIT	#DRDYMSK,T.CS	:CHECK IF DRIVE READY
9	027112	001003			BNE	3\$:YES-SKIP
10	027114	012703	010013		MOV	#MDRDY,R3	:SET NAME MESSAGE POINTER
11	027120	000427			BR	9\$:GO REPORT
12	027122	012703	000005	3\$:	MOV	#5,R3	:SET EXPECTED STATE VALUE
13							
14	027126	020337	002762		CMP	R3,T.STAT	:CHECK IF STATE OK
15	027132	001405			BEQ	5\$:YES-SKIP
16	027134				ERRHRD	701.,,ERR7	:ELSE REPORT STATE ERROR
	027134	104443			TRAP	T\$ERCODE	
	027136	001275			.WORD	701	
	027140	013252			.WORD	ERR7	
17	027142				EXIT	TST	:EXIT
	027142	104032			EMT	C\$EXIT	
	027144	000242			.WORD	L1003!-.	
18	027146	013701	002754	5\$:	MOV	T.MP,R1	:GET MP REG
19	027152	032701	000020		BIT	#HOSTAT,R1	:CHECK HEADS OUT
20	027156	001003			BNE	7\$:YES-SKIP
21	027160	012703	010230		MOV	#MHOSTA,R3	:SET NAME MESSAGE PTR
22	027164	000405			BR	9\$:GO REPORT
23	027166	032701	000010	7\$:	BIT	#BHSTAT,R1	:CHECK BRUSH HOME SET
24	027172	001007			BNE	10\$:YES-SKIP
25	027174	012703	010204		MOV	#MBHSTA,R3	:SET NAME MESSAGE PTR
26	027200			9\$:	ERRHRD	702.,,ERR3	:REPORT ERROR
	027200	104443			TRAP	T\$ERCODE	
	027202	001276			.WORD	702	
	027204	012206			.WORD	ERR3	
27	027206				EXIT	TST	:EXIT
	027206	104032			EMT	C\$EXIT	
	027210	000176			.WORD	L1003!-.	
28	027212	005737	013712	10\$:	TST	MISWIW	:TEST IF MANUAL INTERVENTION RUN
29	027216	100034			BPL	16\$:NO-SKIP
30	027220	005737	003252		TST	PASNUM	:CHECK IF FIRST PASS
31	027224	001031			BNE	16\$:NO-SKIP
32	027226	032701	000100		BIT	#HSSTAT,R1	:ELSE CHECK HD 0 SELECTED
33	027232	001411			BEQ	13\$:YES-SKIP
34	027234	012703	010146		MOV	#MHSTA,R3	:SET NAME MESSAGE PTR
35	027240	012704	011020		MOV	#CCYLUP,R4	:SET CONDITION POINTER
36	027244				ERRHRD	703.,,ERR4	:REPORT ERROR
	027244	104443			TRAP	T\$ERCODE	
	027246	001277			.WORD	703	
	027250	012254			.WORD	ERR4	
37	027252				EXIT	TST	:EXIT
	027252	104032			EMT	C\$EXIT	
	027254	000132			.WORD	L1003!-.	
38	027256	032701	001000	13\$:	BIT	#VCSTAT,R1	:CHECK VOL CHECK SET
39	027262	001003			BNE	15\$:YES-SKIP

```

40 027264 012703 010160      MOV      #MVOLCK,R3      ;ELSE SET NAME MESSAGE PTR
41 027270 000743              BR        9$             ;GO REPORT
42 027272 032737 040000 002746 15$: BIT      #DRVERR,T.CS   ;TEST DRIVE ERROR SET
43 027300 001003              BNE      16$            ;YES-SKIP
44 027302 012703 010135      MOV      #MDRERR,R3     ;ELSE SET NAME MESSAGE PTR
45 027306 000734              BR        9$             ;GO REPORT
46 027310 032701 020000      BIT      #WLSTAT,R1     ;CHECK WRITE LOCK STATUS
47 027314 001405              BEQ      17$            ;SKIP IF RESET
48 027316 012703 010217      MOV      #MWLSTA,R3     ;ELSE SET NAME MESSAGE PTR
49 027322              ERRHRD   705,,,ERR2
    027322 104443              TRAP     T$ERCODE
    027324 001301              .WORD   705
    027326 012140              .WORD   ERR2
50 027330 042701 021177      BIC      #21177,R1      ;CLEAR STAUS EXCEPT FOR ERROR BITS
51 027334 023727 002200 000001 17$: CMP      T.DRIVE,#1
52 027342 001404              BEQ      99$
53 027344 022701 000200      CMP      #200,R1
54 027350 001410              BEQ      19$
55 027352 000402              BR        18$
56 027354 005701              99$:  TST      R1
57 027356 001405              BEQ      19$            ;NO-SKIP
58 027360 18$: ERRHRD   704,,,ERR6   ;ELSE REPORT ALL ERRORS
    027360 104443              TRAP     T$ERCODE
    027362 001300              .WORD   704
    027364 012374              .WORD   ERR6
59 027366              EXIT     TST            ;EXIT
    027366 104032              EMT      C$EXIT
    027370 000016              .WORD   L10031-
60 027372 013701 002746      MOV      T.CS,R1       ;GET CS REG
61 027376 042701 141777      BIC      #141777,R1     ;CLEAR ALL BUT ERROR BITS
62 027402 005701              TST      R1            ;TEST IF ANY ERROR SET
63 027404 001365              BNE      18$           ;YES-SKIP TO REPORT
64 027406              25$:
65 027406              65$:
66 027406              ENDTST
    027406 L10031:
    027406 104001              EMT      C$ETST
    
```

```

1
2
3
4 027410          .SBTTL *TEST 8          INITIAL RESET STATE
   027410          BGNSTST          ;TEST 8
5 027410 012737 006423 002714          MOV #INITST,ERHEAD          T8::
6 027416 004737 015652          JSR PC,TSTINT          ;INITIALIZE TEST
7
8 027422 004737 015670          JSR PC,GSTATR          ;GET STATUS WITH RESET
9 027426 027472          65$
10 027430 005737 013712          TST MISWIW          ;CHECK IF MAN INTERVENTION WAS RUN
11 027434 100016          BPL 4$          ;NO-SKIP
12 027436 005737 003252          TST PASNUM          ;CHECK IF 1ST PASS
13 027442 001013          BNE 4$          ;NO-SKIP
14 027444 032737 000100 002754          BIT #HSSTAT,T.MP          ;CHECK HD SELECT STILL 0
15 027452 001407          BEQ 4$          ;YES-SKIP
16 027454 012703 010146          MOV #MHSTA,R3          ;SET NAME MESSAGE PTR
17 027460 012704 011020          MOV #CCYLUP,R4          ;SET CONDITION POINTER
18 027464          ERRHRD 801,,ERR4          ;REPORT ERROR
   027464 104443          TRAP T$ERRCODE
   027466 001441          .WORD 801
   027470 012254          .WORD ERR4
19 027472          4$:
20 027472          65$:
21 027472          ENDTST
   027472          L10032:
   027472 104001          EMT C$ETST
22
    
```

```

1
2
3
4 027474          .SBTTL *TEST 9          DRIVE READY
   027474          BGNTST                ;TEST 9
5 027474 012737 006451 002714          MOV #T09ERR,ERHEAD ;SET ERROR HEADER
6 027502 012701 003004                MOV #NEWCYL,R1     ;GET POINTER TO DESIRED LOC
7 027506 005021                CLR (R1)+          ;CLEAR NEW CYL
8 027510 005021                CLR (R1)+          ;CLEAR CURRENT CYL
9 027512 005021                CLR (R1)+          ;
10 027514 005011                CLR (R1)           ; DIFFERENCE
11 027516 004737 015652                JSR PC,TSTINT      ; SIGN
12 027522 004737 015670                JSR PC,GSTATR     ;INITIALIZE TEST
13 027526 027772                65$              ;GET STATUS WITH RESET
14 027530 004737 020616                JSR PC,POSHSB     ;POSITION HEAD SELECTED BIT
15 027534 010537 003014                MOV R5,DESHD      ;STORE AS DESIRED HEAD
16 027540 004737 017324                JSR PC,SIMSEK     ;EXECUTE SIMPLE SEEK
17 027544 027772                65$
18 027546 012703 010013                MOV #MDRDY,R3     ;SET NAME MESSAGE PTR
19 027552 012704 010761                MOV #CDRDY,R4     ;SET CONDITION POINTER
20 027556 004737 015720                JSR PC,GSTAT      ;GET STATUS
21 027562 027772                65$
22 027564 032737 000001 002746          BIT #DRDYMSK,T.CS ;TEST READY SET
23 027572 001405                BEQ 4$            ;NO-SKIP
24 027574          ERRHRD 901,,,ERR4          ;REPORT READY ERROR
   027574 104443          TRAP T$ERCODE
   027576 001605          .WORD 901
   027600 012254          .WORD ERR4
25 027602          EXIT TST ;EXIT
   027602 104032          EMT C$EXIT
   027604 000166          .WORD L10033-.
26 027606 012701 000121 4$: MOV #81,,R1 ;SET WAIT COUNT
27 027612 004737 015720 5$: JSR PC,GSTAT ;GET STATUS
28 027616 027772                65$
29 027620 012703 000005                MOV #5,R3         ;SET EXPECTED STATE VALUE
30 027624 023703 002762                CMP T,STAT,R3    ;CHECK STATE IS 5
31 027630 001405                BEQ 7$            ;YES-SKIP
32 027632          ERRHRD 902,,,ERR7          ;ELSE REPORT
   027632 104443          TRAP T$ERCODE
   027634 001606          .WORD 902
   027636 013252          .WORD ERR7
33 027640          EXIT TST
   027640 104032          EMT C$EXIT
   027642 000130          .WORD L10033-.
34 027644 012703 010013 7$: MOV #MDRDY,R3
35 027650 032737 000001 002746          BIT #DRDYMSK,T.CS ;CHECK READY SET
36 027656 001013                BNE 12$          ;YES-SKIP
37 027660 005301                DEC R1            ;ELSE DEC WAIT COUNT
38 027662 001404                BEQ 9$            ;SKIP IF 0
39 027664          WAITUS #1
   027664 012700 000001                MOV #1,R0
   027670 104027          EMT C$WTU
40 027672 000747                BR 5$
41 027674          ERRHRD 903,,,ERR5          ;REPORT READY ERROR
   027674 104443          TRAP T$ERCODE
   027676 001607          .WORD 903
   027700 012324          .WORD ERR5
    
```

```
42 027702          EXIT  TST
    027702 104032   EMT  C$EXIT
    027704 000066   .WORD L10033-.
43
44 027706 005737 002746 12$:  TST  T.CS          ;TEST IF ANY ERROR
45 027712 100005          BPL  15$          ;NO-SKIP
46 027714          ERRHRD 904...ERR6
    027714 104443   TRAP  T$ERCODE
    027716 001610   .WORD 904
    027720 012374   .WORD ERR6
47 027722          EXIT  TST
    027722 104032   EMT  C$EXIT
    027724 000046   .WORD L10033-.
48 027726 012703 010146 15$:  MOV  #MHSTA,R3      ;SET NAME MESSAGE PTR
49 027732 004737 020616   JSR  PC,POSMSB    ;POSITION HEAD SELECT BIT FOR TEST
50 027736 020537 003014   CMP  R5,DESHD    ;CHECK IF CORRECT HEAD SELECTED
51 027742 001413          BEQ  20$          ;YES-SKIP
52 027744 005737 003014   TST  DESHD       ;ELSE TEST IF 1 DESIRED
53 027750 001405          BEQ  17$          ;NO-REPORT SB 0
54 027752          ERRHRD 905...ERR3    ;ELSE REPORT SB 1
    027752 104443   TRAP  T$ERCODE
    027754 001611   .WORD 905
    027756 012206   .WORD ERR3
55 027760          EXIT  TST
    027760 104032   EMT  C$EXIT
    027762 000010   .WORD L10033-.
56 027764          ERRHRD 906...ERR2
    027764 104443   TRAP  T$ERCODEF
    027766 001612   .WORD 906
    027770 012140   .WORD ERR2
57 027772          20$:
58 027772          65$:
59 027772          ENDTST
    027772          L10033:
    027772 104001   EMT  C$ETST
```

1									
2	027774				.SBTTL	*TEST 10	SEEK SIGN SWITCH		
	027774				BGNTST		;TEST 10		
3	027774	012737	006461	002714				T10::	
4	030002	012701	003004		MOV	#T10ERR,ERHEAD	;SET ERROR HEADER		
5	030006	005021			MOV	#NEWCYL,R1			
6	030010	005021			CLR	(R1)*	;CLEAR NEW CYL		
7	030012	005021			CLR	(R1)*	;CLEAR CURRENT CYLINDER		
8	030014	052721	000001		CLR	(R1)*	;CLEAR DIFFERENCE		
9	030020	004737	020616		BIS	#BIT0,(R1)*	;SET FOR SIGN OF 1		
10	030024	010521			JSR	PC,POSMSB	;GET SELECTED HEAD		
11	030026				MOV	R5,(R1)*	;SET AS DESIRED HEAD		
12	030026								
	030026				T1048:				
	030026	104002			BGNSUB			T10.1:	
13	030030	004737	015652		EMT	C\$BSUB			
14	030034	004737	015670		JSR	PC,TSTINT	;INITIALIZE TEST		
15	030040	030274			JSR	PC,GSTATR	;GET STATUS		
16	030042	004737	017324		60\$				
17	030046	030274			JSR	PC,SIMSEK	;DO SEEK		
18	030050	012703	010013		60\$				
19	030054	012704	010761		MOV	#MDRDY,R3	;SET NAME MESSAGE PTR		
20	030060	004737	015720		MOV	#CDRDY,R4	;SET CONDITION MESSAGE PTR		
21	030064	030274			JSR	PC,GSTAT	;GET STATUS		
22	030066	032737	000001	002746	60\$				
23	030074	001405			BIT	#DRDYMSK,T.CS	;CHECK READY RESET		
24	030076				BEQ	4\$;YES-SKIP		
	030076	104443			ERRHRD	1001,,,ERR4	;REPORT READY ERROR		
	030100	001751			TRAP	T\$ERCODE			
	030102	012254			.WORD	1001			
25	030104				.WORD	ERR4			
	030104	104032			EXIT	SUB	;EXIT SUBTEST		
	030106	000166			EMT	C\$EXIT			
26					.WORD	L10035-			
27									
28	030110	012701	000121		4\$:	MOV	#81,R1	;SET WAIT COUNT	
29	030114	004737	015720		5\$:	JSR	PC,GSTAT	;GET STATUS	
30	030120	030274			60\$				
31	030122	012703	000005		MOV	#5,R3	;SET EXPECTED STATE		
32	030126	020337	002762		CMP	R3,T.STAT	;CHECK STATE IS 5		
33	030132	001405			BEQ	7\$;YES-SKIP		
34	030134				ERRHRD	1002,,,ERR7	;REPORT STATE ERROR		
	030134	104443			TRAP	T\$ERCODE			
	030136	001752			.WORD	1002			
	030140	013252			.WORD	ERR7			
35	030142				EXIT	SUB	;EXIT		
	030142	104032			EMT	C\$EXIT			
	030144	000130			.WORD	L10035-			
36	030146	012703	010013		7\$:	MOV	#MDRDY,R3	;SET NAME MESSAGE PTR	
37	030152	032737	000001	002746	BIT	#DRDYMSK,T.CS	;CHECK READY SET		
38	030160	001013			BNE	12\$;YES-SKIP		
39	030162	005301			DEC	R1	;DO WAIT COUNT		
40	030164	001404			BEQ	9\$;SKIP IF 0		
41	030166				WAITUS	#1			
	030166	012700	000001		MOV	#1,R0			
	030172	104027			EMT	C\$WTL			
42	030174	000747			BR	5\$			

```

43
44 030176          98:  ERRHRD 1003...ERR5  ;REPORT READY ERROR
   030176 104443   TRAP  TBERCODE
   030200 001753   .WORD 1003
   030202 012324   .WORD ERR5
45 030204          EXIT  SUB  ;EXIT
   030204 104032   EMT  C$EXIT
   030206 000066   .WORD L10035-.
46 030210 005737 002746 128:  TST  T.CS  ;TEST IF ANY OTHER ERROR
47 030214 100005   BPL  158  ;NO-SKIP
48 030216          ERRHRD 1004...ERR6  ;REPORT ALL ERRORS
   030216 104443   TRAP  TBERCODE
   030220 001754   .WORD 1004
   030222 012374   .WORD ERR6
49 030224          EXIT  SUB  ;EXIT
   030224 104032   EMT  C$EXIT
   030226 000046   .WORD L10035-.
50
51 030230 012703 010146 158:  MOV  #MHSTA,R3  ;SET NAME MESSAGE PTR
52 030234 004737 020616   JSR  PC,POSHSB ;GET SELECTED HEAD BIT
53 030240 020537 003014   CMP  R5,DESHD  ;CHECK IF CORRECT
54 030244 001413   BEQ  208  ;YES - SKIP
55 030246 005737 003014   TST  DESHD  ;WAS IT SET
56 030252 001405   BEQ  178  ;NO-SKIP
57 030254          ERRHRD 1005...ERR3  ;REPORT SB 1
   030254 104443   TRAP  TBERCODE
   030256 001755   .WORD 1005
   030260 012206   .WORD ERR3
58 030262          EXIT  SUB
   030262 104032   EMT  C$EXIT
   030264 000010   .WORD L10035-.
59 030266          ERRHRD 1006...ERR2  ;REPORT SB 0
   030266 104443   TRAP  TBERCODE
   030270 001756   .WORD 1006
   030272 012140   .WORD ERR2
60
61 030274          208:
62 030274          608:
63 030274          ENDSUB
   030274          L10035:
64 030276 005737 003012   EMT  C$ESUB
65 030302 001404   TST  DESSGN  ;CHECK IF BOTH SIGN USED
66 030304 005037 003012   BEQ  258  ;YES-SKIP
67 030310 000137 030026   CLR  DESSGN  ;SET FOR SIGN OF 0
68 030314          JMP  T1048  ;DO TEST AGAIN
69 030314          258:
   030314          ENDTST
   030314          L10034:
   030314 104001   EMT  C$ETST
  
```



```

1
2
3
4 030316          .SBTTL *TEST 11      HEAD ALIGNMENT SUPPORT
   030316          BGNSTST          ;TEST 11
5 030316 032737 000010 013712      BIT      #HDALIGN,MISWIW ;CHECK IF RUN HEAD ALIGNMENT
6 030324 001411          BEQ      1$      ;NO-EXIT
7 030326 005737 003252          TST      PASNUM      ;TEST IF PASS 0
8 030332 001006          BNE      1$      ;NO-EXIT
9 030334 023737 002734 002712      CMP      RLDRV,HADONE ;TEST IF HEAD ALIGN DONE THIS DRIVE
10 030342 001004          BNE      2$      ;NO - SKIP
11 030344 000137 030646          JMP      T11$      ;GO CHECK WRITE LOCK
12 030350          1$:      EXIT      TST
   030350 104032          EMT      C$EXIT
   030352 000374          .WORD    L10036-
13 030354 013737 002734 002712 2$:      MOV      RLDRV,HADONE ;SET HEAD ALIGN DONE FLAG
14 030362          PRINTF   #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
   030362 005046          CLR      -(SP)
   030364 153716 002735          BISB    RLDRV+1,(SP)
   030370 012746 005743          MOV      #DRVNAM,-(SP)
   030374 013746 002730          MOV      RLBAS,-(SP)
   030400 012746 005732          MOV      #BASADD,-(SP)
   030404 012746 011175          MOV      #FMT5,-(SP)
   030410 012746 000005          MOV      #5,-(SP)
   030414 010600          MOV      SP,RO
   030416 104017          EMT      C$PNTF
   030420 062706 000014          ADD     #14,SP
15 030424          PRINTF   #FMT9,#HAMES1 ;TYPE INSTRUCTIONS
   030424 012746 006715          MOV      #HAMES1,-(SP)
   030430 012746 011361          MOV      #FMT9,-(SP)
   030434 012746 000002          MOV      #2,-(SP)
   030440 010600          MOV      SP,RO
   030442 104017          EMT      C$PNTF
   030444 062706 000006          ADD     #6,SP
16 030450          PRINTF   #FMT9,#HAMES2
   030450 012746 007000          MOV      #HAMES2,-(SP)
   030454 012746 011361          MOV      #FMT9,-(SP)
   030460 012746 000002          MOV      #2,-(SP)
   030464 010600          MOV      SP,RO
   030466 104017          EMT      C$PNTF
   030470 062706 000006          ADD     #6,SP
17 030474          BGNSUB
   030474          T11.1:
18 030476 004737 015652          3$:      EMT      C$BSUB
19 030502 005037 002710          JSR      PC,TSTINT ;INITIALIZE TEST
20 030506 013737 002734 002736      CLR      DONE      ;CLEAR DONE
21 030514 052737 000104 002736      MOV      RLDRV,L,CS ;SET UP FOR GET STATUS
22 030522 012737 000013 002742      BIS      #GTSTAT,L,CS
23 030530 013762 002742 000004      MOV      #GETSTAT:DRSET,L,DA
24 030536 013762 002736 000000      MOV      L,DA,RLDA(R2)
25 030544          WAITMS #50. ;DO GET STATUS
   030544 012700 000062          MOV      #50.,RO ;WAIT FOR INTERRUPT
   030550 104026          EMT      C$WTM
26 030552 005737 002710          TST      DONE      ;CHECK IF DONE
27 030556 001747          BEQ     3$      ;NO-GO CLR CONTROLLER
28

```

```

29
30 030560 012737 000021 002742 10$: MOV #HDSEL!MBSET0,L.DA;LOAD FOR HEAD 1
31 030566 032737 020000 002754 BIT #WLSTAT,T.MP ;CHECK IF WRITE LOCK SET
32 030574 001003 BNE 12$ ;YES-SKIP
33 030576 042737 000020 002742 BIC #HDSEL,L.DA ;ELSE CLEAR TO HEAD 0
34 030604 013737 002734 002736 12$: MOV RLDRV,L.CS ;LOAD IN DRIVE NUMBER
35 030612 052737 000106 002736 BIS #SEEK,L.CS ;SET FOR SEEK
36 030620 013762 002742 000004 MOV L.DA,RLDA(R2) ;LOAD & EXECUTE SEEK
37 030626 013762 002736 000000 MOV L.CS,RLCSR(R2)
38 030634 WAITMS #30. ;WAIT FOR INTERRUPT
   030634 012700 000036 MOV #30.,R0
   030640 104026 EMT CSWTM
39 030642 000715 BR 3$ ;LOOP
40 030644 59$: ENDSUB
41 030644 L10037: EMT CS$SUB
   030644 104003
42 030646 T115$: BGNSUB
43 030646 EMT CS$SUB
   030646 104002
44 030650 004737 015652 JSR PC,TSTINT ;INITIALIZE TEST
45 030654 004737 015670 JSR PC,GSTATR ;CLEAR DRIVE
46 030660 030744 60$
47 030662 032737 020000 002754 BIT #WLSTAT,T.MP ;CHECK WRITE LOCK RESET
48 030670 001425 BEQ 19$ ;YES-SKIP
49 030672 18$: PRINTF #FMT9,#OPR12 ;REQUEST WRITE LOCK RESET
   030672 012746 007525 MOV #OPR12,-(SP)
   030676 012746 011361 MOV #FMT9,-(SP)
   030702 012746 000002 MOV #2,-(SP)
   030706 010600 MOV SP,R0
   030710 104017 EMT CSPNTF
   030712 062706 000006 ADD #6,SP
50 030716 005037 004256 CLR OBUFF ;CLEAR FOR RESPONSE
51 030722 GMANIL OPRO02,OBUFF,1,NO ;GET RESPONSE
   030722 104043 EMT CS$GMAN
   030724 000404 BR 10000$
   030726 004256 .WORD OBUFF
   030730 000120 .WORD T$CODE
   030732 007052 .WORD OPRO02
   030734 000001 .WORD 1
   030736 10000$:
52 030736 005737 004256 TST OBUFF ;WAS ANSWER YES
53 030742 001753 BEQ 18$ ;NO-REPEAT REQUEST
54 030744 19$:
55 030744 60$: ENDSUB
56 030744 L10040:
   030744 104003 EMT CS$SUB
57 030746 20$:
58 030746 ENDTST
   030746 L10036:
   030746 104001 EMT CS$TST
  
```

```

1
2
3
4      030750      .SBTTL *TEST 12      HEAD SWITCHING
      030750      BGNTST      ;TEST 12
5      030750      012737      006501      002714      MOV      #T12ERR,ERHEAD      ;SET ERROR HEADER
6      030756      012701      003004      MOV      #NEWCYL,R1      ;GET POINTER TO DESIRED LOCATION
7      030762      005021      CLR      (R1)+      ;CLEAR NEW CYLINDER
8      030764      005021      CLR      (R1)+      ;CLEAR CURRENT CYL.
9      030766      005021      CLR      (R1)+      ;CLEAR DIFFERENCE
10     030770      005021      CLR      (R1)+      ;CLEAR SIGN
11     030772      012721      000001      MOV      #1,(R1)+      ;SET FOR HEAD 1
12     030776
13     030776      T124$:
      030776      BGNSUB
14     031000      004737      015652      EMT      C$BSUB      T12.1:
15     031004      004737      015670      JSR      PC,TSTINT      ;INITIALIZE TEST
16     031010      031244      JSR      PC,GSTATR      ;GET STATUS WITH RESET
17     031012      004737      017324      60$
18     031016      031244      JSR      PC,SIMSEK      ;DO SEEK
19     031020      012703      010013      60$
20     031024      012704      010761      MOV      #MDRDY,R3      ;SET NAME MESSAGE PTR
21     031030      004737      015720      MOV      #CDRDY,R4      ;SET CONDITION POINTER
22     031034      031244      JSR      PC,GSTAT      ;GET STATUS
23     031036      032737      000001      002746      BIT      #DRDYMSK,T.CS      ;CHECK IF READY
24     031044      001405      BEQ      5$      ;NO-SKIP
25     031046      ERRHRD      1201,,,ERR4      ;REPORT READY ERROR
      031046      104443      TRAP      T$ERCODE
      031050      002261      .WORD      1201
      031052      012254      .WORD      ERR4
26     031054      EXIT      SUB      ;EXIT
      031054      104032      EMT      C$EXIT
      031056      000166      .WORD      L10042-.
27
28     031060      012701      000121      5$:      MOV      #81,,R1      ;SET WAIT COUNT
29     031064      004737      015720      6$:      JSR      PC,GSTAT      ;GET STATUS
30     031070      031244      60$
31     031072      012703      000005      MOV      #5,R3      ;SET EXPECTED STATE VALUE
32     031076      020337      002762      CMP      R3,T.STAT      ;CHECK IF STATE IS 5
33     031102      001405      BEQ      7$      ;YES-SKIP
34     031104      ERRHRD      1202,,,ERR7      ;REPORT STATE ERROR
      031104      104443      TRAP      T$ERCODE
      031106      002262      .WORD      1202
      031110      013252      .WORD      ERR7
35     031112      EXIT      SUB
      031112      104032      EMT      C$EXIT
      031114      000130      .WORD      L10042-.
36
37     031116      012703      010013      7$:      MOV      #MDRDY,R3      ;SET NAME MESSAGE PTR
38     031122      032737      000001      002746      BIT      #DRDYMSK,T.CS      ;CHECK DRIVE READY
39     031130      001013      BNE      12$      ;YES-SKIP
40     031132      005301      DEC      R1      ;DEC WAIT COUNT
41     031134      001404      BEQ      9$      ;SKIP IF 0
42     031136      WAITUS      #1
      031136      012700      000001      MOV      #1,R0
      031142      104027      EMT      C$WTU
    
```

```
43 031144 000747 BR 6$
44
45 031146 9$: ERRHRD 1203...ERR5 ;REPORT READY ERROR
   031146 104443 TRAP T$ERCODE
   031150 002263 .WORD 1203
   031152 012324 .WORD ERR5
46 031154 EXIT SUB ;EXIT
   031154 104032 EMT C$EXIT
   031156 000066 .WORD L10042-.
47
48 031160 005737 002746 12$: TST .CS ;TEST IF ANY ERROR
49 031164 100005 BPL 15$ ;NO-SKIP
50 031166 ERRHRD 1204...ERR6 ;REPORT ALL ERRORS
   031166 104443 TRAP T$ERCODE
   031170 002264 .WORD 1204
   031172 012374 .WORD ERR6
51 031174 EXIT SUB
   031174 104032 EMT C$EXIT
   031176 000046 .WORD L10042-.
52 031200 012703 010146 15$: MOV #MHSTA,R3 ;SET NAME MESSAGE PTR
53 031204 004737 020616 JSR PC,POSHSB ;POSITION HEAD SELECT BIT
54 031210 023705 003014 CMP DESHD,R5 ;CHECK IF CORRECT HEAD SELECTED
55 031214 001413 BEQ 20$ ;YES-SKIP
56 031216 005737 003014 TST DESHD ;WAS HEAD 0 SELECTED
57 031222 001405 BEQ 17$ ;YES-SKIP
58 031224 ERRHRD 1205...ERR3 ;REPORT HEAD SB 1
   031224 104443 TRAP T$ERCODE
   031226 002265 .WORD 1205
   031230 012206 .WORD ERR3
59 031232 EXIT SUB ;EXIT
   031232 104032 EMT C$EXIT
   031234 000010 .WORD L10042-.
60 031236 17$: ERRHRD 1206...ERR2 ;ELSE REPORT HEAD SB 0
   031236 104443 TRAP T$ERCODE
   031240 002266 .WORD 1206
   031242 012140 .WORD ERR2
61
62 031244 20$:
63 031244 60$:
64 031244 ENDSUB
   031244 L10042:
   031244 104003 EMT C$ESUB
65 031246 005737 003014 TST DESHD ;CHECK IF HD 0 WAS DONE
66 031252 001404 BEQ 25$ ;YES-SKIP
67 031254 005037 003014 CLR DESHD ;ELSE SET TO HEAD 0
68 031260 000137 030776 JMP T124$ ;REDO TEST
69 031264
70 031264 25$:
   031264 ENDTST
   031264 L10041:
   031264 104001 EMT C$ETST
```

```

1
2
3
4 031266          .SBTTL *TEST 13      READ HEADER (PART 1)
   031266          BGNTST          ;TEST 13
5 031266 012737 006513 002714      MOV #T13ERR,ERHEAD ;SET ERROR HEADER
6 031274 012701 003004              MOV #NEWCYL,R1    ;GET ADDRESS OF DESIRED LOCATIONS
7 031300 005021                    CLR (R1)+         ;CLEAR NEW CYL
8 031302 005021                    CLR (R1)+         ;CLEAR CURRENT CYL
9 031304 005021                    CLR (R1)+         ;CLEAR DIFF
10 031306 005021                   CLR (R1)+         ;CLEAR SIGN
11 031310 005021                   CLR (R1)+         ;CLEAR HEAD
12 031312
13 031312          T134$:
   031312          BGNSUB
   031312          T13.1:
14 031314 004737 015652          EMT C$BSUB
15 031320 004737 015670          JSR PC,TSTINT    ;INITIALIZE TEST
16 031324 031412          JSR PC,GSTATR    ;GET STATUS W/RESET
17 031326 004737 017324          JSR PC,SIMSEK    ;DO SEEK
18 031332 031412          JSR PC,SIMSEK    ;DO SEEK
19 031334 012701 000121          MOV #81.,R1     ;SET WAIT COUNT
20 031340 004737 020646          JSR PC,RDYWAIT  ;WAIT FOR READY
21 031344 031412          JSR PC,RDYWAIT  ;WAIT FOR READY
22
23 031346 004737 020160          JSR PC,XRDHDC    ;DO READ HEADER
24 031352 031412          JSR PC,XRDHDC    ;DO READ HEADER
25 031354 012703 010146          MOV #MHSTA,R3   ;SET NAME MESSAGE PTR
26 031360 004737 020610          JSR PC,POSHW1    ;POSITION HS BIT IN HD WRD 1
27 031364 020537 003014          CMP R5,DESHD    ;CHECK IF HEAD CORRECT
28 031370 001410          BEQ 15$        ;YES-SKIP
29 031372          ERRHRD 1301.,,ERR3 ;REPORT SB 1
   031372 104443          TRAP T$ERCODE
   031374 002425          .WORD 1301
   031376 012206          .WORD ERR3
30 031400          EXIT SUB
   031400 104032          EMT C$EXIT
   031402 000010          .WORD L10044-.
31 031404          ERRHRD 1302.,,ERR2 ;REPORT SB 0
   031404 104443          TRAP T$ERCODE
   031406 002426          .WORD 1302
   031410 012140          .WORD ERR2
32
33 031412          15$:
34 031412          60$:
35 031412          ENDSUB
   031412          L10044:
   031412 104003          EMT C$ESUB
36 031414 005737 003014          TST DESHD       ;TEST IF HEAD 1 DONE
37 031420 001007          BNE 20$        ;YES-SKIP
38 031422 012737 000001 003014      MOV #1,DESHD    ;ELSE SET TO HEAD 1
39 031430 013737 002754 003020      MOV HDWRD1,TEMPO ;STORE HDR WORD 1
40 031436 000725          BR T134$       ;DO TEST AGAIN
41 031440 042737 000177 003020      BIC #177,TEMPO  ;CLEAR ALL BUT CYLINDER IN 1ST HFADER
42 031446 042737 000177 002754      BIC #177,HDWRD1 ;CLEAR ALL BY CYL IN 2ND HEADER
43 031454 023737 003020 002754      CMP TEMPO,HDWRD1 ;COMPARE IF EQUAL
44 031462 001405          BEQ 22$       ;YES-SKIP

```

45 031464 012703 006631
46 031470
031470 104443
031472 002432
031474 012072
47 031476
48 031476
031476
031476 104001

MOV #CYLPER,R3 ;SET NAME MESSAGE PTR
ERRHRD 1306,,,ERR1 ;REPORT HEAD ALIGNMENT PROBLEM
TRAP T\$ERCODE
.WORD 1306
.WORD ERR1
22\$:
ENDTST
L10043:
EMT C\$ETST

```

1          .SBTTL *TEST 14          READ HEADER (PART 2)
2 031500   BGNST          ;TEST 14
3 031500   012737 006527 002714   MOV    #T14ERR,ERHEAD ;SET ERROR HEADER
4 031506   012701 003006          MOV    #CURCYL,R1    ;GET ADDRESS OF DESIRED VALUE
5 031512   005021          CLR    (R1)+          ;CLEAR CURRENT CYL
6 031514   005021          CLR    (R1)+          ;CLEAR DESIRED DIFF
7 031516   005021          CLR    (R1)+          ;CLEAR SIGN
8 031520   005021          CLR    (R1)+          ;CLEAR DESIRED HEAD
9 031522   T153$:
10 031522   BGNST          T14.1:
11 031522   104002          EMT    ^$BSUB
12 031524   004737 015652          JSR    PC,TSTINT    ;INITIALIZE TEST
13 031530   004737 015670          JSR    PC,GSTATR    ;CLEAR DRIVE
14 031534   031730          60$
15 031536   004737 017324          JSR    PC,SIMSEK    ;DO SEEK
16 031542   031730          60$
17 031544   012701 000310          MOV    #200.,R1    ;SET WAIT COUNT FOR 20 MS
18 031550   004737 020646          JSR    PC,RDYWAIT    ;WAIT FOR READY
19 031554   031730          60$
20 031556   004737 021242          JSR    PC,RDALHD    ;DO READ HEADER ALL HEADERS
21 031562   031730          60$
22 031564   005037 002716          CLR    MORECE      ;CLEAR MORE COMPARE ERRORS FOR REPORT
23 031570   052737 000002 002706  BIS    #HDRCMP,OPFLAG ;SET HDR COMPARE FLAG
24 031576   005003          CLR    R3          ;CLEAR FOR HDR COUNT
25 031600   012704 003656          MOV    #IBUFF,R4    ;GET POINTER FOR HDR TO BE CHECKED
26 031604   012705 003020          MOV    #TEMPO,R5    ;GET POINTER TO TEST AREA
27 031610   012701 000050          MOV    #40.,R1     ;SET HDR COUNT
28 031614   011415          MOV    (R4),(R5)    ;GET FIRST HEADER WORD
29 031616   042715 000100          BIC    #HDSSEL,(R5)
30 031622   005737 003014          TST    DESHD       ;TEST IF HD 0 DESIRED
31 031626   001404          BEQ    10$         ;YES-SKIP
32 031630   052715 000100          BIS    #HDSSEL,(R5) ;ELSF SET HEAD BIT
33 031634   005065 000002          CLR    2(R5)       ;CLEAR 2ND WORD OF TEST AREA
34 031640   021524          10$: CMP    (R5),(R4)+    ;COMPARE HEADER WORD
35 031642   001405          BEQ    13$         ;SKIP IF OK
36 031644   005744          TST    -(R4)       ;FLSE POSITION R4 TO BAD WORD
37 031646          ERRHRD 1501.,,ERR10 ;REPORT ERROR
38 031646   104443          TRAP   T$ERCODE
39 031650   002735          .WORD 1501
40 031652   013462          .WORD ERR10
41 031654   005724          TST    (R4)+       ;BUMP R4 TO NEXT WORD
42 031656   005203          13$: INC    R3        ;BUMP WORD COUNT
43 031660   005724          TST    (R4)+       ;TEST 2ND WORD IS 0
44 031662   001405          BEQ    15$         ;YES - SKIP
45 031664   022544          CMP    (R5)+,-(R4) ;POSITION PTRS FOR REPORT
46 031666          ERRHRD 1501.,,ERR10 ;REPORT ERROR
47 031666   104443          TRAP   T$ERCODE
48 031670   002735          .WORD 1501
49 031672   013462          .WORD ERR10
50 031674   024524          CMP    -(R5),(R4)+ ;REPOSITION POINTER
51 031676   005724          15$: TST    (R4)+       ;POSITION R4 PAST ECC WORD
52 031700   005203          INC    R3          ;BUMP WORD COUNT
53 031702   005215          INC    (R5)        ;BUMP SECTOR COUNT
54 031704   011500          MOV    (R5),R0     ;CHECK IF SECTOR IS PAST LAST SECTOR
    
```

49	031706	042700	177700		BIC	#^CHDSEC,RO	
50	031712	022700	000050		CMF	#40.,RO	
51	031716	001002			BNE	17\$:NO-SKIP
52	031720	042715	000077		BIC	#HDSEC,(R5)	:ELSE CLEAR SECTOR TO 0
53	031724	005301		17\$:	DEC	R1	:DEC HDR COUNT
54	031726	001344			BNE	10\$:YES-SKIP
55							
56	031730			60\$:			
57	031730			ENDSUB			
	031730			L10046:			
	031730	104003			EMT	C\$ESUB	
58	031732	005737	003014		TST	DESHD	:CHECK IF HD 1 TESTED
59	031736	001005			BNE	20\$:YES-SKIP
60	031740	012737	000001 003014		MOV	#1,DESHD	:ELSE SET TO HEAD 1
61	031746	000137	031522		JMP	T153\$:REDO TEST
62	031752			20\$:			
63	031752			ENDTST			
	031752			L10045:			
	031752	104001			EMT	C\$ETST	

1			.SBTTL	*TEST 15	DIFFERENCE OF 1 SEEK (PART 1)	
2	031754		BGNTST		;TEST 15	T15::
3	031754	012737	006553	002714	MOV #P2101E,ERHEAD	;SET ERROR HEADER
4	031762	012737	000004	003020	MOV #4,TEMP0	;SET PASS COUNT
5	031770	004737	015652		JSR PC,TSTINT	;INITIALIZE TEST
6	031774	004737	015670		JSR PC,GSTATR	;GET STATUS
7	032000	032274			T1765\$	
8	032002	012737	177777	003024	MOV #-1,TEMP2	;SET -1 INTO DIFF AUGMENT FOR -1 SEEK
9	032010	012704	003006		MOV #CURCYL,R4	;SET POINTERS
10	032014	012705	003004		MOV #NEWCYL,R5	
11	032020	004737	020050		JSR PC,CHOSHD	;GO CHOSE HEAD
12	032024				T172\$:	
13	032024				BGNSUB	T15.1:
	032024	104002			EMT C\$BSUB	
14	032026	004737	021114		JSR PC,GETPOS	;GET POSITION
15	032032	032232			60\$	
16	032034				INLOOP	;CHECK IF IN ERROR LOOP
	032034	104020			EMT C\$INLP	
17	032036				BNCOMPLETE 3\$;NO - SKIP
	032036	103005			BCC 3\$	
18	032040	021415			CMP (R4),(R5)	;CHECK IF CURRENT = NEW
19	032042	001005			BNE 4\$;NO - SKIP
20	032044	004737	020134		JSR PC,ONSWAP	;ELSE SWAP OLD AND NEW
21	032050	000421			BR 9\$;SKIP TO SEEK
22	032052	005437	003024	3\$:	NEG TEMP2	;CHANGE DIFF AUGMENT FOR OPPOSITE DIR
23	032056	011415		4\$:	MOV (R4),(R5)	;MOV CURRENT INTO OLD
24	032060	023714	002204		CMP HLMTW,(R4)	;CHECK IF CURRENT AT 255
25	032064	001004			BNE 7\$;NO - SKIP
26	032066	012737	177777	003024	MOV #-1,TEMP2	;AT MAX CYL, MAKE NEXT SEEK REV
27	032074	000405			BR 8\$;SKIP
28	032076	005714		7\$:	TST (R4)	;TEST IF CURRENT AT 0
29	032100	001003			BNE 8\$;NO - SKIP
30	032102	012737	000001	003024	MOV #1,TEMP2	;AT CYL 0, MAKE NEXT SEEK FWRD
31	032110	063715	003024	8\$:	ADD TEMP2,(R5)	;ADD DIFF TO NEW CYL (+1 OR -1)
32	032114	004737	016562	9\$:	JSR PC,XSEEK	;DO SEEK
33	032120	032232			60\$	
34	032122	004737	015720		JSR PC,GSTAT	;GET STATUS
35	032126	032232			60\$	
36						
37	032130	012703	000004		MOV #4,R3	;SET EXPECTED STATE
38	032134	020337	002762		CMP R3,T.STAT	;CHECK IF STATE COUNT
39	032140	001404			BEQ 10\$;YES-SKIP
40	032142				ERRHRD 101,,,ERR7	;REPORT STATE ERROR
	032142	104443			TRAP T\$ERCODE	
	032144	000145			.WORD 101	
	032146	013252			.WORD ERR7	
41	032150	000423			BR 16\$;EXIT TEST
42	032152	012703	000005	10\$:	MOV #5,R3	;SET EXPECTED STATE
43	032156	012701	000062		MOV #50.,R1	;SET WAIT COUNT FOR 5 MS
44	032162	004737	015720	12\$:	JSR PC,GSTAT	;GET STATUS
45	032166	032232			60\$	
46	032170	020337	002762		CMP R3,T.STAT	;IS STATE 5?
47	032174	001411			BEQ 16\$;YES-SKIP
48	032176	005301			DEC R1	;DEC WAIT COUNT
49	032200	001404			BEQ 14\$;SKIP IF 0

50	032202				WAITUS	#1	
	032202	012700	000001		MOV	#1,R0	
	032206	104027			EMT	C\$WTU	
51	032210	000764			BR	12\$	
52	032212			14\$:	ERRHRD	102,,,ERR7	;REPORT STATE ERROR
	032212	104443			TRAP	T\$ERCODE	
	032214	000146			.WORD	102	
	032216	013252			.WORD	ERR7	
53	032220	012701	000062	16\$:	MOV	#50.,R1	;SET WAIT COUNT FOR 5 MS
54	032224	004737	020646		JSR	PC,RDYWAIT	;GO WAIT FOR DRIVE READY
55	032230	032232			60\$		
56	032232	012737	000002	002720	60\$:	MOV	#2,ERRSWI
57	032240				ENDSUB		
	032240				L10050:		
	032240	104003			EMT	C\$ESUB	
58	032242				ESCAPE	TST	;EXIT TEST IF ERROR
	032242	104010			EMT	C\$ESCAPE	
	032244	000030			.WORD	L10047-	
59	032246	005337	003020		DEC	TEMPO	;DEC PASS COUNT
60	032252	001410			BEQ	24\$;SKIP IF 0-DONE
61							
62	032254	032737	000001	003020	BIT	#BIT0,TEMPO	;TEST IF PASS=2
63	032262	001003			BNE	23\$;NO-SKIP
64	032264	004737	020074		JSR	PC,SWAPHD	;GO SWAP TO HEAD 1 OR END TEST
65	032270	032274			24\$;ABORT RETURN
66	032272	000654			23\$:	BR	T172\$
67	032274				24\$:		
68	032274				T1765\$:		
69	032274				ENDTST		
	032274				L10047:		
	032274	104001			EMT	C\$ETST	


```

50 032522 012737 000002 002720 60$: MOV #2,ERRSWI ;INIT ERROR SWITCH
51 032530 ENDSUB
    032530 L10052:
    032530 104003 EMT C$ESUB
52 032532 ESCAPE TST ;EXIT TEST IF ERROR
    032532 104010 EMT C$ESCAPE
    032534 000030 .WORD L10051-
53 032536 005337 003020 DEC TEMPO ;DEC PASS COUNT
54 032542 001410 BEQ 30$ ;EXIT IF DONE
55
56 032544 032737 000001 003020 BIT #BIT0,TEMPO ;TEST IF PASS 1 OR 3
57 032552 001003 BNE 20$ ;YES-SKIP
58 032554 004737 020074 JSR PC,SWAPHD ;GO SWAP TO HEAD 1 OR END TEST
59 032560 032564 30$ ;ABORT RETURN
60 032562 000673 20$: BR 1187$ ;LOOP
61 032564 30$:
62 032564 11865$:
63 032564 ENDTST
    032564 10051:
    032564 104001 EMT C$ETST
64 032566 ENDMOD
  
```

1				.SBTTL	PARAMETER CODING
2	032566			BGNMOD	HRDPRM
3	032566			BGNHRD	
	032566	000030			.WORD L10053-L\$HARD/2
4	032570			GPRML	CNTYPE,CNT,1,YES
	032570	005130			.WORD T\$CODE
	032572	032724			.WORD CNTYPE
	032574	000001			.WORD 1
5	032576			GPRMA	CSRMSG,CSR,0,160000,177776,YES
	032576	000031			.WORD T\$CODE
	032600	032650			.WORD CSRMSG
	032602	160000			.WORD T\$LLOLIM
	032604	177776			.WORD T\$HILIM
6	032606			GPRMA	VECMMSG,VECT,0,0,776,YES
	032606	001031			.WORD T\$CODE
	032610	032661			.WORD VECMSG
	032612	000000			.WORD T\$LLOLIM
	032614	000776			.WORD T\$HILIM
7	032616			GPRMD	DRMSG,DRSB,0,3400,0,7,YES
	032616	004032			.WORD T\$CODE
	032620	032716			.WORD DRMSG
	032622	003400			.WORD 3400
	032624	000000			.WORD T\$LLOLIM
	032626	000007			.WORD T\$HILIM
8	032630			GPRML	DRTYPE,TYPDR,1,YES
	032630	003130			.WORD T\$CODE
	032632	032701			.WORD DRTYPE
	032634	000001			.WORD 1
9	032636			GPRMD	BRMSG,PRIOR,0,340,0,7,YES
	032636	002032			.WORD T\$CODE
	032640	032670			.WORD BRMSG
	032642	000340			.WORD 340
	032644	000000			.WORD T\$LLOLIM
	032646	000007			.WORD T\$HILIM
10					
11	032650			ENDHRD	
	032650				.EVFN
					L10053:
12					
13	032650	102	125	123	CSRMSG: .ASCIZ /BUS ADDR/
	032653	040	101	104	
	032656	104	122	000	
14	032661	126	105	103	VECMMSG: .ASCIZ /VECTOR/
	032664	124	117	122	
	032667	000			
15	032670	102	122	040	BRMSG: .ASCIZ /BR LEVEL/
	032673	114	105	126	
	032676	105	114	000	
16	032701	104	122	111	DRTYPE: .ASCIZ /DRIVE = RL01/
	032704	126	105	040	
	032707	075	040	122	
	032712	114	060	061	
	032715	000			
17	032716	104	122	111	DRMSG: .ASCIZ /DRIVE/
	032721	126	105	000	
18	032724	122	114	061	CNTYPE: .ASCIZ /RL11/
	032727	061	000		

19	032731				ENDMOD	
20					.EVEN	
21						
22	032732				BGNMOD	SFTPRM
23	032732				BGNSFT	
	032732	000021				.WORD L10054-LSSOFT/2
24						
30	032734				GPRML	SELQ,MISWI,4,YES
	032734	000130				.WORD TSCODE
	032736	032776				.WORD SELQ
	032740	000004				.WORD 4
31	032742				GPRML	ALGNQ,MISWI,10,YES
	032742	000130				.WORD TSCODE
	032744	033031				.WORD ALGNQ
	032746	000010				.WORD 10
33	032750				GPRML	MANQ,MISWI,100000,YES
	032750	000130				.WORD TSCODE
	032752	033070				.WORD MANQ
	032754	100000				.WORD 100000
34						
46	032756				3\$:	GPRMD ERLIMQ,ERLIM,D,377,0,377,YES
	032756	004052				.WORD TSCODE
	032760	033125				.WORD ERLIMQ
	032762	000377				.WORD 377
	032764	000000				.WORD T\$LOLIM
	032766	000377				.WORD T\$HILIM
50	032770				GPRML	AUTOQ,MISWI,20,YES
	032770	000130				.WORD TSCODE
	032772	033147				.WORD AUTOQ
	032774	000020				.WORD 20
51	032776				ENDSFT	
						.EVEN
	032776				L10054:	
52						
58	032776	105	130	105	SELQ:	.ASCIZ /EXECUTE DRIVE SELECT TESTS/
	033001	103	125	124		
	033004	105	040	104		
	033007	122	111	126		
	033012	105	040	123		
	033015	105	114	105		
	033020	103	124	040		
	033023	124	105	123		
	033026	124	123	000		
59	033031	105	130	105	ALGNQ:	.ASCIZ /EXECUTE HEAD ALIGNMENT SUPPORT/
	033034	103	125	124		
	033037	105	040	110		
	033042	105	101	104		
	033045	040	101	114		
	033050	111	107	116		
	033053	115	105	116		
	033056	124	040	123		
	033061	125	120	120		
	033064	117	122	124		
	033067	000				
61	033070	104	117	040	MANQ:	.ASCIZ /DO MANUAL INTERVENTION TESTS/
	033073	115	101	116		
	033076	125	101	114		

	033101	040	111	116	
	033104	124	105	122	
	033107	126	105	116	
	033112	124	111	117	
	033115	116	040	124	
	033120	105	123	124	
	033123	123	000		
69	033125	111	116	120	ERLIMQ: .ASCIZ /INPUT ERROR LIMIT/
	033130	125	124	040	
	033133	105	122	122	
	033136	117	122	040	
	033141	114	111	115	
	033144	111	124	000	
70	033147	104	122	117	AUTOQ: .ASCIZ /DROP DRV IF NO RESPONSE/
	033152	120	040	104	
	033155	122	126	040	
	033160	111	106	040	
	033163	116	117	040	
	033166	122	105	123	
	033171	120	117	116	
	033174	123	105	000	
74					.EVEN
75	033200				ENDMOD
76	033200				LASTAD
					.EVEN
	033200				L\$LAST::

1

.SBTTL DIAGNOSTIC SUPERVISOR -- LOW CORE SET UP

PWR.FAIL: POWER INTERRUPT ROUTINE

35 063774 000000
36 063776 000000
37 064000 000000
38 064002 000000
39 064006
40 000200

.WORD 0
.WORD 0
.WORD 0
.WORD 0
END.SUPV=+.2
.END 200

:SPACE FOR USER POOL POINTER
:SIZE
:CHECKSUM (NOT CURRENTLY USED)
:SIZE OF H.W. PTAB. ALLOCATION

ASSEMBLY ROUTINES
SYMBOL TABLE

ABOFLA	033524	G	BSFVAL	003264		CURR.S	033206	G	C\$RPT	=	000025		EF.RES=	000037	G
ABOPAS	033442	G	BSNSTR	007716		CURR.T	033210	G	C\$SEFG=	000047		EF.STA=	000040	G	
ABO.FM	036004		BYPNM	007647		CYLPER	006631		C\$SPRI=	000041		EF01	=	000001	G
AFSI	033232	G	B\$AAB	042270		CYLTBL	002506		C\$SVEC=	000037		EF02	=	000002	G
ALGNQ	033031		B\$AAF	042202		CYLUP =	000004		C\$TPRI=	000013		EF03	=	000003	G
ALLCYL=	000001		CAFDT	011031		CYLWD	007634		C\$UNBU=	000031		EF04	=	000004	G
ALLOC	054144		CALLPC=	000022		C\$AAD	045546		C\$WTM =	000026		EF05	=	000005	G
ALLSEC=	000002		CALLPS=	000024		C\$AAE	045560		C\$WTU =	000027		EF06	=	000006	G
ANYERR=	100000		CALLSP=	000026		C\$AAK	046556		C10MS	011001		EF07	=	000007	G
APT.ER	035134		CALLTC=	000030		C\$AAL	046722		CSSEC	011042		EF08	=	000010	G
ASSEMB=	000011		CAL.CL	060666		C\$ABRT=	000021		C500MS	011012		EF09	=	000011	G
AUTOQ	033147		CAL.TI	060724	G	C\$ADR =	000020		DANAM	006041		EF10	=	000012	G
AUTOSZ=	000020		CAMSK	002214		C\$AU =	000054		DATA CM=	000001		EF11	=	000013	G
ASAAV	040002		CCYLUP	011020		C\$BRK =	000022		DCKERR=	004000		EF12	=	000014	G
ASAAW	040016		CDRDY	010761		C\$BSEG=	000004		DCLIM =	000012		EF13	=	000015	G
ASAAZ	040030		CHKLUP	042304		C\$BSUB=	000002		DCLIMW	013724		EF14	=	000016	G
ASAAZ	040036		CHKSTR	054506		C\$BUFF=	000030		DECMG	052470		EF15	=	000017	G
ASAAZ	040052		CHKTTY	052574		C\$CEFG=	000046		DESDIF	003010		EF16	=	000020	G
ASABA	040062		CHK.MA	040444		C\$CLEA=	000012		DESHD	003014		EMT.TR	033530	G	
BADADD=	004000		CHK.PC	045574		C\$CLP1=	000006		DESSEC	003016		END.OF	041554		
BAMSK =	000060		CHK.SW	034634		C\$CVEC=	000036		DESSGN	003012		END.SU=	064006		
BANAM	006034		CHOSHD	020050		C\$DCLN=	000044		DEV.CO	033212	G	ENVIRO	033252	G	
BASADD	005732		CHRCNT	054026		C\$DODJ=	000053		DIAGMC=	000000		EOP.CH	062454	G	
BELL	010702		CH.FLA	040152		C\$DRPT=	000024		DIAG.T	033532	G	EOP.FM	036020		
BGN.SU=	033200		CH.PAS	040170		C\$DU =	000055		DIFAUG	003000		EOP.IN	040164		
BHSTAT=	000010		CKDATA=	000102		C\$EDIT=	000002		DIFWD	007610		ERHEAD	002714		
BINMSG	052454		CKERLM	015326		C\$ERDF=	000002		DIRBIT=	000004		ERLIM =	000010		
BIT0 =	000001	G	CLEAR.	041566		C\$ERHR=	000003		DIRMSK	002216		ERLIMQ	033125		
BIT00 =	000001	G	CLKACC	033440	G	C\$ERSF=	000001		DLTERR=	010000		ERLIMW	013722		
BIT01 =	000002	G	CLKBFR	060670		C\$ERSO=	000004		DONE	002710		ERRCNT	003052		
BIT02 =	000004	G	CLKCNT	033436	G	C\$ESCA=	000010		DPDVD	063142	G	ERRFOR	047000		
BIT03 =	000010	G	CLKJUM	061274	G	C\$ESEG=	000005		DPMUL	063030	G	ERRHAN	045600		
BIT04 =	000020	G	CLKRES	062276	G	C\$ESUB=	000003		DRDYMS=	000001		ERRPOI	003050		
BIT05 =	000040	G	CLKSER	062432	G	C\$ETSI=	000001		DRMSG	032716		ERRSWI	002720		
BIT06 =	000100	G	CLKSON	033476	G	C\$EXIT=	000032		DRSB =	000010		ERRVEC	003042		
BIT07 =	000200	G	CLKSE	040246		C\$GMAN=	000043		DRSELT=	000004		ERR.HR	046536		
BIT08 =	000400	G	CLNCOD	015136	G	C\$GPHR=	000042		DRSET =	000010		ERR.NU	033202	G	
BIT09 =	001000	G	CLRBYT	002206		C\$GPRI=	000040		DRTYPE	032701		ERR.SF	046542		
BIT1 =	000002	G	CLRPAR	023034		C\$GTIM=	000052		DRVCNT	002776		ERR1	012072	G	
BIT10 =	002000	G	CLR.MA	040522		C\$INIT=	000011		DRVERR=	040000		ERR1FO	047064		
BIT11 =	004000	G	CNT =	000012		C\$INLP=	000020		DRVNAM	005743		ERR10	013462	G	
BIT12 =	010000	G	CNTYPE	032724		C\$KWF=	000035		DRVNAV	005750		ERR2	012140	G	
BIT13 =	020000	G	CNVT	056744		C\$KWON=	000034		DSESTA=	000400		ERR3	012206	G	
BIT14 =	040000	G	COMMAN	033250	G	C\$LOOP=	000100		DSMSK =	001400		ERR4	012254	G	
BIT15 =	100000	G	COMMTA	056560		C\$MANI=	000051		DSPCOD	013726	G	ERR5	012324	G	
BIT2 =	000004	G	COMPOP=	007777		C\$MSG =	000023		DUNIT.	033446	G	ERR6	012374	G	
BIT3 =	000010	G	CONHNG=	000004		C\$PNTB=	000014		DVC.FT	046526		ERR7	013252	G	
BIT4 =	000020	G	CONTCL	062356	G	C\$PNTF=	000017		D\$AAG	047432		ERR8	013322	G	
BIT5 =	000040	G	CONTIN	014146		C\$PNTS=	000016		D\$AAH	047450		ERR9	013416	G	
BIT6 =	000100	G	COSTAT=	000040		C\$PNTX=	000015		D\$AAI	052216		ESC.PC	045572		
BIT7 =	000200	G	COUNT	003046		C\$POIN=	000040		D\$AAJ	052222		EV.COU	033204	G	
BIT8 =	000400	G	CRDYMS=	000200		C\$QIO =	000377		D\$AAK	052240		EXT05	026202		
BIT9 =	001000	G	CRLF	052656		C\$RDBU=	000007		D\$AAL	052256		FBSFIL	003462		
BLD.HW	040666		CSNAM	006027		C\$REFG=	000050		D\$AAM	052266		FILL	053324		
BLOCK	056300		CSR =	000000		C\$REQT=	000045		EF.CON=	000036	G	FILL.C	000204	G	
BRMSG	032670		CSRMSG	032650		C\$RFSE=	000033		EF.NEW=	000035	G	FLAGS	033244	G	
BSFLAG	002722		CURCYL	003006		C\$REVI=	000002		EF.PWR=	000034	G	FLAGS1	033246	G	

ASSEMBLY ROUTINES
SYMBOL TABLE

FLAGTA	056476	FSSUB =	000002	HDR40 =	100000	KBPTR	033310	G	LSSPCP	002020	G	
FLAG.1	040232	FSSW =	000014	HDSEC =	000077	KBUF	033312	G	LSSPTP	002024	G	
FLA.SE	056444	FSTEST=	000001	HDSEL =	000020	LAB	014120		LSSSTA	002030	G	
FLG.MA	040172	GARBAG	054030	HDWD	007623	LAB1	006053		LSSW	013712	G	
FMTOP1	011050	GBND	002212	HDWRD1	002754	LAB2	006066		LSTIML	002014	G	
FMTOP2	011077	GETCHR	052534	HDWRD2	002756	LCLEXT	027054		LSTIMU	002054	G	
FMTOP3	011121	GETCMN	056120	HDWRD3	002760	LINE.F	033526	G	LSTIMI	002052	G	
FMT1	011142	GETPAR	047612	HEAD =	000006	LOAD.F	040166		LSTSTI	002100	G	
FMT1.1	011147	GETPOS	021114	HEADLM=	010000	LOCERR	003256		LSUNIT	002012	G	
FMT11	011366	GETSTA=	000003	HEADW	013720	LOCYL =	040000		L.BA	002740		
FMT12	011374	GETSWI	055114	HERTZ.	037612	LOGMSG	052476		L.CLK.	037576		
FMT13	011402	GET.TW	054664	HICYL =	020000	LOLIM =	000002		L.CS	002736		
FMT14	011446	GLBDAT	002126	G	HILIM =	000004	LOLIMW	013714	L.DA	002742		
FMT15	011500	GLBEQA	002126	G	HILIMW	013716	LPBFR	033306	G	L.MP	002744	
FMT16	011534	GLBERR	012072	G	HLMTW	002204	LPCNTR	033304	G	L10000	012136	
FMT17	011545	GLBSUB	015262	G	HNFRER=	010000	LPT.AD	037630		L10001	012204	
FMT18	011567	GLBTXT	005134	G	HOLDSP=	000020	LPT.RE	037624		L10002	012252	
FMT19	011621	GSTAT	015720		HOSTAT=	000020	LPT05	025762		L10003	012322	
FMT2	011156	GSTATC	015704		HPTCOD	013672	LSI.RE	037620		L10004	012372	
FMT20	011656	GSTATG	015730		HRDPRM	032566	LUP	060572		L10005	013250	
FMT21	011706	GSTATR	015670		HRDWT5	023064	LUP.AD	045576		L10006	013320	
FMT22	011731	GSTER1	006372		HSM5K =	000100	LSAPT	002036	G	L10007	013414	
FMT23	011765	GTSTAT=	000104		HSSTAT=	000100	LSAUT	002074	G	L10010	013460	
FMT24	012001	GSEXCP=	000400		HW.ADR	033234	LSCCP	002106	G	L10011	013670	
FMT25	012006	GSHILI=	000002		HSAAB	057272	LSCLEA	015136	G	L10012	013710	
FMT26	012016	G\$LOLI=	000001		IBUFF	003656	LSCO	002032	G	L10013	013726	
FMT27	012042	G\$NO =	000000		ININIT	033456	LSDEPO	002011	G	L10014	015134	
FMT28	012061	G\$OFFS=	000400		INITCO	013770	LSDESC	002102	G	L10015	015254	
FMT3	011161	G\$OFSI=	000376		INITIA	052504	LSDEVP	002064	G	L10016	015260	
FMT4	011164	G\$PRMA=	000001		INITST	006423	LSDISP	013730	G	L10017	015324	
FMT5	011175	G\$PRMD=	000002		INIT.M	040570	LSDR	002112	G	L10020	023332	
FMT6	011215	G\$PRML=	000000		INIT.R	033272	LSDRCT	002070	G	L10021	023534	
FMT7	011257	G\$RADA=	000140		INOUTS=	000020	LSDRS	002072	G	L10022	024770	
FMT8	011327	G\$RADB=	000000		INPUTA	053432	LSDRST	002112	G	L10023	025576	
FMT9	011361	G\$RADD=	000040		INTEBL=	000100	SDTP	002040	G	L10024	025474	
FOLWRT=	000100	G\$RADF=	000200		INTFOR	046730	SDU	015256	G	L10025	026202	
FORM.T	047074	G\$RADL=	000120		INTHLR	015270	SDUT	002076	G	L10026	026110	
FREE	054402	G\$RADO=	000020		INVAL.	037536	SDVTY	002114	G	L10027	027054	
FRMWD	007641	G\$RADT=	000100		INVINT	046566	LSEF	002056	G	L10030	027000	
FWDSKO=	002000	G\$XFER=	000004		INV.SW	034570	LSEFLG	002034	G	L10031	027406	
FWDSKS=	000400	G\$YES =	000010		IN.SUF	041540	LSEXP1	002042	G	L10032	027472	
FSAU =	000015	HADONE	002712		ISAU =	000041	LSEXP2	002044	G	L10033	027772	
F\$BGN =	000040	HAMES1	006715		ISCLN =	000041	LSEXP3	002046	G	L10034	030314	
F\$CLEA=	000007	HAMES2	007000		ISDU =	000041	LSHARD	032570	G	L10035	030274	
F\$DU =	000016	HCESTA=	040000		ISHRD =	000041	LSHPCP	002016	G	L10036	030746	
F\$END =	000041	HCORED	037742		ISINIT=	000041	LSHPTP	002022	G	L10037	030644	
F\$HARD=	000004	HCOREQ	037652		ISMOD =	000041	ISHW	013674	G	L10040	030744	
F\$HW =	000013	HCORET	033466	G	ISMSG =	000041	LSICP	002104	G	L10041	031264	
F\$INIT=	000006	HCR CER=	004000		ISPWR =	000041	LSINIT	013770	G	L10042	031244	
F\$JMP =	000050	HC.ADR	033236	G	ISRPT =	000041	LSLADP	002026	G	L10043	031476	
F\$MOD =	000000	HC.DEF	033230	G	ISSEG =	000041	LSLAST	033200	G	L10044	031412	
F\$MSG =	000011	HC.DIA	033226	G	ISSFT =	000041	LSMREV	002050	G	L10045	031752	
F\$PWR =	000017	HDALIG=	000010		ISSRV =	000041	LSNAME	002000	G	L10046	031730	
F\$RPT =	000012	HDCYL	002220		ISSUB =	000041	LSREPP	002066	G	L10047	032274	
F\$SEG =	000003	HDHSEL=	000100		ISTST =	000041	LSREV	002010	G	L10050	032240	
F\$SOFT=	000005	HDMOVF	006572		JJJ	002202	LSSOFT	032734	G	L10051	032564	
F\$SRV =	000010	HDRCMP=	000002		JSJMP =	000167	LSSPC	002062	G	L10052	032530	

ASSEMBLY ROUTINES
SYMBOL TABLE

L10053	032650	MOPER	005306	OLDCYL	003002	PRI04 =	000200	G	SAVEDO=	035134	
L10054	032776	MOPERR	010375	ONSWAP	020134	PRI05 =	000240	G	SBSFIL	003266	
MAJ.IN	033262	MORECE	002716	OPFLAG	002706	PRI06 =	000300	G	SEARCH	054632	
MAJ.LO	060672	MOUTIN	005347	OPIERR=	002000	PRI07 =	000340	G	SECWD	007627	
MAJ.US	033264	MPNAM	006046	OPMSG	002126	PRNTST	053674		SEEK =	000106	
MANQ	033070	MQUALS=	003760	NPRO02	007052	PRO.CM	040144		SEEKOP=	010000	
MAN.TI	001244	MREAD	005142	NPRO03	007077	PSETNM	003254		SEGSTA	033500	G
MAP16	063400	MREADH	005153	NPRO04	007573	PTAB.S	033464	G	SFLQ	032776	
MASK.B	042302	MRESKO	005557	OPR1	007122	PUTCHR	052510		SEQMES	007662	
MASK.W	042300	MREVSK	005435	OPR1A	007544	PWCON	014444		SETDON	014222	
MBADAD	005613	MRLFAL	010556	OPR1B	007550	PWRFLG	003262		SET.MA	040356	
MBADSF	005634	MRSLT	005315	OPR10	007426	PWR.FA	063634	G	SFTPRM	032732	G
MBHSTA	010204	MSEEK	005134	OPR11	007474	PWR.FL	033270	G	SGNWD	007616	
MBSETO=	000501	MSG.AD	033224	OPR12	007525	PWR.MS	063762		SHIFT	063462	G
MCERR	010024	MSG.TY	033200	OPR2	007200	PWR.SA	063756		SIMSEK	017324	
MCONHN	006201	MSPERR	010271	OPR3	007232	PWR.UP	063760		SPDERR	006336	
MCOSTA	010171	MSTERR	010324	OPR5	007246	P.CLK.	037604		SPDSTA=	004000	
MCYLOC	010461	MTMBS	005711	OPR6	007310	P2T01E	006553		SPEC.U	040072	
MCYLUP	005337	MTOSLO	006107	OPR7	007343	P2T02E	006553		SPTCOD	013710	G
MDATCP	005221	MUL	062676	OPR8	007372	RDALHD	021242		SPV.SE	000400	
MDCRC	010046	MULOAD	005326	OPR9	007411	RDDATA=	000114		SSINDX	002704	
MDHEDR	002000	MUNDEF	010511	OUTINS=	000040	RDHEAD=	000110		STAMES	007705	
MDLT	010073	MVOLCK	010160	O\$APTS=	000000	RDNOHR=	000116		STAMSK=	000007	
MDRDY	010013	MWDERR	010360	O\$AU =	000000	RDYCHK	017610		STARTC	062352	G
MDRERR	010135	MWGERR	010307	O\$BGNR=	000000	RDYWAI	020646		STATE2	010731	
MDRRES	006126	MWLSTA	010217	O\$BGNS=	000001	READRL	015466		STATE3	010741	
MDRVST	010256	MWORD	006101	O\$DU =	000001	READ.P	060674	G	STATE5	010751	
MDSERR	010241	MWRCHK	005163	O\$GNSW=	000001	REGBAC	063364	G	STOSTA=	010000	
MEM.SI	037640	MWRITE	005175	O\$POIN=	000001	REGSAV	063350	G	STRCHR	053364	
MERRS	010673	MWRSET	005272	PARSE	056172	RELDWT=	040000		STRT.T	040150	
MEXERS	010633	MWRTAB	010617	PART1 =	000001	REQN.P	033254	G	ST.SET	035002	
MFLERR	010422	M40HDR	005256	PAR.LA	052160	REQN.T	040146		SUBSTK	002306	
MFMTER	005664	NEWCYL	003004	PASCNT	003044	RESE3	010706		SUNIT.	040154	
MFOLWR	005413	NEWPRI	062422	PASNEW	014174	RESE4	010712		SUPERV	036036	
MFWDSK	005470	NEXAR	056662	PASNUM	003252	RESE5	010717		SUPFLA	033444	G
MFWSKO	005523	NOCLR =	000010	PASS.C	033214	RESE6	010724		SUPV.T	033616	G
MGTSTA	005207	NCERCT	003257	PATTBL	002262	RESPAR	002764		SUP.PR	034554	
MHCERR	010342	NOIRPT=	000002	PAT1	004656	RESTAR	014136		SVCBGL=	000001	
MHCRC	010036	NOOP =	000100	PAT10	005132	RESTBL	002222		SVCGBL=	000000	
MHDERR	010405	NOUPR	005767	PAT2	004660	REVSKO=	001000		SVCHAN	042472	
MHDRCP	005240	NO.CLK	037566	PAT3	004720	REVSKS=	000200		SVCINS=	000000	
MHFRC	010105	NO.FLA	056456	PAT4	004760	RE.SET	034736		SVCSUB=	000001	
MHNF	010057	NO.LPT	053774	PAT5	005020	RLBA =	000002		SVCTAG=	000000	
MHOSTA	010230	NO.PTA	037772	PAT6	005026	RLBAS	002730		SVCTST=	000001	
MHSTA	010146	NR =	000000	PAT7	005066	RLCS =	000000		SWAPHD	020074	
MINOUT	005370	NSTACH	006324	PAT8	005070	RLCSR =	000000		SWCHAN	037764	
MIN.IN	033256	NUMBIN	047120	PAT9	005130	RLDA =	000004		SWITCH	056636	
MIN.US	033260	NUM.LA	047266	POSHD0	020622	RLDRV	002734		SW.ADR	033240	G
MISTST	006301	NUM.NO	033242	POSHSB	020616	RLMP =	000006		SW.PTA	037750	
MISWI =	000000	NUM.UN	033650	POSHW1	020610	RLVEC	002732		SYS.FT	046516	
MISWIW	013712	NUNITS	042256	PRINTC	054004	RORWOP=	020000		S\$LSYM=	010000	
MITEST=	100000	NXMERR=	020000	PRINTF	057312	RPTOP	021604		TBLSTR	002726	
MNDRST	010466	NXTFOR	056736	PRIOR =	000004	RPTREM	022600		TBT	002446	
MNEERR	010450	NXTHL	002210	PRI00 =	000000	RPTRES	022372		TCERR	007772	
MNOCLR	006215	NXTPAS	014166	PRI01 =	000040	RSTACK	062624	G	TEMPO	003020	
MNOINT	006146	OBUFF	004256	PRI02 =	000100	RSTRT	014054		TEMP1	003022	
MODR	062742	OCTMSG	052462	PRI03 =	000140	SAMSK =	000077		TEMP2	003024	

ASSEMBLY ROUTINES
SYMBOL TABLE

MACRO V03.01 9-FEB-79 19:14:22 PAGE 171-4

C 11

SEQ 0132

TEMP3	003026	TSEXCP=	000000	T.STAT	002762	T25TBL	002332	WGESTA=	002000
TEMP4	003030	T\$FLAG=	000040	T05ERR	006436	T25TB2	002360	WIDTH	047466
TEMP5	003032	T\$HLI=	000377	T09ERR	006451	T3	023536 G	WLSTAT=	020000
TEMP6	003034	T\$LOLI=	000000	T1	023064 G	T33TBL	002406	WRTSWI	002724
TEMP7	003036	T\$LSYM=	010000	T10	027774 G	T365\$	024770	WTDATA=	000112
TEMP8	003040	T\$NEST=	177777	T10ERR	006461	T4	024772 G	XEQDIA	062510 G
TERM1	060662	T\$NSKO=	000000	T10.1	030026	T4.1	025012	XEQSUB	062476 G
TERML1	056464	T\$NSK1=	000005	T104\$	030026	T465\$	025466	XEQ.CL	042220
TERMTA	052446	T\$NSK2=	000002	T11	030316 G	T5	025600 G	XEQ.CM	037530
TEST.M	040104	T\$SAVL=	177777	T11.1	030474	T5.1	026072	XEQ.IN	041702
TIMFLG	033434 G	T\$SEGL=	177777	T11.2	030646	T504\$	026124	XEQ.LA	035772
TIM.CO	033266 G	T\$SUBN=	000001	T115\$	030646	T6	026204 G	XEQ.OP	041774
TIM.OP	047072	T\$TAGL=	177777	T12	030750 G	T6.1	026600	XEQ.PR	035174
TOO.MA	052426	T\$TAGN=	010055	T12ERR	006501	T7	027056 G	XEQ.TE	042040
TOSLOW=	000001	T\$TEMP=	000000	T12.1	030776	T8	027410 G	XRDHD	020170
TRPFLG	003260	T\$TEST=	000020	T124\$	030776	T9	027474 G	XRDHDC	020160
TRPHAN	015262	T\$TSTM=	177777	T13	031266 G	ULOAD =	000010	XRDHDG	020174
TSTINT	015652	T\$TSTS=	000001	T13ERR	006513	UNDTST	007560	XSEEK	016562
TSTLAB	006273	T\$\$CLE=	010015	T13.1	031312	UNIT.D	033216 G	XSEEKT	016552
TST.AB	042414	T\$\$DU =	010016	T134\$	031312	UNI.MA	040074	XSEEK1	016566
TST.TO	034616	T\$\$HAR=	010053	T14	031500 G	UNXERR	006256	XTIME	061362 G
TYPDR =	000006	T\$\$HW =	010012	T14ERR	006527	USER.P	033460 G	XTIMEN	062206
TYPEC	053022	T\$\$INI=	010014	T14.1	031522	USER.T	033462 G	XTIMST	061404
TYPEPC	046716	T\$\$MSG=	010011	T15	031754 G	VALID.	033720	XXDP.D	037550
TYPFLA	056340	T\$\$SOF=	010054	T15.1	032024	VAL.LA	034540	XSALWA=	000000
TYPLIN	052720	T\$\$SRV=	010017	T153\$	031522	VAL.SW	040204	X\$FALS=	000040
TYPNUM	052302	T\$\$SUB=	010052	T16	032276 G	VCNRST	006235	X\$OFFS=	000400
TYPSTR	052740	T\$\$SW =	010013	T16ERR	006543	VCSTAT=	001000	X\$TRUE=	000020
TYP.ER	046546	T\$\$TES=	010051	T16.1	032352	VECMG	032661	\$BREG	040244
TY.UNI	041560	T.BA	002750	T172\$	032024	VECT =	000002	\$ENDAD	062462 G
T\$ARGC=	000002	T.CS	002746	T1765\$	032274	WAITIN	015520	\$SAV2	063526 G
T\$CODE=	000130	T.DA	002752	T1865\$	032564	WCMSK =	017777	\$SAV3	063542 G
T\$ERCO=	000043	T.DRIV	002200	T187\$	032352	WCRNG =	160000	\$SAV4	063560 G
T\$ERRN=	000311	T.MP	002754	T2	023334 G	WDESTA=	100000	\$SAV5	063600 G

. ABS. 064004 000
000000 001
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 20256 WORDS (80 PAGES)
DYNAMIC MEMORY AVAILABLE FOR 70 PAGES
CZRLIA.BIN,CZRLIA=#SVCRT/M,PART1,C2HEAD,CZRLIA,C2TAIL,DOCTOR