

.REM @

IDENTIFICATION

PRODUCT CODE: AC-F119D-MC  
PRODUCT NAME: CZRLIDO RL01/02 DRIVE TEST 1  
DATE CREATED: 5-JAN-79  
REVISED: 6-NOV-81  
MAINTAINER: DIAGNOSTIC ENGINEERING - COLORADO  
AUTHORS: D. CLAFLIN

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

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

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

COPYRIGHT (C) 1977,1980,1982 DIGITAL EQUIPMENT CORPORATION

-----  
TABLE OF CONTENTS  
-----

1.0	GENERAL INFORMATION
1.1	PROGRAM ABSTRACT
1.1.1	STRUCTURE OF PROGRAM
1.1.2	DIAGNOSTIC INFORMATION
1.1.3	DIAGNOSTIC HISTORY
1.2	SYSTEM REQUIREMENTS
1.2.1	HARDWARE REQUIREMENTS
1.2.2	SOFTWARE 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 FIVE STEPS OF EXECUTION
2.1.2	SAMPLE RUN-THROUGH
2.2	CHAIN MODE OPERATION
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
3.1	ERROR REPORTING
3.1.1	SPECIFIC OPERATION MESSAGES
3.1.2	SPECIFIC RESULT MESSAGES
3.1.3	OTHER MESSAGES
3.2	ERROR HALTS
4.0	PERFORMANCE AND PROGRESS REPORTS
4.1	PERFORMANCE REPORTS
4.2	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 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 2.2 'CHAIN MODE OPERATION' FOR DETAILS OF CHAINING PROCEDURE). IT IS A SINGLE PROGRAM FROM THE STANDPOINT OF THE DIAGNOSTIC USER, WHICH AT RUN TIME IS APPENDED TO A COMMON FRONT-END PIECE OF SUPERVISOR SOFTWARE THROUGH WHICH THE DIAGNOSTIC PROGRAM INTERFACES TO THE ENVIRONMENT AS IT EXECUTES.

WHEN THIS DIAGNOSTIC IS STARTED, 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 (DR>). AT COMMAND MODE THE OPERATOR MAY ENTER ANY OF SEVERAL COMMANDS AS DESCRIBED IN 2.0 'OPERATING INSTRUCTIONS'.

THE DIAGNOSTIC PROGRAM IS LOADED IN THE LOWER 8K OF MEMORY. THE DIAGNOSTIC SUPERVISOR CODING OCCUPIES 6.25K OF THE UPPER PART OF MEMORY JUST BELOW THE XXDP+ MONITOR WHICH RESIDES IN THE UPPERMOST 1.5K OF MEMORY SPACE.

#### 1.1.2 DIAGNOSTIC INFORMATION

-----

THIS PROGRAM TESTS AND EXERCISES RLO1/02 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 RLO1/02 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. WITHOUT MANUAL INTERVENTION, THE TEST REQUIRES APPROXIMATELY 135 SECONDS TO RUN.

### 1.1.3 DIAGNOSTIC HISTORY

REVISION C: MODIFY THE DIAGNOSTIC TO RUN USING THE DRS.  
 REVISION D: THE RL DRIVES HAD THE BRUSH DRIVE REMOVED. THE DIAGNOSTIC CORRECTLY TESTS BOTH DRIVES WITH AND WITHOUT A BRUSH DRIVE. IT ALSO WILL WORK ON A SYSTEM THAT DOES NOT HAVE A KW11P. BREAKS WERE INSERTED TO FACILITATE QUICKER RESPONSE TO A C.

## 1.2 SYSTEM REQUIREMENTS

### 1.2.1 HARDWARE REQUIREMENTS

- \* PDP-11/LSI-11 PROCESSOR WITH 16K OR MORE OF MEMORY
- \* CONSOLE DEVICE (LA30,LA36,VT50,ETC.)
- \* 1 OR 2 RL11/RLV11 CONTROLLER(S) WITH:
  - 1 - 8 RL01 DRIVES WITH RL01K CARTRIDGES CONTAINING A 'BAD SECTOR FILE'
  - 1 - 8 RL02 DRIVES WITH RL02K CARTRIDGES CONTAINING A 'BAD SECTOR FILE'
- \* KW11P CLOCK (P CLOCK) OR KW11L (L CLOCK)
- \* LINE PRINTER (OPTIONAL)

### 1.2.2 SOFTWARE REQUIREMENTS

CZRL100 RL01/02 DRIVE TEST 1

## 1.3 RELATED DOCUMENTS AND STANDARDS

RL01/02 DISK SUBSYSTEM USER'S GUIDE (EK-RL012-UG-002)  
 XXDP+/USER'S MANUAL

## 1.4 DIAGNOSTIC HIERARCHY PREREQUISITES

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

CVRLA	RLV11 RLO1 DISKLESS TEST (RLV11 ONLY)
CZRLG	RL11/RLV11 RLO1/02 CONTROLLER TEST (PART 1)
CZRLH	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 FIVE STEPS OF EXECUTION

THIS DIAGNOSTIC PROGRAM SHOULD BE LOADED AND STARTED USING NORMAL XXDP+ PROCEDURES. START THE EXECUTION OF THE XXDP+ MONITOR BY USING THE APPROPRIATE BOOTSTRAP PROGRAM. THE MONITOR WILL PRINT A MESSAGE IDENTIFYING ITSELF AND REQUESTING THAT THE CURRENT DATE BE ENTERED. AN EXAMPLE OF THIS MESSAGE IS GIVEN BELOW FOR THE XXDP+ MONITOR:

```
CHMDKA0 XXDP+ DK MONITOR NNK
BOOTED VIA UNIT 0
ENTER DATE (DD-MMM-YY):
```

AFTER THE DATE HAS BEEN ACCEPTED BY THE MONITOR THE RESTART ADDRESS OF THE MONITOR IS PRINTED. THEN THE FOLLOWING TWO QUESTIONS ARE ASKED:

```
50 HZ ? N
LSI ? N
```

THE DEFAULTS ARE BOTH 'NO'. TYPE 'R' AND THE PROGRAM NAME TO RUN THE PROGRAM. DO NOT TYPE THE EXTENSION.

WHEN THIS DIAGNOSTIC IS STARTED THE FOLLOWING STEPS WILL OCCUR:

```
*****
* STEP 1 *
*****
```

THE DIAGNOSTIC WILL ISSUE THE PROMPT 'DR>'. 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 'DR>' 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:

PNT	PRINT NUMBER OF TEST BEING EXECUTED
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 2 \*  
\*\*\*\*\*

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 3 \*  
\*\*\*\*\*

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 4 \*  
\*\*\*\*\*

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 5 \*  
\*\*\*\*\*

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 DR>).
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 OCCURRED.

### 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 1, 2, 3, 4, AND 5 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 CZRL!D	0
DRS LOADED	D
DIAG. RUN-TIME SERVICES REV. D APR-79	D
CZRLI-D-0	D
CZRLI TESTS THE RL01-02 INTERFACE AND BASIC DRIVE LOGIC	D
UNIT IS RL01, RL02	D
DR>STA/PASS:1/FLAGS:HOE	D,0
CHANGE HW (L) ? Y	D,0
# UNITS (D) ? 2	D,0
UNIT 0	D
RL11 (L) Y ?	D,0
BUS ADDRESS (O) 174400 ?	D,0
VECTOR (O) 160 ?	D,0
DRIVE (O) 0 ?	D,0
DRIVE TYPE = RL01 (L) Y ?	D,0
BR LEVEL (O) 5 ?	D,0
UNIT 1	D
RL11 (L) Y ?	D,0
BUS ADDRESS (O) 174400 ?	D,0
VECTOR (O) 160 ?	D,0
DRIVE (O) 0 ? 1	D,0
DRIVE TYPE = RL01 (L) ? N	D,0 (N=RL02)
BR LEVEL (O) 5 ?	D,0
CHANGE SW (L) ? N	D,0
EXECUTE DRIVE SELECT TESTS (L) N ?	D,0
EXECUTE HEAD ALIGNMENT SUPPORT (L) N ?	D,0
DO MANUAL INTERVENTION TESTS (L) N ? Y	D,0
INPUT ERROR LIMIT (D) 20 ?	D,0
CZRLI HRD ERR 00004 TST 003 SUB 002 PC:004130	
ERR HLT	
DR>PRO/FLAGS:IER:LOE:HOE=0	D,0

\*\*\*\*\*  
 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
DR>CON/FLAGS:HOE:IER:LOE=0      D,0
CHANGE SW (L) ? N              D,0
CZRLI EOP 1                    D
^C
DR>RESTART/PASS:1              D,0
CHANGE SW (L) ? N              D,0
-----
-----
-----
-----

```

## 2.2 CHAIN MODE OPERATION

CHAIN MODE OPERATION CONSISTS OF THE SEQUENTIAL EXECUTION OF PROGRAMS WITHOUT OPERATOR INTERVENTION. ONLY PROGRAMS THAT HAVE BEEN MODIFIED TO RUN IN CHAIN MODE CAN BE CHAINED. CHAINABLE PROGRAMS ARE IDENTIFIED IN THE DIRECTORY BY A BIC EXTENSION. THE BIC FILES ARE CREATED BY USING THE SETUP UTILITY PROGRAM WHICH IS USED TO PARAMETERIZE THE DIAGNOSTIC PRIOR TO ITS EXECUTION. SETUP PROMPTS THE OPERATOR WITH THE HARDWARE AND SOFTWARE QUESTIONS. THE RESPONSE TO THESE QUESTIONS ARE USED TO BUILD P-TABLES. THE RESULT OF THE SETUP PROCESS IS A FILE WHICH INCLUDES THE DIAGNOSTIC WITH APPENDED P-TABLES. REFER TO THE XXDP+/SUPERVISOR USER'S MANUAL FOR A COMPLETE DESCRIPTION OF THE SETUP UTILITY.

TO RUN CHAIN MODE, THE XXDP+ MONITOR USES AN ASCII FILE (KNOWN AS A CHAIN FILE) LISTING THE PROGRAMS TO BE RUN AND THE NUMBER OF PASSES EACH PROGRAM SHOULD RUN. THIS FILE MUST BE ON THE SYSTEM DEVICE.

A CHAIN FILE MAY BE GENERATED BY USE OF THE XTECO TEXT EDITOR. THIS FILE MUST HAVE A CCC EXTENSION. THE CHAIN FILE MAY CONTAIN ANY OF THE COMMANDS SUPPORTED BY THE XXDP+ MONITOR. THE COMMANDS IN THE ASCII FILE ARE EXECUTED IN THE ORDER IN WHICH THEY ARE ENCOUNTERED.

TO EXECUTE A CHAIN FILE THE USER TYPES:

```
C FILNAM <CR> CR
```

C FILNAM/QV <CR>

IN THE FIRST CASE THE PASS COUNT SPECIFIED IN THE CHAIN FILE IS USED BY THE XXDP+ MONITOR TO DETERMINE THE NUMBER OF PASSES TO EXECUTE EACH PROGRAM. IN THE SECOND CASE THE PROGRAM COUNT IS NOT USED AND EACH PROGRAM IS EXECUTED ONLY ONCE. THE /QV SWITCH PROVIDES A SINGLE EXECUTION MODE OF OPERATION OF QUICK VERIFY.

WHEN PROGRAMS ARE RUN IN CHAIN MODE, THE HARDWARE/SOFTWARE SWITCH REGISTERS SHOULD BE SET TO 000000. THE XXDP+ MONITOR PRINTS EACH COMMAND TAKEN FROM THE CHAIN FILE AND THEN EXECUTES THE COMMAND. WHEN THE LAST COMMAND OTHER THAN ANOTHER C COMMAND HAS BEEN EXECUTED THE XXDP+ MONITOR TERMINATES CHAIN MODE AND TYPES A PROMPT (.). IT IS READY TO ACCEPT ANOTHER COMMAND FROM THE CONSOLE. IF THE LAST COMMAND IS ANOTHER C COMMAND, THE CHAIN MODE WILL CONTINUE AND THE CHAIN FILE SPECIFIED BY THIS NEW C COMMAND WILL BE USED.

IF THE USER WISHES TO TERMINATE CHAIN MODE BEFORE ITS NORMAL TERMINATION HE MAY DO SO BY TYPING A CONTROL/C. HOWEVER, THE MONITOR WILL NOT ABORT THE CHAIN MODE UNTIL IT RECEIVES PROGRAM CONTROL FROM THE PROGRAM CURRENTLY RUNNING.

## 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 EXIT
2. DIAGNOSTIC HAS FINISHED ALL ITS REQUESTED PASSES	START RESTART PRINT DISPLAY FLAGS ZFLAGS EXIT

- |    |   |  |
|----|---|--|
| 3. | OPERATOR INTERRUPTED THE<br>DIAGNOSTIC WITH CTRL/C    | START<br>RESTART<br>CONTINUE<br>PRINT<br>DISPLAY<br>FLAGS<br>ZFLAGS<br>EXIT            |
| 4. | AN ERROR WAS ENCOUNTERED<br>WITH THE HOE FLAG SET SET | START<br>RESTART<br>CONTINUE<br>PROCEED<br>PRINT<br>DISPLAY<br>FLAGS<br>ZFLAGS<br>EXIT |

### 2.3.2 COMMAND SYNTAX

-----

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

THE DIAGNOSTIC IN CORE IS EXECUTED IN ACCORDANCE WITH THE SWITCHES SPECIFIED. THE MESSAGE "% UNITS?" IS PRINTED. THE START COMMAND MAY BE ISSUED WHEN DIAGNOSTIC COMMAND MODE HAS BEEN ENTERED VIA ONE OF THE FOLLOWING: A) OPERATOR TYPED "RUN DIAGNOSTIC" B) DIAGNOSTIC FINISHED EXECUTING C) ERROR WAS ENCOUNTERED WITH HOE FLAG SET D) OPERATOR ENTERED CONTROL/C. AFTER THE OPERATOR RESPONDS TO "% UNITS?", THE HARDWARE DIALOGUE IS INITIATED. WHEN IT IS COMPLETED, THE QUESTIONS "CHANGE SW?" IS ISSUED, AND THE ANSWERS, IF GIVEN, BECOME THE NEW DEFAULTS. THEREFORE IT IS NECESSARY TO RELOAD THE PROGRAM IN ORDER TO RETURN TO THE LOAD DEFAULTS.

THE SWITCH ARGUMENTS ARE AS FOLLOWS:

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

"PASS-CNT" IS A DECIMAL NUMBER INDICATING THE DESIRED NUMBER OF PASSES. A PASS IS DEFINED AS THE EXECUTION OF THE FULL DIAGNOSTIC

(ALL SELECTED TESTS) AGAINST ALL UNITS SUBMITTED. THE DEFAULT IS NON-ENDING 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, SUB-TEST, 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

ADR EXECUTE AUTODROP CODE

LOT LOOP ON TEST

EVL EVALUATE

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 COMMAND MODE HAS BEEN ENTERED VIA A) DIAGNOSTIC IS FINISHED B) HALT ON ERROR C) CONTROL/C.

THE SWITCH ARGUMENTS ARE AS IN THE START COMMAND EXCEPT:

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

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

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

THE SWITCH ARGUMENTS ARE AS IN THE START COMMAND EXCEPT:

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

```
*****
PRO(CCEED)/FLAGS:<FLAG-LIST>
*****
```

COMMAND MODE MUST HAVE BEEN ENTERED VIA A HALT ON ERROR. THE EF-

FFECT 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

\*\*\*\*  
EXIT  
\*\*\*\*

RETURN TO XXDP+ PROMPT MODE.

\*\*\*\*\*  
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 0  
RL11 (L) Y ?  
BUS ADDRESS (O) 174400 ?

VECTOR (O) 160 ?  
 DRIVE (O) 0 ? 0-3  
 DRIVE TYPE = RL01 (L) Y ?  
 BR LEVEL (O) 5 ?

UNIT 4  
 RL11 (L) Y ?  
 BUS ADDRESS (O) 174400 ? 175400  
 VECTOR (O) 160 ? 164  
 DRIVE (O) 0 ? 0-3  
 DRIVE TYPE = RL01 (L) Y ? N  
 BR LEVEL (O) 5 ?

THE FIRST TIME THRU THE P-TABLE QUESTIONS THE DEFAULT VALUES ARE USED FOR THE CONTROLLER TYPE (QUESTION #1), CSR ADDRESS OF THE CONTROLLER (QUESTION #2), THE CONTROLLER VECTOR ASSIGNMENT (QUESTION #3), THE DRIVE TYPE (QUESTION #5), AND THE 'BR LEVEL' (QUESTION #6). THE ACTUAL UNIT NUMBERS OF THE RL01'S FOR QUESTION #4 WAS ASSIGNED 0 THRU 3 FOR THE FIRST 4 P-TABLE SLOTS.

THE SECOND TIME THRU THE P-TABLE QUESTIONS (FOR THE RL02 ASSIGNMENT ON THE SECOND CONTROLLER), THE FIRST QUESTION DEFAULTED TO 'RL11' TYPE CONTROLLER. THE SECOND QUESTION WAS ANSWERED TO REFLECT THE CHANGE IN CSR ADDRESS FOR THE RL02 CONTROLLER (175400). THE SECOND CONTROLLER'S VECTOR WAS ALSO CHANGED TO 164 IN QUESTION #3. THE RL02 TEST UNIT NUMBERS WERE ASSIGNED VALUES 0 TO 3 IN QUESTION #4 AND THE DRIVE TYPE WAS SET FOR RL02'S FOR THE REMAINING 4 UNITS IN QUESTION #5. THE LAST QUESTION WAS DEFAULTED USING THE 'BR LEVEL' FROM THE FIRST PASS.

## 2.5        HARDWARE PARAMETERS

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

RL11 (L) Y?

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

BUS ADDRESS (O) 174400?

ANSWER WITH THE BUS ADDRESS OF THE CONTROLLER.

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

CHANGE S.W. ?

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

EXECUTE DRIVE SELECT TESTS (N)?

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

EXECUTE HEAD ALIGNMENT SUPPORT (N)?

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

EXECUTE MANUAL INTERVENTION TESTS (N)?

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

SPECIFY ERROR LIMIT (DECIMAL) (20)?

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

### 3.0 ERROR INFORMATION

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

#### 3.1 ERROR REPORTING

MOST ERROR REPORTS HAVE THE FOLLOWING FORMAT.

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

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

LINE 1 IS THE ERROR HEADER AND IS PROVIDED BY THE SUPERVISOR. THE PROGRAM IS IDENTIFIED BY NAME WITH THE NUMBER OF TEST AND SUB TEST 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 SEEKS, READ ALL HEADERS FROM BOTH SURFACES, ETC.) THE SUBTEST WILL BE DESCRIPTIVE OF WHERE IN THE TEST THE ERROR OCCURRED.

THE ERROR P.C. IS THE PHYSICAL MEMORY LOCATION WHERE THE ERROR REPORT WAS

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

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

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

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

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

IN ADDITION LINE 5 WILL REPORT ANY HARDWARE DETECTED ERRORS SUCH AS OPERATION INCOMPLETE, HEADER CRC, ETC. IN THIS CASE THE FIRST LINE PRINTED AS RESULT WILL BE DETERMINED BY THE THREE ERROR BITS OPI, HNF/DLT, AND HCRC/DCRC. THE LINE WILL BE DETERMINED AS IN THE FOLLOWING TRUTH TABLE:

HNF/DLT	DCRC/HCRC	OPI	MESSAGE
1	1	1	HDR NOT FND/HDR CRC/OPI ERROR
0	1	1	HDR CRC ERROR
1	0	1	HDR NOT FND ERROR
0	1	0	DATA CRC ERROR
1	0	0	DATA LATE ERROR

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

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

LINE 8 PROVIDES THE CONTENTS OF CONTROLLER REGISTERS WHEN THE OPERATION WAS INITIATED.

LINE 9 PROVIDES THE CONTENTS OF THE CONTROLLER REGISTERS WHEN THE ERROR BEING REPORTED WAS DETECTED. FREQUENTLY THE REGISTER CONTENTS OF OP INIT

AND OP DONE WILL BE DIFFERENT. OP INIT MAY INDICATE A SEEK WAS BEING PERFORMED BUT OP DONE MAY INDICATE THE ERROR WAS DETECTED BY A READ HEADER. THE REASON IS THAT A SEEK WAS EXECUTED AND DID NOT PROPERLY POSITION HEADS AND WHEN THE READ HEADER WAS DONE THE HEADS WERE ON THE WRONG CYLINDER.

LINE 10 IS THE DRIVE STATUS. THIS LINE IS ONLY REPORTED IF THE RLMP REGISTER DOES NOT CONTAIN THE ACTUAL DRIVE STATUS.

LINE 11 AND LINE 12 ARE REPORTED IF THE ERROR WAS DETECTED AS A COMPARE OPERATION, EITHER DATA OR HEADERS. IN ADDITION, GOOD AND BAD DATA IS REPORTED FOR ALL READ ERRORS.

### 3.1.1 SPECIFIC OPERATION MESSAGES

THE OPERATION MESSAGE (LINE 4) IS GENERATED IN A DYNAMIC MANNER BASED ON THE SUBSYSTEM FUNCTION BEING EXECUTED AT THE TIME OF THE ERROR AND THE STATE OF THE FLAGS IN THE LOCATION TAGGED 'OPFLAGS'. THE POSSIBLE OPERATION MESSAGES ARE GIVEN BELOW.

SEEK -  
FROM (CYL NUM) DIFF (CYL DIFF) SGN (0 OR 1) HD (0 OR 1) WHERE THE VALUES ARE GIVEN IN OCTAL. THIS MESSAGE IS THE RESULT OF A SEEK OPERATION THAT WAS VERIFIED BY A READ HEADER AND THE HEAD POSITION AFTER A SEEK IS IN ERROR. (THE ACTUAL HEAD POSITION IN THIS ERROR SITUATION IS GIVEN IN THE RESULT LINE, LINE 5.)

READ DATA -  
IS A READ DATA OPERATION WHERE SOME FORM OF ERROR WAS DETECTED IN THE ACTUAL READ OPERATION. THIS ERROR COULD BE HARDWARE DETECTED SUCH AS DATA CRC, HEADER CRC, HEADER NOT FOUND, ETC., OR A SOFTWARE DETECTED ERROR SUCH AS DRIVE READY RESET AFTER A READ DATA COMPLETED.

READ DATA WITH DATA COMPARE -

IS AN ERROR THAT WAS DETECTED AS BAD DATA IN THE BUFFER AFTER A READ DATA OPERATION. WHEN THIS OPERATION IS REPORTED IT INDICATES THE ACTUAL READ DATA OPERATION COMPLETED WITH NO DETECTED ERRORS BUT THE DATA WAS WRONG.

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

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

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

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

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

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

THE QUALIFIERS "FOL 0 TO CC SEEK" AND "FOL 255 TO CC SEEK" INDICATE THAT  
 THE SEQUENCE OF OPERATIONS INCLUDED A SEEK OF A GIVEN DIRECTION TO THE CYL-  
 INDER 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 DE-  
 FINE 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-EXISTENT 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	(BRUSH HOME)
WRT LCK	(WRITE LOCK)
HDS OUT	(HEADS 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 TOO 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.

"DPI SET-NO DRIVE RESPONSE"

THIS IS REPORTED AS THE RESULT WHEN THE GET STATUS COMMAND IS TIMED OUT (DPI 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 SPECIFICATIONS.)

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:HOE. THERE ARE NO OTHER HALTS.

### 4.0 PERFORMANCE AND PROGRESS REPORTS

#### 4.1 PERFORMANCE REPORTS

THIS PROGRAM WILL NOT GIVE ANY PERFORMANCE REPORTS.

#### 4.2 PROGRESS REPORTS

THIS PROGRAM WILL NOT GIVE ANY PROGRESS REPORTS.

### 5.0 DEVICE INFORMATION TABLES

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

RLCS - CONTROL AND STATUS REGISTER (XXXXX0)

BIT 15 - COMPOSITE ERROR  
 BIT 14 - DRIVE ERROR  
 BIT 13 - NON EXISTANT MEMORY ERROR  
 BIT 12 - HEADER NOT FOUND (WITH BIT 10 SET)  
           - DATA LATE (WITH BIT 10 CLEAR)  
 BIT 11 - HEADER CRC (WITH BIT 10 SET)

- DATA CRC (WITH BIT 10 CLEAR)  
 BIT 10 - OPERATION INCOMPLETE  
 BIT 9/8 - DRIVE SELECT (0-3)  
 BIT 7 - CONTROLLER READY  
 BIT 6 - INTERRUPT ENABLE  
 BIT 5 - EXTENDED BUS ADDRESS (BIT 17)  
 BIT 4 - EXTENDED BUS ADDRESS (BIT 16)  
 BIT 3-1 - FUNCTION CODE  
     0 - NOP (PDP-11) MAINT (LSI-11)  
     1 - WRITE CHECK  
     2 - GET DRIVE STATUS  
     3 - SEEK  
     4 - READ HEADER  
     5 - WRITE DATA  
     6 - READ DATA  
     7 - READ WITHOUT HEADER COMPARE

BIT 0 - DRIVE READY

RLBA - BUS ADDRESS REGISTER (XXXXX2)

-----

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

RLDA - DISK ADDRESS REGISTER (XXXXX4)

-----

FOR READ/WRITE FUNCTIONS

-----

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

FOR SEEK FUNCTION

-----

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

FOR GET STATUS FUNCTION

-----

BIT 15-4 - IGNORED SHOULD BE ZERO (0)

BIT 3 - DRIVE RESET  
 BIT 2 - MUST BE ZERO (0)  
 BIT 1 - MUST BE ONE (1)  
 BIT 0 - MUST BE ONE (1)

RLMP - MULTIPURPOSE REGISTER  
 -----

FOR READ/WRITE FUNCTION  
 -----

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

FOR READ HEADER FUNCTION  
 -----

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

FOR GET STATUS FUNCTION  
 -----

HAS DRIVE STATUS

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

-----

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 OR 3 (WHICH STATE DEPENDS ON WHETHER  
THE DRIVE HAS A BRUSH). IF NOT:

BAD STATE ROM

IF THE DRIVE HAS A BRUSH, 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 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.

CHECK VOLUME CHECK IS SET. IF NOT:

BAD VOLUME CHECK LOGIC

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

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

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

#### TEST 4 HEAD UNLOADING TEST

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

REQUEST OPERATOR TO UNLOAD DRIVE.



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

BAD STATE ROM  
BAD SWITCH

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

BAD STATE ROM

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

NO BRAKING  
BAD AC SERVO

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

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

#### TEST 5 DRIVE SELECT TEST

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

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

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

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

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

#### TEST 6 DRIVE SELECT ERROR TEST

REQUEST OPERATOR INSERT IDENTICAL ADDRESS PLUGS IN TWO DRIVES

(MUST BE IDENTICAL TO NUMBER SPECIFIED EARLIER). REQUEST OPERATOR TYPE CARRIAGE RETURN WHEN READY.

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

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

DRIVE SELECT ERROR DETECTION IS BAD IN THAT DRIVE.

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

#### TEST 7 INITIAL STATE TEST

INSTRUCT OPERATOR TO GO THROUGH A LOAD HEADS CYCLE TO INITIALIZE THE 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

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

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:

BAD

RESET DETECTION, RESET ERROR, OR VOLUME CHECK FLOP

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

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

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

\*\*\*\*\*  
NOTE: THE NULL DETECTOR AND SEEK TIMEOUT SHOULD BE  
GROUNDED ON THOSE DRIVES WHICH LACK THE HEAD  
SELECT TEST POINTS. THE TEST WILL NOT SWITCH  
HEADS IF THERE IS A DRIVE FAULT.  
\*\*\*\*\*

THIS TEST SELECTS THE DRIVE UNDER TEST AND LOOPS ON A GET  
STATUS WITH RESET. THE WRITE LOCK BIT IS MONITORED AND WHEN

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

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

#### TEST 12 HEAD SWITCHING TEST

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

DIFFERENCE COUNTER IS PICKING UP BITS  
ASSOCIATED CIRCUITRY IS BAD

VERIFY DRIVE READY RESET. IF NOT:

ENABLE TIMEOUT OR READY LATCH/ONE SHOT BAD

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

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

VERIFY DRIVE ERROR DID NOT SET.

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

HEAD SELECT REGISTER BAD  
DRIVE COMMAND SHIFT REGISTER BAD

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

#### TEST 13 READ HEADER TEST (PART 1)

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

DO READ HEADER, WAIT FOR INTERRUPT.

CHECK IF HEADER CRC ERROR SET. IF SET:

READ/WRITE BOARD BAD  
READ DATA LINE BAD

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

HEADS ARE SWITCHED (CABLE)  
HEAD SELECT LOGIC

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

REPEAT TESTS USING HEAD 1.

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

HEADS ARE MISALIGNED

#### TEST 14 READ HEADER TEST (PART 2)

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

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

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

BAD READ/WRITE BOARD  
BAD PACK

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

#### 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 20 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  
INTEGRATOR 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.

a



1		000001	PART1==1
2	000000		.ENABLE ABS
3			.LIST MC
4			.NLIST MD,ME,CND,TOC
5	002000		.=2000
6			.MCALL SVC
7			
8	002000		SVC
9		000001	SVCTST=1
10		000001	SVCSUB=1
11		000001	SVCBGL=1
12		000000	SVCINS=0
13		000000	SVCTAG=0
14	002000		POINTER BGNSW,BGNSFT,BGNDU
15			
16	002000		BGNMOD MDHEDR
17	002000		HEADER CZRLI,D,0,1,0
	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	104	.ASCII /D/
	002011	060	.ASCII /O/
	002012	000000	.WORD 0
	002014	000001	.WORD 1
	002016	040376	.WORD L\$HARD
	002020	040552	.WORD L\$SOFT
	002022	014310	.WORD L\$HW
	002024	014326	.WORD L\$SW
	002026	040764	.WORD L\$LAST
	002030	000000	.WORD 0
	002032	000000	.WORD 0
	002034	000000	.WORD 0
	002036	000000	.WORD 0
	002040	014344	.WORD L\$DISPATCH
	002042	000000	.WORD 0
	002044	000000	.WORD 0
	002046	000000	.WORD 0
	002050	003	.BYTE C\$REVISION
	002051	003	.BYTE C\$EDIT
	002052	000000	.WORD 0
	002054	000000	.WORD 0
	002056	000000	.WORD 0
	002060	002212	.WORD L\$DVTYP
	002062	000000	.WORD 0
	002064	000000	.WORD 0
	002066	000000	.WORD 0
	002070	000000	.WORD 0
	002072	016364	.WORD I\$DU
	002074	000000	.WORD 0
	002076	002122	.WORD L\$DESC
	002100	104035	EMT E\$LOAD
	002102	000000	.WORD 0

002104 014412  
 002106 016174  
 002110 015636  
 002112 014404  
 002114 000000  
 002116 000000  
 002120 000000  
 18 002122

.WORD L\$INIT  
 .WORD L\$CLEAN  
 .WORD L\$AUTO  
 .WORD L\$PROT  
 .WORD 0  
 .WORD 0  
 .WORD 0

ENDMOD

19  
 20 002122  
 002122 103 132 122  
 002125 114 111 040  
 002130 124 105 123  
 002133 124 123 040  
 002136 124 110 105  
 002141 040 122 114  
 002144 060 061 055  
 002147 060 062 040  
 002152 111 116 124  
 002155 105 122 106  
 002160 101 103 105  
 002163 040 101 116  
 002166 104 040 102  
 002171 101 123 111  
 002174 103 040 104  
 002177 122 111 126  
 002202 105 040 114  
 002205 117 107 111  
 002210 103 000

DESCRIPT .ASCIZ <CZRL1 TESTS THE RL01-02 INTERFACE AND BASIC DRIVE LOGIC>  
 .ASCIZ /CZRL1 TESTS THE RL01-02 INTERFACE AND BASIC DRIVE LOGIC/

21  
 22 002212  
 002212 122 114 060  
 002215 061 054 122  
 002220 114 060 062  
 002223 000

.EVEN  
 DEVTYP <RL01,RL02>  
 .ASCIZ /RL01,RL02/

23  
 24  
 25  
 26  
 27  
 28  
 29  
 30  
 31  
 32  
 33  
 34  
 35  
 36  
 37  
 38  
 39  
 40

.EVEN  
 :COPYRIGHT (C) 1979  
 :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.

1  
2  
3 002224  
4  
5 002224

.SBTTL BIT AND OFFSET DEFINITIONS

BGNMOD GLBEQAT

EQUALS

; BIT DIFINITIONS

100000	BIT15== 100000
040000	BIT14== 40000
020000	BIT13== 20000
010000	BIT12== 10000
004000	BIT11== 4000
002000	BIT10== 2000
001000	BIT09== 1000
000400	BIT08== 400
000200	BIT07== 200
000100	BIT06== 100
000040	BIT05== 40
000020	BIT04== 20
000010	BIT03== 10
000004	BIT02== 4
000002	BIT01== 2
000001	BIT00== 1

001000	BIT9== BIT09
000400	BIT8== BIT08
000200	BIT7== BIT07
000100	BIT6== BIT06
000040	BIT5== BIT05
000020	BIT4== BIT04
000010	BIT3== BIT03
000004	BIT2== BIT02
000002	BIT1== BIT01
000001	BIT0== BIT00

; EVENT FLAG DEFINITIONS

; EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION

000040	EF.START== 32.	; START COMMAND WAS ISSUED
000037	EF.RESTART== 31.	; RESTART COMMAND WAS ISSUED
000036	EF.CONTINUE== 30.	; CONTINUE COMMAND WAS ISSUED
000035	EF.NEW== 29.	; A NEW PASS HAS BEEN STARTED
000034	EF.PWR== 28.	; A POWER-FAIL/POWER-UP OCCURRED

; PRIORITY LEVEL DEFINITIONS

000340	PRI07== 340
000300	PRI06== 300
000240	PRI05== 240
000200	PRI04== 200
000140	PRI03== 140
000100	PRI02== 100
000040	PRI01== 40
000000	PRI00== 0

;

```

; OPERATOR FLAG BITS
:
000004   EVL==      4
000010   LOT==     10
000020   ADR==     20
000040   IDU==     40
000100   ISR==    100
000200   UAM==    200
000400   BOE==    400
001000   PNT==   1000
002000   PRI==   2000
004000   IXE==   4000
010000   IBE==  10000
020000   IER==  20000
040000   LOE==  40000
100000   HOE== 100000

6
7
8   000000   : OFFSETS FOR HARDWARE P-TABLE
9   000002   CSR      =0           ;BUS ADDRESS
10  000004   VECT     =2           ;VECTOR ADDRESS
11  000006   PRIOR    =4           ;PRIORITY
12  000010   TYPDR    =6           ;DRIVE TYPE
13  000012   DRSB     =10          ;DRIVE SELECT
14  000012   CNT      =12          ;CONTROLLER TYPE

15
16  000000   : OFFSETS FOR SOFTWARE P-TABLE
17  000002   MISWI    =0           ;SOFTWARE PARAMETERS SWITCHES
18  000004   LOLIM    =2           ;CYLINDER LOWER LIMIT
19  000006   HILIM    =4           ;CYLINDER HIGH LIMIT
20  000010   HEAD     =6           ;SELECTED HEAD FOR RUNNING TESTS
21  000012   ERLIM    =10          ;ERROR LIMIT
22  000012   DCLIM    =12          ;DATA COMPARE ERROR LIMIT

23
24  000001   : BIT ASSIGNMENTS FOR SOFTWARE P-TABLE SWITCHES
25  000002   ALLCYL   =BIT00        ;USE ALL CYLINDERS
26  000004   ALLSEC   =BIT01        ;USE ALL SECTORS
27  000010   DRSELT   =BIT02        ;EXECUTE DRIVE SELECT TEST
28  010000   HDALIGN  =BIT03        ;EXECUTE HEAD ALIGNMENT TEST
29  020000   HEADLM   =BIT12        ;HEAD LIMIT SPECIFIED FLAG
30  040000   HICYL   =BIT13        ;HI LIMIT SPECIFIED FLAG
31  040000   LOCYL   =BIT14        ;LO LIMIT SPECIFIED
32  100000   MITEST   =BIT15        ;EXECUTE MANUAL INTERVENTION TESTS

33
34  000102   : SUBSYSTEM FUNCTIONS
35  000104   CKDATA   =102        ;WRITE CHECK
36  000106   GTSTAT   =104        ;GET STATUS
37  000110   SEEK     =106        ;SEEK
38  000112   RDHEAD   =110        ;READ HEADER
39  000114   WTLATA   =112        ;WRITE DATA
40  000116   RDDATA   =114        ;READ DATA
41  000116   RDNOHR   =116        ;READ DATA, IGNORE HEADERS
42  000100   NOOP     =100        ;NO OPERATION

43
44  007777   : OPERATION FLAGS
45  000002   COMPOP   =7777        ;COMPOSITE OPERATION FLAGS
46  000001   HDRCMP   =BIT01        ;HEADER COMPARE OPERATION
      000001   DATACMP  =BIT00        ;DATA COMPARE OPERATION
    
```

47	000004	CYLP	=BIT02	:CYCLE UP OPERATION
48	000010	ULOAD	=BIT03	:UNLOAD OPERATION
49	000020	INOUTS	=BIT04	:IN-OUT SEEK OPERATION
50	000040	OUTINS	=BIT05	:OUT-IN SEEK OPERATION
51	000100	FOLWRT	=BIT06	:FOLLOWING WRITE OPERATION
52	000200	REVSKE	=BIT07	:REV SEEK SEQ (ADJ INTERFERENCE)
53	000400	FWDSKE	=BIT08	:FWD SEEK SEQ (ADJ INTERFERENCE)
54	001000	REVSKO	=BIT09	:REV SEEK SEQ (OVERWRITE)
55	002000	FWDSKO	=BIT10	:FWD SEEK SEQ (OVERWRITE)
56	004000	BADADD	=BIT11	:BAD DISK ADDRESS
57	010000	SEEKOP	=BIT12	:SEEK OPERATION
58	020000	RORWOP	=BIT13	:READ OR WRITE OPERATION
59	040000	RELDWT	=BIT14	:RELOAD WAIT
60	100000	HDR40	=BIT15	:40 HEADER OPERATION
61	003760	MQUALS	=OUTINS!INOUTS!FOLWRT!REVSKE!FWDSKE!REVSKO!FWDSKO	:MESSAGE QUALIFIER BITS
62				
63				
64		:	ERROR FLAGS FROM SUBROUTINES	
65	000001	TOSLOW	=BIT00	:OPERATION TOOK TOO LONG
66	000002	NOIRPT	=BIT01	:NO INTERRUPT FROM OPERATION
67	000004	CONHNG	=BIT02	:CONTROLLER HUNG
68	000010	NOCLR	=BIT03	:BAD CONTROLLER CLEAR
69				
70	000000	RLCS	=0	:CONTROL AND STATUS REGISTER
71	000002	RLBA	=2	:BUS ADDRESS REGISTER
72	000004	RLDA	=4	:DISK ADDRESS REGISTER
73	000006	RLMP	=6	:MULTI-PURPOSE REGISTER
74				
75		:	REGISTER BIT DEFINITIONS - CONTROL STATUS REGISTER	
76	000000	RLCSP	=0	:CONTROL AND STATUS REGISTER
77	100000	ANYERR	=100000	:ANY ERROR BIT
78	040000	DRVERR	=40000	:DRIVE ERROR BIT
79	020000	NXMERR	=20000	:NON-EXISTENT MEMORY ERROR
80	010000	DLTERR	=10000	:DATA LATE ERROR
81	010000	HNFERR	=10000	:HEADER NOT FOUND ERROR
82	004000	DCKERR	=4000	:DATA CHECK ERROR
83	004000	HPCERR	=4000	:HEADER CHECK ERROR
84	002000	OPIERR	=2000	:OPERATION INCOMPLETE ERROR
85	001400	DSMSK	=1400	:DRIVE SELECT MASK
86	000200	CRDYMSK	=200	:CONTROLLER READY MASK
87	000100	INTEBL	=100	:INTERRUPT ENABLE MASK
88	000060	BAMSK	=60	:BUS ADDRESS UPPER MASK
89	000001	DRDYMSK	=1	:DRIVE READY MASK
90				

```
1          : REGISTER BIT DEFINITIONS - DISK ADDRESS FOR DATA XFER
2          :SAMSK =77 ;SECTOR ADDRESS MASK
3          000077 :HSMSK =100 ;HEAD SELECT MASK
4
5          : REGISTER BIT DEFINITIONS - DISK ADDRESS FOR SEEK
6          :MBSETO =1 ;MUST BE SET, BIT 0
7          000001 :DIRBIT =4 ;DIRECTION BIT
8          000004 :HDSEL =20 ;HEAD SELECT BIT
9          000020
10         : REGISTER BIT DEFINITIONS - DISK ADDRESS FOR GET STATUS
11         :GETSTAT =3 ;GET STATUS SETUP
12         000003 :DRSET =10 ;DRIVE RESET MASK
13         000010
14         : REGISTER BIT DEFINITIONS - MP FOR DATA XFER
15         :WCMSK =17777 ;WORD COUNT MASK
16         017777 :WCRNG =160000 ;WORD COUNT RANGE MASK
17         160000
18         : REGISTER BIT DEFINITIONS - MP FOR READ HEADER
19         :HDSEC =77 ;SECTOR MASK
20         000077 :HDHSEL =100 ;HEAD SELECT MASK
21         000100
22         : REGISTER BIT DEFINITIONS - MP FOR GET STATUS
23         :STAMSK =7 ;STATE MASK
24         000007 :BHSTAT =10 ;BRUSH HOME STATUS
25         000010 :HOSTAT =20 ;HEADS OUT STATUS
26         000020 :COSTAT =40 ;COVER OPEN STATUS
27         000040 :HSSSTAT =100 ;HEAD SELECT STATUS
28         000100 :DSESTAT =400 ;DRIVE SELECT ERROR STATUS
29         000400 :VCSTAT =1000 ;VOLUME CHECK STATUS
30         001000 :WGESTAT =2000 ;WRITE GATE ERROR STATUS
31         002000 :SPDSTAT =4000 ;SPIN ERROR STATUS
32         004000 :STOSTAT =10000 ;SEEK TIMEOUT ERROR STATUS
33         010000 :WLSTAT =20000 ;WRITE LOCK STATUS
34         020000 :HCESTAT =40000 ;HEAD CURRENT ERROR STATUS
35         040000 :WDESTAT =100000 ;WRITE DATA ERROR STATUS
36         100000
37         002224 ENDMOD
38
39
```

1  
2  
3  
4  
5  
6  
17  
18  
19  
20  
24  
25  
26  
27  
62

.SBTTL MACRO DEFINITIONS

:DELAY EXECUTION OF PROGRAM A SPECIFIED NUMBER OF 100-MILLISECOND TIME COUNTS.  
:THIS TIMING IS PERFORMED BY SOFTWARE USING CPU TIMING AND IS HIGHLY MACHINE  
:DEPENDENT.

:DELAY EXECUTION OF PROGRAM A SPECIFIED NUMBER OF 100-MICROSECOND TIME COUNTS.  
:THIS TIMING IS PERFORMED BY SOFTWARE USING CPU TIMING AND IS HIGHLY MACHINE  
:DEPENDENT.

:DELAY EXECUTION OF PROGRAM A SPECIFIED NUMBER OF 100-MICROSECOND TIME COUNTS  
:USING A KW11-P PROGRAMMABLE CLOCK OR A LINE CLOCK. THE TIME DELAY IS INVALID  
:IF TOO LARGE AN ARGUMENT IS USED WITH THE LINE CLOCK.

```

1
2
3      .SBTTL  GLOBAL DATA AND CONSTANTS
4
5      BGNMOD  GLBDAT
6
7      ;
8      ; OPMSGS:  TABLE OF OPERATION MESSAGES
9      ;          .WORD  0          ; FILLER
10     ;          .WORD  MWRCHK     ; MESSAGE FOR WRITE CHECK
11     ;          .WORD  MGTSTA     ; GET STATUS
12     ;          .WORD  MSEEK      ; SEEK
13     ;          .WORD  MREADH     ; READ HEADER
14     ;          .WORD  MWRITE     ; WRITE DATA
15     ;          .WORD  MREAD      ; READ DATA
16     ;          .WORD  MWRSET     ; WITH RESET
17     ;          .WORD  MDATCP     ; WITH DATA COMPARE
18     ;          .WORD  MHDRCP     ; WITH HEADER COMPARE
19     ;          .WORD  MCYLUP     ; LOAD HEADS
20     ;          .WORD  MLOAD      ; UNLOAD HEADS
21     ;          .WORD  MINOUT     ; IN-OUT SEQ
22     ;          .WORD  MOUTIN     ; OUT-IN SEQ
23     ;          .WORD  MFOLWRT    ; FOLLOWING WRITE
24     ;          .WORD  MREVSK     ; REV SEEK
25     ;          .WORD  MFWDSK     ; FWD SEEK
26     ;          .WORD  MRESKO     ; REV SEEK
27     ;          .WORD  MFWSKO     ; FWD SEEK
28     ;          .WORD  MBADAD     ; BAD DISK ADD FOR WRITE
29     ;          .WORD  M4OHDR     ; 40 HEADER OPERATION
30
31     T.DRIVE: .WORD  0
32     JJJ:     .WORD  0
33     HLMTW:   .WORD  0
34     CLRBYT:  .WORD  0
35     NXTHL:   .WORD  0
36     GBND:    .WORD  0
37     CAMSK:   .WORD  0
38     DIRMSK:  .WORD  0
39     HDCYL:   .WORD  0
40
41     ;
42     ; RESTBL:  TABLE OF RESULT NAME MESSAGE ADDRESSES
43     ;          .WORD  MCERR       ; CONTROLLER ERROR
44     ;          .WORD  MDRERR     ; DRIVE ERROR
45     ;          .WORD  MNEERR     ; NON-EXISTENT MEMORY ERROR
46     ;          .WORD  MFLERR     ; HEADER NOT FOUND-DATA LATE
47     ;          .WORD  MHDERR     ; HEADER OR DATA ERROR
48     ;          .WORD  MOPERR     ; OPERATION INCOMPLETE
49     ;          .WORD  MWDRST     ; NO DRIVE STATUS AVAILABLE
50     ;          .WORD  0
51     ;          .WORD  MWDERR     ; WRITE DATA ERROR
52     ;          .WORD  MHCERR     ; HEAD CURRENT ERROR
53     ;          .WORD  0
54     ;          .WORD  MSTERR     ; SEEK TIMEOUT ERROR
55     ;          .WORD  MSPERR     ; SPINDLE ERROR
56     ;          .WORD  MWGERR     ; WRITE GATE ERROR
57     ;          .WORD  0
58     ;          .WORD  MDSERR     ; DRIVE SELECT ERROR

```



```

1
2
3 002360 004764      ; PATTBL: PATTERN TABLE
4 002362 004766      .WORD PAT1
5 002364 005026      .WORD PAT2
6 002366 005066      .WORD PAT3
7 002370 005126      .WORD PAT4
8 002372 005134      .WORD PAT5
9 002374 005174      .WORD PAT6
10 002376 005176     .WORD PAT7
11 002400 005236     .WORD PAT8
12 002402 005240     .WORD PAT9
13
14
15
16 002404 000000     ; SUBSTK: SUBROUTINE CALLING STACK
17 002406 000000     .WORD 0 ;STACK IS 12 WORDS LONG
18 002410 000000     .WORD 0
19 002412 000000     .WORD 0
20 002414 000000     .WORD 0
21 002416 000000     .WORD 0
22 002420 000000     .WORD 0
23 002422 000000     .WORD 0
24 002424 000000     .WORD 0
25 002426 000000     .WORD 0
26
27
28 002430 000002     ; RL01 TABLE OF CYLINDERS
29 002432 000006     T25TBL: .WORD 2 ;TABLE OF DIFFERENCES
30 002434 000011     .WORD 6
31 002436 000014     .WORD 9.
32 002440 000021     .WORD 12.
33 002442 000026     .WORD 17.
34 002444 000033     .WORD 22.
35 002446 000042     .WORD 27.
36 002450 000051     .WORD 34.
37 002452 000200     .WORD 41.
38 002454 000377     .WORD 128.
39
40
41 002456 000004     ; RL02 TABLE OF CYLINDERS
42 002460 000014     T25TBL2: .WORD 4
43 002462 000022     .WORD 12.
44 002464 000030     .WORD 18.
45 002466 000042     .WORD 24.
46 002470 000054     .WORD 34.
47 002472 000066     .WORD 44.
48 002474 000104     .WORD 54.
49 002476 000122     .WORD 68.
50 002500 000400     .WORD 82.
51 002502 000777     .WORD 256.
52
53
54
55 002504            ; TABLE TO BE USED TO BUILD AND STORE THE CYLINDERS
56 002544            T33TBL: .BLKW 16.
57
                    TBT: .BLKW 16.
    
```

Line	Address	Value	Label	Unit	Description
58					
59	002604	002	CYLtbl:	.BYTE	2
60	002605	007		.BYTE	7.
61	002606	016		.BYTE	14.
62	002607	024		.BYTE	20.
63	002610	033		.BYTE	27.
64	002611	041		.BYTE	33.
65	002612	046		.BYTE	38.
66	002613	055		.BYTE	45.
67	002614	064		.BYTE	52.
68	002615	072		.BYTE	58.
69	002616	101		.BYTE	65.
70	002617	110		.BYTE	72.
71	002620	115		.BYTE	77.
72	002621	124		.BYTE	84.
73	002622	133		.BYTE	91.
74	002623	141		.BYTE	97.
75	002624	146		.BYTE	102.
76	002625	154		.BYTE	108.
77	002626	161		.BYTE	113.
78	002627	170		.BYTE	120.
79	002630	177		.BYTE	127.
80	002631	206		.BYTE	134.
81	002632	213		.BYTE	139.
82	002633	222		.BYTE	146.
83	002634	230		.BYTE	152.
84	002635	235		.BYTE	157.
85	002636	244		.BYTE	164.
86	002637	252		.BYTE	170.
87	002640	261		.BYTE	177.
88	002641	270		.BYTE	184.
89	002642	275		.BYTE	189.
90	002643	303		.BYTE	195.
91	002644	312		.BYTE	202.
92	002645	317		.BYTE	207.
93	002646	326		.BYTE	214.
94	002647	334		.BYTE	220.
95	002650	343		.BYTE	227.
96	002651	352		.BYTE	234.
97	002652	361		.BYTE	241.
98	002653	367		.BYTE	247.
99	002654	375		.BYTE	253.
100	002655	000		.BYTE	0
101	002656	000401		.WORD	257.
102	002660	000406		.WORD	262.
103	002662	000415		.WORD	269.
104	002664	000423		.WORD	275.
105	002666	000432		.WORD	282.
106	002670	000445		.WORD	293.
107	002672	000454		.WORD	300.
108	002674	000463		.WORD	307.
109	002676	000471		.WORD	313.
110	002700	000500		.WORD	320.
111	002702	000507		.WORD	327.
112	002704	000514		.WORD	332.
113	002706	000523		.WORD	339.
114	002710	000532		.WORD	346.

;TABLE OF DEFAJLT CYLINDERS

```

115 002712 000540 .WORD 352.
116 002714 000545 .WORD 357.
117 002716 000553 .WORD 363.
118 002720 000560 .WORD 368.
119 002722 000567 .WORD 375.
120 002724 000576 .WORD 382.
121 002726 000605 .WORD 389.
122 002730 000612 .WORD 394.
123 002732 000621 .WORD 401.
124 002734 000627 .WORD 407.
125 002736 000634 .WORD 412.
126 002740 000643 .WORD 419.
127 002742 000651 .WORD 425.
128 002744 000660 .WORD 432.
129 002746 000667 .WORD 439.
130 002750 000674 .WORD 444.
131 002752 000702 .WORD 450.
132 002754 000711 .WORD 457.
133 002756 000716 .WORD 462.
134 002760 000725 .WORD 469.
135 002762 000733 .WORD 475.
136 002764 000742 .WORD 482.
137 002766 000751 .WORD 489.
138 002770 000760 .WORD 496.
139 002772 000766 .WORD 502.
140 002774 000774 .WORD 508.
141 002776 000774 .WORD 508.
142 003000 000000 .WORD 0
143 003002 000000 SSINDX: .WORD 0 ;SUBROUTINE STACK INDEX POINTER
144
145 ; OPERATIONAL FLAGS
146 003004 000000 OPFLAG: .WORD 0 ;OPERATION FLAGS
147 003006 000000 DONE: .WORD 0 ;OPERATION COMPLETE FLAG
148 003010 000000 HADONE: .WORD 0 ;HEAD ALIGNMENT DONE FLAG
149 003012 000000 ERHEAD: .WORD 0 ;ADDRESS OF ERROR HEADER
150 003014 000000 MORECE: .WORD 0 ;MORE THAN 1 COMPARE ERROR
151 003016 000000 ERPSWI: .WORD 0 ;ERROR RETURN SWITCH
152 003020 000000 BSFLAG: .WORD 0 ;BAD SECTOR FLAGS
153 003022 000000 WRTSWI: .WORD 0 ;WRITE SWITCH
154 003024 000000 TBLSTR: .WORD 0 ;TABLE STORAGE
155
156 003026 000000 RLBAS: .WORD 0 ;RL11 BASE ADDRESS
157 003030 000000 RLVEC: .WORD 0 ;RL11 VECTOR ADDRESS
158 003032 000000 RLDRV: .WORD 0 ;DRIVE NUMBER UNDER TEST
159
160 003034 000000 L.CS: .WORD 0 ;CONTROLLER REGISTER STORAGE
161 003036 000000 L.BA: .WORD 0 ;BEFORE OPERATION
162 003040 000000 L.DA: .WORD 0
163 003042 000000 L.MP: .WORD 0
164 003044 000000 T.CS: .WORD 0 ;CONTROLLER REGISTER STORAGE
165 003046 000000 T.BA: .WORD 0 ; AFTER OPERATION
166 003050 000000 T.DA: .WORD 0
167 003052 000000 T.MP:
168 003052 000000 HDWRD1: .WORD 0 ;HEADER WORD STORAGE
169 003054 000000 HDWRD2: .WORD 0
170 003056 000000 HDWRD3: .WORD 0
171
    
```

```

172 003060 000000 T.STAT: .WORD 0 ;DRIVE STATE STORAGE
173
174 003062 000000 RESPARM: .WORD 0 ;PARAM BLOCK FOR REASON REPORT
175 003064 000000 .WORD 0
176 003066 000000 .WORD 0
177 003070 000000 .WORD 0
178 003072 000000 .WORD 0
179
180 003074 000000 DRVCNT: .WORD 0 ;DRIVE COUNT FOR DRIVES UNDER TEST
181 003076 000000 DIFAUG: .WORD 0 ;DIFFERENCE ARGUMENT FOR SEEK
182 003100 000000 OLDCYL: .WORD 0 ;OLD CYLINDER
183 003102 000000 NEWCYL: .WORD 0 ;NEW CYLINDER
184 003104 000000 CURCYL: .WORD 0 ;CURRENT CYLINDER
185 003106 000000 DESDIF: .WORD 0 ;DESIRED DIFFERENCE
186 003110 000000 DESSGN: .WORD 0 ;DESIRED SIGN
187 003112 000000 DESHD: .WORD 0 ;DESIRED HEAD
188 003114 000000 DESSEC: .WORD 0 ;DESIRED SECTOR
189 003116 000000 TEMPO: .WORD 0 ;TEMPORARY STORAGE
190 003120 000000 TEMP1: .WORD 0 ;TEMPORARY STARGAGE
191 003122 000000 TEMP2: .WORD 0 ;TEMPORARY STORAGE
192 003124 000000 TEMP3: .WORD 0 ;TEMPORARY STORAGE
193 003126 000000 TEMP4: .WORD 0 ;TEMPORARY STGRAGE
194 003130 000000 TEMP5: .WORD 0 ;TEMPORARY STORAGE
195 003132 000000 TEMP6: .WORD 0 ;TEMPORARY STORAGE
196 003134 000000 TEMP7: .WORD 0 ;TEMPORARY STORAGE
197 003136 000000 TEMP8: .WORD 0 ;TEMPORARY STORAGE
230 003140 000004 ERRVEC: .WORD 4 ;ERROR VECTOR
231 003142 000000 DLYCNT: .WORD 0 ;DELAY COUNTER USED IN TIMING MACROS
232 003144 000000 CLKFLG: .WORD 0 ;FLAG INDICATING PRESENCE OF A L OR P CLOCK
233 003146 000000 CLKADR: .WORD 0 ;POINTER TO DIAGNOSTIC MONITOR CLOCK TABLE
234 003150 000000 LBASE: .WORD 0 ;L CLOCK ITERATION NUMBER TO FAKE P CLOCK
235
236 ; MISCELLANEOUS COUNTERS
237 003152 000000 PASCNT: .WORD 0 ;PASS COUNTER (LOCAL TO A TEST)
238 003154 000000 COUNT: .WORD 0 ;A COUNTER (LOCAL TO A TEST)
239 003156 000000 ERRPOINT: .WORD 0 ;ERROR POINTER
240 003160 000000 ERPCNT: .BLKW 64. ;ERROR COUNTER FOR PROGRAM
241 003360 000000 PASNUM: .WORD 0 ;PASS NUMBER FOR PROGRAM
242 003362 000000 PSETNM: .WORD 0 ;COUNTER FOR PARAMETER SET NUMBER IN USE
243 003364 000 LOCERR: .BYTE 0 ;LOCAL ERROR COUNTER
244 003365 000 NOERCT: .BYTE 0 ;INHIBIT ERROR COUNTING FLAG
245 003366 000000 TRPFLG: .WORD 0 ;HARDWARE TRAP FLAG
246 003370 000000 PWRFLG: .WORD 0 ;POWER FAILURE FLAG
247
248 ; BAD SECTOR TABLES AND POINTERS
249 003372 000000 BSFVAL: .WORD 0 ;BAD SECTORS FILES VALID FLAG
250
251 003374 000000 SBSFIL: .BLKW 76 ;SOFTWARE BAD SECTOR FILE
252 003570 000000 FBSFIL: .BLKW 76 ;FACTORY BAD SECTOR FILE
253
254 003764 000000 IBUFF: .BLKW 200 ;INPUT BUFFER
255 004364 000000 OBUFF: .BLKW 200 ;OUTPUT BUFFER
256
257 004764 000000 PAT1: .WORD 0 ;PATTERN 1 (ALL ZEROS)
258 004766 177772 PAT2: .WORD 177772
259 004770 177777 .WORD 177777
260 004772 177777 .WORD 177777
    
```

261	004774	052525	.WORD	052525
262	004776	052525	.WORD	052525
263	005000	052525	.WORD	052525
264	005002	177777	.WORD	177777
265	005004	177777	.WORD	177777
266	005006	052525	.WORD	052525
267	005010	052525	.WORD	052525
268	005012	177777	.WORD	177777
269	005014	052525	.WORD	052525
270	005016	177252	.WORD	177252
271	005020	177252	.WORD	177252
272	005022	172765	.WORD	172765
273	005024	172765	.WORD	172765
274				
275	005026	000003	PAT3: .WORD	000003
276	005030	000000	.WORD	000000
277	005032	000000	.WORD	000000
278	005034	177777	.WORD	177777
279	005036	177777	.WORD	177777
280	005040	177777	.WORD	177777
281	005042	000000	.WORD	000000
282	005044	000000	.WORD	000000
283	005046	177777	.WORD	177777
284	005050	177777	.WORD	177777
285	005052	000000	.WORD	000000
286	005054	177777	.WORD	177777
287	005056	000000	.WORD	000000
288	005060	177777	.WORD	177777
289	005062	000000	.WORD	000000
290	005064	177777	.WORD	177777
291				
292	005066	025252	PAT4: .WORD	025252
293	005070	052525	.WORD	052525
294	005072	052525	.WORD	052525
295	005074	125252	.WORD	125252
296	005076	125252	.WORD	125252
297	005100	125252	.WORD	125252
298	005102	052525	.WORD	052525
299	005104	052525	.WORD	052525
300	005106	125252	.WORD	125252
301	005110	125252	.WORD	125252
302	005112	052525	.WORD	052525
303	005114	125252	.WORD	125252
304	005116	052525	.WORD	052525
305	005120	125252	.WORD	125252
306	005122	052525	.WORD	052525
307	005124	125252	.WORD	125252
308				
309	005126	155555	PAT5: .WORD	155555
310	005130	133333	.WORD	133333
311	005132	066666	.WORD	066666
312				
313	005134	121105	PAT6: .WORD	121105
314	005136	150442	.WORD	150442
315	005140	064221	.WORD	064221
316	005142	132110	.WORD	132110
317	005144	055044	.WORD	055044

318 005146 026442 .WORD 026442  
 319 005150 013211 .WORD 013211  
 320 005152 105504 .WORD 105504  
 321 005154 042642 .WORD 042642  
 322 005156 021321 .WORD 021321  
 323 005160 110550 .WORD 110550  
 324 005162 044264 .WORD 044264  
 325 005164 022132 .WORD 022132  
 326 005166 011055 .WORD 011055  
 327 005170 104426 .WORD 104426  
 328 005172 042213 .WORD 042213

329  
 330 005174 177777 PAT7: .WORD 177777

331  
 332 005176 045513 PAT8: .WORD 045513  
 333 005200 122645 .WORD 122645  
 334 005202 151322 .WORD 151322  
 335 005204 064551 .WORD 064551  
 336 005206 132264 .WORD 132264  
 337 005210 055132 .WORD 055132  
 338 005212 026455 .WORD 026455  
 339 005214 113226 .WORD 113226  
 340 005216 045513 .WORD 045513  
 341 005220 122645 .WORD 122645  
 342 005222 151322 .WORD 151322  
 343 005224 064551 .WORD 064551  
 344 005226 132264 .WORD 132264  
 345 005230 055132 .WORD 055132  
 346 005232 026455 .WORD 026455  
 347 005234 113226 .WORD 113226

348  
 349 005236 125252 PAT9: .WORD 125252

350  
 351 005240 155555 PAT10: .WORD 155555

352  
 353 005242 ENDMOD

354

355 .SBTTL GLOBAL MESSAGES

356  
 357  
 361 005242 BGNMOD GLBXTT  
 362 005242 123 105 105 MSEEK: .ASCIZ /SEEK /  
 363 005250 122 104 040 MREAD: .ASCIZ /RD DATA /  
 364 005261 122 104 040 MREADH: .ASCIZ /RD HDR /  
 365 005271 127 122 124 MWRCHK: .ASCIZ /WRT CHECK /  
 366 005303 127 122 124 MWRITE: .ASCIZ /WRT DATA /  
 367 005315 107 105 124 MGTSTA: .ASCIZ /GET STAT /  
 368 005327 127 111 124 MDATECP: .ASCIZ /WITH DATA CMP /  
 369 005346 127 111 124 MHDRCP: .ASCIZ /WITH HDR CMP /  
 370 005364 106 117 122 M40HDR: .ASCIZ /FOR 40 HDRS /  
 371 005400 127 111 124 MWRSET: .ASCIZ /WITH RESET /  
 372 005414 117 120 105 MOPER: .ASCIZ /OPER: /  
 373 005423 122 105 123 MRSLT: .ASCIZ /RESULT: /  
 374 005434 125 116 114 MLOAD: .ASCIZ /UNLD DRV /  
 375 005445 114 104 040 MCYLUP: .ASCIZ /LD DRV /  
 376 005455 106 117 114 MOUTIN: .ASCIZ /FOL 0 TO CC SEEK /  
 377 005476 106 117 114 MINOUT: .ASCIZ /FOL 255 TO CC SEEK /

378	005521	106	117	114	MFOLWRT: .ASCIZ /FOL WRT (NO SEEK)/
379	005543	101	104	112	MREVSK: .ASCIZ /ADJ CYL WRTTN AFTER REV SK/
380	005576	101	104	112	MFWDSK: .ASCIZ /ADJ CYL WRTTN AFTER FWD SK/
381	005631	123	113	040	MFWSKO: .ASCIZ /SK FWD,WRT - SK REV,OVERWRT/
382	005665	123	113	040	MRESKO: .ASCIZ /SK REV,WRT - SK FWD,OVERWRT/
383	005721	117	116	040	MBADAD: .ASCIZ /ON BAD SEC FILES/
384	005742	103	101	116	MBADSF: .ASCIZ /CANNOT GET BAD SEC FILES/
385	005773	102	101	104	MFMTERR: .ASCIZ /BAD SEC FILE FMT ERR/
386	006020	124	117	117	MTMBS: .ASCIZ /TOO MANY BAD SEC /
387	006042	102	125	123	BASADD: .ASCIZ /BUS ADD=/
388	006053	104	122	126	DRVNAM: .ASCIZ /DRV=/
389	006060	104	122	126	NOPIR: .ASCIZ /DRV DID NOT REC'R FROM PWR FAIL/
390	006120	122	114	103	CSNAM: .ASCIZ /RLCS/
391	006125	122	114	102	BANAM: .ASCIZ /RLBA/
392	006132	122	114	104	DANAM: .ASCIZ /RLDA/
393	006137	122	114	115	MPNAM: .ASCIZ /RLMP/
394	006144	117	120	040	LAB1: .ASCIZ /OP INIT = /
395	006157	117	120	040	LAB2: .ASCIZ /OP DONE = /
396	006172	127	117	122	MWORD: .ASCIZ /WORD /
397	006200	111	116	124	MTOSLOW: .ASCIZ /INTRPT TOO LATE/
398	006220	116	117	040	MDRRES: .ASCIZ /NO DRV RESPONSE/
399	006240	116	117	040	MNOINT: .ASCIZ /NO INTRPT ON CMND COMPLETE/
400	006273	103	116	124	MCONHNG: .ASCIZ /CNTLR HUNG /
401	006307	105	122	122	MNOCLR: .ASCIZ /ERR DID NOT CLR/
402	006327	126	117	114	VCNRST: .ASCIZ /VOL CHK NOT RSET/
403	006350	125	116	130	UNXERR: .ASCIZ /UNXPCTED ERR/
404	006365	040	124	105	TSTLAB: .ASCIZ / TEST/
406	006373	115	101	116	MISTST: .ASCIZ /MAN INTERVENT STAT/
407	006416	123	124	101	NSTACHG: .ASCIZ /STATE CHG/
408	006430	123	120	116	SPDERR: .ASCIZ /SPNDL TIMEOUT FAILED TO SET/
409	006464	106	101	111	GSTER1: .ASCIZ /FAIL FORCING DRV SEL ERR/
410	006515	111	116	111	INITST: .ASCIZ /INIT STATE/
411	006530	104	122	126	T05ERR: .ASCIZ /DRV SELECT/
412	006543	104	122	126	T09ERR: .ASCIZ /DRV RDY/
413	006553	123	105	105	T10ERR: .ASCIZ /SEEK SGN SWITCH/
414	006573	110	104	040	T12ERR: .ASCIZ /HD SWITCH/
415	006605	122	104	040	T13ERR: .ASCIZ /RD HDR (P1)/
416	006621	122	104	040	T14ERR: .ASCIZ /RD HDR (P2)/
417	006635	127	122	124	T16ERR: .ASCIZ /WRT LCK/
418	006645				P2T01E: .ASCIZ
419	006645	104	111	106	P2T02E: .ASCIZ /DIFF OF 1 SEEK/
420	006664	124	105	123	NOTST: .ASCIZ /TEST CANNOT BE PERFORMED...NO P CLOCK OR SOFTWARE CLOCK/
421	006754	104	122	126	NOCTLR: .ASCIZ /DRV DROPPED - NO CNTLR/
422	007003	104	122	126	NOTRDY: .ASCIZ /DRV DROPPED - NOT RDY/

8	007031	110	104	123	HDMOVF: .ASCIZ	/HDS FAILED TO MOVE IN 10 TRIES/
10	007070	103	131	114	CYLPDR: .ASCIZ	/CYL PORTION OF HDRS DIFFER WHEN READ FROM TRK 0 & 1/
11	007154	110	105	101	HAMES1: .ASCIZ	/HEAD ALIGN. RSET WRT LCK TO SEL HD 0, SET FOR HD 1/
12	007237	124	131	120	HAMES2: .ASCIZ	&TYPE 'CTL/C' TO GET BACK TO SUPVR CMD MODE AND THEN TYPE 'CONT' &
13	007343	111	106	040	HAMES3: .ASCIZ	/IF HD SEL TP (21, 22) DO NOT EXIST/
14	007406	107	116	104	HAMES4: .ASCIZ	/GND NULL DET ON DRV LGC MOD DISABLE SEEK TIME OUT/
15	007470	101	102	117	OPR002: .ASCIZ	/ABOVE CONDITIONS MET/
16	007515	127	101	123	OPR003: .ASCIZ	/WAS LOAD DEPRESSED/
17	007540	103	110	113	OPR1: .ASCIZ	/CHK DRV IS UNLDED, COVER OPN, AND WRTE LCKED /
18	007616	103	114	117	OPR2: .ASCIZ	/CLOSE COVER & RST WRT LCK /
19	007651	120	122	105	OPR3: .ASCIZ	/PRESS LOAD /
20	007665	120	122	105	OPR6: .ASCIZ	/PRESS LOAD & WAIT FOR RDY /
21	007720	122	105	115	OPR7: .ASCIZ	/REMOVE ADR PLGS EXCPT /
22	007747	111	116	123	OPR8: .ASCIZ	/INSRT ADR PLG /
23	007766	111	116	040	OPR9: .ASCIZ	/IN ALL DRVS /
24	010003	111	116	123	OPR10: .ASCIZ	/INSUFFICIENT DRVS FOR DRV SEL ERR TST/
25	010051	122	120	114	OPR11: .ASCIZ	/RPLCE ADR PLGS AS BEFORE/
27	010102	122	105	123	OPR12: .ASCIZ	/RESET WRT LCK /
28	010121	123	105	124	OPR12A: .ASCIZ	/SET WRT LCK/
29	010135	117	116	040	OPR1A: .ASCIZ	/ON /
30	010141	117	116	040	OPR1B: .ASCIZ	/ON DRV /
31	010151	125	116	104	UNDTST: .ASCIZ	/UNDER TEST/
32	010164	123	105	124	OPR004: .ASCIZ	/SET WRT LCK /
33	010201	104	111	106	DIFWD: .ASCIZ	/DIFF /
34	010207	123	107	116	SGNWD: .ASCIZ	/SGN /
35	010214	110	104	040	HDWD: .ASCIZ	/HD /
36	010220	123	105	103	SECWD: .ASCIZ	/SEC /
37	010225	103	131	114	CYLD: .ASCIZ	/CYL /
38	010232	106	122	117	FRMWD: .ASCIZ	/FROM /
39	010240	040	102	131	BYPSNM: .ASCIZ	/ BYPASSED /
40	010253	122	117	125	SEQMES: .ASCIZ	/ROUTINE TRACE SEQ:/
41	010276	104	122	126	STAMES: .ASCIZ	/DRV STAT/
42	010307	102	101	104	BSNSTR: .ASCIZ	/BAD SEC FILES NOT STRD. ALL SEC ASSUMED OK./
43	010363	124	117	124	TCERR: .ASCIZ	/TOTAL CMP ERRS: /
44						
45						
46	010404	104	122	126	MDPDY: .ASCIZ	/DRV RDY /
47	010415	103	117	116	MCERR: .ASCIZ	/CONT ERR /
48	010427	110	104	122	MHCRC: .ASCIZ	/HDR CRC/
49	010437	104	101	124	MDCRC: .ASCIZ	/DATA CRC/
50	010450	110	104	122	MHNF: .ASCIZ	/HDR NOT FND/
51	010464	104	101	124	MDLT: .ASCIZ	/DATA LATE/
52	010476	110	104	122	MHFCRC: .ASCIZ	&HDR NOT FND/HDR CRC/OPI&
53	010526	104	122	126	MDRERR: .ASCIZ	/DRV ERR /
55	010537	123	105	114	MHSTA: .ASCIZ	/SEL'D HD /
56	010551	126	117	114	MVOLCK: .ASCIZ	/VOL CHK /
57	010562	103	117	126	MCOSTA: .ASCIZ	/COVER OPEN/
58	010575	102	122	125	MHSTA: .ASCIZ	/BRUSH HOME/
59	010610	127	122	124	MWLSTA: .ASCIZ	/WRT LCK /
60	010621	110	104	123	MHOSTA: .ASCIZ	/HDS OUT /
62	010632	104	122	126	MDSERR: .ASCIZ	/DRV SEL ERR /
63	010647	104	122	126	MDRVST: .ASCIZ	/DRV STATE /
64	010662	123	120	111	MSPERR: .ASCIZ	/SPIN TIMEOUT /
65	010700	127	122	124	MWGERR: .ASCIZ	/WRT GAT ERR /
66	010715	123	105	105	MSTERR: .ASCIZ	/SEEK TIMEOUT /
67	010733	110	105	101	MHCERR: .ASCIZ	/HEAD CUR ERR /
68	010751	127	122	124	MWDERR: .ASCIZ	/WRT DAT ERR /



69	010766	117	120	122	MOPERR:	.ASCIZ	/OPR-INC/
70	010776	110	104	122	MHDERR:	.ASCIZ	&HDR/DAT ERR &
71	011013	110	104	122	MFLERR:	.ASCIZ	&HDR NOT FND/DAT LATE &
72	011041	116	055	130	MNEERR:	.ASCIZ	/N-X-MEM /
73	011052	103	131	114	MCYLOC:	.ASCIZ	/CYL /
74	011057	103	101	116	MMDRST:	.ASCIZ	/CANNOT GET DRV STAT/
75	011103	125	116	113	MUNDEF:	.ASCIZ	/UNKN DRV STATE-NO RDY,NO ERR,HDS OUT/
76	011150	106	101	111	MRLFAL:	.ASCIZ	/FAIL TO RELD HDS AFTER ERR CLEAR/
77	011211	127	122	124	MWRTAB:	.ASCIZ	/WRT ABORTED/
78	011225	040	117	126	MEXERS:	.ASCIZ	/ OVER ERR LIMIT - UNIT DROPPED /
79	011265	040	105	122	MERRS:	.ASCIZ	/ ERROR/
80	011274	2C7	377	377	BELL:	.ASCIZ	<207><377><377>
81							
82					:	RESULT SETTINGS	
83	011300	111	123	040	RESE3:	.ASCIZ	/IS /
84	011304	040	123	102	RESE4:	.ASCIZ	/ SB /
85							
86					:	RESULT CONDITIONS	
87	011311	040	111	116	RESE5:	.ASCIZ	/ IN /
88	011316	040	117	106	RESE6:	.ASCIZ	/ OF /
89	011323	123	124	101	STATE2:	.ASCIZ	/STATE 2/
90	011333	123	124	101	STATE3:	.ASCIZ	/STATE 3/
91	011343	123	124	101	STATE5:	.ASCIZ	/STATE 5/
93	011353	123	105	105	CDRDY:	.ASCIZ	&SEEK W/O MOTIONS
95	011373	061	123	124	C10MS:	.ASCIZ	/1ST 3 MS/
96	011404	065	060	060	C500MS:	.ASCIZ	/500MS/
97	011412	103	131	103	CCYLUP:	.ASCIZ	/CYCLE UP/
98	011423	104	101	124	CAFDT:	.ASCIZ	/DATA XFR/
99	011434	065	040	123	CSSEC:	.ASCIZ	/5 SEC/
100							
101	011442	045	116	045	FMTOP1:	.ASCIZ	/XNXTXNXTXTX06XSXTX01XN/
102	011471	045	116	045	FMTOP2:	.ASCIZ	/XNXTX01XS1XTX01XN/
103	011513	045	116	045	FMTOP3:	.ASCIZ	/XNXTX01XS1XTXTXN/
104	011534	045	124	045	FMT1:	.ASCIZ	/XTXT/
105	011541	045	116	045	FMT1.1:	.ASCIZ	/XNXTXT/
106	011550	045	124	000	FMT2:	.ASCIZ	/XT/
107	011553	045	116	000	FMT3:	.ASCIZ	/XN/
108	011556	045	116	045	FMT4:	.ASCIZ	/XNXTXTXN/
109	011567	045	116	045	FMT5:	.ASCIZ	/XNXTX06XS1XTX01/
110	011607	045	116	045	FMT6:	.ASCIZ	/XNXS11XTXS4XTXS4XTXS4XTXS4XTXS2XT/
111	011651	045	116	045	FMT7:	.ASCIZ	/XNXTX06XS2X06XS2X06XS2X06XS3X03XS2X01XN/
112	011721	045	116	045	FMT8:	.ASCIZ	/XNXTX06XS2X06XS2X(4XS2X06/
113	011753	045	116	045	FMT9:	.ASCIZ	/XNXT/
114	011760	045	124	045	FMT11:	.ASCIZ	/XTX01/
115	011766	045	124	045	FMT12:	.ASCIZ	/XTX03/
116	011774	045	116	045	FMT13:	.ASCIZ	/XNXS11XTX03XS1XTX03XS1XTX01XS1XTX01/
117	012040	045	116	045	FMT14:	.ASCIZ	/XNXTXTXD3XS1XTX06XS1XTX06/
118	012072	045	116	045	FMT15:	.ASCIZ	/XNXS11XTXD3XS1XTX06XS1XTX06/
119	012126	045	116	045	FMT16:	.ASCIZ	/XNXS5X06/
120	012137	045	123	061	FMT17:	.ASCIZ	/XS10XTXNXS11X06XN/
121	012161	045	116	045	FMT18:	.ASCIZ	/XNXS15XTXS5XTXS4XTXS5XTXN/
122	012213	045	124	045	FMT19:	.ASCIZ	/XTXS4XD6XS4XD6XS4XD6XS4XD6XN/
123	012250	045	124	045	FMT20:	.ASCIZ	/XTXS2XD6XS14XD6XS4XD6XN/
124	012300	045	124	045	FMT21:	.ASCIZ	/XTXS12XD6XS14XD6XN/
125	012323	045	116	045	FMT22:	.ASCIZ	/XNXS11XTX03XS1XTX01XS1XTX02/
126	012357	045	124	045	FMT23:	.ASCIZ	/XTXTXTX01XN/
127	012373	045	116	045	FMT24:	.ASCIZ	/XNXT/

```

128 012400 045 116 045 FMT25: .ASCIZ /%N%D2%T/
129 012410 045 116 045 FMT26: .ASCIZ /%N%S1%T%D4%T%T%D3%N/
130 012434 045 116 045 FMT27: .ASCIZ /%N%T%D3%T%D3%N/
131 012453 045 116 045 FMT28: .ASCIZ /%N%T%T%T/
132 012464 ENDMOD
  
```

.SBTTL ERROR MESSAGES

```

141 012464 BGNMOD GLBERR
142 : ERR1 R3 POINTS TO RESULT MESSAGE
143 : RESULT: (R3)
144 :
145 : ERR2 R3 POINTS TO RESULT NAME
146 : RESULT: (R3) IS 1 SB 0
147 :
148 : ERR3 R3 POINTS TO RESULT NAME
149 : RESULT: (R3) IS 0 SB 1
150 :
151 : ERR4 R3 POINTS TO RESULT NAME
152 : R4 POINTS TO RESULT CONLITIONS
153 : RESULT: (R3) IS 1 SB 0 (R4)
154 :
155 : ERR5 R3 POINTS TO RESULT NAME
156 : R4 POINTS TO RESULT CONDITIONS
157 : RESULT: (R3) IS 0 SB 1 (R4)
158 :
159 : ERR6 RESULT ROUTINE DETERMINES WHICH ERROR(S) ARE SET AND
160 : REPORTS ALL
161 : RESULT: "ERROR" IS 1 SB 0
162 :
163 : ERR7 DRIVE STATE ERROR REPORT
164 : R3 CONTAINS EXPECTED STATE
165 : T.STAT CONTAINS BAD STATE
166 : RESULT: DRIVE STATE IS (T.STAT) SB (R3)
167 :
168 : ERR8 HEAD POSITIONING ERROR REPORT
169 : NEWCYL CONTAINS EXPECTED CYLINDER
170 : HDWRD1 CONTAINS BAD CYLINDER
171 : RESULT: CYLINDER IS (HDWRD1) SB (NEWCYL)
172 :
173 : ERR9 UTILITY RESULT REPORT
174 : R3 POINTS TO RESULT NAME
175 : R4 POINTS TO VALUE 1
176 : R5 POINTS TO VALUE 2
177 : RESULT: (R3-NAME) IS (R4-VALUE 1) SB (R5-VALUE 2)
178 :
179 : ERR10 COMPARE ERROR REPORT
180 : R3 CONTAINS THE BAD WORD NUMBRER
181 : R4 POINTS TO BAD WORD
182 : R5 POINTS TO GOOD WORD
183 : RESULT: WORD (R3) IS (R4) SB (R5)
  
```

.NLIST MD,ME

186  
187  
188

189	012464			BGNMSG	ERR1			
190	012464	105767	170675		TSTB	NOERCT		:TEST IF ERROR COUNTING INHIBITED
191	012470	001002			BNE	1\$		:YES - SKIP
192	012472	005277	170460		INC	@ERRPOINT		:ELSE BUMP ERROR COUNT
193	012476	010146		1\$:	MOV	R1,-(SP)		:STORE R1
194	012500	004767	011212		JSR	PC,RPTOP		:REPORT OPERATION
195	012504	012721	000001		MOV	#1,(R1)+		:SET PARAM NUMBER
196	012510	010321			MOV	R3,(R1)+		:INSERT MESSAGE ADDRESS POINTER
197	012512	004767	011766		JSR	PC,RPTRES		:REPORT RESULTS
198	012516	004767	012170		JSR	PC,RPTREM		:REPORT REMAINDER
199	012522	012601			MOV	(SP)+,R1		:RESTORE R1
200	012524	004767	003714		JSR	PC,CKERLM		:GO CHECK IF ERROR COUNT EXCEEDED
201	012530			ENDMSG				
	012530			L10000:				
	012530	104423			TRAP	C\$MSG		
202								
203	012532			BGNMSG	ERR2			
204	012532	005277	170420		INC	@ERRPOINT		:BUMP ERROR COUNT
205	012536	010146			MOV	R1,-(SP)		:STORE R1
206	012540	004767	011152		JSR	PC,RPTOP		:REPORT OPERATION
207	012544	012721	000003		MOV	#3,(R1)+		:SET PARAM NUMBER
208	012550	010321			MOV	R3,(R1)+		:INSERT NAME ADD POINTER
209	012552	012721	000001		MOV	#1,(R1)+		:SET IS VALUE
210	012556	005021			CLR	(R1)+		:SET SB VALUE
211	012560	004767	011720		JSR	PC,RPTRES		:REPORT RESULTS
212	012564	004767	012122		JSR	PC,RPTREM		:REPORT REMAINDER
213	012570	012601			MOV	(SP)+,R1		:RESTORE R1
214	012572	004767	003646		JSR	PC,CKERLM		:GO CHECK IF ERROR COUNT EXCEEDED
215	012576			ENDMSG				
	012576			L10001:				
	012576	104423			TRAP	C\$MSG		
216								
217	012600			BGNMSG	ERR3			
218	012600	005277	170352		INC	@ERRPOINT		:BUMP ERROR COUNT
219	012604	010146			MOV	R1,-(SP)		:STORE R1
220	012606	004767	011104		JSR	PC,RPTOP		:REPORT OPERATION
221	012612	012721	000003		MOV	#3,(R1)+		:SET PARAM NUMBER
222	012616	010321			MOV	R3,(R1)+		:INSERT NAME ADD POINTER
223	012620	005021			CLR	(R1)+		:SET IS VALUE
224	012622	012721	000001		MOV	#1,(R1)+		:SET SB VALUE
225	012626	004767	011652		JSR	PC,RPTRES		:REPORT RESULTS
226	012632	004767	012054		JSR	PC,RPTREM		:REPORT REMAINDER
227	012636	012601			MOV	(SP)+,R1		:RESTORE R1
228	012640	004767	003600		JSR	PC,CKERLM		:GO CHECK IF ERROR COUNT EXCEEDED
229	012644			ENDMSG				
	012644			L10002:				
	012644	104423			TRAP	C\$MSG		
230								
231	012646			BGNMSG	ERR4			
232	012646	005277	170304		INC	@ERRPOINT		:BUMP ERROR COUNT
233	012652	010146			MOV	R1,-(SP)		:STORE R1
234	012654	004767	011036		JSR	PC,RPTOP		:REPORT OPERATION
235	012660	012721	000004		MOV	#4,(R1)+		:SET PARAM NUMBER
236	012664	010321			MOV	R3,(R1)+		:INSERT NAME ADD POINTER
237	012666	012721	000001		MOV	#1,(R1)+		:SET IS VALUE
238	012672	005021			CLR	(R1)+		:SET SB VALUE
239	012674	010411			MOV	R4,(R1)		:INSERT ADD OF CONDITION POINTER

ERROR MESSAGES

240	012676	004767	011302		JSR	PC,RPTRES	;REPORT RESULTS
241	012702	004767	012004		JSR	PC,RPTREM	;REPORT REMAINDER
242	012706	012601			MOV	(SP)+,R1	;RESTORE R1
243	012710	004767	003530		JSR	PC,CKERLM	;GO CHECK IF ERROR COUNT EXCEEDED
244	012714			ENDMSG			
	012714			L10003:			
	012714	104423			TRAP	C\$MSG	
245							
246	012716			BGNMSG	ERR5		
247	012716	005277	170234		INC	@ERRPOINT	;BUMP ERROR COUNT
248	012722	010146			MOV	R1,-(SP)	;STORE R1
249	012724	004767	010766		JSR	PC,RPTOP	;REPORT OPERATION
250	012730	012721	000004		MOV	#4,(R1)+	;SET PARAM NUMBER
251	012734	010321			MOV	R3,(R1)+	;INSERT NAME ADD POINTER
252	012736	005021			CLR	(R1)+	;SET IS VALUE
253	012740	012721	000001		MOV	#1,(R1)+	;SET SB VALUE
254	012744	010411			MOV	R4,(R1)	;INSERT ADD OF CONDITION POINTER
255	012746	004767	011532		JSR	PC,RPTRES	;REPORT RESULTS
256	012752	004767	011734		JSR	PC,RPTREM	;REPORT REMAINDER
257	012756	012601			MOV	(SP)+,R1	;RESTORE R1
258	012760	004767	003460		JSR	PC,CKERLM	;GO CHECK IF ERROR COUNT EXCEEDED
259	012764			ENDMSG			
	012764			L10004:			
	012764	104423			TRAP	C\$MSG	
260							
261	012766			BGNMSG	ERR6		
262	012766	105767	170373		TSTB	NOERCT	;TEST IF ERROR COUNTING INHIBITED
263	012772	001002			BNE	17\$	;YES - SKIP
264	012774	005277	170156		INC	@ERRPOINT	;ELSE BUMP ERROR COUNT
265	013000	010146		17\$:	MOV	R1,-(SP)	;STORE R1
266	013002	010346			MOV	R3,-(SP)	;STORE R3
267	013004	010446			MOV	R4,-(SP)	;STORE R4
268	013006	010546			MOV	R5,-(SP)	;STORE R5
269	013010	004767	010702		JSR	PC,RPTOP	;REPORT OPERATION
270	013014	012721	000003		MOV	#3,(R1)+	;SET PARAM NUMBER
271	013020	012761	000001	000002	MOV	#1,2(R1)	;INSERT IS VALUE
272	013026	005067	170072		CLR	TEMP3	;CLEAR FOR STATUS STORAGE
273	013032	016703	170006		MOV	T.CS,R3	;GET T.CS
274	013036	042703	177761		BIC	#177761,R3	;AND CLEAR ALL BUT FUNCTION
275	013042	022703	000004		CMF	#4,R3	;CHECK IF IT WAS GET STATUS
276	013046	001443			BEQ	1\$	;YES - STATUS IS IN T.MP, SKIP
277	013050	012762	000003	000004	MOV	#GETSTAT,RLDA(R2)	;ELSE DO GET STATUS
278	013056	012703	000004		MOV	#4,R3	
279	013062	056703	167744		BIS	RLDRV,R3	
280	013066	010362	000000		MOV	R3,RLCS(R2)	
281	013072				WAITUS	10.	;WAIT FOR CONTROLLER READY
	013072	012727	000012		MOV	#10.,(PC)+	
	013076	000000			.WORD	0	
	013100	016727	167012		MOV	L\$DLY,(PC)+	
	013104	000000			.WORD	0	
	013106	005367	177772		DEC	-6(PC)	
	013112	001375			BNE	-.4	
	013114	005367	177756		DEC	-22(PC)	
	013120	001367			BNE	-.20	
282	013122	032762	000200	000000	BIT	#CRDYMSK,RLCS(R2)	;TEST IF READY
283	013130	001003			BNE	10\$	;YES - SKIP
284	013132	012703	001000	9\$:	MOV	#BIT9,R3	;ELSE SET NO DRIVE STATUS BIT

285	013136	000413		BR	2\$	:IN MESSAGE WORD AND SKIP
286	013140	016203	000006	10\$:	MOV RLMP(R2),R3	:STORE STATUS FOR REPORT
287	013144	010367	167754		MOV R3,TEMP3	
288	013150	116703	167751		MOVB TEMP3+1,R3	:GET ERROR BITS IN PROPER POSITION
289	013154	000402		BR	13\$	
290	013156	116703	167671	1\$:	MOVB T.MP+1,R3	:GET ERROR BITS FROM MP REG
291	013162	042703	177442	13\$:	BIC #177442,R3	:CLEAR UNUSED BITS
292	013166	016704	167652	2\$:	MOV T.CS,R4	:GET ERROR BITS FROM CS REG
293	013172	042704	001777		BIC #1777,R4	:CLEAR UNUSED BITS
294	013176	050403			BIS R4,R3	:MAKE ONE WORD OF POSSIBLE ERRORS
295	013200	032703	002000		BIT #OPIERR,R3	:TEST IF OPI SET
296	013204	001442			BEQ 115\$	:NO - SKIP
297	013206	032703	010000		BIT #HNFERR,R3	:TEST IF HDR NOT FOUND ERROR
298	013212	001026			BNE 107\$	:YES - SKIP
299	013214	032703	004000		BIT #HCRCERR,R3	:TEST IF HDR CRC ERR
300	013220	001020			BNE 105\$	:YES - SKIP
301	013222	012704	010766		MOV #MOPERR,R4	:SET OPI ALONE MESSAGE
302	013226			100\$:	PRINTB #FMT28,#MRSLT,R4,#MERRS	:REPORT ERROR
	013226	012746	011265		MOV #MERRS,-(SP)	
	013232	010446			MOV R4,-(SP)	
	013234	012746	005423		MOV #MRSLT,-(SP)	
	013240	012746	012453		MOV #FMT28,-(SP)	
	013244	012746	000004		MOV #4,-(SP)	
	013250	010600			MOV SP,R0	
	013252	104414			TRAP C\$PNTB	
	013254	062706	000012		ADD #12,SP	
303	013260	000430			BR 120\$	:SKIP
304	013262	012704	010427	105\$:	MOV #MHCRC,R4	:HDR CRC MESSAGE
305	013266	000757			BR 100\$	
306	013270	032703	004000	107\$:	BIT #HCRCERR,R3	:TEST IF HCRC WITH HDR NOT FND
307	013274	001003			BNE 109\$	:YES - SKIP
308	013276	012704	010450		MOV #MHNFR,R4	:MESSAGE HEADER NOT FOUND
309	013302	000751			BR 100\$	
310	013304	012704	010476	109\$:	MOV #MHFCRC,R4	:HNF AND HCRC MESSAGE
311	013310	000746			BR 100\$	:SKIP
312	013312	032703	004000	115\$:	BIT #DCKERR,R3	:TEST IF DATA CHECK SET, NOT OPI
313	013316	001403			BEQ 118\$	:NO - SKIP
314	013320	012704	010437		MOV #MDCRC,R4	:SET MESSAGE DATA CHECK
315	013324	000740			BR 100\$	:SKIP
316	013326	032703	010000	118\$:	BIT #DLTERR,R3	:TEST IF DATA LATE ERROR
317	013332	001403			BEQ 120\$	:NO - SKIP
318	013334	012704	010464		MOV #DLT,R4	:SET MESSAGE DATA LATE
319	013340	000732			BR 100\$	:SKIP
320	013342	012705	100000	120\$:	MOV #BIT15,R5	:SET BIT POINTER FOR TEST
321	013346	005004			CLR R4	:CLEAR R4 FOR TABLE COUNT
322	013350	030503		3\$:	BIT R5,R3	:TEST IF BIT IS SET
323	013352	001005			BNE 6\$	:YES - SKIP TO REPORT
324	013354	005724		4\$:	TST (R4)+	:ELSE BUMP TABLE POINTER
325	013356	000241			CLC	:CLEAR CARRY
326	013360	006005			ROR R5	:SHIFT BIT POINTER TO NEXT BIT
327	013362	001372			BNE 3\$	:LOOP IF NOT 0
328	013364	000405			BR 7\$	:ELSE REPORT REMAINDER
329	013366	016411	002320	6\$:	MOV RESTBL(R4),(R1)	:INSERT NAME ADDRESS
330	013372	004767	011106		JSR PC,RPTRES	:REPORT RESULTS
331	013376	000766			BR 4\$	:GET NEXT BIT
332	013400	004767	011306	7\$:	JSR PC,RPTREM	:REPORT REMAINDER
333	013404	005767	167514		TST TEMP3	:TEST IF ANY NEW STATUS

334	013410	001414				BEQ	15\$		:NO - SKIP
335	013412					PRINTB	#FMT17,#STAMES,TEMP3		
	013412	016746	167506			MOV	TEMP3,-(SP)		
	013416	012746	010276			MOV	#STAMES,-(SP)		
	013422	012746	012137			MOV	#FMT17,-(SP)		
	013426	012746	000003			MOV	#3,-(SP)		
	013432	010600				MOV	SP,R0		
	013434	104414				TRAP	C\$PNTB		
	013436	062706	000010			ADD	#10,SP		
336	013442	032767	004000	167374	15\$:	BIT	#DCKERR,T.CS		:TEST IF DATA CHECK ERROR
337	013450	001453				BEQ	25\$		:NO - SKIP
338	013452	032767	002000	167364		BIT	#OPIERR,T.CS		:TEST IF OPI SET
339	013460	001047				BNE	25\$		:YES - SKIP
340	013462	005067	167326			CLR	MORECE		:CLEAR COMPARE ERROR COUNT
341	013466	012701	000200			MOV	#128,R1		:SET COMPARE LENGTH
342	013472	012703	000001			MOV	#1,R3		:SET WORD COUNT
343	013476	012705	004364			MOV	#OBUFF,R5		:SET GOOD WORD POINTER
344	013502	012704	003764			MOV	#IBUFF,R4		:SET TEST WORD POINTER
345	013506	021514			18\$:	CMP	(R5),(R4)		:CHECK WORD
346	013510	001427				BEQ	19\$		:GOOD - SKIP
347	013512	026727	167276	000012		CMP	MORECE,#10.		:TEST IF COMPARE LIMIT REACHED
348	013520	003021				BGT	20\$		:YES - SKIP
349	013522					PRINTB	#FMT15,#MWORD,R3,#RESE3,(R4),#RESE4,(R5)		
	013522	011546				MOV	(R5),(R4)		
	013524	012746	011304			MOV	#RESE4,-(SP)		
	013530	011446				MOV	(R4),(R3)		
	013532	012746	011300			MOV	#RESE3,-(SP)		
	013536	010346				MOV	R3,-(SP)		
	013540	012746	006172			MOV	#MWORD,-(SP)		
	013544	012746	012072			MOV	#FMT15,-(SP)		
	013550	012746	000007			MOV	#7,-(SP)		
	013554	010600				MOV	SP,R0		
	013556	104414				TRAP	C\$PNTB		
	013560	062706	000020			ADD	#20,SP		
350	013564	005267	167224		20\$:	INC	MORECE		:BUMP ERROR COUNTER
351	013570	022524			19\$:	CMP	(R5)+,(R4)+		:BUMP POINTERS
352	013572	005203				INC	R3		:BUMP COUNTER
353	013574	005301				DEC	R1		:DEC LENGTH COUNT
354	013576	001343				BNE	18\$		:LOOP IF NOT DONE
355	013600	005767	167210		25\$:	TST	MORECE		:TEST IF ANY COMPARE ERRORS
356	013604	001421				BEQ	27\$		:NO - SKIP
357	013606	012701	000200			MOV	#128,R1		:SET COMPARE LENGTH
358	013612					PRINTB	#FMT27,#TCERR,MORECE,#RESE6,R1		
	013612	010146				MOV	R1,-(SP)		
	013614	012746	011316			MOV	#RESE6,-(SP)		
	013620	016746	167170			MOV	MORECE,-(SP)		
	013624	012746	010363			MOV	#TCERR,-(SP)		
	013630	012746	012434			MOV	#FMT27,-(SP)		
	013634	012746	000005			MOV	#5,-(SP)		
	013640	010600				MOV	SP,R0		
	013642	104414				TRAP	C\$PNTB		
	013644	062706	000014			ADD	#1,SP		
359	013650	012605			27\$:	MOV	(SP)+,R5		:RESTORE R5, 4, 3, 1
360	013652	012604				MOV	(SP)+,R4		
361	013654	012603				MOV	(SP)+,R3		
362	013656	012601				MOV	(SP)+,R1		
363	013660	004767	002560			JSR	PC,CKERM		:GO CHECK IF ERROR COUNT EXCEEDED

```

364 013664          ENDMSG
      013664          L10005:
      013664 104423   TRAP      C$MSG

365
366 013666          BGNMSG  ERR7
367 013666 005277 167264   INC      @ERRPOINT      ;BUMP ERROR COUNT
368 013672 010146          MOV      R1,-(SP)      ;STORE R1
369 013674 004767          JSR      PC,RPTOP      ;REPORT OPERATION
370 013700 012721 000003   MOV      #3,(R1)+      ;SET PARAM NUMBER
371 013704 012721 010647   MOV      #MDRVST,(R1)+ ;INSERT NAME ADD POINTER
372 013710 016721 167144   MOV      T,STAT,(R1)+ ;INSERT IS VALUE
373 013714 010311          MOV      R3,(R1)      ;INSERT SB VALUE
374 013716 004767 010562   JSR      PC,RPTRES     ;REPORT RESULTS
375 013722 004767 010764   JSR      PC,RPTREM     ;REPORT REMAINDER
376 013726 012601          MOV      (SP)+,R1      ;RESTORE R1
377 013730 004767 002510   JSR      PC,CKERLM     ;GO CHECK IF ERROR COUNT EXCEEDED
378 013734          ENDMSG
      013734          L10006:
      013734 104423   TRAP      C$MSG

379
380 013736          BGNMSG  ERR8
381 013736 005277 167214   INC      @ERRPOINT      ;BUMP ERROR COUNT
382 013742 010146          MOV      R1,-(SP)      ;STORE R1
383 013744 010346          MOV      R3,-(SP)      ;STORE R3
384 013746 004767 007744   JSR      PC,RPTOP      ;REPORT OPERATION
385 013752 012721 000003   MOV      #3,(R1)+      ;SET PARAM NUMBER
386 013756 012721 011052   MOV      #MCYLOC,(R1)+ ;INSERT NAME ADD POINTER
387 013762 016711 167064   MOV      HDWRD1,(R1)   ;GET HEADER WORD
388 013766 012703 000007   MOV      #7,R3         ;SET SHIFT COUNT
389 013772 000241          3$:   CLC
390 013774 006011          ROR      (R1)          ;ALIGN CHAR FOR PRINTING
391 013776 005303          DEC      R3           ; AS IS VALUE
392 014000 001374          BNE      3$
393 014002 005721          TST      (R1)+        ;BUMP PARAM POINTER
394 014004 016711 167072   MOV      NEWCYL,(R1)   ;INSERT SB VALUE
395 014010 004767 010470   JSR      PC,RPTRES     ;REPORT RESULTS
396 014014 004767 010672   JSR      PC,RPTREM     ;REPORT REMAINDER
397 014020 012603          MOV      (SP)+,R3      ;RESTORE R3
398 014022 012601          MOV      (SP)+,R1      ;RESTORE R1
399 014024 004767 002414   JSR      PC,CKERLM     ;GO CHECK IF ERROR COUNT EXCEEDED
400 014030          ENDMSG
      014030          L10007:
      014030 104423   TRAP      C$MSG

401
402 014032          BGNMSG  ERR9
403 014032 005277 167120   INC      @ERRPOINT      ;BUMP ERROR COUNT
404 014036 010146          MOV      R1,-(SP)      ;STORE R1
405 014040 004767 007652   JSR      PC,RPTOP      ;REPORT OPERATION
406 014044 012721 000003   MOV      #3,(R1)+      ;SET PARAM NUMBER
407 014050 010321          MOV      R3,(R1)+      ;INSERT NAME ADD POINTER
408 014052 010421          MOV      R4,(R1)+      ;SET IS VALUE
409 014054 010521          MOV      R5,(R1)+      ;SET SB VALUE
410 014056 004767 010422   JSR      PC,RPTRES     ;REPORT RESULTS
411 014062 004767 010624   JSR      PC,RPTREM     ;REPORT REMAINDER
412 014066 012601          MOV      (SP)+,R1      ;RESTORE R1
413 014070 004767 002350   JSR      PC,CKERLM     ;GO CHECK IF ERROR COUNT EXCEEDED
414 014074          ENDMSG
    
```

```

014074
014074 104423
415 014076
416 014076 010146
417 014100 005767 166710
418 014104 001051
419 014106 005277 167044
420 014112 004767 007600
421 014116
    014116 005046
    014120 156716 166707
    014124 012746 006053
    014130 016746 166672
    014134 012746 006042
    014140 012746 011567
    014144 012746 000005
    014150 010600
    014152 104414
    014154 062706 000014
422 014160
    014160 011546
    014162 012746 011304
    014166 011446
    014170 012746 011300
    014174 010346
    014176 012746 006172
    014202 012746 005423
    014206 012746 012040
    014212 012746 000010
    014216 010600
    014220 104414
    014222 062706 000022
423 014226 000421
424 014230
    014230 011546
    014232 012746 011304
    014236 011446
    014240 012746 011300
    014244 010346
    014246 012746 006172
    014252 012746 012072
    014256 012746 000007
    014262 010600
    014264 104414
    014266 062706 000020
425 014272 005267 166516
426 014276 012601
427 014300 004767 002140
428 014304
    014304
    014304 104423
429 014306
430
431
432 014306
433 014306
    014306 000006
  
```

```

L10010:
BGNMSG TRAP C$MSG
ERR10
MOV R1,-(SP) ;STORE R1
TST MORECE ;TEST IF 2ND BAD LINE
BNE 3$ ;YES - SKIP
INC @ERRPOINT ;BUMP ERROR COUNT
JSR PC,RPTOP ;REPORT OPERATION
PRINTB #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1> ;REPORT ID
CLR -(SP)
BISB RLDRV+1,(SP)
MOV #DRVNAM,-(SP)
MOV RLBAS,-(SP)
MOV #BASADD,-(SP)
MOV #FMT5,-(SP)
MOV #5,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD #14,SP
PRINTB #FMT14,#MRSLT,#MWORD,R3,#RESE3,(R4),#RESE4,(R5)
MOV (R5),-(SP)
MOV #RESE4,-(SP)
MOV (R4),-(SP)
MOV #RESE3,-(SP)
MOV R3,-(SP)
MOV #MWORD,-(SP)
MOV #MRSLT,-(SP)
MOV #FMT14,-(SP)
MOV #10,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD #22,SP
BR 4$
3$: PRINTB #FMT15,#MWORD,R3,#RESE3,(R4),#RESE4,(R5) ;REPORT DATA
MOV (R5),-(SP)
MOV #RESE4,-(SP)
MOV (R4),-(SP)
MOV #RESE3,-(SP)
MOV R3,-(SP)
MOV #MWORD,-(SP)
MOV #FMT15,-(SP)
MOV #7,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD #20,SP
4$: INC MORECE ;INC COMPARE ERROR COUNT
MOV (SP)+,R1 ;RESTORE R1
JSR PC,CKERLM ;GO CHECK IF ERROR COUNT EXCEEDED

ENDMSG
L10011:
ENDMOD TRAP C$MSG
.EVEN

BGNMOD HPTCODE
BGNIIW .WORD L10012-L$HW/2
  
```



434 014310 174400  
 435 014312 000160  
 436 014314 000240  
 437 014316 000001  
 438 014320 000000  
 439 014322 000001  
 440 014324  
 441 014324  
 442  
 443 014324  
 444 014324  
 445 014324 000006  
 446 014326 000000  
 447  
 448  
 449  
 450  
 451  
 452  
 453 014330 000000  
 454 014332 000377  
 455 014334 000000  
 456 014336 000024  
 457 014340 000012  
 458 014342  
 459 014342  
 460  
 461 014342  
 463 014342  
 014342 000020  
 014344 025176  
 014346 025456  
 014350 025664  
 014352 030344  
 014354 031532  
 014356 032136  
 014360 033410  
 014362 034302  
 014364 034370  
 014366 035024  
 014370 035502  
 014372 036260  
 014374 036732  
 014376 037152  
 014400 037432  
 014402 040100  
 468 014404  
 469  
 470  
 471 014404  
 472 014404 000000  
 473 014406 177777  
 474 014410 000010  
 475 014412

.WORD 174400  
 .WORD 160  
 .WORD 240  
 .WORD 1  
 .WORD 0  
 .WORD 1  
 ENDMOD  
 L10012:  
 ENDMOD  
 BGNMOD SP\*CODE  
 BGNSW  
 MISWIW: .WORD L10013-LSSW/2  
 .WORD 0  
 LOLIMW: .WORD 0  
 HILIMW: .WORD 255.  
 HEADW: .WORD 0  
 ERLIMW: .WORD 20.  
 DCLIMW: .WORD 10.  
 ENDSW  
 L10013:  
 ENDMOD  
 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  
 ENDMOD  
 ;LOAD PROTECTION TABLE  
 BGNPROT  
 .WORD 0  
 .WORD -1  
 .WORD 10  
 ENDPROT

;CSR BASE ADDRESS DEFAULT  
 ;VECTOR DEFAULT  
 ;PRIORITY DEFAULT  
 ;TYPE OF DRIVE, RL01=1, RL02=2  
 ;DRIVE NUMBER DEFAULT  
 ;RL11 CONTROLLER  
 ;BIT 0 = USE ALL CYLINDERS  
 ;BIT 1 = USE ALL SECTORS  
 ;BIT 2 = EXECUTE DRIVE SELECT TEST  
 ;BIT 3 = EXECUTE HEAD ALIGNMENT  
 ;BIT 12 = HEAD SELECT SUPPLIED FLAG  
 ;BIT 13 = HILIMIT SPECIFIED FLAG  
 ;BIT 14 = LO LIMIT SPECIFIED FLAG  
 ;BIT 15 = DO MANUAL INTERVENTION  
 ;ERROR LIMIT  
 ;COMPARE ERROR LIMIT  
 ;P-TABLE OFFSET OF CSR  
 ;NOT A MASS-BUSS DRIVE  
 ;P-TABLE OFFSET OF DRIVE

```

476
477
478          .SBTTL  INITIALIZATION CODE
479
480 014412    BGNMOD  INITCODE
481 014412    BGNINIT
482          ;CHECK FOR PRESENCE OF A CLOCK
483 014412    005067  166526    PCLK:  CLR      CLKFLG          ;CLEAR CLOCK FLAG
484
485 014416    012700  000120          MOV      #'P,R0
486 014422    104462          TRAP     C$CLCK
487 014424    010067  166516          MOV      R0,CLKADR
488 014430    103004          BCC     NOPCLK
489 014432    010057  000001  166504    MOV      #1,CLKFLG          ;INDICATE PRESENCE OF A P-CLOCK
490 014440    000451          BR      TCLK                ;P CLOCK EXISTS, DO NOT USE L CLOCK.
491
492 014442    012737  014556  000004    NOPCLK: MOV     #TSTCLK,@#4          ;TEST FOR L CLOCK. IF NO CLOCK - SKIP.
493 014450    005737  177546          TST     @#177546
494
495 014454    012767  000011  166462    MOV     #11,CLKFLG          ;INDICATE THE PRESENCE OF AN L CLOCK.
496
497 014462    012737  014514  000100    MOV     #LCLK,@#100         ;L CLOCK VECTGR POINTS TO LCLK.
498
499 014470    010146          MOV     R1,-(SP)           ;SAVE  R1 AND R2 ON THE STACK.
500 014472    010246          MOV     R2,-(SP)
501
502 014474    005002          CLR     R2
503 014476    012737  000100  177546    MOV     #100,@#177546       ;START THE L CLOCK.
504
505 014504    062702  000001          1$:   ADD     #1,R2           ;BUILD SOFTWARE LOOP. USE ADD TO SET FLAGS.
506 014510    000240          NOP
507 014512    000774          BR      1$
508
509 014514    012716  014522          LCLK:  MOV     #LCLK1,@SP     ;MODIFY THE STACK TO RETURN TO LCLK1.
510 014520    000002          RTI
511 014522    005037  177546          LCLK1: CLR     @#177546       ;STOP THE L CLOCK.
512
513 014526    012701  000246          MOV     #166.,R1           ;THIS IS THE DIVISOR TO GET 100 US.
514
515 014532    005067  166412          CLR     LBASE
516 014536    005267  166406          1$:   INC     LBASE           ;LBASE IS THE APPROXIMATE NUMBER OF ITERATIONS
517 014542    160102          SUB     R1,R2             ;NEEDED TO GIVE 100 US.
518 014544    100401          BMI    2$
519 014546    000773          BR      1$
520
521 014550    012602          2$:   MOV     (SP)+,R2         ;RESTORE R1 AND R2.
522 014552    012601          MOV     (SP)+,R1
523 014554    000403          BR      TCLK              ;SKIP RTI HANDLER
524
525 014556    012716  014564          TSTCLK: MOV    #TCLK,(SP)     ;ADJUST STACK FOR RTI
526 014562    000002          RTI
527 014564    005767  166354          TCLK:  TST     CLKFLG         ;IF THERE IS NO P OR L CLOCK, DO NOT DO THE
528 014570    001015          BNE    1$                 ;TEST. PRINT A MESSAGE SAYING WHY THE TEST IS
529 014572    012746  006664          MOV     #NOTST,-(SP)       ;ABORTED.
530 014576    012746  011753          MOV     #FMT9,-(SP)
531 014602    012746  000002          MOV     #2,-(SP)
532 014606    010600          MOV     SP,R0

```

```

533 014610 104417          TRAP    C$PNTF
534 014612 062706 000006  ADD     #6,SP
535 014616 012701 000200  MOV     #200,R1
536 014622 000111          JMP     @R1
537
538 014624          1$:   SETPRI #340          ;SET PRIORITY TO 7 TO INHIBIT INTERRUPTS
    014624 012700 000340  MOV     #340,RO
    014630 104441          TRAP    C$SPRI
539 014632          MANUAL          ;CHECK IF MANUAL INTERVENTION ALLOWED
    014632 104450          TRAP    C$MANI
540 014634          BCOMPLETE 2$          ;YES - SKIP
    014634 103403          BCS     2$
541 014636 042767 100014 177462  BIC     #MITEST!DRSELT!HDALIGN,MISWIW ;CLEAR ALL MANUAL
542                                     ; INTERVENTION FLAGS
543 014644 005067 166132          2$:   CLR     SSINDX          ;CLEAR SUBROUTINE STACK INDEX
544 014650          READEF #EF.PWR          ;POWER FAILURE?
    014650 012700 000034  MOV     #EF.PWR,RO
    014654 104447          TRAP    C$REFG
545 014656          BNCOMPLETE 4$          ;NO, GO CHECK NEW PASS
    014656 103005          BCC     4$
546 014660 016767 165126 166502  MOV     LSUNIT,PWRFLG ;SET POWER FAIL FLAG
547 014666 000167 000406          JMP     PWCON          ;GO SERVICE PCWER FAIL
548
549 014672          ;"START" COMMAND SEQUENCE
    014672 012700 000040          4$:   READEF #EF.START          ;CHECK IF START
    014676 104447          MOV     #EF.START,RO
    014700          TRAP    C$REFG
550 014700          BNCOMPLETE RESTART ;NO - SKIP
    014700 103034          BCC     RESTART
551                                     ; ON START INITIALIZE TO START AT FIRST DRIVE, CLEAR INTERNAL
552                                     ; PASS COUNT, AND ERROR COUNT.
553 014702 016767 165104 166164  MOV     LSUNIT,DRVCNT ;SET UP UNIT COUNT
554 014710 005067 166444          RSTRT: CLR     PASNUM          ;CLEAR PASS NUMBER
555 014714 012700 003160          MOV     #ERRCNT,RO
556 014720 012701 000100          MOV     #64,R1
557 014724 005020          1$:   CLR     (RO)+          ;GET A COUNT
558 014726 005301          DEC     R1
559 014730 001375          BNE     1$
560 014732 012767 003156 166216  MOV     #ERRCNT-2,ERRPOINT ;LOOP TILL ALL CLEARED
561 014740 012767 177777 166414  MOV     #-1,PSETNM      ;INIT ERROR POINTER
562 014746 012767 177777 166034  MOV     #-1,HADONE      ;SET PARAM SELECT TO INITIAL VALUE
563 014754 032767 040000 177344  LAB:   BIT     #LOCYL,MISWIW ;PRESET HEAD ALIGN DONE FLAG
564 014762 001002          BNE     5$
565 014764 005067 177340          BCS     5$
566 014770 000432          CLR     LOLIMW          ;TEST IF LO LIMIT SET
567 014772          5$:   BR     SETDON          ;YES - SKIP
568 014772          RSTRT:  BR     SETDON          ;ELSE CLEAR LO LIMIT
    014772 012700 000037          READEF #EF.RESTART          ;CHECK IF RESTART
    014776 104447          MOV     #EF.RESTART,RO
    015000          TRAP    C$REFG
569 015000          BNCOMPLETE RSTRT      ;NO - SKIP
    015000 103743          BCS     RSTRT
570                                     ;"CONTINUE" COMMAND SEQUENCE
571 015002          CONTINUE:
572 015002          READEF #EF.CONTINUE          ;TEST IF CONTINUE
    015002 012700 000036  MOV     #EF.CONTINUE,RO
    015006 104447          TRAP    C$REFG
573 015010          BNCOMPLETE PWCON
    015010 103533          BCS     PWCON
    
```

```

574 : ON CONTINUE PICK UP UNIT LAST UNDER TEST
575 015012 READDEF #EF.NEW ;CHECK IF STARTING NEW PASS
      015012 012700 000035 MOV #EF.NEW,RO
      015016 104447 TRAP C$REFG
576 015020 BCOMPLETE PASNEW
      015020 103403 BCS PASNEW
577 015022 NXPAS:
578 015022 005767 166046 TST DRVCNT ;TEST IF ALL UNITS CHECKED
579 015026 001013 BNE SETDON ;NO - SKIP
580 015030 005267 166324 PASNEW: INC PASNUM ;ELSE BUMP PASS COUNT
581 015034 012767 003156 166114 MOV #ERRCNT-2,ERRPOINT ;INIT ERROR POINTER
582 015042 016767 164744 166024 MOV L$UNIT,DRVCNT ;GET ALL DRIVES
583 015050 012767 177777 166304 MOV #-1,PSETNM ;SET PARAM SELECT TO INITIAL
584 015056 005267 166300 SETDON: INC PSETNM ;NEXT SET OF PARAMETERS
585 015062 005367 166006 DEC DRVCNT ;DOWN COUNT DRIVE TOTAL
586 015066 062767 000002 166062 ADD #2,ERRPOINT ;UPDATE THE ERROR POINTER
587 015074 016700 166262 MOV PSETNM,RO ;SET UP TO GET PARAMETERS
588 015100 012702 003026 MOV #RLBAS,R2 ;GET POINTER TO RL11 BASE ADDRESS
589 015104 GPHARD RO,R1
      015104 104442 TRAP C$GPHRD
      015106 010001 MOV RO,R1
590 015110 BCOMPLETE 7$ ;SKIP IF GOOD PARAM
      015110 103406 BCS 7$
591 015112 005767 166252 TST PWRFLG ;RECENT POWER FAILURE
592 015116 001741 BEQ NXPAS ;NO
593 015120 005367 166244 DEC PWRFLG ;ACCOUNT FOR DRIVE
594 015124 000736 BR NXPAS
595 :MOVE P-TABLE CONTENTS TO LOCAL STORAGE
596 015126 012122 7$: MOV (R1)+,(R2)+ ;STORE CSR
597 015130 012122 MOV (R1)+,(R2)+ ;STORE VECTOR
598 015132 005721 TST (R1)+ ;BUMP PAST PRIORITY
599 015134 012167 165136 MOV (R1)+,T.DRIVE ;STORE DRIVE TYPE
600 015140 012122 MOV (R1)+,(R2)+
601 015142 022767 000001 165126 CMP #1,T.DRIVE
602 015150 001426 BEQ 65$
603 :INITIALIZE RLO2 PARAMETERS
604 015152 012767 000776 165126 MOV #510.,NXTHL
605 015160 012767 000777 165114 MOV #511.,HLMTW
606 015166 012767 001000 165114 MOV #512.,GBND
607 015174 012767 177600 165110 MOV #177600,CAMSK
608 015202 012767 177600 165104 MOV #177600,DIRMSK
609 015210 012767 177600 165100 MOV #177600,HDCYL
610 015216 012767 177000 165060 MOV #177000,CLRBYT
611 015224 000425 BR PWCON
612 :INITIALIZE RLO1 PARAMETERS
613 015226 012767 000377 165046 65$: MOV #255.,HLMTW
614 015234 012767 000400 165046 MOV #256.,GBND
615 015242 012767 077600 165042 MOV #77600,CAMSK
616 015250 012767 077600 165036 MOV #77600,DIRMSK
617 015256 012767 077600 165032 MOV #77600,HDCYL
618 015264 012767 000376 165014 MOV #254.,NXTHL
619 015272 012767 177400 165004 MOV #177400,CLRBYT
620
621 015300 032767 020000 177020 PWCON: BIT #HICYL,MISWIW
622 015306 001003 BNE 1$
623 015310 016767 164766 177014 MOV HLMTW,HILIMW
624 015316 1$: SETVEC RLVEC,#INTHLR,#340 ;SET UP INTERRUPT VECTOR FOR DRIVE
    
```

```

015316 012746 00C340      MOV      #340,-(SP)
015322 012746 016370      MOV      #INTHLR,-(SP)
015326 016746 165476      MOV      RLVEC,-(SP)
015332 012746 000003      MOV      #3,-(SP)
015336 104437              TRAP     C$SVEC
015340 062706 000010      ADD      #10,SP
625 015344              SETPRI  #0                ;SET PRIORITY TO 0 TO ALLOW INTERRUPTS
015344 012700 000000      MOV      #0,R0
015350 104441              TRAP     C$SPRI
626 015352 016702 165450      MOV      RLBAS,R2        ;SET RL11 BASE ADDRESS POINTER
627
628
630
631 015356              MANUAL              ;MANUAL INTERVENTION ALLOWED?
015356 104450      TRAP     C$MANI
632 015360      BNCOMPLETE 4$        ;NO
015360 103004      BCC      4$
633
634 015362 005767 165772      TST      PASNUM          ;YES, CHECK PASS NUMBER
635 015366 001001      BNE      4$              ;NOT FIRST PASS, NEED DRIVE UP
636 015370 000521      BR       8$              ;FIRST PASS, PROGRAM WILL INSTRUCT USER
637
639              ;CHECK IF POWER FAILURE WAIT IS NEEDED
640
641 015372 005767 165772      4$: TST      PWRFLG          ;NEEDED?
642 015376 001516      BEQ      8$              ;NO, SKIP
643
644 015400 016705 165426      MOV      RLDRV,R5        ;DRIVE SELECT
645 015404 052705 000200      BIS      #CRDYMSK,R5    ;SET CRDY
646 015410 010562 000000      MOV      R5,RLCS(R2)    ;SELECT DRIVE
647 015414 012701 000170      MOV      #120.,R1       ;INITIALIZE WAIT COUNT
648 015420 032762 000001 000000 9$: BIT      #DRDYMSK,RLCS(R2) ;DRIVE UP YET
649 015426 001102      BNE      8$              ;YES START TEST
650
651 015430      WAITMS  10.           ;WAIT A SECOND
015446 012727 000372      MOV      #250.,(PC)+
015452 000000      .WORD   0
015454 016727 164436      MOV      L$DLY,(PC)+
015460 000000      .WORD   0
015462 005367 177772      DEC      -6(PC)
015466 001375      BNE      -4
015470 005367 177756      DEC      -22(PC)
015474 001367      BNE      -20
015476 104422      TRAP     C$BRK
652 015506 005301      DEC      R1              ;SIXTY GONE BY
653 015510 001343      BNE      9$              ;NO
654 015512      PRINTF  #FMT24,#NOPWR  ;REPORT 'DRV DID NOT REC'R FROM PWR FAIL'
015512 012746 006060      MOV      #NOPWR,-(SP)
015516 012746 012373      MOV      #FMT24,-(SP)
015522 012746 000002      MOV      #2,-(SP)
015526 010600      MOV      SP,R0
015530 104417      TRAP     C$PNTF
015532 062706 000006      ADD      #6,SP
655 015536      PRINTF  #FMT5,#BASADD,RLBAS,#DRVNM,<B,RLDRV+1> ;REPORT DRIVE UNIBUS
015536 005046      CLR      -(SP)
015540 156716 165267      BISB    RLDRV+1,(SP)
015544 012746 006053      MOV      #DRVNM,-(SP)

```

```

015550 016746 165252      MOV      RLBAS,-(SP)
015554 012746 006042      MOV      #BASADD,-(SP)
015560 012746 011567      MCV      #FMT5,-(SP)
015564 012746 000005      MOV      #5,-(SP)
015570 010600      MOV      SP,RO
015572 104417      TRAP     C$PNTF
015574 062706 000014      ADD      #14,SP

656
657 015600      PRINTF  #FMT3      ;NEW LINE
015600 012746 011553      MOV      #FMT3,-(SP)
015604 012746 000001      MOV      #1,-(SP)
015610 010600      MOV      SP,RO
015612 104417      TRAP     C$PNTF
015614 062706 000004      ADD      #4,SP
658 015620      DODU     PSETNM      ;DO DROP UNIT ON DRIVE
015620 016700 165536      MOV      PSETNM,RO
015624 104451      TRAP     C$DODU
659 015626      DOCLN   ;INVOKE CLEAN-UP CODE TO RESTORE DRIVE
015626 104444      TRAP     C$DCLN
660
661 015630 005067 165304      CLR      ERRVEC      ;/TO STATIC STATE
662
663 015634      8$:      ;CLEAR ERROR VECTOR
664
665 015634      ENDINIT
015634      L10015:
015634 104411      TRAP     C$INIT
666
667 015636      ENDMOD
668
669
670      .SBTTL  AUTO DROP SECTION
671
672      ;THE AUTO DROP SECTION IS INVOKED BY THE DIAGNOSTIC SUPERVISOR WHENEVER THE
673      ;"ADR" FLAG IS SET BY THE OPERATOR. IT IS EXECUTED AFTER THE INITIALIZATION
674      ;CODE AND CHECKS THE DRIVE TO DETERMINE IF IT IS READY TO RECEIVE A COMMAND.
675      ;IF THE DRIVE IS NOT READY IT IS DROPPED FROM THE TEST CYCLE AND THE NEXT
676      ;DRIVE IS ACCESSED. IF THE DRIVE IS READY THE HARDWARE TESTS ARE PERFORMED
677      ;AFTER WHICH THE NEXT DRIVE IS ACCESSED.
678
679 015636      BGNAUTO
680 015636 005067 165524      CLR      TRPFLG      ;CLEAR TRAP FLAG
681 015642      SETVEC  ERRVEC,#TRPHAN,#340 ;SET UP TRAP VECTOR TO DETECT
015642 012746 000340      MOV      #340,-(SP)
015646 012746 016436      MOV      #TRPHAN,-(SP)
015652 016746 165262      MOV      ERRVEC,-(SP)
015656 012746 000003      MOV      #3,-(SP)
015662 104437      TRAP     C$SVEC
015664 062706 000010      ADD      #10,SP

682
683
684 015670 016702 165132      MOV      RLBAS,R2      ;NON-EXISTENT CONTROLLER UNIBUS
685 015674 005762 000000      TST      RLCS(R2)      ;/ADDRESS
686 015700 005767 165462      TST      TRPFLG      ;GET RL11 BASE ADDRESS
687 015704 001447      BEQ      1$           ;ACCESS DRIVE CONTROLLER UNIBUS ADDRESS
688 015706 012746 006754      PRINTF  #FMT24,#NOCTLR ;DID TRAP OCCUR?
015706 012746 006754      MOV      #NOCILR,-(SP) ;BRANCH TO CHECK DRIVE IF TRAP DID NOT OCCUR
                                ;ELSE, PRINT MSG. 'DRV DROPPED - NO CNTLR'

```

	015712	012746	012373		MOV	#FMT24,-(SP)	
	015716	012746	000002		MOV	#2,-(SP)	
	015722	010600			MOV	SP,RO	
	015724	104417			TRAP	C\$PNTF	
	015726	062706	000006		ADD	#6,SP	
689							:PRINT DRIVE INFORMATION
690	015732				PRINTF	#FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>	
	015732	005046			CLR	-(SP)	
	015734	156716	165073		BISB	RLDRV+1,(SP)	
	015740	012746	006053		MOV	#DRVNAM,-(SP)	
	015744	016746	165056		MOV	RLBAS,-(SP)	
	015750	012746	006042		MOV	#BASADD,-(SP)	
	015754	012746	011567		MOV	#FMT5,-(SP)	
	015760	012746	000005		MOV	#5,-(SP)	
	015764	010600			MOV	SP,RO	
	015766	104417			TRAP	C\$PNTF	
	015770	062706	000014		ADD	#14,SP	
691	015774				PRINTF	#FMT3	
	015774	012746	011553		MOV	#FMT3,-(SP)	
	016000	012746	000001		MOV	#1,-(SP)	
	016004	010600			MOV	SP,RO	
	016006	104417			TRAP	C\$PNTF	
	016010	062706	000004		ADD	#4,SP	
692	016014				DODU	PSETNM	:DO DROP UNIT ON DRIVE
	016014	016700	165342		MOV	PSETNM,RO	
	016020	104451			TRAP	C\$DODU	
693	016022	000460			BR	2\$	:BRANCH TO EXIT
694	016024	016705	165002	1\$:	MOV	RLDRV,R5	:ELSE, GET DRIVE NUMBER
695	016030	052705	000200		BIS	#CRDYMSK,R5	:SET CONTROLLER READY
696	016034	010562	000000		MOV	R5,RLCS(R2)	:LOAD IN THE DRIVE NUMBER
697	016040	032762	000001	000000	BIT	#DRDYMSK,RLCS(R2)	:IS DRIVE READY?
698	016046	001046			BNE	2\$	:BRANCH TO PERFORM TESTS IF DRIVE IS READY
699	016050				PRINTF	#FMT24,#NOTRDY	:PRINT MSG. 'DRV DROPPED - NOT RDY'
	016050	012746	007003		MOV	#NOTRDY,-(SP)	
	016054	012746	012373		MOV	#FMT24,-(SP)	
	016060	012746	000002		MOV	#2,-(SP)	
	016064	010600			MOV	SP,RO	
	016066	104417			TRAP	C\$PNTF	
	016070	062706	000006		ADD	#6,SP	
700							:PRINT DRIVE INFORMATION
701	016074				PRINTF	#FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>	
	016074	005046			CLR	-(SP)	
	016076	156716	164731		BISB	RLDRV+1,(SP)	
	016102	012746	006053		MOV	#DRVNAM,-(SP)	
	016106	016746	164714		MOV	RLBAS,-(SP)	
	016112	012746	006042		MOV	#BASADD,-(SP)	
	016116	012746	011567		MOV	#FMT5,-(SP)	
	016122	012746	000005		MOV	#5,-(SP)	
	016126	010600			MOV	SP,RO	
	016130	104417			TRAP	C\$PNTF	
	016132	062706	000014		ADD	#14,SP	
702	016136				PRINTF	#FMT3	
	016136	012746	011553		MOV	#FMT3,-(SP)	
	016142	012746	000001		MOV	#1,-(SP)	
	016146	010600			MOV	SP,RO	
	016150	104417			TRAP	C\$PNTF	
	016152	062706	000004		ADD	#4,SP	

```

703 016156          DODU      PSETNM      ;DO DROP UNIT ON DRIVE
    016156 016700 165200      MOV      PSETNM,RO
    016162 104451          TRAP     C$DODU
704 016164          2$:      CLRVEC    ;RELEASE THE ERROR VECTOR
    016164 016700 164750      MOV      ERRVEC,RO
    016170 104436          TRAP     C$CVEC
705 016172          ENDAUTO
    016172          L10016:
    016172 104461          TRAP     C$AUTO
706
707
708
709
710 016174          BGNMOD   CLNCODE
711 016174          BGNCLN
712
713 016174          SETVEC   ERRVEC,#TRPHAN,#340
    016174 012746 000340      MOV      #340,-(SP)
    016200 012746 016436      MOV      #TRPHAN,-(SP)
    016204 016746 164730      MOV      ERRVEC,-(SP)
    016210 012746 000003      MOV      #3,-(SP)
    016214 104437          TRAP     C$SVEC
    016216 062706 000010      ADD      #10,SP
714
715 016222          SETPRI   #7          ;SET PRIORITY TO 7
    016222 012700 000007      MOV      #7,RO
    016226 104441          TRAP     C$SPRI
716 016230 032762 000200 000000 2$:  BIT      #CRDYMSK,RLCS(R2)      ;TEST IF CONTROLLER READY
717 016236 001407          BEQ     3$          ;NO LOOP UNTIL READY
718 016240 056762 164566 000000      BIS      RLDRV,RLCS(R2)      ;SET DRIVE NUMBER
719 016246 032762 000001 000000      BIT      #DRDYMSK,RLCS(R2)    ;TEST IF DRIVE BUSY
720 016254 001027          BNE     5$          ;NO - SKIP
721 016256          3$:      WAITMS   3          ;WAIT 300 MS
    016274 012727 000372      MOV      #250.,(PC)+
    016300 000000          .WORD   0
    016302 016727 163610      MOV      L$DLY,(PC)+
    016306 000000          .WORD   0
    016310 005367 177772      DEC      -6(PC)
    016314 001375          BNE     -4
    016316 005367 177756      DEC      -22(PC)
    016322 001367          BNE     -20
722 016334          5$:      TRAP     C$BRK
    016334 016700 164470      CLRVEC   RLVEC          ;RELEASE DRIVE VECTOR
    016340 104436          MOV      RLVEC,RO
    016342 005767 165022      TRAP     C$CVEC
723 016346 001402          TST     PWRFLG          ;PWR FAIL SET
724 016350 005367 165014          BEQ     7$          ;NO
725 016354          7$:      DEC      PWRFLG
726 016354 016700 164560      CLRVEC   ERRVEC
    016354 104436          MOV      ERRVEC,RO
    016360          TRAP     C$CVEC
727 016362          ENDCLN
    016362          L10017:
    016362 104412          TRAP     C$CLEAN
728
729 016364          BGN$DU
730 016364 000240          NOP
    
```



```

731 016366          ENDDU
      016366          L10020:
      016366 104453    TRAP    C$DU
732
733 016370          ENDMOD
734
735
736
737
738
739 016370          .SBTTL  INTERRUPT SERVICE ROUTINES
740          BGNSRV  INTHLR
      016370 005067 164546 ;INTERRUPT HANDLER FOR DRIVE ABORTS WAIT TIMER AND STORES ALL RL11 REGISTERS
741 016370 005067 164546 CLR    DLYCNT          ;CLEAR UNELAPSED DELAY COUNT
742 016374 012267 164444 MOV    (R2)+,T.CS      ;STORE RL REGISTERS
743 016400 012267 164442 MOV    (R2)+,T.BA
744 016404 012267 164440 MOV    (R2)+,T.DA
745 016410 011267 164436 MOV    (R2),T.MP
746 016414 012767 164364 MOV    #-1,DONE        ;SET DONE FLAG
747 016422 016702 164400 MOV    RLBAS,R2        ;RESTORE R2
748 016426          ENDSRV
      016426          L10021:
      016426 000002    RTI
749
750          ;INTERRUPT SERVICE ROUTINE FOR P-CLOCK DECREMENTS DELAY COUNTER AT 100-MICROSECOND
751          ;TIME INTERVALS
752 016430          BGNSRV  CLKINT
      016430 005367 164506 DEC    DLYCNT          ;DECREMENT CLOCK DELAY COUNTER
753 016430 005367 164506
754 016434          ENDSRV
      016434          L10022:
      016434 000002    RTI
755
756          ;INTERRUPT SERVICE ROUTINE SETS TRAP FLAG WHEN A NON-EXISTENT UNIBUS ADDRESS IS
757          ;ACCESSED
758 016436          BGNSRV  TRPHAN
      016436 005267 164724 INC    TRPFLG          ;INDICATE THAT TRAP OCCURRED
759 016436 005267 164724
760 016442          ENDSRV
      016442          L10023:
      016442 000002    RTI
761
762
    
```

```

1          .SBTTL GLOBAL SUBROUTINES
2
3
4 016444   BGNMOD GLBSUB
5
6
7
8          :
9          : ERROR LIMIT CHECKING ROUTINE
10         : DROPS DRIVE IF ERROR LIMIT EXCEEDED
11 016444 027767 164506 175664 CKERLM: CMP @ERRPOINT,ERLIMW ;TEST IF ERROR LIMIT EXCEEDED
12 016452 002453 BLT 1$ ;NO - SKIP
13 016454 INLOOP ;CHECK IF IN ERROR LOOP
14 016454 104420 TRAP C$INLP
15 016456 BCOMPLETE 1$ ;YES - SKIP
16 016456 103451 BCS 1$
17 016460 PRINTF #FMT25,ERLIMW,#MEXERS ;PRINT MSG. 'OVER ERROR LIMIT - UNIT DROPPED'
18 016460 012746 011225 MOV #MEXERS,-(SP)
19 016464 016746 175646 MOV ERLIMW,-(SP)
20 016470 012746 012400 MOV #FMT25,-(SP)
21 016474 012746 000003 MOV #3,-(SP)
22 016500 010600 MOV SP,RO
23 016502 104417 TRAP C$PNTF
24 016504 062706 000010 ADD #10,SP
25 016510 PRINTF #FMT5,#BASADD,RLBAS,#DRVNM,<B,RLDRV+1> ;PRINT DRIVE INFORMATION
26 016510 005046 CLR -(SP)
27 016512 156716 164315 BISB RLDRV+1,(SP)
28 016516 012746 006053 MOV #DRVNM,-(SP)
29 016522 016746 164300 MOV RLBAS,-(SP)
30 016526 012746 006042 MOV #BASADD,-(SP)
31 016532 012746 011567 MOV #FMT5,-(SP)
32 016536 012746 000005 MOV #5,-(SP)
33 016542 010600 MOV SP,RO
34 016544 104417 TRAP C$PNTF
35 016546 062706 000014 ADD #14,SP
36 016552 PRINTF #FMT3
37 016552 012746 011553 MOV #FMT3,-(SP)
38 016556 012746 000001 MOV #1,-(SP)
39 016562 010600 MOV SP,RO
40 016564 104417 TRAP C$PNTF
41 016566 062706 000004 ADD #4,SP
42 016572 DODU PSETNM ;DROP DRIVE
43 016572 016700 164564 MOV PSETNM,RO
44 016576 104451 TRAP C$DODU
45 016600 DOCLN ;GO TO CLEAN UP
46 016600 104444 TRAP C$DCLN
47 016602 000207 1$: RTS PC
48
49
50         : READ AND STORE ALL RL11 REGISTERS
51 016604 016267 000000 164232 READRL: MOV RLCSR(R2),T,CS ;GET CS REG
52 016612 016267 000002 164226 MOV RLBA(R2),T,BA ;GET BUS ADDRESS REG
53 016620 016267 000004 164222 MOV RLDA(R2),T,DA ;GET DISK ADDRESS
54 016626 016267 000006 164216 MOV RLMP(R2),T,MP ;GET MULTI-PURPOSE REG
55 016634 000207 RTS PC ;RETURN
56
57
58
59
60
    
```

```

31
32
33 016636 011646          ; WAITIN: WAIT FOR CONTROLLER TIMEOUT TO FORCE INTERRUPT ROUTINE
34 016640 005066 000002   ; MOV (SP),-(SP) ;MAKE ROOM FOR ERROR POINTER
35 016644 032762 000200 000000 ; CLR 2(SP) ;CLEAR FOR POINTER
36 016652 001420          ; BIT #CRDYMSK,RLCSR(R2) ;TEST IF CONTROLLER READY
37 016654 004767 177724   ; BEQ 4$ ;NO - SKIP TO WAIT
38 016660 005767 164122   ; JSR PC,READRL ;READ ALL RL REGS
39 016664 001453          ; TST DONE ;TEST IF INTERRUPT OCCURRED
40 016666 012766 006200 000002 1$: ; MOV #MTOSLOW,2(SP) ;ELSE SET TOO SLOW ERROR POINTER
41 016674 032767 002000 164142 ; BIT #OPIERR,T.CS ;TEST IF OPI SET
42 016702 001403          ; BEQ 2$ ;NO - SKIP
43 016704 012766 006220 000002 2$: ; MOV #MDRRES,2(SP) ;SET MESSAGE FOR NO DRIVE RESPONSE
44 016712 000207          ; RTS PC ;RETURN
45 016714 012767 000001 164220 4$: ; MOV #1,DLYCNT ;INITIALIZE DELAY COUNT
46 016722 006367 164214   ; ASL DLYCNT ;MULTIPLY BY 2
47 016726 006367 164210   ; ASL DLYCNT ;MULTIPLY BY 2 AGAIN
48 016732 012727 000012   ; MOV #10.,(PC)+ ;IMPLEMENT TIME DELAY LOGP
49 016736 000000          ; .WORD 0
50 016740 016727 163152   ; MOV L$DLY,(PC)+
51 016744 000000          ; .WORD 0
52 016746 005367 177772   ; DEC -6(PC)
53 016752 001375          ; BNE -4
54 016754 005367 177756   ; DEC -22(PC)
55 016760 001367          ; BNE -20
56 016762 032762 000200 000000 ; BIT #CRDYMSK,RLCSR(R2) ;TEST IF READY NOW SET
57 016770 001906          ; BNE 3$ ;YES - SKIP
58 016772 004767 177606   ; JSR PC,READRL ;READ RL REGS
59 016776 012766 006273 000002 ; MOV #MCONHNG,2(SP) ;SET MESSAGE FOR CONTROLLER HUNG
60 017004 000742          ; BR 2$ ;SKIP
61 017006 005767 163774   ; TST DONE ;ELSE CHECK IF INTERRUPT OCCURRED
62 017012 001325          ; BNE 1$ ;YES - SKIP TO SET TOO SLOW
63 017014 004767 177564   ; JSR PC,READRL ;READ RL REGS
64 017020 012766 006240 000002 ; MOV #MNOINT,2(SP) ;ELSE SET NO INTERRUPT FLAG
65 017026 000731          ; BR 2$ ;GO TO RETURN
66
67
68
69
70 017030 005067 163750   ; TSTINT: OPERATION AND TEST INITIALIZE ROUTINE
71 017034 105067 164325   ; CLR OPFLAG ;CLEAR OPERATION FLAGS
72 017040 005067 163750   ; CLRB NOERCT ;RESET INHIBIT ERROR COUNTING
73 017044 000207          ; CLR MORECE ;RESET MORE COMPARE ERRORS
74
75
76
77
78 017046 016746 164054   ; GSTATR: GET STATUS AND GET STATUS WITH RESET ROUTINE
79 017052 012767 000013 164046 ; MOV TEMP4,-(SP) ;STORE TEMP4
80 017060 000412          ; BR #GETSTAT!DRSET,TEMP4 ;SET FOR RESET
81 017062 016746 164040   ; GSTATC: MOV GSTATG
82 017066 012767 000003 164032 ; MOV TEMP4,-(SP) ;STORE TEMP4
83 017074 000404          ; BR #GETSTAT,TEMP4 ;SET FOR NO RESET
84 017076 016746 164024   ; GSTAT: MOV GSTATG
85 017102 005067 164020   ; MOV TEMP4,-(SP) ;STORE TEMP4
86 017106 010346          ; CLR TEMP4 ;SET FOR SAVE L. AND T. REGS
87 017110 016703 163666   ; GSTATG: MOV R3,-(SP) ;STORE R3
                        ; MOV SSINDX,R3 ;GET SUBROUTINE INDEX
    
```

```

88 017114 005723          TST      (R3)+          ;BUMP IT FOR NEXT ENTRY
89 017116 016663 000004 002404  MOV     4(SP),SUBSTK(R3) ;INSERT THIS CALL
90 017124 162763 000004 002404  SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
91 017132 010367 163644          MOV     R3,SSINDX      ;STORE IT BACK
92 017136 010046          MOV     R0,-(SP)       ;STORE R0
93 017140 010146          MOV     R1,-(SP)       ;STORE R1
94 017142 012767 000002 163646  MOV     #2,ERRSW1     ;SET FOR NO ERROR RETURN
95 017150 032767 000010 163750  BIT     #DRSET,TEMP4   ;TEST IF DRIVE RESET
96 017156 001525          BEQ     11$            ;NO - SKIP
97 017160 032762 040000 000000  BIT     #DRVERR,RLCS(R2) ;TEST IF DRIVE ERROR SET
98 017166 001427          BEQ     49$            ;NO - SKIP
99 017170          WAITMS 3             ;WAIT FOR DRIVE TO SETTLE
    017206 012727 000372          MOV     #250.,(PC)+
    017212 000000          .WORD 0
    017214 016727 162676          MOV     L$DLY,(PC)+
    017220 000000          .WORD 0
    017222 005367 177772          DEC     -6(PC)
    017226 001375          BNE     -4
    017230 005367 177756          DEC     -22(PC)
    017234 001367          BNE     -20
    017236 104422          TRAP   C$BRK
100          .NLIST
101 017246 012701 000062 49$:  MOV     #50.,R1        ;INITIALIZE WAIT COUNTER
102 017252 004767 177620 50$:  JSR     PC,GSTAT      ;GET DRIVE STATUS
103 017256 020110          3$
104 017260 032767 000001 163556  BIT     #DRDYMSK,T.CS  ;TEST IF DRIVE READY
105 017266 001077          BNE     5$            ;YES - GO DO CLEAR
106 017270 032767 000020 163554  BIT     #HOSTAT,T.MP   ;ELSE TEST IF HEADS OUT
107 017276 001010          BNE     51$           ;YES - BYPASS RELOAD WAIT FLAG SETTING
108 017300 032767 144000 163544  BIT     #SPDSTAT!HCESTAT!WDESTAT,T.MP ;TEST IF DRIVE HAS ERROR
109          ;THAT CAUSED HEADS TO
110          ;UNLOAD
111 017306 001467          BEQ     5$            ;NO - SKIP
112 017310 052767 040000 163466  BIS     #RELDWT,OPFLAG ;ELSE SET WAIT FLAG
113 017316 000463          BR     5$            ;SKIP TO CLEAR
114 017320 032767 040000 163516 51$: BIT     #DRVERR,T.CS  ;TEST IF DRIVE ERROR NOW
115 017326 001057          BNE     5$            ;YES - SKIP TO CLEAR
116 017330          WAITMS 1             ;WAIT FOR DRIVE TO GET ERROR, READY, OR HEADS OUT
    017346 012727 000372          MOV     #250.,(PC)+
    017352 000000          .WORD 0
    017354 016727 162536          MOV     L$DLY,(PC)+
    017360 000000          .WORD 0
    017362 005367 177772          DEC     -6(PC)
    017366 001375          BNE     -4
    017370 005367 177756          DEC     -22(PC)
    017374 001367          BNE     -20
    017376 104422          TRAP   C$BRK
117 017406 005301          DEC     R1            ;DEC WAIT COUNTER
118 017410 001320          BNE     50$           ;IF NOT DONE, LOOP
119 017412 012703 011103          MOV     #MUNDEF,R3    ;MESSAGE FOR UNDEFINED STATE
120 017416          ERRHRD 10001.,,ERR1
    017416 104456          TRAP   C$ERHRD
    017420 023421          .WORD 10001
    017422 000000          .WORD 0
    017424 012464          .WORD ERR1
121 017426 000167 000452          JMP     14$            ;EXIT
122 017432 005767 163470 11$: TST     TEMP4         ;TEST IF SAVE REGISTERS
    
```

123	017436	001013				BNE	5\$		;NO SKIP
124	017440	012701	000004			MOV	#4,R1		;SET SAVE COUNT
125	017444	012703	003044			MOV	#L.MP+2,R3		;SET ADDRESS OF FIRST SAVE
126	017450	014346			8\$:	MOV	-(R3),-(SP)		;PUT REG ON STACK
127	017452	005301				DEC	R1		;DEC COUNT
128	017454	001375				BNE	8\$		;LOOP UNTIL ALL SAVED
129	017456	012767	000003	163354		MOV	#GETSTAT,L.DA		;SET FOR GET STATUS
130	017464	000403				BR	6\$		;SKIP
131	017466	016767	163434	163344	5\$:	MOV	TEMP4,L.DA		;INSERT PRESET FOR STATUS
132	017474				6\$:				
133	017474	005067	163306			CLR	DONE		;CLEAR INTERRUPT FLAG
134	017500	016767	163326	163326		MOV	RLDRV,L.CS		;SET UP TO GET STATUS
135	017506	042767	002000	163320		BIC	#BIT10,L.CS		;CLEAR FOR DRIVE 4 - 7 SPEC'D
136	017514	052767	000104	163312		BIS	#GTSTAT,L.CS		
137	017522	016762	163312	000004		MOV	L.DA,RLDA(R2)		;LOAD RL REGS
138	017530	016762	163300	000000		MOV	L.CS,RLCSR(R2)		;LOAD CS REG
139	017536					WAITUS	1		;WAIT FOR INTERRUPT
	017536	012727	000001			MOV	#1,(PC)+		
	017542	000000				.WORD	0		
	017544	016727	162346			MOV	L\$DLY,(PC)+		
	017550	000000				.WORD	0		
	017552	005367	177772			DEC	-6(PC)		
	017556	001375				BNE	-.4		
	017560	005367	177756			DEC	-22(PC)		
	017564	001367				BNE	-.20		
140	017566	005767	163214			TST	DONE		;CHECK IF INTERRUPT OCCURRED
141	017572	001535				BEQ	1\$		;NO - SKIP
142	017574	016767	163252	163256	4\$:	MOV	T.MP,T.STAT		;STORE MP REGISTER
143	017602	042767	177770	163250		BIC	#C<STAMSK>,T.STAT		;CLEAR ALL BUT STATE
144	017610	032767	000010	163222		BIT	#DRSET,L.DA		;TEST IF RESET WAS SPECIFIED
145	017616	001534				BEQ	3\$		;NO - SKIP TO EXIT
146	017620	032767	040000	163156		BIT	#RELDWT,OPFLAG		;TEST IF RELOAD WAIT FLAG SET
147	017626	001451				BEQ	12\$		;NO - SKIP
148	017630	012701	001130			MOV	#600.,R1		;INITIALIZE WAIT COUNTER
149	017634	032762	000001	000000	13\$:	BIT	#DRDYMSK,RLCS(R2)		;TEST IF DRIVE NOW READY
150	017642	001043				BNE	12\$		;YES - SKIP
151	017644					WAITMS	1		;CALL WAIT
	017662	012727	000372			MOV	#250.,(PC)+		
	017666	000000				.WORD	0		
	017670	016727	162222			MOV	L\$DLY,(PC)+		
	017674	000000				.WORD	0		
	017676	005367	177772			DEC	-6(PC)		
	017702	001375				BNE	-.4		
	017704	005367	177756			DEC	-22(PC)		
	017710	001367				BNE	-.20		
	017712	104422				TRAP	C\$BRK		
152	017722	005301				DEC	R1		;DEC COUNT
153	017724	001343				BNE	13\$		;LOOP IF NOT 0
154	017726	004767	177144			JSR	PC,GSTAT		;GET DRIVE STATUS
155	017732	020110				3\$			;ERROR RETURN
156	017734	012703	011150			MOV	#MRLFAL,R3		;SET RESULT MESSAGE POINTER
157	017740					ERRHRD	10003.,,ERR1		
	017740	104456				TRAP	C\$ERHRD		
	017742	023423				.WORD	10003		
	017744	000000				.WORD	0		
	017746	012464				.WORD	ERR1		
158	017750	000455				BR	14\$		;GO TO EXIT

159	017752			12\$:	WAITUS	10.		:WAIT
	017752	012727	000012		MOV	#10.,(PC)+		
	017756	000000			.WORD	0		
	017760	016727	162132		MOV	L\$DLY,(PC)+		
	017764	000000			.WORD	0		
	017766	005367	177772		DEC	-6(PC)		
	017772	001375			BNE	-.4		
	017774	005367	177756		DEC	-22(PC)		
	020000	001367			BNE	.-20		
160	020002	004767	177070		JSR	PC,GSTAT		:GET DRIVE STATUS
161	020006	020110			3\$			
162	020010	032767	100000	163026	BIT	#ANYERR,T.CS		:TEST IF ANY ERROR
163	020016	001434			BEQ	3\$		:NO - SKIP
164	020020	032767	001000	163024	BIT	#VCSTAT,T.MP		:CHECK IF VOLUME CHECK RESET
165	020026	001403			BEQ	7\$		:YES SKIP
166	020030	012703	006327		MOV	#VCNRST,R3		:SET REASON POINTER
167	020034	000417			BR	2\$		:EXIT
168	020036	032767	040000	163000	7\$:	BIT	#DRVERR,T.CS	:CHECK IF DRIVE ERROR
169	020044	0C1405			BEQ	9\$		:NO - SKIP
170	020046				ERRHRD	10004.,,ERR6		
	020046	104456			TRAP	C\$ERHRD		
	020050	023424			.WORD	10004		
	020052	00C070			.WORD	0		
	020054	012766			.WORD	ERR6		
171	020056	000412			BR	14\$		:EXIT
172	020060	012703	006350	9\$:	MOV	#UNXERR,R3		:SET REASON POINTER
173	020064	000403			BR	2\$		:EXIT
174	020066	004767	176544	1\$:	JSR	PC,WAITIN		:WAIT FOR INTERRUPT
175	020072	012603			MOV	(SP)+,R3		:STORE REASON POINTER FOR RETURN
176	020074			2\$:	ERRHRD	10002.,,ERR1		
	020074	104456			TRAP	C\$ERHRD		
	020076	023422			.WORD	10002		
	020100	000000			.WORD	0		
	020102	012464			.WORD	ERR1		
177	020104	005067	162706	14\$:	CLR	ERRSWI		:CLEAR FOR ERROR RETURN
178	020110	005767	163012	3\$:	TST	TEMP4		:TEST IF REGISTERS WERE SAVED
179	020114	001007			BNE	22\$		:NO - SKIP
180	020116	012703	003034		MOV	#L.CS,R3		:SET POINTER TO RESTORE
181	020122	012701	000004		MOV	#4,R1		:SET REGISTER COUNT
182	020126	012623		20\$:	MOV	(SP)+,(R3)+		:RESTORE REG
183	020130	005301			DEC	R1		:DEC COUNT
184	020132	001375			BNE	20\$		:LOOP UNTIL ALL ARE RESTORED
185	020134	162767	000002	162640	22\$:	SUB	#2,SSIDX	:REMOVE ENTRY FROM SUBROUTINE STACK
186	020142	012601			MOV	(SP)+,R1		:RESTORE R1
187	020144	012600			MOV	(SP)+,R0		:RESTORE R0
188	020146	012603			MOV	(SP)+,R3		:RESTORE R3
189	020150	012667	162752		MOV	(SP)+,TEMP4		:RESTORE TEMP4
190	020154	005767	162636		TST	ERRSWI		:TEST IF ERROR RETURN
191	020160	001404			BEQ	99\$		:YES - SKIP
192	020162	066716	162630		ADD	ERRSWI,(SP)		:ADD IN FOR NO ERROR RETURN
193	020166	000240			NOP			
194	020170	000207			RTS	PC		
195	020172	017616	000000	99\$:	MOV	@(SP),(SP)		:SET ERROR RETURN ADDRESS
196	020176	000240			NOP			
197	020200	000207			RTS	PC		
198								
199								

```

200
201
202 020202 010346          : GDRSTA: GET DRIVE STATE ROUTINE
203 020204 012701 000004  : MOV R3,-(SP) ;SAVE R3
204 020210 012703 003044  : MOV #4,R1 ;INITIALIZE REGISTER SAVE COUNT
205 020214 014346          : MOV #L.MP+2,R3 ;INITIALIZE ADDRESS OF FIRST SAVE
206 020216 005301          : 1$: MOV -(R3),-(SP) ;SAVE REGISTER ON STACK
207 020220 001375          : DEC R1 ;DECREMENT REGISTER SAVE COUNT
208 020222 012767 000003 162610 : BNE 1$ ;LOOP UNTIL ALL 4 REGISTERS ARE SAVED
209          : MOV #GETSTAT,L.DA ;SET UP DISK ADDRESS REGISTER FOR GET STATUS
210 020230 005067 162552          : CLR DONE ;/COMMAND
211 020234 016767 162572 162572 : MOV RLDRV,L.CS ;CLEAR INTERRUPT FLAG
212          : ;SET UP CONTROL STATUS REGISTER WITH
213 020242 042767 002000 162564 : BIC #BIT10,L.CS ;DRIVE NUMBER
214 020250 052767 000104 162556 : BIS #GTSTAT,L.CS ;CLEAR FOR DRIVES 4-7 SPECIFIED
215          : ;INITIALIZE CONTROL STATUS REGISTER FOR
216 020256 016762 162556 000004 : MOV L.DA,RLDA(R2) ;INITIALIZE DISK ADDRESS REGISTER FOR
217          : ;/GET STATUS COMMAND
218 020264 016762 162544 000000 : MOV L.CS,RLCSR(R2) ;LOAD CONTROL STATUS REGISTER TO EXECUTE
219          : ;/GET STATUS COMMAND
220 020272 105762 000000          : 5$: TSTB RLCS(R2) ;WAIT FOR CONTROLLER READY INDICATING
221 020276 001775          : BEQ 5$ ;/RECEIPT OF GET STATUS COMMAND
222 020300 005767 162502          : TST DONE ;INTERRUPT OCCURRED?
223 020304 001416          : BEQ 3$ ;BRANCH IF NOT
224 020306 016767 162540 162544 : MOV T.MP,T.STAT ;GET CONTENTS OF MULTI-PURPOSE REGISTER
225 020314 042767 177770 162536 : BIC #*C<STAMSK>,T.STAT ;CLEAR ALL BUT STATE DRIVE BITS
226 020322 012703 003034          : MOV #L.CS,R3 ;INITIALIZE POINTER TO RESTORE RL REGISTERS
227 020326 012701 000004          : MOV #4,R1 ;INITIALIZE REGISTER SAVE COUNT
228 020332 012623          : 2$: MOV (SP)+,(R3)+ ;RESTORE REGISTERS
229 020334 005301          : DEC R1 ;DECREMENT REGISTER SAVE COUNT
230 020336 001375          : BNE 2$ ;LOOP UNTIL ALL 4 REGISTERS ARE RESTORED
231 020340 000402          : BR 4$
232 020342 004767 176270          : 3$: JSR PC,WAITIN ;WAIT FOR INTERRUPT
233 020346 012603          : 4$: MOV (SP)+,R3 ;RESTORE R3
234 020350 000207          : RTS PC ;RETURN
235
236
237
238
239 020352 012767 177777 162540 : XSEEK: SEEK ROUTINE
240 020360 000402          : BR XSEEK1 ;SET SPECIAL TIMING SEEK FLAG
241 020362 005067 162532          : XSEEK: CLR TEMP1 ;CLEAR SPECIAL TIMING SEEK FLAG
242 020366 010346          : XSEEK1: MOV R3,-(SP) ;STORE R3
243 020370 016703 162406          : MOV SSINDX,R3 ;GET SUBROUTINE INDEX
244 020374 005723          : TST (R3)+ ;BUMP IT FOR NEXT ENTRY
245 020376 016663 000002 002404 : MOV 2(SP),SUBSTK(R3) ;INSERT THIS CALL
246 020404 162763 000004 002404 : SUB #4,SUBSTK(R3) ;ADJUST IT TO CALLING LOCATION
247 020412 010367 162364          : MOV R3,SSINDX ;STORE IT BACK
248 020416 010046          : MOV R0,-(SP)
249 020420 010146          : MOV R1,-(SP)
250 020422 010546          : MOV R5,-(SP) ;STORE REG
251 020424 012767 000002 162364 : MOV #2,ERRSWI ;SET FOR NO ERROR RETURN
252 020432 005067 162440          : CLR DIFAUG ;CLEAR DIFFERENCE ARGUMENT (FOR SEEKING
253          : ; PAST GUARD BAND)
254 020436 004767 002560          : JSR PC,GETPOS ;GET PRESENT POSITION
255 020442 021112          : 65$
256 020444 016767 162434 162426 : MOV CURCYL,OLDCYL ;MOVE CURRENT TO OLD CYLINDER
    
```

```

257 020452 026767 162424 161622    CMP    NEWCYL,HLMTW    ;TEST IF NEW IS GREATER THAN 255
258 020460 003427                    BLE    3$             ;NO - SKIP
259 020462 166767 161614 162412    SUB    HLMTW,NEWCYL   ;ELSE SUBTRACT 255.
260 020470 016767 162406 162400    MOV    NEWCYL,DIFAUG ;STORE DIFFERENCE AS ARGUMENT
261 020476 016767 161600 162376    MOV    HLMTW,NEWCYL  ;SET NEWCYL AS 255.
262 020504 022767 000001 161564    CMP    #1,T.DRIVE
263 020512 001424                    BEQ    6$
264 020514 162767 000001 162360    SUB    #1,NEWCYL
265 020522 012767 000001 162360    MOV    #1,DESSGN
266 020530 012767 000001 162350    MOV    #1,DESDIF
267 020536 000451                    BR     18$
268 020540 005767 162336            3$:    TST    NEWCYL        ;TEST IF NEWCYL HAS NEGATIVE VALUE
269 020544 100007                    BPL    6$             ;NO - SKIP
270 020546 005467 162330            NEG    NEWCYL        ;ELSE MAKE IT POSITIVE
271 020552 016767 162324 162316    MOV    NEWCYL,DIFAUG ;AND STORE IT AS ARGUMENT
272 020560 005067 162316            CLR    NEWCYL        ;AND SET NEWCYL TO 0
273 020564 016705 162314            6$:    MOV    CURCYL,R5    ;COMPUTE DIFFERENCE AND NEW CYLINDER
274 020570 166705 162306            SUB    NEWCYL,R5    ;SUB NEWCYL FROM CURCYL
275 020574 100005                    BPL    13$           ;IF DIFF IS POSITIVE - SKIP(REV SEEK)
276 020576 012767 000001 162304    MOV    #1,DESSGN    ;ELSE SET SIGN FOR FORWARD
277 020604 005405                    NEG    R5            ;MAKE DIFFERENCE POSITIVE
278 020606 000402                    BR     14$           ;SKIP
279 020610 005067 162274            13$:   CLR    DESSGN        ;SET SIGN FOR REVERSE
280 020614 010567 162266            14$:   MOV    R5,DESDIF    ;STORE DIFFERENCE
281 020620 005767 162252            TST    DIFAUG        ;IS THERE A DIFFERENCE ARGUMENT
282 020624 001416                    BEQ    18$           ;NO - SKIP
283 020626 026767 162250 161446    CMP    NEWCYL,HLMTW ;CHECK IF NEW CYL IS 255.
284 020634 001007                    BNE    17$           ;NO - SKIP
285 020636 012767 000001 162244    MOV    #1,DESSGN    ;ELSE FORCE SIGN FOR FORWARD
286                                     ;(INNER GUARD BAND)
287 020644 022767 000001 161424    CMP    #1,T.DRIVE
288 020652 001003                    BNE    18$
289 020654 066767 162216 162224    17$:   ADD    DIFAUG,DESDIF
290 020662                                18$:
291 020662 012705 003034            MOV    #L.CS,R5     ;GET RL REG ADDRESS
292 020666 012715 000106            MOV    #SEEK,(R5)   ;SET FOR SEEK
293 020672 056715 162134            BIS    RLDRV,(R5)   ;INSERT DRIVE NUMBER
294 020676 042725 002000            BIC    #BIT10,(R5)+ ;CLEAR IF DRIVE 4 - 7 SPEC'D
295 020702 005025                    CLR    (R5)+        ;CLEAR BUS ADDRESS
296 020704 016715 162176            MOV    DESDIF,(R5) ;LOAD DIFFERENCE
297 020710 012700 000007            MOV    #7,R0        ;SET TO SHIFT DIFFERENCE
298 020714 006315                    21$:   ASL    (R5)
299 020716 005300                    DEC    R0
300 020720 001375                    BNE    21$           ;LOOP UNTIL ALIGNED
301 020722 005767 162162            TST    DESSGN        ;TEST SIGN
302 020726 001402                    BEQ    23$           ;SKIP IF 0
303 020730 052715 000004            BIS    #DIRBIT,(R5) ;ELSE INSERT SIGN
304 020734 005767 162152            23$:   TST    DESHD        ;TEST IF HEAD 0
305 020740 001402                    BEQ    25$           ;YES - SKIP
306 020742 052715 000020            BIS    #HDSSEL,(R5) ;ELSE SET HEAD BIT
307 020746 052725 000001            25$:   BIS    #MBSSET0,(R5)+ ;INSERT MARKER BIT
308 020752 004767 000504            JSR    PC,RDYCHK    ;CHECK IF DRIVE READY
309 020756 021112                    65$:   CLR    DONE
310 020760 005067 162022            CLR    TEMP1        ;CLEAR INTERRUPT FLAG
311 020764 005767 162130            TST    TEMP1        ;CHECK IF SPECIAL SEEK FLAG SET
312 020770 001050                    BNE    65$           ;YES - SKIP, DO NOT START SEEK
313 020772 014562 000004            MOV    -(R5),RLDA(R2) ;LOAD RL REGISTERS
    
```



314	020776	014562	00C002		MOV	-(R5),RLBA(R2)	
315	021002	014562	000000		MOV	-(R5),RLCS(R2)	;PERFORM SEEK OPERATION
316	021006			30\$:	WAITUS	1	;ALLOW TIME FOR RECEIPT OF SEEK COMMAND
	021006	012727	000001		MOV	#1,(PC)+	
	021012	000000			.WORD	0	
	021014	016727	161076		MOV	L\$DLY,(PC)+	
	021020	000000			.WORD	0	
	021022	005367	177772		DEC	-6(PC)	
	021026	001375			BNE	.-4	
	021030	005367	177756		DEC	-22(PC)	
	021034	001367			BNE	.-20	
317	021036	005767	161744		TST	DONE	;TEST IF INTERRUPT DONE
318	021042	001012			BNE	32\$	;YES - SKIP
319	021044	004767	175566		JSR	PC,WAITIN	;GO WAIT FOR INTERRUPT
320	021050	012603			MOV	(SP)+,R3	;GET RESULT MESSAGE POINTER
321	021052				ERRHRD	10005,,,ERR1	
	021052	104456			TRAP	C\$ERHRD	
	021054	023425			.WORD	10005	
	021056	000000			.WORD	0	
	021060	012464			.WORD	ERR1	
322	021062	005067	161730		CLR	ERRSWI	;CLEAR FOR ERROR RETURN
323	021066	000411			BR	65\$	
324	021070	005767	161750	32\$:	TST	T.CS	;TEST IF ANY ERROR
325	021074	100006			BPL	65\$	;NO - SKIP
326	021076				ERRHRD	10006,,,ERR6	
	021076	104456			TRAP	C\$ERHRD	
	021100	023426			.WORD	10006	
	021102	000000			.WORD	0	
	021104	012766			.WORD	ERR6	
327	021106	005067	161704		CLR	ERRSWI	;CLEAR FOR ERROR RETURN
328	021112	162767	000002	161662	65\$:	SUB	#2,SSINDX
329	021120	012605			MOV	(SP)+,R5	;RESTORE REGISTER
330	021122	012601			MOV	(SP)+,R1	
331	021124	012600			MOV	(SP)+,R0	
332	021126	012603			MOV	(SP)+,R3	;RESTORE R3
333	021130	005767	161662		TST	ERRSWI	;TEST IF ERROR RETURN
334	021134	001403			BEO	99\$	;YES - SKIP
335	021136	066716	161654		ADD	ERRSWI,(SP)	;ADD IN ERROR RETURN
336	021142	000207			RTS	PC	
337	021144	017616	000000	99\$:	MOV	@(SP),(SP)	;SET ERROR RETURN ADDRESS
338	021150	000207			RTS	PC	
339							
341							
342							
343	021152	010346		SIMSEK:	MOV	R3,-(SP)	;STORE REGISTERS
344	021154	016703	161622		MOV	SSINDX,R3	;GET SUBROUTINE INDEX
345	021160	005723			TST	(R3)+	;BUMP IT FOR NEXT ENTRY
346	021162	016663	000002	002404	MOV	2(SP),SUBSTK(R3)	;INSERT THIS CALL
347	021170	162763	000004	002404	SUB	#4,SUBSTK(R3)	;ADJUST IT TO CALLING LOCATION
348	021176	010367	161600		MOV	R3,SSINDX	;STORE IT BACK
349	021202	010046			MOV	R0,-(SP)	
350	021204	010446			MOV	R4,-(SP)	
351	021206	012767	000002	161602	MOV	#2,ERRSWI	;SET FOR NO ERROR RETURN
352	021214	004767	000242		JSR	PC,RDYCHK	;CHECK IF DRIVE READY
353	021220	021424			65\$		
354	021222	012704	003034		MOV	#L.CS,R4	;GET POINTER TO L REGS
355	021226	012714	000106		MOV	#SEEK,(R4)	;SET FOP SEEK

```

356 021232 056714 161574      BIS      RLDRV,(R4)      ;INSERT DRIVE NUMBER
357 021236 042724 002000      BIC      #BIT10,(R4)+   ;CLEAR FOR DRIVE 4 - 7 SPEC'D
358 021242 005024              CLR      (R4)+          ;CLEAR BUS ADDRESS
359 021244 016714 161636      MOV      DESDIF,(R4)    ;LOAD DIFFERENCE
360 021250 012703 000007      MOV      #7,R3         ;SET COUNT FOR SHIFT TO ALIGN
361 021254 006314              3$:     ASL      (R4)        ;ALIGN DIFFERENCE IN DA
362 021256 005303              DEC      R3
363 021260 001375              BNE     3$
364 021262 005767 161622      TST     DESSGN         ;TEST IF SIGN SET
365 021266 001402              BEQ     5$             ;NO - SKIP
366 021270 052714 000004      BIS      #DIRBIT,(R4)   ;INSERT SIGN
367 021274 005767 161612      5$:     TST     DESHD     ;TEST IF HEAD 0
368 021300 001402              BEQ     7$             ;YES - SKIP
369 021302 052714 000020      BIS      #HSEL,(R4)     ;INSERT HEAD BIT
370 021306 052724 000001      7$:     BIS      #MBSET0,(R4)+ ;INSERT MARKER BIT
371 021312 005067 161470      CLR     DONE           ;CLEAR INTERRUPT FLAG
372 021316 012701 000012      MOV     #10,R1         ;SET WAIT COUNT FOR 800US
373 021322 014462 000004      MOV     -(R4),RLDA(R2) ;LOAD RL REGISTERS
374 021326 014462 000002      MOV     -(R4),RLBA(R2)
375 021332 014462 000000      MOV     -(R4),RLCS(R2)
376 021336 005767 161444      10$:    TST     DONE         ;CHECK IF INTERRUPTED
377 021342 001030              BNE     65$           ;YES - SKIP
378 021344 005301              DEC     R1             ;DEC WAIT COUNT
379 021346 001415              BEQ     13$           ;IF 0 - SKIP
380 021350              WAITUS 1
      021350 012727 000001      MOV     #1,(PC)+
      021354 000000              .WORD 0
      021356 016727 160534      MOV     L$DLY,(PC)+
      021362 000000              .WORD 0
      021364 005367 177772      DEC     -6(PC)
      021370 001375              BNE     -4
      021372 005367 177756      DEC     -22(PC)
      021376 001367              BNE     -20
381 021400 000756              BR      10$           ;GO CHECK DONE
382 021402 004767 175230      13$:    JSR     PC,WAITIN  ;GO WAIT FOR TIMEOUT
383 021406 012603              MOV     (SP)+,R3      ;GET RESULT MESSAGE POINTER
384 021410              ERRHRD 10011,ERR1
      021410 104456              TRAP   C$ERRRD
      021412 023433              .WORD 10011
      021414 000000              .WORD 0
      021416 012464              .WORD ERR1
385 021420 005067 161372      CLR     ERRSWI        ;CLEAR FOR ERROR RETURN
386 021424              14$:
387 021424 162767 000002 161350 65$:    SUB     #2,SSINDEX    ;REMOVE ENTRY FROM SUBROUT STACK
388 021432 012604              MOV     (SP)+,R4     ;RESTORE REGS
389 021434 012600              MOV     (SP)+,R0
390 021436 012603              MOV     (SP)+,R3
391 021440 005767 161352      TST     ERRSWI       ;TEST IF ERROR RETURN
392 021444 001403              BEQ     99$           ;YES - SKIP
393 021446 066716 161344      ADD     ERRSWI,(SP)   ;ADD IN ERROR RETURN
394 021452 000207              RTS     PC
395 021454 017616 000000      99$:    MOV     @ (SP),(SP)  ;SET ERROR RETURN ADDRESS
396 021460 000207              RTS     PC
398
474
475
476
; DRIVE READY TEST ROUTINE. CHECKS DRIVE IS READY. IF NOT, WAIT
    
```

```

477      ;RDYCHK: 500MS FOR READY TO SET.
478 021462 010346      MOV R3,-(SP)      ;STORE REGS
479 021464 016703 161312  MOV SSINDY,R3      ;GET SUBROUTINE INDEX
480 021470 005723      TST (R3)+          ;BUMP IT FGR NEXT ENTRY
481 021472 016663 000002 002404  MOV 2(SP),SUBSTK(R3) ;INSERT THIS CALL
482 021500 162763 000004 002404  SUB #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
483 021506 010367 161270      MOV R3,SSINDX      ;STORE IT BACK
484 021512 010046      MOV R0,-(SP)
485 021514 010146      MOV R1,-(SP)
486 021516 010446      MOV R4,-(SP)
487 021520 012767 000002 161270  MOV #2,ERRSWI      ;SET FOR NO ERROR RETURN
488 021526 012701 011610      MOV #5000.,R1      ;SET WAIT COUNT
489 021532 004767 175340      JSR PC,GSTAT      ;GET DRIVE STATUS
490 021536 021754      4$
491 021540 032767 000001 161276  BIT #DRDYMSK,T.CS ;TEST IF DRIVE READY
492 021546 001104      BNE 5$            ;YES - EXIT
493 021550      WAITUS 1
      MOV #1,(PC)+
      .WORD 0
      MOV L$DLY,(PC)+
      .WORD 0
      DEC -6(PC)
      BNE -4
      DEC -22(PC)
      BNE -20
494 021600 005301      DEC R1            ;DEC WAIT COUNT
495 021602 001353      BNE 1$          ;LOOP IF NOT 0
496 021604 012703 010404      MOV #MDRDY,R3    ;SET RESULT MESSAGE POINTER
497 021610 012704 011404      MOV #C500MS,R4  ;SET CONDITION MESSAGE POINTER
498 021614      ERRHRD 10010.,,ERR5
      TRAP C$ERHRD
      .WORD 10010
      .WORD 0
      .WORD ERR5
499 021624 012701 000030      MOV #24.,R1      ;INITIALIZE WAIT COUNT
500 021630 004767 175242      JSR PC,GSTAT      ;GET DRIVE STATUS
501 021634 021754      4$
502 021636 032767 000001 161200  BIT #DRDYMSK,T.CS ;TEST IF DRIVE READY
503 021644 001031      BNE 3$          ;YES - SKIP
504 021646      WAITMS 1      ;WAIT FOR 100MS
      MOV #250.,(PC)+
      .WORD 0
      MOV L$DLY,(PC)+
      .WORD 0
      DEC -6(PC)
      BNE -4
      DEC -22(PC)
      BNE -20
      TRAP C$BRK
505 021724 005301      DEC R1            ;DEC WAIT COUNTER
506 021726 001340      BNE 2$          ;LOOP UNTIL TIME DONE
507 021730 032767 100000 161106 3$: BIT #ANYERR,T.CS ;TEST IF ANYERR SET
508 021736 001406      BEQ 4$          ;NO - SKIP
509 021740      ERRHRD 10011.,,ERR6 ;REPORT ALL ERRORS
      TRAP C$ERHRD
      .WORD 10011
      .WORD 0
    
```

```

021746 012766 .WORD ERR6
510 021750 005367 161204 DEC ERRCNT ;REDUCE ERROR COUNT FOR DUAL ERRORS
511 021754 005067 161036 4$: CLR ERRSWI ;CLEAR FOR ERROR RETURN
512 021760 162767 000002 161014 5$: SUB #2,SSINDX ;REMOVE ENTRY FROM SUBROUT STACK
513 021766 012604 MOV (SP)+,R4 ;RESTORE REGS
514 021770 012601 MOV (SP)+,R1
515 021772 012600 MOV (SP)+,R0
516 021774 012603 MOV (SP)+,R3
517 021776 005767 161014 TST ERRSWI ;TEST IF ERROR RETURN
518 022002 001403 BEQ 99$ ;YES - SKIP
519 022004 066716 161006 ADD ERRSWI,(SP) ;ADD IN ERROR RETURN
520 022010 000207 RTS PC
521 022012 017616 000000 99$: MOV @ (SP),(SP) ;SET ERROR RETURN ADDRESS
522 022016 000207 RTS PC
523
524 : CHOOSE HEAD ROUTINE. PICKS HEAD 0 UNLESS SPECIFIC HEAD IS
525 : SELECTED BY SOFTWARE PARAMETER.
526 022020 005067 161066 CHOSHD: CLR DESHD ;CLEAR TO HEAD 0
527 022024 032767 010000 172274 BIT #HEADLM,MISWIW ;TEST IF HEAD SPECIFIED
528 022032 001403 BEQ 1$ ;NO - SKIP
529 022034 016767 172274 161050 MOV HEADW,DESHD ;INSERT SPECIFIED HEAD
530 022042 000207 1$: RTS PC
531
532
533
534 : SWAP HEAD ROUTINE. CHANGES SELECTED HEAD TO HEAD 1
535 : UNLESS HEAD 0 SPECIFICALLY SELECTED BY SOFTWARE PARAMETER.
536 022044 032767 010000 172254 SWAPHD: BIT #HEADLM,MISWIW ;TEST IF HEAD SPECIFIED
537 022052 001011 BNE 2$ ;YES - TAKE ABORT EXIT
538 022054 005767 161032 TST DESHD ;TEST IF HEAD ONE USED
539 022060 001006 BNE 2$ ;YES - TAKE ABORT EXIT
540 022062 012767 000001 161022 MOV #1,DESHD ;ELSE SET FOR HEAD ONE
541 022070 062716 000002 ADD #2,(SP) ;BUMP PAST ABORT RETURN
542 022074 000207 RTS PC ;RETURN
543 022076 017616 000000 2$: MOV @ (SP),(SP) ;GET ABORT DESTINATION
544 022102 000207 3$: RTS PC
545
546
547
548 : SWAP OLD CYLINDER AND NEW CYLINDER ROUTINE.
549 022104 010046 ONSWAP: MOV R0,-(SP) ;STORE R0
550 022106 016700 160766 MOV OLDCYL,R0 ;MOVE OLD TO R0
551 022112 016767 160764 160760 MOV NEWCYL,OLDCYL ;MOVE NEW TO OLD
552 022120 010067 160756 MOV R0,NEWCYL ;PUT OLD IN NEW
553 022124 012600 MOV (SP)+,R0 ;RESTORE R0
554 022126 000207 RTS PC
555
556
557
558 : READ HEADERS ROUTINE.
559 022130 012767 000001 160770 XRDHDC: MOV #1,TEMP4 ;SET FLAG TO BYPASS REG STORAGE
560 022136 000402 BR XRDHDG ;GO DO IT
561 022140 005067 160762 XRDHD: CLR TEMP4 ;SET FLAG TO SAVE T. AND L. REGS
562 022144 010346 XRDHDG: MOV R3,-(SP) ;STORE REGISTERS
563 022146 016703 160630 MOV SSINDX,R3 ;GET SUBROUTINE INDEX
564 022152 005723 TST (R3)+ ;BUMP IT FOR NEXT ENTRY
565 022154 016663 000002 002404 MOV 2(SP),SUBSTK(R3) ;INSERT THIS CALL

```

580	022162	162763	00C004	002404	SUB	#4,SUBSTK(R3)	:ADJUST IT TO CALLING LOCATION
581	022170	010367	160606		MOV	R3,SSINDX	:STORE IT BACK
582	022174	010046			MOV	R0,-(SP)	
583	022176	010146			MOV	R1,-(SP)	
584	022200	010446			MOV	R4,-(SP)	
585	022202	012767	000002	160606	MOV	#2,ERRSWI	:SET FOR NO ERROR RETURN
586	022210	005767	160712		TST	TEMP4	:TEST IF REGISTERS TO BE SAVED
587	022214	001007			BNE	2\$	:NO - SKIP
588	022216	012703	003044		MOV	#L.MP+2,R3	:SET POINTER FOR REGS
589	022222	012701	000004		MOV	#4,R1	:SET COUNT
590	022226	014346		1\$:	MOV	-(R3),-(SP)	:SAVE REGISTER
591	022230	005301			DEC	R1	:DEC COUNT
592	022232	001375			BNE	1\$	:LOOP UNTIL ALL ARE SAVED
593	022234	004767	177222	2\$:	JSR	PC,RDYCHK	:CHECK DRIVE READY
594	022240	022526			65\$		
595	022242	005067	160540		CLR	DONE	:CLEAR INTERRUPT FLAG
596	022246	012701	003034		MOV	#L.CS,R1	:GET ADDRESS OF LOAD REGS
597	022252	016711	160554		MOV	RLDRV,(R1)	:LOAD DRIVE NUMBER
598	022256	042711	002000		BIC	#BIT10,(R1)	:CLEAR FOR DRIVE 4 - 7 SPEC'D
599	022262	052721	000110		BIS	#RDHEAD,(R1)+	:INSERT COMMAND
600	022266	005021			CLR	(R1)+	:CLEAR BA
601	022270	005021			CLR	(R1)+	:CLEAR DA
602	022272	014162	000004		MOV	-(R1),RLDA(R2)	:LOAD RL11 REGS
603	022276	014162	000002		MOV	-(R1),RLBA(R2)	
604	022302	014162	000000		MOV	-(R1),RLCSR(R2)	
605	022306			3\$:	WAITUS	10.	:WAIT 1 MS FOR INTERRUPT
	022306	012727	000012		MOV	#10.,(PC)+	
	022312	000000			.WORD	0	
	022314	016727	157576		MOV	L\$DLY,(PC)+	
	022320	000000			.WORD	0	
	022322	005367	177772		DEC	-6(PC)	
	022326	001375			BNE	-4	
	022330	005367	177756		DEC	-22(PC)	
	022334	001367			BNE	-20	
606	022336	005767	160444		TST	DONE	:TEST IF INTERRUPT FLAG SET
607	022342	001460			BEQ	14\$	:NO - SKIP
608	022344	032767	000001	160472	5\$:	BIT	#DRDYMSK,T.CS
609	022352	001035			BNE	10\$	:TEST IF DRIVE READY
610	022354	012703	010404		MOV	#MDRDY,R3	:YES - SKIP
611	022360	012704	011423		MOV	#CAFDT,R4	:SET NO READY MESSAGE
612	022364				ERRHRD	10017,,,ERR5	:CONDITION OF AFTER DATA XFER
	022364	104456			TRAP	C\$ERHRD	
	022366	023441			.WORD	10017	
	022370	000000			.WORD	0	
	022372	012716			.WORD	ERR5	
613	022374	012701	000030		MOV	#24.,R1	:INITIALIZE WAIT COUNT
614	022400	004767	174472	4\$:	JSR	PC,G\$STAT	:GET STATUS
615	022404	022522			60\$		
616	022406	032767	000001	160430	BIT	#DRDYMSK,T.CS	:TEST IF DRIVE HAS COME READY
617	022414	001403			BEQ	11\$	:NO - SKIP
618	022416	005067	160374		CLR	ERRSWI	:CLEAR ERROR SWITCH
619	022422	000411			BR	10\$	:SKIP
620	022424	005301		11\$:	DEC	R1	:DEC WAIT COUNT
621	022426	001364			BNE	4\$	:LOOP UNTIL TIME DONE
622	022430	012704	011434		MOV	#C\$SEC,R4	:SET CONDITION AFTER 5 SECONDS
623	022434				ERRHRD	10014,,,ERR5	
	022434	104456			TRAP	C\$ERHRD	

```

022436 023436 .WORD 10014
022440 000000 .WORD 0
022442 012716 .WORD ERR5
624 022444 000426 BR 60$ ;EXIT
625 022446 005767 160372 10$: TST T.CS ;CHECK FOR ANY ERRORS
626 022452 100005 BPL 12$ ;NO - SKIP
627 022454 ERRHRD: 10016...ERR6 ;REPORT ALL ERRORS
022454 104456 TRAP (SERHRD
022456 023440 .WORD 10016
022460 000000 .WORD 0
022462 012766 .WORD ERR6
628 022464 000416 BR 60$
629 022466 012701 003054 12$: MOV #HDWRD2,R1 ;GET POINTER
630 022472 016221 000006 MOV RLMP(R2),(R1)+ ;STORE LAST TWO HEADER WORDS
631 022476 016221 000006 MOV RLMP(R2),(R1)+
632 022502 000411 BR 65$ ;EXIT
633 022504 004767 174126 14$: JSR PC,WAITIN ;WAIT FOR INTERRUPT
634 022510 012603 MOV (SP)+,R3 ;GET RESULTS
635 022512 ERRHRD: 10015...ERR1 ;REPORT
022512 104456 TRAP (SERHRD
022514 023437 .WORD 10015
022516 000000 .WORD 0
022520 012464 .WORD ERR1
636 022522 005067 160270 60$: CLR ERRSWI ;CLEAR FOR ERROR RETURN
637 022526 005767 160374 65$: TST TEMP4 ;TEST IF REGISTERS WERE SAVED
638 022532 001007 BNE 22$ ;NO - SKIP
639 022534 012703 003034 MOV #L.CS,R3 ;SET POINTER TO RESTORE REGS
640 022540 012701 000004 MOV #4,R1 ;SET COUNT
641 022544 012623 20$: MOV (SP)+,(R3)+ ;RESTORE REGISTER
642 022546 005301 DEC R1 ;DEC COUNT
643 022550 001375 BNE 20$ ;LOOP UNTIL ALL ARE RESTORED
644 022552 162767 000002 160222 22$: SUB #2,SSINDX ;REMOVE ENTRY FROM SUBROUT STACK
645 022560 012604 MOV (' )+,R4 ;RESTORE REGS
646 022562 012601 MOV (,P)+,R1
647 022564 012600 MOV (SP)+,R0
648 022566 012603 MOV (SP)+,R3
649 022570 005767 160222 TST ERRSWI ;TEST IF ERROR RETURN
650 022574 001403 BEQ 99$ ;YES - SKIP
651 022576 066716 160214 ADD ERRSWI,(SP) ;ADD IN ERROR RETURN
652 022602 000207 RTS PC
653 022604 017616 000000 99$: MOV @ (SP),(SP) ;SET ERROR RETURN ADDRESS
654 022610 000207 RTS PC
655
731
732
733 ; POSITION HEAD BIT FROM HEADER OR MULTIPURPOSE REGISTER TO LSB.
734 022612 016705 160234 POSHW1: MOV HDWRD1,R5 ;START FOR POSITION HD BIT IN WD 1
735 022616 000402 BR POSHDO ;SKIP
736 022620 016705 160226 POSHSB: MOV T.MP,R5 ;START FOR POSITION HD BIT IN MP
737 022624 010146 POSHDO: MOV R1,-(SP) ;STORE R1
738 022626 042705 177677 BIC #*(CHSSTAT,R5 ;CLEAR ALL BUT HEAD SEL BIT
739 022632 012701 000006 MOV #6,R1 ;SET SHIFT COUNT
740 022636 006205 1$: ASR R5 ;SHIFT FOR RIGHT JUSTIFY
741 022640 005301 DEC R1
742 022642 001375 BNE 1$
743 022644 012601 MOV (SP)+,R1 ;RESTORE R1
744 022646 000207 RTS PC ;RETURN
    
```

```

745
746
747
748      ;      WAIT FOR READY ROUTINE. DURATION OF WAIT PASSED TO THE ROUTINE
749      ;      FROM THE CALLING ROUTINE IN R1.
750 022650 010346      RDYWAIT:  MOV     R3,-(SP)      ;STORE R3
751 022652 016703 160124  MOV     SSINDX,R3      ;GET SUBROUTINE INDEX
752 022656 005723      TST     (R3)+          ;BUMP IT FOR NEXT ENTRY
753 022660 016663 000002 002404  MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
754 022666 162763 000004 002404  SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
755 022674 010367 160102      MOV     R3,SSINDX      ;STORE IT BACK
756 022700 010046      MOV     R0,-(SP)
757 022702 010146      MOV     R1,-(SP)
758 022704 010446      MOV     R4,-(SP)
759 022706 012767 000002 160102  MOV     #2,ERRSWI      ;SET FOR NO ERROR RETURN
760
761 022714      5$:  BREAK      ;ALLOW A ^C
      022714      TRAP     CSBRK
762 022716 004767 174154      JSR     PC,GSTAT      ;GET DRIVE STATUS
763 022722 023156      10$
764 022724 032767 000001 160112  BIT     #DRDYMSK,T.CS  ;CHECK IF READY
765 022732 001113      BNE     9$           ;YES - SKIP
766 022734 005301      DEC     R1           ;DEC WAIT COUNT
767 022736 001415      BEQ     7$           ;SKIP IF 0
768 022740      WAITUS  1
      022740 012727 000001      MOV     #1,(PC)+
      022744 000000      .WORD  0
      022746 016727 157144      MOV     LSDLY,(PC)+
      022752 000000      .WORD  0
      022754 005367 177772      DEC     -6(PC)
      022760 001375      BNE     -4
      022762 005367 177756      DEC     -22(PC)
      022766 001367      BNE     -20
769 022770 000751      BR      5$
770 022772 012703 010404      7$:  MOV     #MDRDY,R3      ;SET NAME MESSAGE PTR
771 022776      ERRHRD  10020,,,ERR3 ;REPORT READY ERROR
      022776 104456      TRAP     CSERHRD
      023000 023444      .WORD  10020
      023002 000000      .WORD  0
      023004 012600      .WORD  ERR3
772 023006 012701 000030      MOV     #24.,R1      ;INITIALIZE WAIT COUNT
773
774 023012      6$:  BREAK      ;ALLOW A ^C
      023012 104422      TRAP     CSBRK
775 023014 004767 174056      JSR     PC,GSTAT      ;GET DRIVE STATUS
776 023020 023156      10$
777 023022 032767 000001 160014  BIT     #DRDYMSK,T.CS  ;TEST IF DRIVE READY
778 023030 001040      BNE     8$           ;YES - SKIP
779 023032      WAITMS  1           ;WAIT 100 MS
      023050 012727 000372      MOV     #250.,(PC)+
      023054 000000      .WORD  0
      023056 016727 157034      MOV     LSDLY,(PC)+
      023062 000000      .WORD  0
      023064 005367 177772      DEC     -6(PC)
      023070 001375      BNE     -4
      023072 005367 177756      DEC     -22(PC)
      023076 001367      BNE     -20
    
```

```

800 023100 104422 TRAP C$BRK
780 023110 005301 DEC R1 ;DEC WAIT COUNT
781 023112 001337 BNE 6$ ;LOOP UNTIL TIME DONE
782 023114 012704 011434 MOV #C5SEC,R4 ;SET CONDITION AFTER 5 SECDS
783 023120 ERRHRD 10021,,,ERR5
    023120 104456 TRAP C$ERHRD
    023122 023445 .WORD 10021
    023124 000000 .WORD 0
    023126 012716 .WORD ERR5
784 023130 000410 BR 11$ ;EXIT
785 023132 032767 100000 157704 8$: BIT #ANYERR,T.CS ;TEST IF ANY ERROR SET
786 023140 001406 BEQ 10$ ;NO - SKIP
787 023142 ERRHRD 10022,,,ERR6 ;REPORT ALL ERRORS
    023142 104456 TRAP C$ERHRD
    023144 023446 .WORD 10022
    023146 000000 .WORD 0
    023150 012766 .WORD ERR6
788 023152 005367 160002 11$: DEC ERRCNT ;DECREMENT FOR DOUBLE ERROR REPORT
789 023156 005067 157634 10$: CLR ERRSWI ;CLEAR FOR ERROR RETURN
790 023162 162767 000002 157612 9$: SUB #2,SSINDX ;REMOVE ENTRY FROM SUBROUT STACK
791 023170 012604 MOV (SP)+,R4 ;RESTORE REGISTERS
792 023172 012601 MOV (SP)+,R1
793 023174 012600 MOV (SP)+,R0
794 023176 012603 MOV (SP)+,R3 ;RESTORE R3
795 023200 005767 157612 TST ERRSWI ;TEST IF ERROR RETURN
796 023204 001403 BEQ 99$ ;YES - SKIP
797 023206 066716 157604 ADD ERRSWI,(SP) ;ADD IN ERROR RETURN
798 023212 000207 RTS PC
799 023214 017616 000000 99$: MOV @ (SP),(SP) ;SET ERROR RETURN ADDRESS
800 023220 000207 RTS PC
801
802
803
804 ; GET POSITION ROUTINE. READS A HEADER FROM CURRENT CYLINDER
805 ; (WHERE IT IS PRESENTLY POSITIONED) AND STORES CYLINDER
806 ; NUMBER IN CURCYL.
807 023222 010346 GETPOS: MOV R3,-(SP) ;STORE REGISTERS
808 023224 016703 157552 MOV SSINDX,R3 ;GET SUBROUTINE INDEX
809 023230 005723 TST (R3)+ ;BUMP IT FOR NEXT ENTRY
810 023232 016663 000002 002404 MOV 2(SP),SUBSTK(R3) ;INSERT THIS CALL
811 023240 162763 000004 002404 SUB #4,SUBSTK(R3) ;ADJUST IT TO CALLING LOCATION
812 023246 010367 157530 MOV R3,SSINDX ;STORE IT BACK
813 023252 010046 MOV R0,-(SP)
814 023254 010546 MOV R5,-(SP)
815 023256 004767 176656 JSR PC,XRDHD ;DO READ HEADER
816 023262 023312 65$
817 023264 016703 157562 MOV HDWRD1,R3 ;GET HEADER WORD
818 023270 012705 000007 MOV #7,R5 ;SET SHIFT COUNT
819 023274 006203 4$: ASR R3 ;SHIFT TO RIGHT JUSTIFY
820 023276 005305 DEC R5
821 023300 001375 BNE 4$
822 023302 042703 177000 BIC #177000,R3
823 023306 010367 157572 MOV R3,CURCYL ;STORE AS CURRENT CYLINDER
824 023312 162767 000002 157462 65$: SUB #2,SSINDX ;REMOVE ENTRY FROM SUBROUT STACK
825 023320 012605 MOV (SP)+,R5 ;RESTORE REGISTERS
826 023322 012600 MOV (SP)+,R0
827 023324 012603 MOV (SP)+,R3
    
```



```

828 023326 005767 157464      TST  ERRSWI      ;TEST IF ERROR RETURN
829 023332 001403              BEQ  99$        ;YES - SKIP
830 023334 066716 157456      ADD  ERRSWI,(SP) ;ADD IN ERROR RETURN
831 023340 000207              RTS  PC
832 023342 017616 000000      99$: MOV  @ (SP), (SP) ;SET ERROR RETURN ADDRESS
833 023346 000207              RTS  PC
834
863
864
865 ; READ ALL HEADERS ROUTINE. 40 HEADERS ARE READ AND STORED
866 ; IN Ibuff.
867 023350 010346      RDALHD: MOV  R3, -(SP) ;STORE REGISTERS
868 023352 016703 157224      MOV  SSINDX,R3 ;GET SUBROUTINE INDEX
869 023356 005723              TST  (R3)+      ;BUMP IT FOR NEXT ENTRY
870 023360 016663 000002 002404  MOV  2(SP),SUBSTK(R3) ;INSERT THIS CALL
871 023366 162763 000004 002404  SUB  #4,SUBSTK(R3) ;ADJUST IT TO CALLING LOCATION
872 023374 010367 157402      MOV  R3,SSINDX ;STORE IT BACK
873 023400 010046      MOV  R0, -(SP)
874 023402 010146      MOV  R1, -(SP)
875 023404 010446      MOV  R4, -(SP)
876 023406 012767 000002 157402  MOV  #2,ERRSWI ;SET FOR NO ERROR RETURN
877 023414 012701 000050      MOV  #40,R1 ;SET HEADER COUNT
878 023420 052767 100000 157356  BIS  #HDR40,OPFLAG ;SET 40 HDR OP FLAG
879 023426 012703 003764      MOV  #IBUFF,R3 ;SET POINTER TO STORE HDRS
880 023432 016704 157370      MOV  RLBAS,R4 ;GET BASE ADDRESS
881 023436 062704 000006      ADD  #RLMP,R4 ;MAKE IT POINT TO MP REG
882 023442 012767 000010 157364  MOV  #10,LCS ;LOAD FOR READ HEADER, NO INTERRUPT
883 023450 056767 157356 157356  BIS  RLDV,LCS ;INSERT DRIVE NUMBER
884 023456 042767 002000 157350  BIC  #BIT10,LCS ;CLEAR FOR DRIVE 4 - 7 SPEC'D
885 023464 005067 157346      CLR  L.BA ;CLEAR BA
886 023470 005067 157344      CLR  L.DA ;CLEAR DA
887 023474 005767 157412      TST  DESHD ;TEST IF HEAD 0
888 023500 001403              BEQ  3$        ;YES - SKIP
889 023502 052767 000020 157330  BIS  #HDSSEL,L.DA ;ELSE INSERT HEAD 0
890 023510 016762 157324 000004 3$: MOV  L.DA,RLDA(R2) ;LOAD RLDA REG
891 023516 016762 157314 000002  MOV  L.BA,RLBA(R2) ;LOAD RLBA
892 023524 032762 000200 000000  BIT  #CRDYMSK,RLCS(R2) ;TEST IF CONTROLLER READY
893 023532 001003              BNE  6$        ;YES - SKIP
894 023534 004767 175722      JSR  PC,RDYCHK ;ELSE CHECK READY
895 023540 023656      65$
896 023542 016762 157266 000000 6$: MOV  L.CS,RLCS(R2) ;LOAD RLCS REG
897 023550 012700 077777      MOV  #77777,R0 ;SET COUNT FOR WAIT
898 023554 032762 000200 000000 7$: BIT  #CRDYMSK,RLCS(R2) ;CHECK THAT OPERATION COMPLETED
899 023562 001016              BNE  8$        ;YES - SKIP
900 023564 005300              DEC  R0 ;DEC COUNT
901 023566 001372              BNE  7$        ;SKIP IF NOT YET 0
902 023570 004767 173010      JSR  PC,READRL ;ELSE GET ALL REGISTERS
903 023574 004767 173036      JSR  PC,WAITIN ;ELSE WAIT FOR TIMEOUT
904 023600 012603      MOV  (SP)+,R3 ;GET RESULT MESSAGE POINTER
905 023602      ERRHRD 10025,,ERR1
      TRAP C$ERHRD
      .WORD 10025
      .WORD 0
      .WORD ERR1
906 023612 005067 157200      CLR  ERRSWI ;CLEAR FOR ERROR RETURN
907 023616 000417      BR  65$
908 023620 005767 157220      8$: TST  T.CS ;TEST FOR ANY ERRORS
    
```

```

909 023624 100007      BPL      12$      ;NO - SKIP
910 023626      ERRHRD 10026.,,ERR6
      023626 104456      TRAP    C$ERHRD
      023630 023452      .WORD  10026
      023632 000000      .WORD  0
      023634 012766      .WORD  ERR6
911 023636 005067 157154 CLR      ERRSWI      ;CLEAR FOR ERROR RETURN
912 023642 000405      BR      65$
913 023644 011423      12$:    MOV     (R4),(R3)+    ;STORE HEADER WORDS
914 023646 011423      MOV     (R4),(R3)+
915 023650 011423      MOV     (R4),(R3)+
916 023652 005301      DEC     R1          ;DEC HEADER COUNT
917 023654 001332      BNE     6$
918 023656 162767 000002 157116 65$:    SUB     #2,SSINDX    ;REMOVE ENTRY FROM SUBROUT STACK
919 023664 012604      MOV     (SP)+,R4    ;RESTORE REGISTERS
920 023666 012601      MOV     (SP)+,R1
921 023670 012600      MOV     (SP)+,R0
922 023672 012603      MOV     (SP)+,R3
923 023674 005767 157116      TST     ERRSWI      ;TEST IF ERROR RETURN
924 023700 001403      BEQ     99$        ;YES - SKIP
925 023702 066716 157110      ADD     ERRSWI,(SP) ;ADD IN ERROR RETURN
926 023706 000207      RTS     PC
927 023710 017616 000000 99$:    MOV     @ (SP),(SP)  ;SET ERROR RETURN ADDRESS
928 023714 000207      RTS     PC
929
930
1158
1159
1160      :      REPORT OPERATION ROUTINE. PRINTS SUBROUTINE TRACE SEQUENCE AND
1161      :      OPERATION BEING PERFORMED PORTION OF ALL
1162      :      ERROR MESSAGES.
1162 023716 010446      RPTOP: MOV     R4,-(SP)
1163 023720 005767 157056      TST     SSINDX      ;TEST SUBROUTINE INDEX 0
1164 023724 001433      BEQ     1$          ;SKIP IF 0
1165 023726 012704 000002      MOV     #2,R4       ;SET INDEXER TO FIRST ENTRY
1166 023732      PRINTB #FMT9,#SEQMES ;PRINT 'SUBROUTINE CALL SEQ'
      023732 012746 010253      MOV     #SEQMES,-(SP)
      023736 012746 011753      MOV     #FMT9,-(SP)
      023742 012746 000002      MOV     #2,-(SP)
      023746 010600      MOV     SP,R0
      023750 104414      TRAP    C$PNTB
      023752 062706 000006      ADD     #6,SP
1167 023756      PRINTB #FMT16,SUBSTK(R4) ;PRINT CALLING LOCATION
      023756 016446 002404      MOV     SUBSTK(R4),-(SP)
      023762 012746 012126      MOV     #FMT16,-(SP)
      023766 012746 000002      MOV     #2,-(SP)
      023772 010600      MOV     SP,R0
      023774 104414      TRAP    C$PNTB
      023776 062706 000006      ADD     #6,SP
1168 024002 062704 000002      ADD     #2,R4       ;BUMP INDEX
1169 024006 020467 156770      CMP     R4,SSINDX   ;CHECK IF ALL PRINTED
1170 024012 003761      BLE     3$         ;LOOP IF NOT ALL PRINTED YET
1171 024014      PRINTB #FMT4,ERHEAD,#TSTLAB ;PRINT ERROR HEADER
      024014 012746 006365      MOV     #TSTLAB,-(SP)
      024020 016746      MOV     ERHEAD,-(SP)
      024024 012746 011556      MOV     #FMT4,-(SP)
      024030 012746 000003      MOV     #3,-(SP)
      024034 010600      MOV     SP,R0
    
```

	024036	104414			TRAP	C\$PNTB	
	024040	062706	000010		ADD	#10,SP	
1172	024044	042767	030000	156732	BIC	#SEEKOP,RORWOP,OPFLAG	;CLEAR SK & RD OR WRT FLAG
1173	024052	016701	156756		MOV	L,CS,R1	;GET COMMAND EXECUTED
1174	024056	042701	177741		BIC	#177741,R1	;STRIP ALL BUT FUNCTION CODE
1175	024062	022701	000006		CMP	#6,R1	;TEST IF SEEK OPERATION
1176	024066	001003			BNE	2\$	;NO - SKIP
1177	024070	052767	010000	156706	BIS	#SEEKOP,OPFLAG	;ELSE SET SEEK FLAG
1178	024076	022701	000012		CMP	#12,R1	;TEST IF WRITE
1179	024102	001003			BNE	20\$	;NO - SKIP
1180	024104	052767	020000	156672	BIS	#RORWOP,OPFLAG	;SET RD OR WRT FLAG
1181	024112	022701	000014		CMP	#14,R1	;TEST IF READ
1182	024116	001003			BNE	22\$	;NO - SKIP
1183	024120	052767	020000	156656	BIS	#RORWOP,OPFLAG	;SET RD OR WRT FLAG
1184	024126				PRINTB	#FMT1,#MOPER,OPMSG\$(R1)	;PRINT OPERATION
	024126	016146	002224		MOV	OPMSG\$(R1),-(SP)	
	024132	012746	005414		MOV	#MOPER,-(SP)	
	024136	012746	011534		MOV	#FMT1,-(SP)	
	024142	012746	000003		MOV	#3,-(SP)	
	024146	010600			MOV	SP,R0	
	024150	104414			TRAP	C\$PNTB	
	024152	062706	000010		ADD	#10,SP	
1185	024156	020127	000004		CMP	R1,#4	;CHECK IF GET STATUS
1186	024162	001007			BNE	4\$	;NO - SKIP
1187	024164	032767	000010	156646	BIT	#DRSET,L.DA	;TEST IF RESET INCLUDED
1188	024172	001403			BEQ	4\$	;NO - SKIP
1189	024174	012701	000016		MOV	#16,R1	;SET TO PRINT WITH RESET
1190	024200	000436			BR	9\$	
1191	024202	032767	007777	156574	4\$:	BIT	#COMPOP,OPFLAG ;TEST IF ANY OTHER OPERATION
1192	024210	001424			BEQ	8\$	;NO - SKIP
1193	024212	016704	156566		MOV	OPFLAG,R4	;SET UP TO DETERMINE WHICH ONE
1194	024216	012701	000020		MOV	#20,R1	;PRESET THE POINTER
1195	024222	032704	000001		5\$:	BIT	#BIT00,R4 ;CHECK THE BIT
1196	024226	001003			BNE	6\$	;IF SET - SKIP
1197	024230	005721			TST	(R1)+	;BUMP POINTER
1198	024232	006204			ASR	R4	
1199	024234	000772			BR	5\$	
1200	024236				6\$:	PRINTB	#FMT2,OPMSG\$(R1)
	024236	016146	002224		MOV	OPMSG\$(R1),-(SP)	
	024242	012746	011550		MOV	#FMT2,-(SP)	
	024246	012746	000002		MOV	#2,-(SP)	
	024252	010600			MOV	SP,R0	
	024254	104414			TRAP	C\$PNTB	
	024256	062706	000006		ADD	#6,SP	
1201	024262	032767	100000	156514	8\$:	BIT	#HDR40,OPFLAG ;TEST IF 40 HEADER OPERATION
1202	024270	001415			BEQ	1C\$	;NO - SKIP
1203	024272	012701	000050		MOV	#50,R1	;ELSE PRINT IT
1204	024276				9\$:	PRINTB	#FMT2,OPMSG\$(R1)
	024276	016146	002224		MOV	OPMSG\$(R1),-(SP)	
	024302	012746	011550		MOV	#FMT2,-(SP)	
	024306	012746	000002		MOV	#2,-(SP)	
	024312	010600			MOV	SP,R0	
	024314	104414			TRAP	C\$PNTB	
	024316	062706	000006		ADD	#6,SP	
1205	024322	000434			BR	15\$	;SKIP
1206	024324	032767	000000	156452	10\$:	BIT	#SEEKOP,OPFLAG ;TEST IF SEEK
1207	024332	001430			BEQ	15\$	;NO - SKIP

```

1208 024334          PRINTB #FMT13,#FRMWD,OLDCYL,#DIFWD,DESDIF,#SGNWD,DESSGN,#HDWD,DESHD
      024334 016746 156552      MOV     DESHD,-(SP)
      024340 012746 010214      MOV     #HDWD,-(SP)
      024344 016746 156540      MOV     DESSGN,-(SP)
      024350 012746 010207      MOV     #SGNWD,-(SP)
      024354 016746 156526      MOV     DESDIF,-(SP)
      024360 012746 010201      MOV     #DIFWD,-(SP)
      024364 016746 156510      MOV     OLDCYL,-(SP)
      024370 012746 010232      MOV     #FRMWD,-(SP)
      024374 012746 011774      MOV     #FMT13,-(SP)
      024400 012746 000011      MOV     #11,-(SP)
      024404 010600          MOV     SP,R0
      024406 104414          TRAP   C$PNTB
      024410 062706 000024      ADD     #24,SP
1209 024414 032767 020000 156362 15$: BIT     #RORWOP,OPFLAG ;TEST IF READ OR WRITE SET
1210 024422 001424          BEQ    17$ ;NO - SKIP
1211 024424          PRINTB #FMT22,#CYLWD,CURCYL,#HDWD,DESHD,#SECWD,DESSEC
      024424 016746 156464      MOV     DESSEC,-(SP)
      024430 012746 010220      MOV     #SECWD,-(SP)
      024434 016746 156452      MOV     DESHD,-(SP)
      024440 012746 010214      MOV     #HDWD,-(SP)
      024444 016746 156434      MOV     CURCYL,-(SP)
      024450 012746 010225      MOV     #CYLWD,-(SP)
      024454 012746 012323      MOV     #FMT22,-(SP)
      024460 012746 000007      MOV     #7,-(SP)
      024464 010600          MOV     SP,R0
      024466 104414          TRAP   C$PNTB
      024470 062706 000020      ADD     #20,SP
1212 024474 004767 000446      JSR    PC,CLRPARM ;CLEAR PARAM TABLE
1213 024500 012604          MOV     (SP)+,R4 ;RESTORE R4
1214 024502 000207          RTS    PC
1215
1216
1217
1218 ;
1219 ; REPORT REASON ROUTINE
      RPTRES: PRINTS REASON PORTION FOR ALL ERROR REPORTS.
1220 024504 010146          MOV     R1,-(SP) ;STORE R1
1221 024506 010346          MOV     R3,-(SP) ;STORE R3
1222 024510 010446          MOV     R4,-(SP) ;STORE R4
1223 024512 012701 003062      MOV     #RESPARM,R1 ;GET START OF PARAM
1224 024516 012103          MOV     (R1)+,R3 ;GET NUMBER OF PARAM
1225 024520          PRINTB #FMT1.1,#MRSLT,(R1) ;PRINT NAME
      024520 011146          MOV     (R1),-(SP)
      024522 012746 005423      MOV     #MRSLT,-(SP)
      024526 012746 011541      MOV     #FMT1.1,-(SP)
      024532 012746 000003      MOV     #3,-(SP)
      024536 010600          MOV     SP,R0
      024540 104414          TRAP   C$PNTB
      024542 062706 000010      ADD     #10,SP
1226 024546 021127 011057      CMP     (R1),#MNRST ;TEST IF MESSAGE IS NO DRV STATUS
1227 024552 001453          BEQ    6$ ;YES - SKIP REST OF REPORT
1228 024554 012704 011760      MOV     #FMT11,R4 ;PRESET FOR FORMAT 11
1229 024560 022127 011052      CMP     (R1)+,#MCYLOC ;CHECK IF REPORTING CYLINDER LOC
1230 024564 001002          BNE    3$ ;NO - SKIP
1231 024566 012704 011766      MOV     #FMT12,R4 ;ELSE CHANGE TO FORMAT 12
1232 024572 005303          DEC     R3 ;DEC PARAM COUNT
1233 024574 001442          BEQ    6$ ;IF 0 - EXIT
    
```

```

1234 024576          PRINTB R4,#RESE3,(R1)+ ;REPORT IS VALUE
      024576 012146    MOV      (R1)+,-(SP)
      024600 012746 011300    MOV      #RESE3,-(SP)
      024604 010446    MOV      R4,-(SP)
      024606 012746 000003    MOV      #3,-(SP)
      024612 010600    MOV      SP,R0
      024614 104414    TRAP     C$PNTB
      024616 062706 000010    ADD      #10,SP
1235 024622          PRINTB R4,#RESE4,(R1)+ ;REPORT SB VALUE
      024622 012146    MOV      (R1)+,-(SP)
      024624 012746 011304    MOV      #RESE4,-(SP)
      024630 010446    MOV      R4,-(SP)
      024632 012746 000003    MOV      #3,-(SP)
      024636 010600    MOV      SP,R0
      024640 104414    TRAP     C$PNTB
      024642 062706 000010    ADD      #10,SP
1236 024646 162703 000002    SUB      #2,R3 ;DEC PARAM COUNT
1237 024652 001413          BEQ     6$ ;IF 0 - EXIT
1238 024654          PRINTB #FMT1,#RESE5,(R1)+ ;REPORT CONDITION
      024654 012146    MOV      (R1)+,-(SP)
      024656 012746 011311    MOV      #RESE5,-(SP)
      024662 012746 011534    MOV      #FMT1,-(SP)
      024666 012746 000003    MOV      #3,-(SP)
      024672 010600    MOV      SP,R0
      024674 104414    TRAP     C$PNTB
      024676 062706 000010    ADD      #10,SP
1239 024702 012604          6$: MOV      (SP)+,R4 ;RESTORE REGS
1240 024704 012603          MOV      (SP)+,R3
1241 024706 012601          MOV      (SP)+,R1
1242 024710 000207          RTS     PC ;RETURN
1243
1244
1245
1246 ; REPORT PHYSICAL ADDRESS OF DEVICE UNDER TEST
1247 ; AND ALL REGISTER CONTENTS.
1248 RPTREM: PRINTB #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
      024712 005046    CLR      -(SP)
      024714 156716 156113    BISB    RLDRV+1,(SP)
      024720 012746 006053    MOV      #DRVNAM,-(SP)
      024724 016746 156076    MOV      RLBAS,-(SP)
      024730 012746 006042    MOV      #BASADD,-(SP)
      024734 012746 011567    MOV      #FMT5,-(SP)
      024740 012746 000005    MOV      #5,-(SP)
      024744 010600    MOV      SP,R0
      024746 104414    TRAP     C$PNTB
      024750 062706 000014    ADD      #14,SP
1249 ; REPORT RL11 REGISTERS
1250 PRINTB #FMT6,#CSNAM,#DANAM,#BANAM,#MPNAM,#CYLWD,#HDWD
      024754 012746 010214    MOV      #HDWD,-(SP)
      024760 012746 010225    MOV      #CYLWD,-(SP)
      024764 012746 006137    MOV      #MPNAM,-(SP)
      024770 012746 006125    MOV      #BANAM,-(SP)
      024774 012746 006132    MOV      #DANAM,-(SP)
      025000 012746 006120    MOV      #CSNAM,-(SP)
      025004 012746 011607    MOV      #FMT6,-(SP)
      025010 012746 000007    MOV      #7,-(SP)
      025014 010600    MOV      SP,R0
    
```

1251	025016	104414	000020	TRAP	C\$PNTB
	025020	062706		ADD	#20,SP
	025024			PRINTB	#FMT8,#LAB1,L.CS,L.DA,L.BA,L.MP
	025024	016746	156012	MOV	L.MP,-(SP)
	025030	016746	156002	MOV	L.BA,-(SP)
	025034	016746	156000	MOV	L.DA,-(SP)
	025040	016746	155770	MOV	L.CS,-(SP)
	025044	012746	006144	MOV	#LAB1,-(SP)
	025050	012746	011721	MOV	#FMT8,-(SP)
	025054	012746	000006	MOV	#6,-(SP)
	025060	010600		MOV	SP,RO
	025062	104414		TRAP	C\$PNTB
1252	025064	062706	000016	ADD	#16,SP
	025070			PRINTB	#FMT7,#LAB2,T.CS,T.DA,T.BA,T.MP,CURCYL,DESHD
	025070	016746	156016	MOV	DESHD,-(SP)
	025074	016746	156004	MOV	CURCYL,-(SP)
	025100	016746	155746	MOV	T.MP,-(SP)
	025104	016746	155736	MOV	T.BA,-(SP)
	025110	016746	155734	MOV	T.DA,-(SP)
	025114	016746	155724	MOV	T.CS,-(SP)
	025120	012746	006157	MOV	#LAB2,-(SP)
	025124	012746	011651	MOV	#FMT7,-(SP)
	025130	012746	000010	MOV	#10,-(SP)
	025134	010600		MOV	SP,RO
	025136	104414		TRAP	C\$PNTB
	025140	062706	000022	ADD	#22,SP
1253	025144	000207		RTS	PC

1254  
 1255  
 1256  
 1257  
 1258  
 1259  
 1260  
 1261  
 1262  
 1263  
 1264  
 1265  
 1266  
 1267  
 1268  
 1269  
 1270  
 1271  
 1272  
 1273  
 1274  
 1275  
 1276  
 1277  
 1278  
 1279  
 1280  
 1281  
 1282  
 1283

```

: CLEAR PARAMETER BLOCK FOR REPORTING
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

.TITLE CZRLIDO RL01/02 DRIVE TEST 1

:DISK STATE FUNCTIONS

:BITS 0-2 OF THE MULTIPURPOSE REGISTER DURING GET STATUS COMMAND DEFINE THE  
 :STATE OF THE DRIVE

```

:
: STATE 0 LOAD STATE
: STATE 1 SPIN UP
    
```

1284	:	STATE	2	BRUSH CYCLE
1285	:	STATE	3	LOAD HEADS
1286	:	STATE	4	SEEK
1287	:	STATE	5	LOCK ON
1288	:	STATE	6	UNLOAD HEADS
1289	:	STATE	7	SPIN DOWN
1290	:			
1291	:			
1292	:			

```

1 025176          BGNMOD  HRDWTST
2
3
4
5          .SBTTL  *TEST 1          BASIC INTERFACE (PART 1)
6
7 025176          BGNTST          ;TEST01
8
9          ;TEST THAT UNLOAD, COVER OPEN AND WRITE PROTECT START
10          ;IN THE PROPER STATE.
10 025176 005767 156156      TST    PASNUM          ;CHECK IF FIRST PASS
11 025202 001124          BNE    65$          ;EXIT IF NO
12 025204 005767 167116      TST    MISWIW          ;CHECK IF MANUAL INTERVENTION
13 025210 100121          BPL    65$          ;NO - EXIT TEST
14 025212 012767 006373 155572 2$:  MOV    #MISTST,ERHEAD ;LOAD ERR HEADER
15          ;PROMPT CHK DRV IS UNLDED, COVR OPN, AND
16          ;WRTE LCKED
17 025220          PRINTF  #FMTOP1,#OPR1,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
18 025220 005046          CLR    -(SP)
19 025222 156716 155605      BISB  RLDRV+1,(SP)
20 025226 012746 006053      MOV    #DRVNAM,-(SP)
21 025232 015746 155570      MOV    RLBAS,-(SP)
22 025236 012746 006042      MOV    #BASADD,-(SP)
23 025242 012746 010135      MOV    #OPR1A,-(SP)
24 025246 012746 007540      MOV    #OPR1,-(SP)
25 025252 012746 011442      MOV    #FMTOP1,-(SP)
26 025256 012746 000007      MOV    #7,-(SP)
27 025262 010600          MOV    SP,R0
28 025264 104417          TRAP  C$PNTF
29 025266 062706 000020      ADD    #20,SP
30
31 025272 005067 157066      CLR    OBUFF          ;CLEAR FOR RESPONSE
32 025276          GMANIL  OPRO02,OBUFF,1,NO
33 025276 104443          TRAP  C$GMAN
34 025300 000404          BR    10000$
35 025302 004364          .WORD OBUFF
36 025304 000120          .WORD T$CODE
37 025306 007470          .WORD OPRO02
38 025310 000001          .WORD 1
39
40 025312          10000$:
41 025312 005767 157046      TST    OBUFF          ;TEST RESPONSE YES
42 025316 001735          BEQ  2$             ;YES - SKIP
43 025320 004767 171504      JSR    PC,TSTINT      ;INITIALIZE TEST
44 025324 004767 171516      JSR    PC,GSTATR      ;GO GET STATUS WITH RESET
45 025330 025454          65$
46 025332 032767 000040 155512  BIT    #COSTAT,T.MP      ;CHECK IF COVER OPEN SET
47 025340 001006          BNE  7$             ;YES - SKIP
48 025342 012703 010562      MOV    #MCOSTA,R3      ;SET NAME POINTER
49 025346          ERRHRD  101,,,ERR3
50 025346 104456          TRAP  C$ERHRD
51 025350 000145          .WORD 101
52 025352 000000          .WORD 0
53 025354 012600          .WORD ERR3
54 025356 032767 000010 155466 7$:  BIT    #BHSTAT,T.MP      ;TEST IF BRUSHES HOME
55 025364 001006          BNE  9$             ;YES - SKIP
56 025366 012703 010575      MOV    #MBHSTA,R3      ;SET POINTER FOR BRUSH HOME ERROR
57 025372          ERRHRD  102,,,ERR3
    
```



```

025372 104456 TRAP C$ERHRD
025374 000146 .WORD 102
025376 000000 .WORD 0
025400 012600 .WORD ERR3
34 025402 032767 020000 155442 9$: BIT #WLSTAT,T.MP ;TEST IF WRITE LOCK SET
35 025410 001006 BNE 11$ ;YES - SKIP
36 025412 012703 010610 MOV #MWLSTA,R3 ;SET NAME POINTER
37 025416 ERRHRD 103,,,ERR3
025416 104456 TRAP C$ERHRD
025420 000147 .WORD 103
025422 000000 .WORD 0
025424 012600 .WORD ERR3
38 025426 005767 155426 11$: TST T.STAT ;TEST IF STATE ZERO
39 025432 001405 BEQ 15$ ;YES - SKIP
40 025434 005003 CLR R3 ;SET STATE EXPECTED
41 025436 ERRHRD 104,,,ERR7
025436 104456 TRAP C$ERHRD
025440 000150 .WORD 104
025442 000000 .WORD 0
025444 013666 .WORD ERR7
42 025446 004767 171374 15$: JSR PC,GSTATR ;DO DRIVE RESET
43 025452 025454 65$:
44 025454 ENDTST
45 025454 L10024:
025454 104401 TRAP C$ETST
46
47
48
49
50 .SBTTL *TEST 2 BASIC INTERFACE (PART 2)
51 025456 BGNTST ;TEST 2
025456 T2::
52 ;VERIFY THAT COVER OPEN AND WRITE PROTECT WORK.
53 025456 005767 155676 TST PASNUM ;TEST IF PASS 0
54 025462 001077 BNE 65$ ;NO - SKIP
55 025464 005767 166636 TST MISWIW ;TEST IF MANUAL INTERVENTION
56 025470 100074 BPL 65$ ;NO - SKIP
57 025472 012767 006373 155312 MOV #MISTST,ERHEAD ;SET ERROR HEADER
58
59 025500 2$: ;PROMPT CLOSE COVER AND RESET WRITE LOCK.
60 025500 PRINTF #FMTOP1,#OPR2,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
025500 CLR -(SP)
025502 156716 155325 BISB RLDRV+1,(SP)
025506 012746 006053 MOV #DRVNAM,-(SP)
025512 016746 155310 MOV RLBAS,-(SP)
025516 012746 006042 MOV #BASADD,-(SP)
025522 012746 010135 MOV #OPR1A,-(SP)
025526 012746 007616 MOV #OPR2,-(SP)
025532 012746 011442 MOV #FMTOP1,-(SP)
025536 012746 000007 MOV #7,-(SP)
025542 010600 MOV SP,RO
025544 104417 TRAP C$PNTF
025546 062706 000020 ADD #20,SP
61 025552 005067 156606 CLR OBUF ;CLEAR FOR RESPONSE
62 025556 G$MANIL OPROO2,C$BUFF,1,NO
025556 104443 TRAP C$G$MAN
  
```

```

025560 000404 BR 10000$
025562 004364 .WORD OBUFF
025564 000120 .WORD T$CODE
025566 007470 .WORD OPRO02
025570 000001 .WORD 1
025572 10000$:
63 025572 005767 156566 TST OBUFF ;TEST IF RESPONSE YES
64 025576 001740 BEQ 2$ ;NO - SKIP
65
66 025600 004767 171224 1$: JSR PC,TSTINT ;INITIALIZE TEST
67 025604 004767 171236 JSR PC,GSTATR ;GET STATUS WITH RESET
68 025610 025662 65$
69 025612 032767 000040 155232 BIT #COSTAT,T.MP ;TEST IF COVER OPEN RESET
70 025620 001406 BEQ 9$ ;YES - SKIP
71 025622 012703 010562 MOV #MCOSTA,R3 ;SET NAME MESSAGE POINTER
72 025626 ERRHRD 201...ERR2
025626 104456 TRAP C$ERHRD
025630 000311 .WORD 201
025632 000000 .WORD 0
025634 012532 .WORD ERR2
73
74 025636 032767 020000 155206 9$: BIT #WLSTAT,T.MP ;TEST IF WRITE LOCK RESET
75 025644 001406 BEQ 65$ ;YES - SKIP
76 025646 012703 010610 MOV #MWLSTA,R3 ;SET NAME MESSAGE POINTER
77 025652 ERRHRD 202...ERR2
025652 104456 TRAP C$ERHRD
025654 000312 .WORD 202
025656 000000 .WORD 0
025660 012532 .WORD ERR2
78 025662 65$:
79 025662 ENDTST
025662 L10025:
025662 104401 TRAP C$ETST
80
81
82
83 .SETTL *TEST 3 HEAD LOADING
84 025664 BGNTST ;TEST03
85 T3::
86 ;SPIN UP THE DRIVE. VERIFY THAT THE DRIVE GOES FROM
87 ;STATE 0 TO STATE 5 PROPERLY.
87 025664 005767 155470 TST PASNUM ;TEST IF PASS 0
88 025670 001003 BNE 1$ ;NO - SKIP
89 025672 005767 166430 TST MISWIW ;TEST IF MANUAL INTERVENTION
90 025676 100402 BMI 2$ ;YES - SKIP
91 025700 1$: EXIT TST
025700 104432 TRAP C$EXIT
025702 002440 .WORD L10026-.
92 025704 004767 171120 2$: JSR PC,TSTINT ;INITIALIZE TEST
93 025710 004767 171132 JSR PC,GSTATR ;GET STATUS
94 025714 030342 T365$
95 025716 005767 155136 TST T.STAT ;TEST IF STATE 0
96 025722 001426 BEQ 4$ ;YES - SKIP
97 025724 3$: ;PRINT UNEXPECTED ERROR AND EXIT TEST
98 025724 PRINTF #FMTOP1,#UNXERR,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
025724 CLR -(SP)
025726 156716 155101 BISB RLDRV+1,(SP)

```

```

025732 012746 006353      MOV      #DRVNAM,-(SP)
025736 016746 155064      MOV      RLBAS,-(SP)
025742 012746 006042      MOV      #BASADD,-(SP)
025746 012746 010135      MOV      #OPR1A,-(SP)
025752 012746 006350      MOV      #UNXERR,-(SP)
025756 012746 011442      MOV      #FMTOP1,-(SP)
025762 012746 000007      MOV      #7,-(SP)
025766 010600      MOV      SP,RO
025770 104417      TRAP    C$PNTF
025772 062706 000020      ADD     #20,SP
99 025776 104401      TRAP    C$E1ST
100
101 026000      4$:      PRINTF  #FMTOP1,#OPR3,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
102 026000      CLR     -(SP)
026000 005046      B1SB   RLDRV+1,(SP)
026002 156716 155025      MOV      #DRVNAM,-(SP)
026006 012746 006053      MOV      RLBAS,-(SP)
026012 016746 155010      MOV      #BASADD,-(SP)
026016 012746 006042      MOV      #OPR1A,-(SP)
026022 012746 010135      MOV      #OPR3,-(SP)
026026 012746 007651      MOV      #FMTOP1,-(SP)
026032 012746 011442      MOV      #7,-(SP)
026036 012746 000007      MOV      SP,RO
026042 010600      TRAP    C$PNTF
026044 104417      ADD     #20,SP
026046 062706 000020
103
104 026052 012767 000004 154724      MOV      #CYLUP,OPFLAG      :SET CYCLE UP FLAG
105 026060 012703 000001      MOV      #1,R3              :SET EXPECTED STATE VALUE
106 026064 012767 006416 154720      MOV      #NSTACHG,ERHEAD    :SET ERROR HEADER
107 026072 012701 000454      MOV      #300.,R1          :WAIT COUNT R1*TIMDLY= 30 SECONDS.
108 026076 004767 170760      JSR     PC,GSTATC          :GET STATUS
109 026102 030342      T365$
110 026104 005767 154750      TST     T,STAT             :TEST IF STATE IS STILL 0
111 026110 001072      BNE    10$                 :NO - SKIP
112 026112 005301      DEC    R1                  :DEC WAIT COUNT
113 026114 001453      BEQ    7$                  :EXIT IF WAIT DONE
114 026116      TIMDLY 1000.
115 026242 000715      BR     6$
116
117 026244 005067 156114      7$:      CLR     OBUFF              :CLEAR FOR RESPONSE
118 026250      GMANIL OPRO03,OBUFF,1,NO
026250 104443      TRAP    C$GMAN
026252 000404      BR     10000$
026254 004364      .WORD  OBUFF
026256 000120      .WORD  T$CODE
026260 007515      .WORD  OPRO03
026262 000001      .WORD  1
026264
119 026264 005767 156074      10000$: TST     OBUFF              :TEST IF RESPONSE YES
120 026270 001005      BNE    11$                 :YES - REPORT
121 026272 000167 177426      JMP     3$
122 026276 020367 154556      10$:    CMP     R3,T,STAT       :CHECK IF NOW STATE 1
123 026302 001406      BEQ    13$                 :YES - SKIP
124 026304      11$:    FRRHRD 301.,,ERR7
026304 104456      TRAP    C$ERHRD
026306 000455      .WORD  301
    
```

	026310	000000			.WORD	0	
	026312	013666			.WORD	ERR7	
125	026314				EXIT	TST	
	026314	104432			TRAP	C\$EXIT	
	026316	002024			.WORD	L10026-	
126	026320	012701	000454	13\$:	MOV	#300.,R1	;INITIALIZE WAIT COUNT FOR 30 SECONDS
127	026324	012703	000002		MOV	#2,R3	;SET EXPECTED STATE VALUE
128	026330	004767	170526	14\$:	JSR	PC,GSTATC	;GET STATUS
129	026334	030342			T365\$		
130	026336	020367	154516		CMP	R3,T.STAT	;CHECK IF STATE 2
131	026342	001503			BEQ	20\$	;YES - GO TO STATE 2
132	026344	002002			BGE	17\$	;CHECK IF NO CHANGE CONTINUE WAIT
133	026346	000167	001000		JMP	32\$	;GO TO STATE 3.
134	026352	005301		17\$:	DEC	R1	;DEC WAIT COUNT
135	026354	001453			BEQ	18\$	;SKIP IF 0
136	026356				TIMDLY	1000.	
137	026502	000712			BR	14\$	;CHECK FOR STATE CHANGE
138	026504			18\$:	ERRHRD	303.,,ERR7	
	026504	104456			TRAP	C\$ERHRD	
	026506	000457			.WORD	303	
	026510	000000			.WORD	0	
	026512	013666			.WORD	ERR7	
139	026514	032767	004000	154330	BIT	#SPDSTAT,T.MP	;TEST IF SPINDLE TIMEOUT
140	026522	001011			BNE	19\$	;YES - SKIP
141	026524	012767	006430	154260	MOV	#SPDERR,ERHEAD	;SET ERROR HEADER
142	026532	012703	010662		MOV	#MSPERR,R3	;SET NAME MESSAGE POINTER
143	026536				ERRHRD	304.,,ERR3	
	026536	104456			TRAP	C\$ERHRD	
	026540	000460			.WORD	304	
	026542	000000			.WORD	0	
	026544	012600			.WORD	ERR3	
144	026546			19\$:	EXIT	TST	
	026546	104432			TRAP	C\$EXIT	
	026550	001572			.WORD	L10026-	
145							
146	026552	012701	000005	20\$:	MOV	#5,R1	;WAIT .5 SECONDS
147	026556			21\$:	TIMDLY	1000.	
148	026702	005301			DEC	R1	
149	026704	001324			BNE	21\$	
150							
151	026706	004767	170150		JSR	PC,GSTATC	;CHECK TO SEE IF STATE 3, IF YES GO TO STATE 3
152	026712	030342			T365\$		
153	026714	022767	000003	154136	CMP	#3,T.STAT	
154	026722	003002			BGT	22\$	
155	026724	000167	000422		JMP	32\$	
156							
157	026730	012767	006373	154054	22\$:	MOV	#MISTST,ERHEAD
158	026736	012704	011323		MOV	#STATE2,R4	;SET CONDITION MESSAGE POINTER
159	026742	012703	010575		MOV	#MBHSTA,R3	;SET NAME MESSAGE POINTER
160	026746	032767	000010	154076	BIT	#BHSTAT,T.MP	;TEST IF BRUSH HOME STILL SET
161	026754	001006			BNE	23\$	;YES - SKIP
162	026756				ERRHRD	305.,,ERR5	
	026756	104456			TRAP	C\$ERHRD	
	026760	000461			.WORD	305	
	026762	000000			.WORD	0	
	026764	012716			.WORD	ERR5	
163	026766				EXIT	TST	

026766	104432				TRAP	CSEXIT		
026770	001352				.WORD	L10026-		
164 026772	012701	000062		23\$:	MOV	#50.,R1		;SET WAIT COUNT FOR 5 SECONDS
165 026776	004767	170060		24\$:	JSR	PC,GSTATC		;GET STATUS
166 027002	030342				T365\$			
167 027004	032767	000010	154040		BIT	#BHSTAT,T.MP		;TEST IF BRUSH HOME RESET
168 027012	001463				BEQ	27\$		;YES - SKIP
169 027014	005301				DEC	R1		;DEC WAIT COUNT
170 027016	001453				BEQ	26\$		;SKIP IF ZERO
171 027020					TIMDLY	1000.		
172 027144	000714				BR	24\$		;LOOP
173 027146				26\$:	ERRHRD	306.,,ERR4		
027146	104456				TRAP	C\$ERHRD		
027150	000462				.WORD	306		
027152	000000				.WORD	0		
027154	012646				.WORD	ERR4		
174 027156					EXIT	TST		
027156	104432				TRAP	CSEXIT		
027160	001162				.WORD	L10026-		
175 027162	012701	000454		27\$:	MOV	#300.,R1		;INITIALIZE WAIT COUNT FOR 30 SECONDS
176 027166	004767	167670		28\$:	JSR	PC,GSTATC		;GET STATUS
177 027172	030342				T365\$			
178 027174	032767	000010	153650		BIT	#BHSTAT,T.MP		;TEST IF BRUSH HOME SET AGAIN
179 027202	001063				BNE	32\$		;YES - SKIP
180 027204	005301				DEC	R1		;ELSE DEC WAIT COUNT
181 027206	001453				BEQ	30\$		;SKIP IF 0
182 027210					TIMDLY	1000.		
183 027334	000714				BR	28\$		
184 027336				30\$:	ERRHRD	307.,,ERR5		
027336	104456				TRAP	C\$ERHRD		
027340	000463				.WORD	307		
027342	000000				.WORD	0		
027344	012716				.WORD	ERR5		
185 027346					EXIT	TST		
027346	104432				TRAP	CSEXIT		
027350	000772				.WORD	L10026-		
186 027352	012767	006416	153432	32\$:	MOV	#NSTACHG,ERHEAD		;SET ERROR HEADER
187 027360	012703	000003			MOV	#3,R3		;SET EXPECTED STATE VALUE
188 027364	004767	167472			JSR	PC,GSTATC		;GET STATUS
189 027370	030342				T365\$			
190 027372	020367	153462			CMP	R3,T.STAT		;CHECK IF STATE 3
191 027376	001406				BEQ	36\$		;YES - SKIP
192 027400					ERRHRD	308.,,ERR7		
027400	104456				TRAP	C\$ERHRD		
027402	000464				.WORD	308		
027404	000000				.WORD	0		
027406	013666				.WORD	ERR7		
193 027410					EXIT	TST		
027410	104432				TRAP	CSEXIT		
027412	000730				.WORD	L10026-		
194 027414	012767	006373	153370	36\$:	MOV	#MISTST,ERHEAD		;SET ERROR HEADER
195 027422	012704	011333			MOV	#STATE3,R4		;SET CONDITION MESSAGE POINTER
196 027426	012703	010621			MOV	#MHOSTA,R3		;SET NAME MESSAGE POINTER
197 027432	004767	167424			JSR	PC,GSTATC		;GET STATUS
198 027436	030342				T365\$			
199 027440	032767	000020	153404		BIT	#HOSTAT,T.MP		;TEST IF HEADS OUT SET
200 027446	001006				BNE	38\$		;YES - SKIP

201	027450				ERRHRD	309...	ERR5	
	027450	104456			TRAP	C\$ERHRD		
	027452	000465			.WORD	309		
	027454	000000			.WORD	0		
	027456	012716			.WORD	ERR5		
202	027460				EXIT	TST		
	027460	104432			TRAP	C\$EXIT		
	027462	000660			.WORD	L10026-		
203	027464	012701	005670	38\$:	MOV	#3000.,R1		:SET WAIT COUNT FOR 300 MS
204	027470	012767	006416	153314	MOV	#NSTACHG,ERHEAD		:SET ERROR HEADER
205	027476	012703	000004		MOV	#4,R3		:SET EXPECTED STATE VALUE
206	027502	004767	167354	43\$:	JSR	PC,GSTATC		:GET STATUS
207	027506	030342			T365\$			
208	027510	020367	153344		CMP	R3,T.STAT		:CHECK IF STATE 4
209	027514	001463			BEQ	49\$		:YES - SKIP
210	027516	005301			DEC	R1		:DEC WAIT COUNT
211	027520	001453			BEQ	47\$		:SKIP IF 0
212	027522				TIMDLY	1		
213	027646	000715			BR	43\$		
214	027650			47\$:	ERRHRD	312...	ERR7	
	027650	104456			TRAP	C\$ERHRD		
	027652	000470			.WORD	312		
	027654	000000			.WORD	0		
	027656	013666			.WORD	ERR7		
215	027660				EXIT	TST		
	027660	104432			TRAP	C\$EXIT		
	027662	000460			.WORD	L10026-		
216	027664	012701	000454	49\$:	MOV	#300.,R1		:SET WAIT COUNT FOR 30 MS
217	027670	012703	000005		MOV	#5,R3		:SET EXPECTED STATE VALUE
218	027674	004767	167162	50\$:	JSR	PC,GSTATC		:GET STATUS
219	027700	030342			T365\$			
220	027702	020367	153152		CMP	R3,T.STAT		:CHECK IF STATE 5
221	027706	001463			BEQ	53\$		:YES - SKIP
222	027710	005301			DEC	R1		:DEC WAIT COUNT
223	027712	001453			BEQ	52\$		:ELSE SKIP
224	027714				TIMDLY	1		
225	030040	000715			BR	50\$		
226	030042			52\$:	ERRHRD	313...	ERR7	
	030042	104456			TRAP	C\$ERHRD		
	030044	000471			.WORD	313		
	030046	000000			.WORD	0		
	030050	013666			.WORD	ERR7		
227	030052				EXIT	TST		
	030052	104432			TRAP	C\$EXIT		
	030054	000266			.WORD	L10026-		
228	030056	032767	001000	152766	53\$:	BIT	#VCSTAT,T.MP	:VOLUME CHECK SHOULD BE SET FOR
229	030064	001010			BNE	54\$		:STATE 5, IF NOT GIVE ERROR.
230	030066	012703	010551		MOV	#MVOLCK,R3		:SET NAME MESSAGE POINTER
231	030072				ERRHRD	310...	ERR5	
	030072	104456			TRAP	C\$ERHRD		
	030074	000466			.WORD	310		
	030076	000000			.WORD	0		
	030100	012716			.WORD	ERR5		
232	030102				EXIT	TST		
	030102	104432			TRAP	C\$EXIT		
	030104	000276			.WORD	L10026-		
233	030106	032767	040000	152730	54\$:	BIT	#DRVERR,T.CS	:TEST IF DRIVE ERROR SET

234	030114	001010			BNE	57\$		:YES - SKIP
235	030116	012703	010526		MOV	#MDRERR,R3		:SET NAME MESSAGE POINTER
236	030122				ERRHRD	315,,ERR5		
	030122	104456			TRAP	C\$ERHRD		
	030124	000473			.WORD	315		
	030126	000000			.WORD	0		
	030130	012716			.WORD	ERR5		
237	030132				EXIT	TST		
	030132	104432			TRAP	C\$EXIT		
	030134	000206			.WORD	L10026-		
238	030136	012701	000120	57\$:	MOV	#80,,R1		:SET WAIT FOR 8 MS
239	030142	004767	166714	56\$:	JSR	PC,GSTATC		:GET STATUS
240	030146	030342			T365\$			
241	030150	032767	000001	152666	BIT	#DRDYMSK,T.CS		:CHECK IF DRIVE READY
242	030156	001071			BNE	172\$		:YES - SKIP
243	030160	005301			DEC	R1		:DEC COUNT
244	030162	001453			BEQ	58\$		:SKIP IF 0
245	030164				TIMDLY	1		
246	030310	000714			BR	56\$		
247	030312	012767	006373	152472	58\$:	MOV	#MISTST,ERHEAD	:SET ERROR HEADER
248	030320	012704	011343		MOV	#STAT5,R4		:SET CONDITION MESSAGE POINTER
249	030324	012703	010404		MOV	#MDRDY,R3		:SET NAME MESSAGE POINTER
250	030330				ERRHRD	316,,ERR5		
	030330	104456			TRAP	C\$ERHRD		
	030332	000474			.WORD	316		
	030334	000000			.WORD	0		
	030336	012716			.WORD	ERR5		
251	030340	000400			BR	172\$		:EXIT TEST
252	030342				172\$:			
253	030342				T365\$:			
254	030342				ENDTST			
	030342				L10026:			
	030342	104401			TRAP	C\$ETST		

255								
256								
257								
258					.SETTL	*TEST 4	HEAD UNLOADING	
259	030344				BGNTST		;TEST04	
	030344							T4::
260								:SPIN DOWN AND UNLOAD HEADS. VERIFY THAT THE DRIVE
261								:GOES FROM STATE 5 TO STATE 7 PROPERLY.
262	030344	005767	153010		TST	PASNUM		:TEST IF FIRST PASS
263	030350	001003			BNE	8\$		:NO - SKIP
264	030352	005767	163750		TST	MISWIW		:TEST IF MANUAL INTERVENTION
265	030356	100403			BMI	TST4		:YES - SKIP
266	030360				8\$:	EXIT		
	030360	104432			TRAP	C\$EXIT		
	030362	001146			.WORD	L10027-		
267								
268	030364				BGNSUB			14.1:
	030364							
	030364	104402			TRAP	C\$BSUB		
269	030366	012767	006416	152416	TST4:	MOV	#NSTACHG,ERHEAD	:SET ERROR HEADER
270	030374	004767	166430		JSR	PC,TSTINT		:INITIALIZE TEST
271	030400	004767	166442		JSR	PC,GSTATR		:GET STATUS
272	030404	031420			T465\$			
273	030406	032767	000001	152430	BIT	#DRDYMSK,T.CS		:CHECK IF DRIVE READY

```

274 030414 001040          BNE      3$          ;YES - SKIP
275                                     ;PROMPT PRESS LD AND WAIT FOR RDY
276 030416          1$: PRINTF  #FMTOP1,#OPR6,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
    030416 005046          CLR      -(SP)
    030420 156716 152407    BISB   RLDRV+1,(SP)
    030424 012746 006053    MOV    #DRVNAM,-(SP)
    030430 016746 152372    MOV    RLBAS,-(SP)
    030434 012746 006042    MOV    #BASADD,-(SP)
    030440 012746 010135    MOV    #OPR1A,-(SP)
    030444 012746 007665    MOV    #OPR6,-(SP)
    030450 012746 011442    MOV    #FMTOP1,-(SP)
    030454 012746 000007    MOV    #7,-(SP)
    030460 010600          MOV    SP,R0
    030462 104417          TRAP   C$PNTF
    030464 062706 000020    ADD    #20,SP
277 030470 005067 153670    CLR    OBUF          ;CLEAR FOR RESPONSE
278 030474          GMANIL  OPRO02,OBUF,1,NO
    030474 104443          TRAP   C$GMAN
    030476 000404          BR     10000$
    030500 004364          .WORD OBUF
    030502 000120          .WORD T$CODE
    030504 007470          .WORD OPRO02
    030506 000001          .WORD 1
279 030510 005767 153650    10000$: TST   OBUF          ;TST RESPONSE YES
280 030514 001740          BEQ    1$          ;NO - SKIP
281
282 030516 052767 000010 152260 3$: BIS    #UNLOAD,OPFLAG ;SET UNLOAD OPERATION
283                                     ;PROMPT PRESS LOAD
284 030524          4$: PRINTF  #FMTOP1,#OPR3,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
    030524 005046          CLR      -(SP)
    030526 156716 152301    BISB   RLDRV+1,(SP)
    030532 012746 006053    MOV    #DRVNAM,-(SP)
    030536 016746 152264    MOV    RLBAS,-(SP)
    030542 012746 006042    MOV    #BASADD,-(SP)
    030546 012746 010135    MOV    #OPR1A,-(SP)
    030552 012746 007651    MOV    #OPR3,-(SP)
    030556 012746 011442    MOV    #FMTOP1,-(SP)
    030562 012746 000007    MOV    #7,-(SP)
    030566 010600          MOV    SP,R0
    030570 104417          TRAP   C$PNTF
    030572 062706 000020    ADD    #20,SP
285
286 030576 012703 000006          MOV    #6,R3          ;SET EXPECTED STATE VALUE
287 030602 012704 000144          MOV    #100.,R4       ;SET SECOND LEVEL COUNT
288 030606 012701 001274          MOV    #700.,R1       ;SET WAIT COUNT FOR 30 SECONDS
289 030612 004767 166244          5$: JSR    PC,GSTATC   ;GET STATUS
290 030616 031420          T465$
291 030620 020367 152234          CMP    R3,T.STAT      ;WATCH FOR STATE CHANGE FROM 5 TO 6
292 030624 001506          BEQ    11$           ;YES - SKIP
293 030626 022767 000005 152224    CMP    #5,T.STAT      ;TEST IF STATE 5
294 030634 001074          BNE    9$            ;NO - REPORT WRONG STATE
295 030636 005304          8$: DEC    R4          ;DEC 2ND LEVEL COUNT
296 030640 001004          BNE    6$            ;SKIP IF NOT 0
297 030642 005301          DEC    R1            ;ELSE DEC 1ST LEVEL COUNT
298 030644 001455          BEQ    7$            ;IF 0 - SKIP TO QUESTION
299 030646 012704 000144          MOV    #100.,R4       ;ELSE RESET 2ND LEVEL
  
```



300	030652			6\$:	TIMDLY	1		;WAIT 100 US
301	030776	000705			BR	5\$		
302	031000	005067	153360	7\$:	CLR	OBUFF		;CLEAR FOR RESPONSE
303	031004				GMANIL	OPR003,OBUFF,1,NO		
	031004	104443			TRAP	C\$GMAN		
	031006	000404			BR	10001\$		
	031010	004364			.WORD	OBUFF		
	031012	000120			.WORD	T\$CODE		
	031014	007515			.WORD	OPR003		
	031016	000001			.WORD	1		
	031020			10001\$:				
304	031020	005767	153340		TST	OBUFF		;TEST IF RESPONSE YES
305	031024	001637			BEQ	4\$		;NO - SKIP
306	031026			9\$:	ERRHRD	401,,,ERR7		;ELSE REPORT STATE CHANGE WRONG
	031026	104456			TRAP	C\$ERRHRD		
	031030	000621			.WORD	401		
	031032	000000			.WORD	0		
	031034	013666			.WORD	ERR7		
307	031036				EXIT	SUB		
	031036	104432			TRAP	C\$EXIT		
	031040	000366			.WORD	L10030-		
308	031042	012703	000007	11\$:	MOV	#7,R3		;SET EXPECTED STATE VALUE
309	031046	012701	005670		MOV	#3000,,R1		;SET COUNT FOR 300MS
310	031052	004767	166004	12\$:	JSR	PC,G\$STATC		;GET STATUS
311	031056	031420			T465\$			
312	031060	020367	151774		CMP	R3,T\$STAT		;CHECK IF STATE 7
313	031064	001463			BEQ	18\$		;YES - SKIP
314	031066	005301			DEC	R1		;DEC WAIT COUNT
315	031070	001453			BEQ	16\$		;TIME OUT GIVE ERROR MESSAGE
316	031072				TIMDLY	1		
317	031216	000715			BR	12\$		
318	031220			16\$:	ERRHRD	402,,,ERR7		;REPORT WRONG STATE CHANGE
	031220	104456			TRAP	C\$ERRHRD		
	031222	000622			.WORD	402		
	031224	000000			.WORD	0		
	031226	013666			.WORD	ERR7		
319	031230				EXIT	SUB		
	031230	104432			TRAP	C\$EXIT		
	031232	000174			.WORD	L10030-		
320	031234	005003		18\$:	CLR	R3		;SET EXPECTED STATE VALUE
321	031236	012701	013560		MOV	#6000,,R1		;SET WAIT COUNT FOR 60 SECONDS
322	031242	004767	165614	20\$:	JSR	PC,G\$STATC		;GET STATUS
323	031246	031420			T465\$			
324	031250	005767	151604		TST	T\$STAT		;CHECK IF STATE 0
325	031254	001461			BEQ	24\$		;YES - SKIP
326	031256	005301			DEC	R1		;DEC WAIT COUNT
327	031260	001453			BEQ	22\$		;SKIP IF 0
328	031262				TIMDLY	100.		
329	031406	000715			BR	20\$		
330	031410			22\$:	ERRHRD	403,,,ERR7		;REPORT WRONG STATE CHANGE
	031410	104456			TRAP	C\$ERRHRD		
	031412	000623			.WORD	403		
	031414	000000			.WORD	0		
	031416	013666			.WORD	ERR7		
331	031420			24\$:				
332	031420	012767	000002	T465\$:	MOV	#2,ERRSWI		;INIT ERROR SWITCH
333			151370					

```

334 031426          ENDSUB
      031426          L10030:
      031426 104403  TRAP    C$ESUB
335                                     :PROMPT PRESS LD AND WAIT FOR RDY
336 031430          26$: PRINTF #FMTOP1,#OPR6,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
      031430 005046          CLR    -(SP)
      031432 156716          BISR  RLDRV+1,(SP)
      031436 012746 006053  MOV    #DRVNAM,-(SP)
      031442 016746 151360  MOV    RLBAS,-(SP)
      031446 012746 006042  MOV    #BASADD,-(SP)
      031452 012746 010135  MOV    #OPR1A,-(SP)
      031456 012746 007665  MOV    #OPR6,-(SP)
      031462 012746 011442  MOV    #FMTOP1,-(SP)
      031466 012746 000007  MOV    #7,-(SP)
      031472 010600          MOV    SP,RO
      031474 104417          TRAP  C$PNTF
      031476 062706 000020  ADD    #20,SP
337
338 031502 005067 152656  CLR    OBUF          :CLEAR FOR RESPONSE
339 031506          GMANIL OPRO02,OBUF,1,NO
      031506 104443  TRAP  C$GMAN
      031510 000404  BR    10000$
      031512 004364  .WORD OBUF
      031514 000120  .WORD T$CODE
      031516 007470  .WORD OPRO02
      031520 000001  .WORD 1
      031522          10000$:
340 031522 005767 152636  TST   OBUF          :TEST IF RESPONSE YES
341 031526 001740          BEQ   26$           :NO - SKIP
342 031530
343
344 031530          ENDTST
      031530          L10027:
      031530 104401  TRAP  C$ETST
345
346
347
348
349 031532          .SBTTL *TEST 5          DRIVE SELECT
      031532          BGNTST          :TEST05
350 031532 012767 000002 151256  MOV    #2,ERRSWI    :SET FOR NO ERROR RETURN
351 031540 005767 151614          IST   PASNUM       :TEST IF FIRST PASS
352 031544 001173          BNE   EXT05        :NO - SKIP
353 031546 032767 000004 162552  BIT   #DRSELT,MISWIW :TEST IF SELECT TESTS
354 031554 001567          BEQ   EXT05        :NO - SKIP
355 031556          1$: PRINTF #FMTOP1,#OPR7,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
      031556 005046          CLR    -(SP)
      031560 156716          BISR  RLDRV+1,(SP)
      031564 012746 006053  MOV    #DRVNAM,-(SP)
      031570 016746 151232  MOV    RLBAS,-(SP)
      031574 012746 006042  MOV    #BASADD,-(SP)
      031600 012746 010135  MOV    #OPR1A,-(SP)
      031604 012746 007720  MOV    #OPR7,-(SP)
      031610 012746 011442  MOV    #FMTOP1,-(SP)
      031614 012746 000007  MOV    #7,-(SP)
      031620 010600          MOV    SP,RO
      031622 104417          TRAP  C$PNTF
    
```

```

031624 062706 00CJ20      ADD      #20,SP
356 031630 005067 152530      CLR      OBUFF
357 031634 104443      GMANIL  OPRO02,OBUFF,1,NO
358 031634 000404      TRAP    C$GMAN
031636 004364      ER      10000$
031640 000120      .WORD  OBUFF
031642 0007470      .WORD  T$CODE
031644 000001      .WORD  OPRO02
031646 10000$      .WORD  1
359 031650 005767 152510      TST     OBUFF
360 031654 001740      BEQ     1$
361 031656 012767 006530 151126 3$:  MOV     #TOSERR,EPHEAD
362 031664 004767 165140      JSR     PC,TSTINT
363 031670 004767 165166      JSR     PC,GSTATC
364 031674 032056      T504$
365 031676 016767 151130 151212  MOV     RLDRV,TEMPO
366 031704 016701 151122      MOV     RLDRV,R1
367 031710 012704 000004      MOV     #4,R4
368 031714 062701 000400      LPT05: ADD     #400,R1
369 031720 022701 002000      CMP     #2000,R1
370 031724 001001      BNE     4$
371 031726 005001      CLR     R1
372 031730 010167 151076      4$:  MOV     R1,RLDRV
373 031734 012746 010151      5$:  PRINTF #FMTOP3,#OPR8,<B,RLDRV+1>,#OPR1B,#UNDTST
031734 012746 010141      MOV     #UNDTST,-(SP)
031740 005046      MOV     #OPR1B,-(SP)
031744 156716 151061      CLR     -(SP)
031746 007747      BISB   RLDRV+1,(SP)
031752 012746 007747      MOV     #OPR8,-(SP)
031756 012746 011513      MOV     #FMTOP3,-(SP)
031762 010600      MOV     #5,-(SP)
031766 104417      MOV     SP,R0
031770 062706 000014      TRAP   C$PnTF
374 031772 005067 152362      ADD     #14,SP
375 031776 005067 152362      CLR     OBUFF
376 032002 104443      GMANIL  OPRO02,OBUFF,1,NO
032002 000404      TRAP   C$GMAN
032004 004364      BR     10001$
032006 000120      .WORD  OBUFF
032010 0007470      .WORD  T$CODE
032012 000001      .WORD  OPRO02
032014 10001$:  .WORD  1
377 032016 005767 152342      TST     OBUFF
378 032022 001744      BEQ     5$
379 032024 104402      BGNSUB
032024 004767 165030      TRAP   C$BSUB
380 032032 000002 150754 60$:  JSR     PC,GSTATC
381 032034 012767 000002 150754 60$:  MOV     #2,ERRSWI
382 032042 100032:  ENDSUB
383 032042
384 032042
  
```

T5.1:

;GET STATUS - REPORT ANY ERROR

;INIT ERROR SWITCH

```

032042 104403 TRAP C$ESUB
385 032044 005304 DEC R4 ;DEC COUNT
386 032046 001322 BNE LPT05 ;LOOP IF NOT ZERO
387 032050 016767 151042 150754 MOV TEMPO,RLDRV ;ELSE RESTORE RLDRV
388 032056 T504$:
389 032056 4$: PRINTF #FMT4,#OPR8,#OPR9
032056 012746 007766 MOV #OPR9,-(SP)
032062 012746 007747 MOV #OPR8,-(SP)
032066 012746 011556 MOV #FMT4,-(SP)
032072 012746 000003 MOV #3,-(SP)
032076 010600 MOV SP,R0
032100 104417 TRAP C$PNTF
032102 062706 000010 ADD #10,SP
390 032106 005067 152252 CLR OBUFF ;CLEAR FOR RESPONSE
391 032112 GMANIL OPRO02,OBUFF,1,N0
032112 104443 TRAP C$GMAN
032114 000404 BR 10000$
032116 004364 .WORD OBUFF
032120 000120 .WORD T$CODE
032122 007470 .WORD OPRO02
032124 000001 .WORD 1
032126 10000$:
392 032126 005767 152232 TST OBUFF ;TEST IF RESPONSE YES
393 032132 001751 BEQ 4$ ;NO - SKIP
394 032134 EXT05:
395 032134 ENDTST
032134 L10031:
396 032134 104401 TRAP C$ETST
397
398
399 .SBTTL *TEST 6 DRIVE SELECT ERROR TEST
400 BGNSTST ;TEST06
032136 T6::
032136 005767 151216 TST PASNUM ;CHECK IF FIRST PASS
401 032136 005767 151216 TST PASNUM ;CHECK IF FIRST PASS
402 032142 001004 BNE 1$ ;NO - SKIP
403 032144 032767 000004 162154 BIT #DRSELT,MISWIW ;CHECK IF TEST DRIVE SELECT
404 032152 001002 BNE 6$ ;YES - SKIP
405 032154 1$: EXIT
032154 104432 TRAP C$EXIT
032156 001230 .WORD L10033-
406 032160 012767 006464 150624 6$: MOV #GSTER1,ERHEAD ;SET ERROR HEADER
407 032166 004767 164636 JSR PC,TSTINT ;INITIALIZE TEST
408 032172 016703 151164 MOV PS$TNM,R3 ;GET PARAM SET NUMBER
409 032176 026727 147610 000001 CMP L$UNIT,#1 ;TEST IF MORE THAN 1 UNIT
410 032204 101517 BLOS 5$ ;NO - SKIP
411 032206 005203 2$: INC R3 ;BUMP PARAMETER SET NUMBER
412 032210 020367 147576 CMP R3,L$UNIT ;CHECK IF PAST VALID PARAMETER TABLE
413 032214 101401 BLOS 3$ ;NO - SKIP
414 032216 005003 CLR R3 ;ELSE CLEAR TO POINT TO ENTRY 0
415 032220 3$: GPHARD R3,R0
032220 010300 MOV R3,R0
032222 104442 TRAP C$GPHRD
416 032224 BNCOMPLETE 2$ ;SKIP IF NOT AVAILABLE
032224 103370 BCC 2$
417 032226 010004 MOV R0,R4 ;PUT POINTER INTO R4
418 032230 021467 150572 CMP (R4),RLBAS ;CHECK IF SAME CONTROLLER
  
```

```

419 032234 001364 BNE 2$ ;NO - SKIP
420 032236 005067 150544 CLR DONE ;CLEAR DONE FLAG
421 032242 012767 000104 150564 MOV #GTSTAT,L,CS ;LOAD GET STATUS
422 032250 056467 000010 150556 BIS 10(R4),L,CS ;INSERT DRIVE
423 032256 012767 000013 150554 MOV #GETSTAT!DRSET,L,DA ;SET UP TO CLEAR DRIVE
424 032264 016762 150550 000004 MOV L,DA,RLDA(R2) ;LOAD DA REG
425 032272 016762 150536 000000 MOV L,CS,RLCS(R2) ;LOAD CS REG
426 032300 TIMDLY 30. ;WAIT 3 MS
427 032424 005767 150356 TST DONE ;TEST IF INTERRUPT
428 032430 001666 BEQ 2$ ;NO - SKIP
429 032432 032767 100000 150404 BIT #ANYERR,T,CS ;TEST IF ANY ERROR SET
430 032440 001415 BEQ 7$ ;NO - GO TEST
431 032442 000661 BR 2$ ;ELSE CHECK NEXT DRIVE
432 032444 5$: PRINTF #FMT9,#OPR10 ;REPORT CAN'T FIND 2ND DRIVE
    032444 012746 010003 MOV #OPR10,-(SP)
    032450 012746 011753 MOV #FMT9,-(SP)
    032454 012746 000002 MOV #2,-(SP)
    032460 010600 MOV SP,RO
    032462 104417 TRAP C$PNTF
    032464 062706 000006 ADD #6,SP
433 032470 000167 000712 JMP LCLEXT
434 032474 016467 000010 150416 7$: MOV 10(R4),TEMP1 ;STORE NEW ADDRESS
435 ;ASK FOR PLUG CHANGE
436 032502 016700 150324 9$: MOV RLDRV,RU ;GET DRIVE UNDER TEST
437 032506 016705 150406 MOV TEMP1,R5 ;GET NEW ADDRESS
438 032512 042700 002000 BIC #2000,R0 ;CLEAR FOR ADDRESS 0 TO 3
439 032516 042705 002000 BIC #2000,R5
440 032522 020527 001400 20$: CMP R5,#1400 ;TEST IF DRIVE NUMBER 3
441 032526 001001 BNE 21$ ;NO - SKIP
442 032530 005005 CLR R5 ;ELSE SET TO DRIVE NUMBER 0
443 032532 062705 000400 21$: ADD #400,R5 ;BUMP TO NEXT ADDRESS
444 032536 020500 CMP R5,R0 ;THIS EQUAL TO NEW ADDRESS?
445 032540 001770 BEQ 20$ ;YES - SKIP
446 032542 052705 000200 BIS #CRDYMSK,R5 ;ELSE SET CONTROLLER READY BIT
447 032546 010562 000000 MOV R5,RLCS(R2) ;AND LOAD CS REG
448 ;PROMPT INSRT ADR PLG AN DRV
449 032552 PRINTF #FMTOP2,#OPR8,<B,RLDRV+1>,#OPR1B,<B,TEMP1+1>
    032552 005046 CLR -(SP)
    032554 156716 150341 BISB TEMP1+1,(SP)
    032560 012746 010141 MOV #OPR1B,-(SP)
    032564 005046 CLR -(SP)
    032566 156716 150241 BISB RLDRV+1,(SP)
    032572 012746 007747 MOV #OPR8,-(SP)
    032576 012746 011471 MOV #FMTOP2,-(SP)
    032602 012746 000005 MOV #5,-(SP)
    032606 010600 MOV SP,RO
    032610 104417 TRAP C$PNTF
    032612 062706 000014 ADD #14,SP
450 032616 005067 151542 CLR OBUFF ;CLEAR FOR RESPONSE
451 032622 GMANIL OPR002,OBUFF,1,NO
    032622 104443 TRAP C$GMAN
    032624 000404 BR 10000$
    032626 004364 .WORD OBUFF
    032630 000120 .WORD T$CODE
    032632 007470 .WORD OPR002
    032634 000001 .WORD 1
    032636 10000$:
  
```

```

452 032636 005767 15 522          TST      OBUFF          ;TEST IF RESPONSE YES
453 032642 001717                    BEQ      9$            ;NO - SKIP
454 032644 012704 0G0012          MOV      #10.,R4       ;SET COUNT
455 032650                    BGNSUB
    032650                    T6.1:
    032650 104402          TRAP     C$BSUB
456 032652 016767 150154 150154 8$:  MOV     RLDRV,L.CS    ;SET UP TO SELECT MULTIPLE DRIVES
457 032660 016762 150150 000000  MOV     L.CS,RLCSR(R2) ;DO IT
458 032666                    TIMDLY  100.
459 033012 052767 000104 150014  BIS     #GTSTAT,L.CS   ;SET GET STATUS
460 033020 012767 000013 150012  MOV     #GETSTAT!DRSET,L.DA ;SET RESET BIT 3 IN THE DA REG FOR THE
461                    ;/DRIVE TO CLEAR ITS ERROR REGISTER
462                    ;/BEFORE SENDING A STATUS WORD TO THE
463                    ;/MP REG DURING GET STATUS COMMAND
464
465 033026 016762 150006 000004  MOV     L.DA,RLDA(R2)
466 033034 005067 147746          CLR     DONE
467 033040 016762 147770 000000  MOV     L.CS,RLCSR(R2) ;DO GET STATUS
468 033046                    WAITUS  1          ;WAIT FOR INTERRUPT
    033046 012727 000001  MOV     #1,(PC)+
    033052 000000          .WORD  0
    033054 016727 147036  MOV     L$DLY,(PC)+
    033060 000000          .WORD  0
    033062 005367 177772  DEC     -6(PC)
    033066 001375          BNE     -4
    033070 005367 177756  DEC     -22(PC)
    033074 001367          BNE     -20
469 033076 005767 147704          TST     DONE          ;CHECK IF INTERRUPTED
470 033102 001012          BNE     12$          ;YES - SKIP
471 033104 004767 163526          JSR     PC,WAITIN    ;WAIT FOR TIMEOUT
472 033110 012603          MOV     (SP)+,R3     ;GET ERROR POINTER
473 033112 001406          BEQ     12$          ;SKIP IF 0
474 033114          ERRHRD  601.,GSTER1,ERR1
    033114 104456          TRAP     C$ERHRD
    033116 001131          .WORD  601
    033120 006464          .WORD  GSTER1
    033122 012464          .WORD  ERR1
475 033124          EXIT     SUB
    033124 104432          TRAP     C$EXIT
    033126 000204          .WORD  L10034-.
476 033130                    TIMDLY  20.          ;WAIT FOR DSE TO SET
477 033254 004767 164722          JSR     PC,GDRSTA    ;GET STATUS
478 033260 032767 000400 147564  BIT     #DSESTAT,T.MP ;TEST IF DRIVE SELECT ERROR SET
479 033266 001010          BNE     16$          ;YES - SKIP
480 033270 012703 010632          MOV     #MDSEERR,R3 ;SET NAME MESSAGE POINTER
481 033274          ERRHRD  602.,ERR3
    033274 104456          TRAP     C$ERHRD
    033276 001132          .WORD  602
    033300 000000          .WORD  0
    033302 012600          .WORD  ERR3
482 033304          EXIT     SUB
    033304 104432          TRAP     C$EXIT
    033306 000C24          .WORD  L10034-.
483 033310 010562 000000          16$:  MOV     R5,RLCS(R2)   ;LOAD IN DIFFERENT ADDRESS
484 033314 005304          DEC     R4          ;DEC COUNT
485 033316 001402          BEQ     60$          ;LOOP IF NOT ZERO
486 033320 000167 177326          JMP     8$
  
```

```

487 033324 012767 00C002 147464 60$: MOV #2,ERRSWI ;INIT ERROR SWITCH
488 033332 ENDSUB
033332 L10034:
033332 104403
489 033334 15$: TRAP C$ESUB ;REQUEST PLUG CHANGE
033334 C12746 010051 PRINTF #FMT9,#OPR11
033340 012746 011753 MOV #OPR11,-(SP)
033344 012746 000002 MOV #FMT9,-(SP)
033350 010600 MOV #2,-(SP)
033352 104417 MOV SP,R0
033354 062706 00C006 TRAP C$PNTF
490 033360 005067 151000 CLR #6,SP ;CLEAR FOR RESPONSE
491 033364 GMANIL OPR002,OBUFF,1,NO
033364 104443 TRAP C$GMAN
033366 000404 BR 10000$
033370 004364 .WORD OBUFF
033372 000120 .WORD T$CODE
033374 007470 .WORD OPR002
033376 000001 .WORD 1
033400 10000$:
492 033400 005767 150760 TST OBUFF ;TEST IF RESPONSE YES
493 033404 001753 BEQ 15$ ;NO - SKIP
494 033406
495 033406 LCLEXT:
033406 ENDTST
033406 L10033:
496 033406 104401 TRAP C$ETST
497
498
499
500 .SBTTL *TEST 7 INITIAL STATE
BGNTST ;TEST 07
501 033410 005767 147744 TST PASNUM ;CHECK IF FIRST PASS
502 033414 001003 BNE 1$ ;NO - EXIT TEST
503 033416 005767 160704 TST MISWIW ;CHECK IF MANUAL INTERVENTION
504 033422 100402 BMI 3$ ;PERFORM TEST IF MANUAL INTERVENTION
505 033424 1$: EXIT TST
033424 104432 TRAP C$EXIT
033426 000652 .WORD L10035-
506 033430 012767 006515 147354 3$: MOV #INITST,ERHEAD ;SET ERROR HEADER
507 033436 004767 163366 JSR PC,TSTINT ;INITIALIZE TEST
508 033442 TIMDLY 10. ;WAIT 1 MS
509 033566 004767 163254 JSR PC,GSTATR ;GET STATUS WITH RESET
510 033572 034300 100$
511 033574 032767 000001 147242 BIT #DRDYMSK,T.CS ;CHECK IF DRIVE IS READY
512 033602 001432 BEQ 20$ ;BRANCH IF DRIVE IS NOT READY
513
514 033604 052767 000010 147172 BIS #ULOAD,OPFLAG ;SET UNLOAD OPERATION
515 ;PROMPT OPERATOR TO 'PRESS LOAD'
516 033612 PRINTF #FMTOP1,#OPR3,#OPR1A,#BASADD,RLBAS,#DRVNM,<B,RLDRV+1>
033612 005046 CLR -(SP)
033614 156716 147213 BISB RLDRV+1,(SP)
033620 012746 006053 MOV #DRVNM,-(SP)
033624 016746 147176 MOV RLBAS,-(SP)
033630 012746 006042 MOV #BASADD,-(SP)
033634 012746 010135 MOV #OPR1A,-(SP)
033640 012746 007651 MOV #OPR3,-(SP)
  
```

	033644	012746	011442		MOV	#FMTOP1,-(SP)	
	033650	012746	000007		MOV	#7,-(SP)	
	033654	010600			MOV	SP,R0	
	033656	104417			TRAP	C\$PNTF	
	033660	062706	000020		ADD	#20,SP	
517	033664	012703	000000		MOV	#0,R3	;SET 'LOAD CARTRIDGE' STATE VALUE 0
518							
519	033670	004767	163166	20\$:	JSR	PC,GSTATC	;GET STATUS
520	033674	034300			100\$		
521	033676				BREAK		;MAKE A SUPERVISOR CALL
	033676	104422			TRAP	C\$BRK	
522	033700	022767	000000	147152	CMP	#0,T.STAT	;TEST IF STATE 0
523	033706	001370			BNE	20\$	;WAIT FOR STATE 0
524							
525							;PROMPT OPERATOR TO 'PRESS LOAD &
526							;WAIT FOR READY'
527	033710			21\$:	PRINTF	#FMTOP1,#OPR6,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>	
	033710	005046			CLR	-(SP)	
	033712	156716	147115		BISB	RLDRV+1,(SP)	
	033716	012746	006053		MOV	#DRVNAM,-(SP)	
	033722	016746	147100		MOV	RLBAS,-(SP)	
	033726	012746	006042		MOV	#BASADD,-(SP)	
	033732	012746	010135		MOV	#OPR1A,-(SP)	
	033736	012746	007665		MOV	#OPR6,-(SP)	
	033742	012746	011442		MOV	#FMTOP1,-(SP)	
	033746	012746	000007		MOV	#7,-(SP)	
	033752	010600			MOV	SP,R0	
	033754	104417			TRAP	C\$PNTF	
	033756	062706	000020		ADD	#20,SP	
528	033762	005067	150376		CLR	OBUFF	;CLEAR FOR RESPONSE
529	033766				GMANIL	OPR002,OBUFF,1,NO	;PROMPT OPERATOR FOR RESPONSE
	033766	104443			TRAP	C\$GMAN	
	033770	000404			BR	10000\$	
	033772	004364			.WORD	OBUFF	
	033774	000120			.WORD	T\$CGDE	
	033776	007470			.WORD	OPR002	
	034000	000001			.WORD	1	
	034002			10000\$:			
530	034002	005767	150356		TST	OBUFF	;TEST IF RESPONSE IS YES
531	034006	001740			BEQ	21\$	;BRANCH IF NOT READY
532							
533	034010	004767	163046	22\$:	JSR	PC,GSTATC	;GET STATUS
534	034014	034300			100\$		
535	034016				BREAK		;MAKE A SUPERVISOR CALL
	034016	104422			TRAP	C\$BRK	
536	034020	022767	000005	147032	CMP	#5,T.STAT	;CHECK IF STATE 5
537	034026	001370			BNE	22\$	;WAIT FOR STATE 5
538							
539	034030	016701	147016		MOV	T.MP,R1	;GET MP REG
540	034034	032701	000020		BIT	#HOSTAT,R1	;CHECK HEADS OUT
541	034040	001003			BNE	7\$	;YES-SKIP
542	034042	012703	010621		MOV	#MHOSTA,R3	;SET NAME MESSAGE PTR
543	034046	000405			BR	9\$	;GO REPORT
544	034050	032701	000010	7\$:	BIT	#HSTAT,R1	;CHECK BRUSH HOME SET
545	034054	001010			BNE	10\$	;YES-SKIP
546	034056	012703	010575		MOV	#MBHSTA,R3	;SET NAME MESSAGE PTR
547	034062			9\$:	ERRHRD	702.,,ERR3	;REPORT ERROR



	034062	104456			TRAP	C\$ERHRD		
	034064	001276			.WORD	702		
	034066	000000			.WORD	0		
	034070	012600			.WORD	ERR3		
548	034072				EXIT	TST		;EXIT
	034072	104432			TRAP	C\$EXIT		
	034074	000204			.WORD	L10035-		
549	034076	005767	160224	10\$:	TST	MISWIW		;TEST IF MANUAL INTERVENTION RUN
550	034102	100035			BPL	16\$		;NO-SKIP
551	034104	005767	147250		TST	PASNUM		;CHECK IF FIRST PASS
552	034110	001032			BNE	16\$		;NO-SKIP
553	034112	032701	000100		BIT	#HSSTAT,R1		;ELSE CHECK HD 0 SELECTED
554	034116	001412			BEQ	13\$		;YES-SKIP
555	034120	012703	010537		MOV	#MHSTA,R3		;SET NAME MESSAGE PTR
556	034124	012704	011412		MOV	#CCYLUP,R4		;SET CONDITION POINTER
557	034130				ERRHRD	703,,,ERR4		;REPORT ERRGR
	034130	104456			TRAP	C\$ERHRD		
	034132	001277			.WORD	703		
	034134	000000			.WORD	0		
	034136	012646			.WORD	ERR4		
558	034140				EXIT	TST		;EXIT
	034140	104432			TRAP	C\$EXIT		
	034142	000136			.WORD	L10035-		
559	034144	032701	001000	13\$:	BIT	#VCSTAT,R1		;CHECK VOL CHECK SET
560	034150	001003			BNE	15\$		;YES-SKIP
561	034152	012703	010551		MOV	#MVOLCK,R3		;ELSE SET NAME MESSAGE PTR
562	034156	000741			BR	9\$		;GO REPORT
563	034160	032767	040000	146656	15\$:	BIT	#DRVERR,T.CS	;TEST DRIVE ERROR SET
564	034166	001003			BNE	16\$		;YES-SKIP
565	034170	012703	010526		MOV	#MDRERR,R3		;ELSE SET NAME MESSAGE PTR
566	034174	000732			BR	9\$		;GO REPORT
567	034176	032701	020000	16\$:	BIT	#WLSTAT,R1		;CHECK WRITE LOCK STATUS
568	034202	001406			BEQ	17\$		;SKIP IF RESET
569	034204	012703	010610		MOV	#MWLSTA,R3		;ELSE SET NAME MESSAGE PTR
570	034210				ERRHRD	705,,,ERR2		
	034210	104456			TRAP	C\$ERHRD		
	034212	001301			.WORD	705		
	034214	000000			.WORD	0		
	034216	012532			.WORD	ERR2		
571	034220	042701	021177	17\$:	BIC	#21177,R1		;CLEAR STAU8 EXCEPT FOR ERROR BITS
572	034224	026727	146046	000001	CMP	T.DRIVE,#1		
573	034232	001404			BEQ	99\$		
574	034234	022701	000200		CMP	#200,R1		
575	034240	001411			BEQ	19\$		
576	034242	000402			BR	18\$		
577	034244	005701		99\$:	TST	R1		
578	034246	001406			BEQ	19\$		;NO-SKIP
579	034250			18\$:	ERRHRD	704,,,ERR6		;ELSE REPORT ALL ERRORS
	034250	104456			TRAP	C\$ERHRD		
	034252	001300			.WORD	704		
	034254	000000			.WORD	0		
	034256	012766			.WORD	ERR6		
580	034260				EXIT	TST		;EXIT
	034260	104432			TRAP	C\$EXIT		
	034262	000016			.WORD	L10035-		
581	034264	016701	146554	19\$:	MOV	T.CS,R1		;GET CS REG
582	034270	042701	141777		BIC	#141777,R1		;CLEAR ALL BUT ERROR BITS

```

583 034274 005701          TST      R1          ;TEST IF ANY ERROR SET
584 034276 001364          BNE      18$         ;YES-SKIP TO REPORT
585 034300                25$:
586 034300                100$:
587 034300                ENDTST
    034300                L10035:
    034300 104401          TRAP     C$ETST

588
589
590
591
592 034302                .SBTTL  *TEST 8      INITIAL RESET STATE
    034302                BGNSTST ;TEST 8
593 034302 012767 006515 146502  MOV     #INITST,ERHEAD
594 034310 004767 162514        JSR     PC,TSTINT      ;INITIALIZE TEST
595
596 034314 004767 162526        JSR     PC,GSTATR     ;GET STATUS WITH RESET
597 034320 034366        65$:
598 034322 005767 160000        TST     MISWIW        ;CHECK IF MAN INTERVENTION WAS RUN
599 034326 100017        BPL     4$            ;NO-SKIP
600 034330 005767 147024        TST     PASNUM        ;CHECK IF 1ST PASS
601 034334 001014        BNE     4$            ;NO-SKIP
602 034336 032767 000100 146506  BIT     #HSSTAT,T.MP  ;CHECK HD SELECT STILL 0
603 034344 001410        BEQ     4$            ;YES-SKIP
604 034346 012703 010537        MOV     #MHSTA,R3     ;SET NAME MESSAGE PTR
605 034352 012704 011412        MOV     #CCYLUP,R4    ;SET CONDITION POINTER
606 034356                ERRHRD  801,,ERR4    ;REPORT ERROR
    034356 104456        TRAP     C$ERHRD
    034360 001441        .WORD   801
    034362 000000        .WORD   0
    034364 012646        .WORD   ERR4

607 034366                4$:
608 034366                65$:
609 034366                ENDTST
    034366                L10036:
    034366 104401          TRAP     C$ETST
  
```

```

610
611
612
613
614 034370                .SBTTL  *TEST 9      DRIVE READY
    034370                BGNSTST ;TEST 9
615 034370 012767 006543 146414  MOV     #T09ERR,ERHEAD ;SET ERROR HEADER
616 034376 012701 003102        MOV     #NEWCYL,R1    ;GET POINTER TO DESIRED LOC
617 034402 005021          CLR     (R1)+         ;CLEAR NEW CYL
618 034404 005021          CLR     (R1)+         ;CLEAR CURRENT CYL
619 034406 005021          CLR     (R1)+         ;      DIFFERENCE
620 034410 005011          CLR     (R1)          ;      SIGN
621 034412 004767 162412        JSR     PC,TSTINT     ;INITIALIZE TEST
622 034416 004767 162424        JSR     PC,GSTATR     ;GET STATUS WITH RESET
623 034422 035022        100$:
624 034424 004767 166170        JSR     PC,POSHSB     ;POSITION HEAD SELECTED BIT
625 034430 010567 146456        MOV     R5,DESHD     ;STORE AS DESIRED HEAD
626 034434 004767 164512        JSR     PC,SIMSEK     ;EXECUTE SIMPLE SEEK
627 034440 035022        100$:
628 034442 012703 010404        MOV     #MDRDY,R3    ;SET NAME MESSAGE PTR
629 034446 012704 011353        MOV     #CDRDY,R4    ;SET CONDITION POINTER
  
```

630	034452	004767	162420		JSR	PC,GSTAT		;GET STATUS
631	034456	035022			100\$			
632	034460	032767	000001	146356	BIT	#DRDYMSK,T.CS		;TEST READY SET
633	034466	001406			BEQ	4\$		;NO-SKIP
634	034470				ERRHRD	901,,,ERR4		;REPORT READY ERROR
	034470	104456			TRAP	C\$ERHRD		
	034472	001605			.WORD	901		
	034474	000000			.WORD	0		
	034476	012646			.WORD	ERR4		
635	034500				EXIT	TST		;EXIT
	034500	104432			TRAP	C\$EXIT		
	034502	000320			.WORD	L10037-		
636	034504	012701	000121		MOV	#81,,R1		;SET WAIT COUNT
637	034510	004767	162362		JSR	PC,GSTAT		;GET STATUS
638	034514	035022			100\$			
639	034516				BREAK			;ALLOW FOR A ^C
	034516	104422			TRAP	C\$BRK		
640								
641	034520	012703	000005		MOV	#5,R3		;SET EXPECTED STATE VALUE
642	034524	026703	146330		CMP	T,STAT,R3		;CHECK STATE IS 5
643	034530	001406			BEQ	7\$		;YES-SKIP
644	034532				ERRHRD	902,,,ERR7		;ELSE REPORT
	034532	104456			TRAP	C\$ERHRD		
	034534	001606			.WORD	902		
	034536	000000			.WORD	0		
	034540	013666			.WORD	ERR7		
645	034542				EXIT	TST		
	034542	104432			TRAP	C\$EXIT		
	034544	000256			.WORD	L10037-		
646	034546	012703	010404		MOV	#MDRDY,R3		
647	034552	032767	000001	146264	BIT	#DRDYMSK,T.CS		;CHECK READY SET
648	034560	001063			BNE	12\$		;YES-SKIP
649	034562	005301			DEC	R1		;ELSE DEC WAIT COUNT
650	034564	001403			BEQ	9\$		;SKIP IF 0
651	034566				TIMDLY	1		
652	034712	000676			BR	5\$		
653	034714				ERRHRD	903,,,ERR5		;REPORT READY ERROR
	034714	104456			TRAP	C\$ERHRD		
	034716	001607			.WORD	903		
	034720	000000			.WORD	0		
	034722	012716			.WORD	ERR5		
654	034724				EXIT	TST		
	034724	104432			TRAP	C\$EXIT		
	034726	000074			.WORD	L10037-		
655								
656	034730	005767	146110		TST	T.CS		;TEST IF ANY ERROR
657	034734	100006			BPL	15\$		;NO-SKIP
658	034736				ERRHRD	904,,,ERR6		
	034736	104456			TRAP	C\$ERHRD		
	034740	001610			.WORD	904		
	034742	000000			.WORD	0		
	034744	012766			.WORD	ERR6		
659	034746				EXIT	TST		
	034746	104432			TRAP	C\$EXIT		
	034750	000052			.WORD	L10037-		
660	034752	012703	010537		MOV	#MHSTA,R3		;SET NAME MESSAGE PTR
661	034756	004767	165636		JSR	PC,POSHSB		;POSITION HEAD SELECT BIT FOR TEST

```

662 034762 020567 146124      CMP      R5,DESHD      ;CHECK IF CORRECT HEAD SELECTED
663 034766 001415      BEQ      20$           ;YES-SKIP
664 034770 005767 146116      TST      DESHD        ;ELSE TEST IF 1 DESIRED
665 034774 001406      BEQ      17$           ;NO-REPORT SB 0
666 034776      ERRHRD  905,,,ERR3    ;ELSE REPORT SB 1
      034776 104456      TRAP     C$ERRHRD
      035000 001611      .WORD   905
      035002 000000      .WORD   0
      035004 012600      .WORD   ERR3
667 035006      EXIT     TST
      035006 104432      TRAP     C$EXIT
      035010 000012      .WORD   L10037-
668 035012      17$:  ERRHRD  906,,,ERR2
      035012 104456      TRAP     C$ERRHRD
      035014 001612      .WORD   906
      035016 000000      .WORD   0
      035020 012532      .WORD   ERR2
669 035022      20$:
670 035022      100$:
671 035022      ENDTST
      035022      L10037:
      035022 104401      TRAP     C$E1ST
672
673
674
675
676 035024      .SBTTL  *TEST 10      SEEK SIGN SWITCH
      035024      BGNSTST             ;TEST 10
677 035024 012767 006553 145760      MOV      #T10ERR,ERHEAD ;SET ERROR HEADER
678 035032 012701 003102      MOV      #NEWCYL,R1
679 035036 005021      CLR      (R1)+         ;CLEAR NEW CYL
680 035040 005021      CLR      (R1)+         ;CLEAR CURRENT CYLINDER
681 035042 005021      CLR      (R1)+         ;CLEAR DIFFERENCE
682 035044 052721 000001      BIS      #BIT0,(R1)+   ;SET FOR SIGN OF 1
683 035050 004767 165544      JSR      PC,POSHSB     ;GET SELECTED HEAD
684 035054 010521      MOV      R5,(R1)+     ;SET AS DESIRED HEAD
685 035056      T104$:
686 035056      BGNSSUB
      035056      T10.1:
      035056 104402      TRAP     C$BSUB
687 035060 004767 161744      JSR      PC,TSTINT     ;INITIALIZE TEST
688 035064 004767 161756      JSR      PC,GSTATR     ;GET STATUS
689 035070 035460      60$
690 035072 004767 164054      JSR      PC,SIMSEK     ;DO SEEK
691 035076 035460      60$
692 035100 012703 010404      MOV      #MDRDY,R3     ;SET NAME MESSAGE PTR
693 035104 012704 011353      MOV      #CDRDY,R4     ;SET CONDITION MESSAGE PTR
694 035110 004767 161762      JSR      PC,GSTAT      ;GET STATUS
695 035114 035460      60$
696 035116 032767 000001 145720      BIT      #DRDYMSK,T.CS ;CHECK READY RESET
697 035124 001406      BEQ      4$           ;YES-SKIP
698 035126      ERRHRD  1001,,,ERR4  ;REPORT READY ERROR
      035126 104456      TRAP     C$ERRHRD
      035130 001751      .WORD   1001
      035132 000000      .WORD   0
      035134 012646      .WORD   ERR4
699 035136      EXIT     SUB          ;EXIT SUBTEST
  
```

035136	104432				TRAP	C\$EXIT	
035140	000320				.WORD	L10041-.	
700							
701							
702	035142	012701	000121	4\$:	MOV	#81.,R1	;SET WAIT COUNT
703	035146	004767	161724	5\$:	JSR	PC,GSTAT	;GET STATUS
704	035152	035460			60\$		
705	035154				BREAK		;ALLOW FOR A ^C
	035154	104422			TRAP	C\$BRK	
706							
707	035156	012703	000005		MOV	#5,R3	;SET EXPECTED STATE
708	035162	020367	145672		CMP	R3,T.STAT	;CHECK STATE IS 5
709	035166	001406			BEQ	7\$	;YES-SKIP
710	035170				ERRHRD	1002...ERR7	;REPORT STATE ERROR
	035170	104456			TRAP	C\$ERHRD	
	035172	001752			.WORD	1002	
	035174	000000			.WORD	0	
	035176	013666			.WORD	ERR7	
711	035200				EXIT	SUB	;EXIT
	035200	104432			TRAP	C\$EXIT	
	035202	000256			.WORD	L10041-.	
712	035204	012703	010404	7\$:	MOV	#MDRDY,R3	;SET NAME MESSAGE PTR
713	035210	032767	000001	145626	BIT	#DRDYMSK,T.CS	;CHECK READY SET
714	035216	001063			BNE	12\$	;YES-SKIP
715	035220	005301			DEC	R1	;DO WAIT COUNT
716	035222	001453			BEQ	9\$	;SKIP IF 0
717	035224				TIMDLY	1	
718	035350	000676			BR	5\$	
719							
720	035352			9\$:	ERRHRD	1003...ERR5	;REPORT READY ERROR
	035352	104456			TRAP	C\$ERHRD	
	035354	001753			.WORD	1003	
	035356	000000			.WORD	0	
	035360	012716			.WORD	ERR5	
721	035362				EXIT	SUB	;EXIT
	035362	104432			TRAP	C\$EXIT	
	035364	000074			.WORD	L10041-.	
722	035366	005767	145452	12\$:	TST	T.CS	;TEST IF ANY OTHER ERROR
723	035372	100006			BPL	15\$	;NO-SKIP
724	035374				ERRHRD	1004...ERR6	;REPORT ALL ERRORS
	035374	104456			TRAP	C\$ERHRD	
	035376	001754			.WORD	1004	
	035400	000000			.WORD	0	
	035402	012766			.WORD	ERR6	
725	035404				EXIT	SUB	;EXIT
	035404	104432			TRAP	C\$EXIT	
	035406	000052			.WORD	L10041-.	
726							
727	035410	012703	010537	15\$:	MOV	#MHSTA,R3	;SET NAME MESSAGE PTR
728	035414	004767	165200		JSR	PC,POSHSB	;GET SELECTED HEAD BIT
729	035420	020567	145466		CMP	R5,DESHD	;CHECK IF CORRECT
730	035424	001415			BEQ	20\$	;YES - SKIP
731	035426	005767	145460		TST	DESHD	;WAS IT SET
732	035432	001406			BEQ	17\$	;NO-SKIP
733	035434				ERRHRD	1005...ERR3	;REPORT SB 1
	035434	104456			TRAP	C\$ERHRD	
	035436	001755			.WORD	1005	

```

035440 000000      .WORD 0
035442 012600      .WORD ERR3
734 035444      EXIT SUB
035444 104432      TRAP C$EXIT
035446 000012      .WORD L10041-
735 035450      17$: ERRHRD 1006.,ERR2 ;REPORT SB 0
035450 104456      TRAP C$ERRRD
035452 001756      .WORD 1006
035454 000000      .WORD 0
035456 012532      .WORD ERR2

736
737 035460      20$:
738 035460      60$:
739 035460      ENDSUB
035460      L10041:
035460 104403      TRAP C$ESUB
740 035462 005767 145422      TST DESSGN ;CHECK IF BOTH SIGN USED
741 035466 001404      BEQ 25$ ;YES-SKIP
742 035470 005067 145414      CLR DESSGN ;SET FOR SIGN OF 0
743 035474 000167 177356      JMP T104$ ;DO TEST AGAIN
744 035500
745 035500      25$:
035500      ENDTST
035500      L10040:
035500 104401      TRAP C$ETST

746
747
748
749
750 035502      .SBTTL *TEST 11 HEAD ALIGNMENT SUPPORT
BGNTST ;TEST 11
035502      T11::
751 035502 032767 000010 156616      BIT #HDALIGN,MISWIW ;CHECK IF RUN HEAD ALIGNMENT
752 035510 001411      BEQ 1$ ;NO-EXIT
753 035512 005767 145642      TST PASNUM ;TEST IF PASS 0
754 035516 001006      BNE 1$ ;NO-EXIT
755 035520 026767 145306 145262      CMP RLDRV,HADONE ;TEST IF HEAD ALIGN DONE THIS DRIVE
756 035526 001004      BNE 2$ ;NO - SKIP
757 035530 000167 000422      JMP T115$ ;GO CHECK WRITE LOCK
758 035534      1$:
035534 104432      EXIT TST
035536 000520      TRAP C$EXIT
759 035540 015767 145266 145242      2$: MOV RLDRV,HADONE ;SET HEAD ALIGN DONE FLAG
760 ;TELL DRV AND CNTRL HD ALIGN TO BE DONE ON
761 035546      PRINTF #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
035546 005046      CLR -(SP)
035550 156716 145257      BISB RLDRV+1,(SP)
035554 012746 006053      MOV #DRVNAM,-(SP)
035560 016746 145242      MOV RLBAS,-(SP)
035564 012746 006042      MOV #BASADD,-(SP)
035570 012746 011567      MOV #FMT5,-(SP)
035574 012746 000005      MOV #5,-(SP)
035600 010600      MOV SP,RO
035602 104417      TRAP C$PNTF
035604 062706 000014      ADD #14,SP

762 ;HD ALIGN. RSETWRT LCK TO SEL HD 0, SET HD 1
763 035610      PRINTF #FMT9,#HAME$1
035610 012746 007154      MOV #HAME$1,-(SP)
035614 012746 011753      MOV #FMT9,-(SP)
    
```

```

035620 012746 00C002      MOV      #2,-(SP)
035624 010600      MOV      SP,R0
035626 104417      TRAP    C$PNTF
035630 062706 000006      ADD     #6,SP
764                                     ;^C TO RET TO SUPVR CMD MODE
765 035634      PRINTF  #FMT9,#HAMES2
035634 012746 007237      MOV      #HAMES2,-(SP)
035640 012746 011753      MOV      #FMT9,-(SP)
035644 012746 000002      MOV      #2,-(SP)
035650 010600      MOV      SP,R0
035652 104417      TRAP    C$PNTF
035654 062706 000006      ADD     #6,SP
766                                     ;IF HD SEL TP (21, 22) DO NOT EXIST
767 035660      PRINTF  #FMT9,#HAMES3
035660 012746 007343      MOV      #HAMES3,-(SP)
035664 012746 011753      MOV      #FMT9,-(SP)
035670 012746 000002      MOV      #2,-(SP)
035674 010600      MOV      SP,R0
035676 104417      TRAP    C$PNTF
035700 062706 000006      ADD     #6,SP
768                                     ;JUMPER DRV RDY AND SEEK INCOMPLETE ON DRV
769                                     ;LOGIC MOD
770 035704      PRINTF  #FMT9,#HAMES4
035704 012746 007406      MOV      #HAMES4,-(SP)
035710 012746 011753      MOV      #FMT9,-(SP)
035714 012746 000002      MOV      #2,-(SP)
035720 010600      MOV      SP,R0
035722 104417      TRAP    C$PNTF
035724 062706 000006      ADD     #6,SP
771                                     ;SET WRITE LOCK
772 035730      PRINTF  #FMTOP1,#OPR12A,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
035730 005046      CLR     -(SP)
035732 156716 145075      BISB    RLDRV+1,(SP)
035736 012746 006053      MOV      #DRVNAM,-(SP)
035742 016746 145060      MOV      RLBAS,-(SP)
035746 012746 006042      MOV      #BASADD,-(SP)
035752 012746 010135      MOV      #OPR1A,-(SP)
035756 012746 010121      MOV      #OPR12A,-(SP)
035762 012746 011442      MOV      #FMTOP1,-(SP)
035766 012746 000007      MOV      #7,-(SP)
035772 010600      MOV      SP,R0
035774 104417      TRAP    C$PNTF
035776 062706 000020      ADD     #20,SP
773
774 036002      BGNSUB
036002
036002 104402      TRAP    C$BSUR
775 036004 004767 161020      JSR     PC,TS,INT ;INITIALIZE TEST
776 036010 005067 144772      CLR     DONE ;CLEAR DONE
777
778 036014 016767 145012 145012      MOV     RLDRV,L,CS ;SET UP FOR GET STATUS
779 036022 052767 000104 145004      BIS     #GTSTAT,L,CS
780 036030 012767 000013 145002      MOV     #GETSTAT!DRSET,L,DA
781
782 036036 016762 144776 000004      MOV     L,DA,RLDA(R2) ;DO GET STATUS
783 036044 016762 144764 000000      MOV     L,CS,RLCSR(R2)
784

```

T11.1:

```

785
786 036052 005767 144730      13$:  TST      DONE      ;CHECK IF DONE
787 036056 001775              BEQ      13$      ;NO-GO CLR CONTROLLER
788
789 036060 005067 144722      CLR      DONE
790 036064 012767 000021 144746 20$:  MOV      #HDSEL!MBSET0,L.DA;LOAD FOR HEAD 1
791 036072 000240              NOP
792 036074 032767 020000 144750      BIT      #WLSTAT,T.MP      ;CHECK IF WRITE LOCK SET
793 036102 001003              BNE      22$      ;YES-SKIP
794 036104 042767 000020 144726      BIC      #HDSEL,L.DA      ;ELSE CLEAR TO HEAD 0
795 036112 016767 144714 144714 22$:  MOV      RLDRV,L.CS      ;LOAD IN DRIVE NUMBER
796 036120 052767 000106 144706      BIS      #SEEK,L.CS      ;SET FOR SEEK
797 036126 016762 144706 000004      MOV      L.DA,RLDA(R2)    ;LOAD & EXECUTE SEEK
798 036134 016762 144674 000000      MOV      L.CS,RLCSR(R2)
799 036142              BREAK
       036142 104422          TRAP      C$BRK      ;ALLOW OPERATOR TO INTERRUPT PROGRAM TO GET
800
801 036144 005767 144636      30$:  TST      DONE
802 036150 001775              BEQ      30$
803 036152 000716              BR       11$      ;LOOP
804 036154
805 036154          59$:  ENDSUB
       036154          L10043:
       036154 104403          TRAP      C$ESUB
806 036156          T115$:
807 036156          BGNSUB
       036156          T11.2:
       036156 104402          TRAP      C$BSUB
808 036160 004767 160644      JSR      PC,TSTINT      ;INITIALIZE TEST
809 036164 004767 160656      JSR      PC,GSTATR      ;CLEAR DRIVE
810 036170 036254          60$
811 036172 032767 020000 144652      BIT      #WLSTAT,T.MP      ;CHECK WRITE LOCK RESET
812 036200 001425              BEQ      19$      ;YES-SKIP
813 036202          18$:  PRINTF  #FMT9,#OPR12      ;REQUEST WRITE LOCK RESET
       036202 012746 010102      MOV      #OPR12,-(SP)
       036206 012746 011753      MOV      #FMT9,-(SP)
       036212 012746 000002      MOV      #2,-(SP)
       036216 010600          MOV      SP,RO
       036220 104417          TRAP      C$PNTF
       036222 062706 000006      ADD      #6,SP
814 036226 005067 146132      CLR      OBUF      ;CLEAR FOR RESPONSE
815 036232          GMANIL  OPRO02,OBUF,1,NO      ;GET RESPONSE
       036232 104443          TRAP      C$GMAN
       036234 000404          BR       10000$
       036236 004364          .WORD  OBUF
       036240 000120          .WORD  T$CODE
       036242 007470          .WORD  OPRO02
       036244 000001          .WORD  1
       036246          10000$:
816 036246 005767 146112      TST      OBUF      ;WAS ANSWER YES
817 036252 001753              BEQ      18$      ;NO-REPEAT REQUEST
818 036254          19$:
819 036254          50$:
820 036254          ENDSUB
       036254          L10044:
       036254 104403          TRAP      C$ESUB
821 036256          20$:
    
```



```

822 036256          ENDTST
      036256          L10042:
      036256 104401 TRAP    C$ETST

823
824
825
826
827 036260          .SBTTL *TEST 12          HEAD SWITCHING
      036260          BGNTST          ;TEST 12

828 036260 012767 006573 144524 MOV    #T12ERR,ERHEAD ;SET ERROR HEADER          T12::
829 036266 012701 003102          MOV    #NEWCYL,R1      ;GET POINTER TO DESIRED LOCATION
830 036272 005021          CLR    (R1)+          ;CLEAR NEW CYLINDER
831 036274 005021          CLR    (R1)+          ;CLEAR CURRENT CYL.
832 036276 005021          CLR    (R1)+          ;CLEAR DIFFERENCE
833 036300 005021          CLR    (R1)+          ;CLEAR SIGN
834 036302 012721 000001          MOV    #1,(R1)+      ;SET FOR HEAD 1
835 036306
836 036306          T124$:
      036306          BGNSUB

837 036310 004767 160514          TRAP   C$BSUB          T12.1:
838 036314 004767 160526          JSR   PC,TSTINT      ;INITIALIZE TEST
839 036320 036710          JSR   PC,GSTATR     ;GET STATUS WITH RESET
840 036322 004767 162624          JSR   PC,SIMSEK     ;DO SEEK
841 036326 036710          60$
842 036330 012703 010404          MOV    #MDRDY,R3    ;SET NAME MESSAGE PTR
843 036334 012704 011353          MOV    #CDRDY,R4    ;SET CONDITION POINTER
844 036340 004767 160532          JSR   PC,GSTAT      ;GET STATUS
845 036344 036710          60$
846 036346 032767 000001 144470 BIT    #DRDYMSK,T.CS ;CHECK IF READY
847 036354 001406          BEQ   5$            ;NO-SKIP
848 036356          ERRHRD 1201,,,ERR4 ;REPORT READY ERROR
      036356 104456          TRAP   C$ERHRD
      036360 002261          .WORD 1201
      036362 000000          .WORD 0
      036364 012646          .WORD ERR4
849 036366          EXIT   SUB          ;EXIT
      036366 104432          TRAP   C$EXIT
      036370 000320          .WORD L10046-.

850
851 036372 012701 000121          5$:  MOV    #81.,R1      ;SET WAIT COUNT
852 036376 004767 160474          6$:  JSR   PC,GSTAT      ;GET STATUS
853 036402 036710          60$
854 036404          BREAK
      036404 104422          TRAP   C$BRK          ;ALLOW FOR A ^C

855
856 036406 012703 000005          MOV    #5,R3        ;SET EXPECTED STATE VALUE
857 036412 020367 144442          CMP    R3,T.STAT    ;CHECK IF STATE IS 5
858 036416 001406          BEQ   7$            ;YES-SKIP
859 036420          ERRHRD 1202,,,ERR7 ;REPORT STATE ERROR
      036420 104456          TRAP   C$ERHRD
      036422 002262          .WORD 1202
      036424 000000          .WORD 0
      036426 013666          .WORD ERR7
860 036430          EXIT   SUB
      036430 104432          TRAP   C$EXIT
      036432 000256          .WORD L10046-.
    
```

```

861
862 036434 012703 010404      7$:  MOV    #MDRDY,R3      ;SET NAME MESSAGE PTR
863 036440 032767 000001 144376 BIT    #DRDYMSK,T.CS ;CHECK DRIVE READY
864 036446 001063          BNE    12$           ;YES-SKIP
865 036450 005301          DEC    R1            ;DEC WAIT COUNT
866 036452 001453          BEQ    9$           ;SKIP IF 0
867 036454          TIMDLY 1
868 036600 000676          BR     6$
869
870 036602          9$:  ERRHRD 1203...ERR5 ;REPORT READY ERROR
      036602 104456      TRAP  C$ERHRD
      036604 002263      .WORD 1203
      036606 000000      .WORD 0
      036610 012716      .WORD ERR5
871 036612          EXIT  SUB           ;EXIT
      036612 104432      TRAP  C$EXIT
      036614 000074      .WORD L10046-.
872
873 036616 005767 144222      12$: TST    T.CS        ;TEST IF ANY ERROR
874 036622 100006          BPL    15$          ;NO-SKIP
875 036624          ERRHRD 1204...ERR6 ;REPORT ALL ERRORS
      036624 104456      TRAP  C$ERHRD
      036626 002264      .WORD 1204
      036630 000000      .WORD 0
      036632 012766      .WORD ERR6
876 036634          EXIT  SUB
      036634 104432      TRAP  C$EXIT
      036636 000052      .WORD L10046-.
877 036640 012703 010537      15$: MOV    #MHSTA,R3    ;SET NAME MESSAGE PTR
878 036644 004767 163750      JSR    PC,POSHSB   ;POSITION HEAD SELECT BIT
879 036650 026705 144236      CMP    DESHD,R5   ;CHECK IF CORRECT HEAD SELECTED
880 036654 001415          DEQ    20$         ;YES-SKIP
881 036656 005767 144230      TST    DESHD      ;WAS HEAD 0 SELECTED
882 036662 001406          BEQ    17$        ;YES-SKIP
883 036664          ERRHRD 1205...ERR3 ;REPORT HEAD SB 1
      036664 104456      TRAP  C$ERHRD
      036666 002265      .WORD 1205
      036670 000000      .WORD 0
      036672 012600      .WORD ERR3
884 036674          EXIT  SUB           ;EXIT
      036674 104432      TRAP  C$EXIT
      036676 000012      .WORD L10046-.
885 036700          17$: ERRHRD 1206...ERR2 ;ELSE REPORT HEAD SB 0
      036700 104456      TRAP  C$ERHRD
      036702 002266      .WORD 1206
      036704 000000      .WORD 0
      036706 012532      .WORD ERR2
886
887 036710          20$:
888 036710          60$:
889 036710          ENDSUB
      036710          L10046:
      036710 104403      TRAP  C$ESUB
890 036712 005767 144174      TST    DESHD      ;CHECK IF HD 0 WAS DONE
891 036716 001404          BEQ    25$        ;YES-SKIP
892 036720 005067 144166      CLR    DESHD      ;ELSE SET TO HEAD 0
893 036724 000167 177356      JMP    T124$     ;REDO TEST
    
```

```

894 036730          25$:
895 036730          ENDTST
      036730          L10045:
      036730 104401 TRAP   C$ETST

896
897
898
899
900 036732          .SBTTL +TEST 13      READ HEADER (PART 1)
      036732          BGNTST          ;TEST 13
      036732          144052          T13::
901 036732 012767 006605 144052 MOV   #T13ERR,ERHEAD ;SET ERROR HEADER
902 036740 012701 003102 MOV   #NEWCYL,R1    ;GET ADDRESS OF DESIRED LOCATIONS
903 036744 005021 CLR   (R1)+        ;CLEAR NEW CYL
904 036746 005021 CLR   (R1)+        ;CLEAR CURRENT CYL
905 036750 005021 CLR   (R1)+        ;CLEAR DIFF
906 036752 005021 CLR   (R1)+        ;CLEAR SIGN
907 036754 005021 CLR   (R1)+        ;CLEAR HEAD
908 036756          T134$:
909 036756          BGNSUB
      036756          104402          TRAP   C$BSUB          T13.1:
910 036760 004767 160044 JSR   PC,TSTINT    ;INITIALIZE TEST
911 036764 004767 160056 JSR   PC,GSTATR    ;GET STATUS W/RESET
912 036770 037062 60$
913 036772 004767 162154 JSR   PC,SIMSEK    ;DO SEEK
914 036776 037062 60$
915 037000 012701 000121 MOV   #81.,R1      ;SET WAIT COUNT
916 037004 004767 163640 JSR   PC,RDYWAIT   ;WAIT FOR READY
917 037010 037062 60$
918
919 037012 004767 163112 JSR   PC,XRDHDC    ;DO READ HEADER
920 037016 037062 60$
921 037020 012703 010537 MOV   #MHSTA,R3    ;SET NAME MESSAGE PTR
922 037024 004767 163562 JSR   PC,POSHW1    ;POSITION HS BIT IN HD WRD 1
923 037030 020567 144056 CMP   R5,DESHD     ;CHECK IF HEAD CORRECT
924 037034 001412 BEQ   15$          ;YES-SKIP
925 037036          ERRHRD 1301.,,ERR3 ;REPORT SB 1
      037036 104456 TRAP   C$ERHRD
      037040 002425 .WORD 1301
      037042 000000 .WORD 0
      037044 012600 .WORD ERR3
926 037046          EXIT SUB
      037046 104432 TRAP   C$EXIT
      037050 000012 .WORD L10050-.
927 037052          17$: ERRHRD 1302.,,ERR2 ;REPORT SB 0
      037052 104456 TRAP   C$ERHRD
      037054 002426 .WORD 1302
      037056 000000 .WORD 0
      037060 012532 .WORD ERR2

928
929 037062          15$:
930 037062          60$:
931 037062          ENDSUB
      037062          L10050:
      037062 104403 TRAP   C$ESUB
932 037064 005767 144022 TST   DESHD        ;TEST IF HEAD 1 DONE
933 037070 001007 BNE   20$          ;YES-SKIP
    
```

934	037072	012767	00C001	144012	MOV	#1,DESHD	:ELSE SET TO HEAD 1	
935	037100	016767	143746	144010	MOV	HDWRD1,TEMPO	:STORE HDR WORD 1	
936	037106	000723			BR	T134\$	:DO TEST AGAIN	
937	037110	042767	000177	144000	20\$:	BIC	#177,TEMPO	:CLEAR ALL BUT CYLINDER IN 1ST HEADER
938	037116	042767	000177	143726	BIC	#177,HDWRD1	:CLEAR ALL BY CYL IN 2ND HEADER	
939	037124	026767	143766	143720	COMP	TEMPO,HDWRD1	:COMPARE IF EQUAL	
940	037132	001406			BEQ	22\$	:YES-SKIP	
941	037134	012703	007070		MOV	#CYLPER,R3	:SET NAME MESSAGE PTR	
942	037140				ERRHRD	1306,ERR1	:REPORT HEAD ALIGNMENT PROBLEM	
	037140	104456			TRAP	C\$ERHRD		
	037142	002432			.WORD	1306		
	037144	000000			.WORD	0		
	037146	012464			.WORD	ERR1		
943	037150				22\$:			
944	037150				ENDTST			
	037150				L10047:			
	037150	104401			TRAP	C\$ETST		

945  
946  
947

948 .SBTTL \*TEST 14 READ HEADER (PART 2)  
 949 BGNST ;TEST 14

950	037152	012767	006621	143632	MOV	#T14ERR,ERHEAD	T14::	:SET ERROR HEADER
951	037160	012701	003104		MOV	#CURCYL,R1		:GET ADDRESS OF DESIRED VALUE
952	037164	005021			CLR	(R1)+		:CLEAR CURRENT CYL
953	037166	005021			CLR	(R1)+		:CLEAR DESIRED DIFF
954	037170	005021			CLR	(R1)+		:CLEAR SIGN
955	037172	005021			CLR	(R1)+		:CLEAR DESIRED HEAD

956 037174 T153\$:  
 957 037174 BGNSUB

037174 T14.1:

958	037176	004767	157626		TRAP	C\$BSUB	
959	037202	004767	157640		JSR	PC,TSTINT	:INITIALIZE TEST
960	037206	037406			JSR	PC,GSTATR	:CLEAR DRIVE
961	037210	004767	161736		60\$		
962	037214	037406			JSR	PC,SIMSEK	:DO SEEK
963	037216	012701	000310		60\$		
964	037222	004767	163422		MOV	#200,R1	:SET WAIT COUNT FOR 20 MS
965	037226	037406			JSR	PC,RDYWAIT	:WAIT FOR READY
966	037230	004767	164114		60\$		
967	037234	037406			JSR	PC,RDALHD	:DO READ HEADER ALL HEADERS
968	037236	005067	143552		60\$		
969	037242	052767	000002	143534	CLR	MORECE	:CLEAR MORE COMPARE ERRORS FOR REPORT
970	037250	005003			BIS	#HDCMP,OPFLAG	:SET HDR COMPARE FLAG
971	037252	012704	003764		CLR	R3	:CLEAR FOR HDR COUNT
972	037256	012705	003116		MOV	#IBUFF,R4	:GET POINTER FOR HDR TO BE CHECKED
973	037262	012701	000050		MOV	#TEMPO,R5	:GET POINTER TO TEST AREA
974	037266	011415			MOV	#40,R1	:SET HDR COUNT
975					MOV	(R4),(R5)	:GET FIRST HEADER WORD

976	037270	042715	000100		BIC	#HDHSEL,(R5)	
977	037274	005767	143612		TST	DESHD	:TEST IF HD 0 DESIRED
978	037300	001404			BEQ	10\$	:YES-SKIP
979	037302	052715	000100		BIS	#HDHSEL,(R5)	:ELSE SET HEAD BIT
980	037306	005065	000002		CLR	2(R5)	:CLEAR 2ND WORD OF TEST AREA
981							

```

982 037312 021524      10$:  CMP      (R5),(R4)+      ;COMPARE HEADER WORD
983 037314 001406      BEQ      13$              ;SKIP IF OK
984 037316 005744      TST      -(R4)           ;ELSE POSITION R4 TO BAD WORD
985 037320      ERRHRD 1501...ERR10 ;REPORT ERROR
    037320 104456      TRAP     C$ERHRD
    037322 002735      .WORD   1501
    037324 000000      .WORD   0
    037326 014076      .WORD   ERR10
986 037330 005724      TST      (R4)+           ;BUMP R4 TO NEXT WORD
987 037332 005203      13$:  INC      R3           ;BUMP WORD COUNT
988 037334 005724      TST      (R4)+           ;TEST 2ND WORD IS 0
989 037336 001406      BEQ      15$              ;YES - SKIP
990 037340 022544      CMP      (R5)+,-(R4)     ;POSITION PTRS FOR REPORT
991 037342      ERRHRD 1501...ERR10 ;REPORT ERROR
    037342 104456      TRAP     C$ERHRD
    037344 002735      .WORD   1501
    037346 000000      .WORD   0
    037350 014076      .WORD   ERR10
992 037352 024524      CMP      -(R5),(R4)+     ;REPOSITION POINTER
993 037354 005724      15$:  TST      (R4)+           ;POSITION R4 PAST ECC WORD
994 037356 005203      INC      R3           ;BUMP WORD COUNT
995 037360 005215      INC      (R5)           ;BUMP SECTOR COUNT
996 037362 011500      MOV      (R5),R0        ;CHECK IF SECTOR IS PAST LAST SECTOR
997 037364 042700 177700 BIC      #^CHDSEC,R0
998 037370 022700 000050 CMP      #40.,R0
999 037374 001002      BNE     17$              ;NO-SKIP
1000 037376 042715 000077 BIC      #HDSEC,(R5)     ;ELSE CLEAR SECTOR TO 0
1001 037402 005301      17$:  DEC      R1           ;DEC HDR COUNT
1002 037404 001342      BNE     10$              ;YES-SKIP
1003
1004 037406
1005 037406      60$:  ENDSUB
    037406 L10052:
    037406 104403      TRAP     C$ESUB
1006 037410 005767 143476 TST      DESHD           ;CHECK IF HD 1 TESTED
1007 037414 001005      BNE     20$              ;YES-SKIP
1008 037416 012767 000001 143466 MOV      #1,DESHD
1009 037424 000167 177544 JMP      T153$           ;REDO TEST
1010 037430
1011 037430      20$:  ENDTST
    037430 L10051:
    037430 104401      TRAP     C$ETST
1012
1013
1014
1015
1016 037432      .SBTTL *TEST 15      DIFFERENCE OF 1 SEEK (PART 1)
    037432 BGNTST ;TEST 15
1017
1018 037432 012767 006645 143352 MOV      #P2T01E,ERHEAD ;SET ERROR HEADER
1019 037440 012767 000004 143450 MOV      #4,TEMP0       ;SET PASS COUNT
1020 037446 004767 157356 JSR      PC,TSTINT      ;INITIALIZE TEST
1021 037452 004767 157370 JSR      PC,GSTATR      ;GET STATUS
1022 037456 040076      T1765$
1023 037460 022767 000001 142610 CMP      #1,T.DRIVE     ;RLO1 OR RLO2?
1024 037466 001404      BEQ     2$              ;BRANCH TO SET UP DIFF ARGUMENT FOR RLO1
1025 037470 012767 177776 143424 MOV      #-2,TEMP2      ;ELSE, SET -2 INTO DIFF ARGUMENT FOR RLO2
  
```

T15::

```

1026
1027 037476 000403 BR 5$ ;/(RLO2 HAS DOUBLE THE TRACK DENSITY OF RLO1)
1028 037500 012767 177777 143414 2$: MOV #-1,TEMP2 ;SET -1 INTO DIFF ARGUMENT FOR -1 SEEK
1029 037506 012704 003104 5$: MOV #CURCYL,R4 ;SET POINTERS
1030 037512 012705 003102 MOV #NEWCYL,R5
1031 037516 G04767 162276 JSR PC,CHOSHD ;GO CHOOSE HEAD
1032 037522 T172:
1033 037522 BGN SUB T15.1:
037522 TRAP C$BSUB
037522 104402 JSR PC,GETPOS ;GET POSITION
1034 037524 004767 163472 60$
1035 037530 040032 BREAK
1036 037532 104422 TRAP C$BRK ;ALLOW FOR A ^C
1037
1038 037534 INLOOP ;CHECK IF IN ERROR LOOP
037534 104420 TRAP C$INLP
1039 037536 BNCOMPLETE 3$ ;NO - SKIP
037536 103005 BCC 3$
1040 037540 021415 CMP (R4),(R5) ;CHECK IF CURRENT = NEW
1041 037542 001005 BNE 4$ ;NO - SKIP
1042 037544 004767 162334 JSR PC,ONSWAP ;ELSE SWAP OLD AND NEW
1043 037550 000441 BR 9$ ;SKIP TO SEEK
1044 037552 005467 143344 3$: NEG TEMP2 ;CHANGE DIFF ARGUMENT FOR OPPOSITE DIR
1045 037556 011415 4$: MOV (R4),(R5) ;MOVE CURRENT INTO OLD
1046 037560 026714 142516 CMP HLMTW,(R4) ;CHECK IF CURRENT AT 255
1047 037564 001014 BNE 7$ ;NO - SKIP
1048 037566 022767 000001 142502 CMP #1,T.DRIVE ;RLO1 OR RLO2?
1049 037574 001404 BEQ 6$ ;BRANCH IF RLO1
1050 037576 012767 177776 143316 MOV #-2,TEMP2 ;ELSE, SET UP DIFF ARGUMENT FOR RLO2
1051 037604 000421 BR 8$
1052 037606 012767 177777 143306 6$: MOV #-1,TEMP2 ;AT MAX CYL, MAKE NEXT SEEK REV
1053 037614 000415 BR 8$ ;SKIP
1054 037616 005714 7$: TST (R4) ;TEST IF CURRENT AT 0
1055 037620 001013 BNE 8$ ;NO - SKIP
1056 037622 022767 000001 142446 CMP #1,T.DRIVE ;RLO1 OR RLO2?
1057 037630 001404 BEQ 11$ ;BRANCH IF RLO1
1058 037632 012767 000002 143262 MOV #2,TEMP2 ;ELSE, SET UP DIFF ARGUMENT FOR RLO2
1059 037640 000403 BR 8$
1060 037642 012767 000001 143252 11$: MOV #1,TEMP2 ;AT CYL 0, MAKE NEXT SEEK FWRD
1061 037650 066715 143246 8$: ADD TEMP2,(R5) ;ADD DIFF TO NEW CYL (+1 OR -1 FOR RLO1,
;+2 OR -2 FOR RLO2)
1062
1063 037654 9$: BREAK ;ALLOW A ^C
037654 104422 TRAP C$BRK
1064 037656 004767 160500 JSR PC,XSEEK ;DO SEEK
1065 037662 040032 60$
1066 037664 004767 160312 JSR PC,GDRSTA ;GET DRIVE STATE
1067
1068 037670 012703 000004 MOV #4,R3 ;SET EXPECTED STATE
1069 037674 020367 143160 CMP R3,T.STAT ;CHECK DRIVE STATE
1070 037700 001405 BEQ 10$ ;YES-SKIP
1071 037702 ERRHRD 101,ERR7 ;REPORT STATE ERROR
037702 104456 TRAP C$ERHRD
037704 000145 .WORD 101
037706 000000 .WORD 0
037710 013666 .WORD ERR7
1072 037712 000442 BR 16$ ;EXIT TEST
  
```

1073	037714	012703	00CJ05	10\$:	MOV #5,R3	:SET EXPECTED STATE
1074	037720				WAITMS 20	:WAIT 20 MS FOR DRIVE STATE CHANGE FROM 4 TO 5
	037736	012727	000372		MOV #250.,(PC)+	
	037742	000000			.WORD 0	
	037744	016727	142146		MOV L\$DLY,(PC)+	
	037750	000000			.WORD 0	
	037752	005367	177772		DEC -6(PC)	
	037756	001375			BNE -4	
	037760	005367	177756		DEC -22(PC)	
	037764	001367			BNE -20	
	037766	104422			TRAP C\$BRK	
1075	037776	004767	160200		JSR PC,GDRSTA	:GET DRIVE STATE
1076	040002	020367	143052		CMP R3,T.STAT	:IS STATE 5?
1077	040006	001404			BEQ 16\$	:YES-SKIP
1078	040010				ERRHRD 102.,,ERR7	:REPORT STATE ERROR
	040010	104456			TRAP C\$ERHRD	
	040012	000146			.WORD 102	
	040014	000000			.WORD 0	
	040016	013666			.WORD ERR7	
1079	040020	012701	090062	16\$:	MOV #50.,R1	:INITIALIZE WAIT COUNT
1080	040024	004767	162620		JSR PC,RDYWAIT	:GO WAIT FOR DRIVE READY
1081	040030	040032			60\$	
1082	040032	012767	000002	142756	60\$: MOV #2,ERRSWI	:INIT ERROR SWITCH
1083	040040				ENDSUB	
	040040				L10054:	
	040040	104403			TRAP C\$ESUB	
1084	040042				ESCAPE TST	:EXIT TEST IF ERROR
	040042	104410			TRAP C\$ESCAPE	
	040044	000032			.WORD L10053-	
1085	040046	005367	143044		DEC TEMPO	:DEC PASS COUNT
1086	040052	001411			BEQ 24\$	:SKIP IF 0-DONE
1087						
1088	040054	032767	000001	143034	BIT #BIT0,TEMPO	:TEST IF PASS=2
1089	040062	001003			BNE 23\$	:NO-SKIP
1090	040064	004767	161754		JSR PC,SWAPHD	:GO SWAP TO HEAD 1 OR END TEST
1091	040070	040076			24\$	:ABORT RETURN
1092	040072	000167	177424	23\$:	JMP T172\$	
1093	040076				24\$:	
1094	040076				T1765\$:	
1095	040076				ENDTST	
	040076				L10053:	
	040076	104401			TRAP C\$ETST	

1096						
1097						
1098						
1099						
1100	040100				.SBTTL *TEST 16	DIFFERENCE OF 1 SEEK (PART 2)
	040100				BGNTST	:TEST 16
						T16::
1101	040100	012767	006645	142704	MOV #P2TO2E,ERHEAD	:SET ERROR HEADER
1102	040106	012767	000004	143002	MOV #4,TEMPO	:SET PASS COUNT
1103	040114	004767	156710		JSR PC,TSTINT	:INITIALIZE TEST
1104	040120	004767	156722		JSR PC,GSTATR	:GET STATUS, CLEAR DRIVE
1105	040124	040372			T1865\$	
1106	040126	004767	161666		JSR PC,CHOSHD	:GO CHOOSE HEAD
1107	040132	012767	177777	142762	MOV #-1,TEMP2	:SET DIFF ARGUMENT TO -1 (REVERSE)
1108	040140	012703	003102		MOV #NEWCYL,R3	:GET ADDRESSES
1109	040144	012704	003104		MOV #CURCYL,R4	

1110	040150	012705	000100		MOV	#OLDCYL,R5	
1111	040154				T187\$:		
1112	040154				BGNSUB		
	040154						T16.1:
	040154	104402			TRAP	C\$BSUB	
1113	040156	004767	163040		JSR	PC,GETPOS	;GET CURRENT POSITION
1114	040162	040330			60\$		
1115	040164				BREAK		;ALLOW FOR A ^C
	040164	104422			TRAP	C\$BRK	
1116							
1117	040166				INLOOP		;CHECK IF IN ERROR LOOP
	040166	104420			TRAP	C\$INLP	
1118	040170				BNCOMPLETE	3\$	;NO - SKIP
	040170	103005			BCC	3\$	
1119	040172	021413			CMP	(R4),(R3)	;CHECK IF CURRENT = NEW
1120	040174	001005			BNE	4\$	;NO - SKIP
1121	040176	004767	161702		JSR	PC,ONSWAP	;ELSE SWAP OLD AND NEW
1122	040202	000421			BR	9\$	;SKIP TO SEEK
1123	040204	005467	142712	3\$:	NEG	TEMP2	;CHANGE DIFF ARGUMENT FOR OPPOSITE DIR
1124	040210	011413		4\$:	MOV	(R4),(R3)	;MOV CURRENT INTO NEW
1125	040212	026714	142064		CMP	HLMTW,(R4)	;CHECK IF CURRENT AT 255
1126	040216	001004			BNE	7\$	;NO - SKIP
1127	040220	012767	177777	142674	MOV	#-1,TEMP2	;AT MAX CYL, MAKE NEXT SEEK REV
1128	040226	000405			BR	8\$	;SKIP
1129	040230	005714		7\$:	TST	(R4)	;TEST IF CURRENT AT 0
1130	040232	001003			BNE	8\$	;NO - SKIP
1131	040234	012767	000001	142660	MOV	#1,TEMP2	;AT CYL 0, MAKE NEXT SEEK FWRD
1132	040242	066713	142654	8\$:	ADD	TEMP2,(R3)	;ADD DIFF TO NEW CYL (+1 OR -1)
1133	040246	004767	160110	9\$:	JSR	PC,XSEEK	;DO SEEK
1134	040252	040330			60\$		
1135	040254	012701	000226		MOV	#150,,R1	;SET WAIT COUNT FOR 15 MS
1136	040260	004767	162364		JSR	PC,RDYWAIT	;WAIT FOR READY
1137	040264	040330			60\$		
1138	040266	004767	162730		JSR	PC,GETPOS	;STORE POSITION
1139	040272	040330			60\$		
1140	040274	011501			MOV	(R5),R1	;GET OLD POSITION
1141	040276	161401			SUB	(R4),R1	;SUBTRACT FROM NEW POINTER (FORWARD)
1142	040300	005767	142604		TST	DESSGN	;CHECK IF SIGN FORWARD
1143	040304	001402			BEQ	10\$	;YES-SKIP, ELSE SUB FOR SEEK REVERSE
1144	040306	011401			MOV	(R4),R1	;GET NEW CYLINDER
1145	040310	161501			SUB	(R5),R1	;SUBTRACT FROM OLD CYL
1146	040312	022701	000001	10\$:	CMP	#1,R1	;CHECK IF RESULT IS DIFFERENCE OF 1
1147	040316	001404			BEQ	12\$	;YES-SKIP
1148	040320				ERRHRD	201,,,ERR8	;ELSE REPORT ERROR
	040320	104456			TRAP	C\$ERHRD	
	040322	000311			.WORD	201	
	040324	000000			.WORD	0	
	040326	013736			.WORD	ERR8	
1149	040330			12\$:			
1150	040330	012767	000002	142460	60\$:	MOV	#2,ERRSWI
1151	040336				ENDSUB		;INIT ERROR SWITCH
	040336				L10056:		
	040336	104403			TRAP	C\$ESUB	
1152	040340				ESCAPE	TST	;EXIT TEST IF ERROR
	040340	104410			TRAP	C\$ESCAPE	
	040342	000030			.WORD	L10055-	
1153	040344	005367	142546		DEC	TEMPO	;DEC PASS COUNT



```

1154 040350 001410          BEQ      30$          ;EXIT IF DONE
1155
1156 040352 032767 000001 142536 BIT      #BIT0,TEMPO ;TEST IF PASS 1 OR 3
1157 040360 001003          BNE      20$          ;YES-SKIP
1158 040362 004767 161456 JSR      PC,SWAPHD   ;GO SWAP TO HEAD 1 OR END TEST
1159 040366 040372          30$          ;ABORT RETURN
1160 040370 000671          20$: BR      T187$    ;LOOP
1161 040372          30$:
1162 040372          T1865$:
1163 040372          ENDTST
      040372          L10055:
      040372 104401          TRAP     C$ETST
1164 040374          ENDMOD
1165
1166          .SBTTL  PARAMETER CODING
1167 040374          BGNMOD  HRDPRM
1168 040374          BGNHRD
      040374 000030          .WORD   L10057-L$HARD/2
1169
1170 040376          GPRML   CNTYPE,CNT,1,YES
      040376 005130          .WORD   T$CODE
      040400 040542          .WORD   CNTYPE
      040402 000001          .WORD   1
1171
1172 040404          GPRMA   CSRMSG,CSR,0,160000,177776,YES
      040404 000031          .WORD   T$CODE
      040406 040456          .WORD   CSRMSG
      040410 160000          .WORD   T$LLOLIM
      040412 177776          .WORD   T$HILIM
1173
1174 040414          GPRMA   VECMSG,VECT,0,0,776,YES
      040414 001031          .WORD   T$CODE
      040416 040472          .WORD   VECMSG
      040420 000000          .WORD   T$LLOLIM
      040422 000776          .WORD   T$HILIM
1175
1176 040424          GPPMD   DRMSG,DRSB,0,3400,0,7,YES
      040424 004032          .WORD   T$CODE
      040426 040534          .WORD   DRMSG
      040430 003400          .WORD   3400
      040432 000000          .WORD   T$LLOLIM
      040434 000007          .WORD   T$HILIM
1177
1178 040436          GPRML   DRTYPE,TYPDR,1,YES
      040436 003130          .WORD   T$CODE
      040440 040512          .WORD   DRTYPE
      040442 000001          .WORD   1
1179
1180 040444          GPRMD   BRMSG,PRIOR,0,340,0,7,YES
      040444 002032          .WORD   T$CODE
      040446 040501          .WORD   BRMSG
      040450 000340          .WORD   340
      040452 000000          .WORD   T$LLOLIM
      040454 000007          .WORD   T$HILIM
1181
1182 040456          ENDIIRD
      .EVEN
  
```

1183	040456				L10057:
1184					.EVEN
1185					
1186	040456	102	125	123	CSRMSG: .ASCIZ /BUS ADDRESS/
	040461	040	101	104	
	040464	104	122	105	
	040467	123	123	000	
1187					
1188	040472	126	105	103	VECMMSG: .ASCIZ /VECTOR/
	040475	124	117	122	
	040500	000			
1189					
1190	040501	102	122	040	BRMSG: .ASCIZ /BR LEVEL/
	040504	114	105	126	
	040507	105	114	000	
1191					
1192	040512	104	122	111	DRTYPE: .ASCIZ /DRIVE TYPE = RL01/
	040515	126	105	040	
	040520	124	131	120	
	040523	105	040	075	
	040526	040	122	114	
	040531	060	061	000	
1193					
1194	040534	104	122	111	DRMSG: .ASCIZ /DRIVE/
	040537	126	105	000	
1195					
1196	040542	122	114	061	CNTYPE: .ASCIZ /RL11/
	040545	061	000		
1197					
1198	040547				ENDMOD
1199					.EVEN
1200					
1201					
1202	040550				BGNMOD SFTPRM
1203	040550				BGNSFT
	040550	000016			.WORD L1G060-L\$SOFT/2
1204					
1205	040552				GPRML SELQ,MISWI,4,YES
	040552	000130			.WORD T\$CODE
	040554	040606			.WORD SELQ
	040556	000004			.WORD 4
1206					
1207	040560				GPRML ALGNQ,MISWI,10,YES
	040560	000130			.WORD T\$CODE
	040562	040641			.WORD ALGNQ
	040564	000010			.WORD 10
1208					
1209	040566				GPRML MANQ,MISWI,100000,YES
	040566	000130			.WORD T\$CODE
	040570	040700			.WORD MANQ
	040572	100000			.WORD 100000
1210					
1211	040574				3\$: GPRMD ERLIMQ,ERLIM,D,377,0,377,YES
	040574	004052			.WORD T\$CODE
	040576	040735			.WORD ERLIMQ
	040500	000377			.WORD 377

```

040602 000000          .WORD  T$LOLIM
040604 000377          .WORD  T$HILIM
1212
1213 040606          ENDSFT
                                .EVEN
                                L10060:
1214
1215          .EVEN
1216
1217 040606          105      130      105  SELQ:  .ASCIZ  /EXECUTE DRIVE SELECT TESTS/
040611          103      125      124
040614          105      040      104
040617          122      111      126
040622          105      040      123
040625          105      114      105
040630          103      124      040
040633          124      105      123
040636          124      123      000
1218
1219 040641          105      130      105  ALGNQ:  .ASCIZ  /EXECUTE HEAD ALIGNMENT SUPPORT/
040644          103      125      124
040647          105      040      110
040652          105      101      104
040655          040      101      114
040660          111      107      116
040663          115      105      116
040666          124      040      123
040671          125      120      120
040674          117      122      124
040677          000
1220
1221 040700          104      117      040  MANQ:  .ASCIZ  /DO MANUAL INTERVENTION TESTS/
040703          115      101      116
040706          125      101      114
040711          040      111      116
040714          124      105      122
040717          126      105      116
040722          124      111      117
040725          116      040      124
040730          105      123      124
040733          123      000
1222
1223 040735          111      116      120  ERLIMQ: .ASCIZ  /INPUT ERROR LIMIT/
040740          125      124      040
040743          105      122      122
040746          117      122      040
040751          114      111      115
040754          111      124      000
1224
1225          .EVEN
1226
1227 040760          ENDMOD
1228
1229 040760          LASTAD
                                .EVEN
040760          000000          .WORD  0
040762          000000          .WORD  0

```

1230 040764  
1231  
1232 040764  
1233  
1234 009001

L\$LAST::  
.EVEN  
L\$LAST::  
.END

ADR = 000020 G  
 ALGNO 040641  
 ALLCYL = 000001  
 ALLSEC = 000002  
 ANYERR = 100000  
 ASSEMB = 000010  
 BADADD = 004000  
 BASK = 000060  
 BANAM 006125  
 BASADD 006042  
 BELL 011274  
 BHSTAT = 000010  
 BIT0 = 000001 G  
 BIT00 = 000001 G  
 BIT01 = 000002 G  
 BIT02 = 000004 G  
 BIT03 = 000010 G  
 BIT04 = 000020 G  
 BIT05 = 000040 G  
 BIT06 = 000100 G  
 BIT07 = 000200 G  
 BIT08 = 000400 G  
 BIT09 = 001000 G  
 BIT1 = 000002 G  
 BIT10 = 002000 G  
 BIT11 = 004000 G  
 BIT12 = 010000 G  
 BIT13 = 020000 G  
 BIT14 = 040000 G  
 BIT15 = 100000 G  
 BIT2 = 000004 G  
 BIT3 = 000010 G  
 BIT4 = 000020 G  
 BIT5 = 000040 G  
 BIT6 = 000100 G  
 BIT7 = 000200 G  
 BIT8 = 000400 G  
 BIT9 = 001000 G  
 BOE = 000400 G  
 BRMSG 040501  
 BSFLAG 003020  
 BSFVAL 003372  
 BSNSTR 010307  
 BYPSNM 010240  
 CAFDT 011423  
 CAMSK 002312  
 CCYLUP 011412  
 CDRDY 011353  
 CHOSHD 022020  
 CKDATA = 000102  
 CKRLM 016444  
 CLKADR 003146  
 CLKFLG 003144  
 CLKINT 016430 G  
 CLNCOD 016174 G  
 CLRBYT 002304  
 CLRPAR 025146

CNT = 000012  
 CNTYPE 040542  
 COMPOP = 007777  
 CONHNG = 000004  
 CONTIN 015002  
 COSTAT = 000040  
 COUNT 003154  
 CRDYS = 000200  
 CSNAM 006120  
 CSR = 000000  
 CSRMSG 040456  
 CURCYL 003104  
 CYLPER 007070  
 CYLTBL 002604  
 CYLUP = 000004  
 CYLWD 010225  
 CSAU = 000052  
 CSAUTO = 000061  
 CSBRK = 000022  
 CSBSEG = 000004  
 CSBSUB = 000020  
 CSEFG = 000045  
 C\$CLCK = 000062  
 C\$CLEA = 000012  
 C\$CLOS = 000035  
 C\$CLP1 = 000006  
 C\$CVEC = 000036  
 C\$DCLN = 000044  
 C\$DODU = 000051  
 C\$DRPT = 000024  
 C\$DU = 000053  
 C\$EDIT = 000003  
 C\$ERDF = 000055  
 C\$ERHR = 000056  
 C\$ERRO = 000060  
 C\$ERSF = 000054  
 C\$ERSO = 000057  
 C\$ESCA = 000010  
 C\$ESEG = 000005  
 C\$ESUB = 000003  
 C\$ETST = 000001  
 C\$EXIT = 000032  
 C\$GETB = 000026  
 C\$GETW = 000027  
 C\$GMAN = 000043  
 C\$GPHR = 000042  
 C\$GPLO = 000030  
 C\$GPRI = 000040  
 C\$INIT = 000011  
 C\$INLP = 000020  
 C\$MANI = 000050  
 C\$MEM = 000031  
 C\$MSG = 000023  
 C\$OPEN = 000034  
 C\$PNTB = 000014  
 C\$PNTF = 000017  
 C\$PNTS = 000016

C\$PNTX = 000015  
 C\$QIO = 000377  
 C\$RDBU = 000007  
 C\$REFG = 000047  
 C\$RESE = 000033  
 C\$REVI = 000003  
 C\$RFLA = 000021  
 C\$RPT = 000025  
 C\$SEFG = 000046  
 C\$SPRI = 000041  
 C\$SVEC = 000037  
 C\$TPRI = 000013  
 C10MS 011373  
 C5SEC 011434  
 C500MS 011404  
 DANAM 006132  
 DATACM = 000001  
 DCKERR = 004000  
 DCLIM = 000012  
 DCLIMW 014340  
 DESDIF 003106  
 DESHD 003112  
 DESSEC 003114  
 DESSGN 003110  
 DIAGMC = 000000  
 DIFAug 003076  
 DIFWD 010201  
 DIRBIT = 000004  
 DIRMSK 002314  
 DLTRP = 010000  
 DLYCNT 003142  
 DOM = 003006  
 DRDMS = 000001  
 DRMSG 040534  
 DRSB = 000010  
 DRSEL = 000004  
 DRSET = 000010  
 DRTYPE 040512  
 DRVCNT 003074  
 DRVERR = 040000  
 DRVNAM 006053  
 DSESTA = 000400  
 DSMSK = 001400  
 DSPCOD 014342 G  
 EF.CON = 000036 G  
 EF.NEW = 000035 G  
 EF.PWR = 000034 G  
 EF.RES = 000037 G  
 EF.STA = 000040 G  
 ERHEAD 003012  
 F\$AU = 000010  
 ERLIM 040735  
 ERLIMW 014336  
 ERRCNT 003160  
 ERRPOI 003156  
 ERRSWI 003016  
 ERRVEC 003140

ERR1 012464 G  
 ERR10 014076 G  
 ERR2 012532 G  
 ERR3 012600 G  
 ERR4 012646 G  
 ERR5 012716 G  
 ERR6 012766 G  
 ERR7 013666 G  
 ERR8 013736 G  
 ERR9 014032 G  
 EVL = 000004 G  
 EXT05 032134  
 E\$END = 002100  
 E\$LOAD = 000035  
 FBSFIL 003570  
 FMTOP1 011442  
 FMTOP2 011471  
 FMTOP3 011513  
 FMT1 011534  
 FMT1.1 011541  
 FMT11 011760  
 FMT12 011766  
 FMT13 011774  
 FMT14 012040  
 FMT15 012072  
 FMT16 012126  
 FMT17 012137  
 FMT18 012161  
 FMT19 012213  
 FMT2 011550  
 FMT20 012250  
 FMT21 012300  
 FMT22 012323  
 FMT23 012357  
 FMT24 012373  
 FMT25 012400  
 FMT26 012410  
 FMT27 012434  
 FMT28 012453  
 FMT3 011553  
 FMT4 011556  
 FMT5 011567  
 FMT6 011607  
 FMT7 011651  
 FMT8 011721  
 FMT9 011753  
 FOLWRT = 000100  
 FRMWD 010232  
 FWDSKO = 002000  
 FWDSKS = 000400  
 F\$AU = 000015  
 F\$AUTO = 000020  
 F\$BGN = 000040  
 F\$CLIA = 000007  
 F\$DU = 000016  
 F\$END = 000041  
 F\$HARD = 000004

F\$HW = 000013  
 F\$INIT = 000006  
 F\$JMP = 000050  
 F\$MOD = 000000  
 F\$MSG = 000011  
 F\$PROT = 000021  
 F\$PWR = 000017  
 F\$RPT = 000012  
 F\$SEG = 000003  
 F\$SOFT = 000005  
 F\$SRV = 000010  
 F\$SUB = 000002  
 F\$SW = 000014  
 F\$TEST = 000001  
 GBND 002310  
 GDRSTA 020202  
 GETPOS 023222  
 GETSTA = 000003  
 GLBDAT 002224 G  
 GLBEQA 002224 G  
 GLBERR 012464 G  
 GLBSUB 016444 G  
 GLBXT 005242 G  
 GSTAT 017076  
 GSTATC 017062  
 GSTATG 017106  
 GSTATR 017046  
 GSTER1 006464  
 GTSTAT = 000104  
 G\$CNT0 = 000200  
 G\$DELM = 000372  
 G\$DISP = 000003  
 G\$EXCP = 000400  
 G\$HILI = 000002  
 G\$LOLI = 000001  
 G\$NO = 000000  
 G\$OFFS = 000400  
 G\$OFSI = 000376  
 G\$PRMA = 000001  
 G\$PRMD = 000002  
 G\$PRML = 000000  
 G\$RADA = 000140  
 G\$RADR = 000000  
 G\$RADD = 000040  
 G\$RADL = 000120  
 G\$RADO = 000020  
 G\$XFER = 000004  
 G\$YES = 000010  
 HADONE 003010  
 HAMES1 007154  
 HAMES2 007237  
 HAMES3 007343  
 HAMES4 007406  
 HCESTA = 040000  
 HCR CER = 004000  
 HDALIG = 000010  
 HDCYL 002316

HDHSEL = 000100	LAE1 006144	L\$SPCP 002020 G	L10057 040456	MRSLT 005423
HDMOVF 007031	LAB2 006157	L\$SPTP 002024 G	L10060 040606	MSEEK 005242
HDRCMP= 000002	LBASE 003150	L\$STA 002030 G	MANQ 040700	MSPERR 010662
HDR40 = 100000	LCLEXT 033406	L\$SW 014326 G	MBADAD 005721	MSTERR 010715
HDSEC = 000077	LCLK 014514	L\$TEST 002114 G	MBADSF 005742	MTMBS 006020
HDSEL = 000020	LCLK1 014522	L\$TIML 002014 G	MBHSTA 010575	MTOSLO 006200
HDWD 010214	LOCERR 003364	L\$UNIT 002012 G	MBSETO= 000001	MULOAD 005434
HDWRD1 003052	LOCYL = 040000	L.BA 003036	MCERR 010415	MUNDEF 011103
HDWRD2 003054	LOE = 040000 G	L.CS 003034	MCONHN 006273	MVOLCK 010551
HDWRD3 003056	LOLIM = 000002	L.DA 003040	MCOSTA 010562	MWDERR 010751
HEAD = 000006	LOIMW 014330	L.MP 003042	MCYLOC 011052	MWGERR 010700
HEADLM= 010000	LOT = 000010 G	L10000 012530	MCYLUP 005445	MWLSTA 010610
HEADW 014334	LPTOS 031714	L10001 012576	MDATCP 005327	MWORD 006172
HICYL = 020000	L\$ACP 002110 G	L10002 012644	MDCRL 010437	MWRCHK 005271
HILIM = 000004	L\$APT 002036 G	L10003 012714	MDHEDR 002000 G	MWRITE 005303
HILIMW 014332	L\$AUT 002070 G	L10004 012764	MDLT 010464	MWRSET 005400
HLMTW 002302	L\$AUTO 015636 G	L10005 013664	MDRDY 010404	MWRTAB 011211
HNFERR= 010000	L\$CCP 002106 G	L10006 013734	MDRERR 010526	M4OHDR 005364
HOE = 100000 G	L\$CLEA 016174 G	L10007 014030	MDRRES 006220	NEWCYL 003102
HOSTAT= 000020	L\$CO 002032 G	L10010 014074	MDRVST 010647	NOCLR = 000010
HPTCOD 014306 G	L\$DEPO 002011 G	L10011 014304	MDSERR 010632	NOCTLR 006754
HRDPRM 040374 G	L\$DESC 002122 G	L10012 014324	MERRS 011265	NOERCT 003365
HRDWT5 025176 G	L\$DESP 002076 G	L10013 014342	MEXERS 011225	NOIRPT= 000002
HSMSK - 000100	L\$DEVP 002060 G	L10015 015634	MFERR 011013	NOOP = 000100
HSSTAT= 000100	L\$DISP 014344 G	L10016 016172	MFMTER 005773	NOPCLK 014442
IBE = 010000 G	L\$DLY 002116 G	L10017 016362	MFOLWR 005521	NOPIR 006060
IBUFF 003764	L\$DTP 002040 G	L10020 016366	MFWD5K 005576	NOTRDY 007003
IDU = 000040 G	L\$DTP 002034 G	L10021 016426	MFWSKO 005631	NOTST 006664
IER = 020000 G	L\$DU 016364 G	L10022 016434	MGTSTA 005315	NSTACH 006416
INITCO 014412 G	L\$DUT 002072 G	L10023 016442	MHCERR 010733	NXMERR= 020000
INITST 006515	L\$DVTY 002212 G	L10024 025454	MHCRC 010427	NXTHL 002306
INOUTS= 000020	L\$EF 002052 G	L10025 025662	MHDERR 010776	NXTPAS 015022
INTEBL= 000100	L\$ENVI 002044 G	L10026 030342	MHDRCP 005346	OBUFF 004364
INTHLR 016370 G	L\$ETP 002102 G	L10027 031530	MHF CRC 010476	OLD CYL 003100
ISR = 000100 G	L\$EXP1 002046 G	L10030 031426	MHNF 010450	ONSWAP 022104
IXE = 004000 G	L\$EXP4 002064 G	L10031 032134	MHOSTA 010621	OPFLAG 003004
I\$AU = 000041	L\$EXP5 002066 G	L10032 032042	MHSTA 010537	OPIERR= 002000
I\$AUTO= 000041	L\$SHARD 040376 G	L10033 033406	MINOUT 005476	OPMSG5 002224
I\$CLN = 000041	L\$HIME 002120 G	L10034 033332	MISTST 006373	OPR002 007470
I\$DU = 000041	L\$HPCP 002016 G	L10035 034300	MISWI = 000000	OPR003 007515
I\$HRD = 000041	L\$HPTP 002022 G	L10036 034366	MISWIW 014326	OPR004 010164
I\$INIT= 000041	L\$HW 014310 G	L10037 035022	MITEST= 100000	OPR1 007540
I\$MOD = 000041	L\$ICP 002104 G	L10040 035500	MNDRST 011057	OPR1A 010135
I\$MSG = 000041	L\$INIT 014412 G	L10041 035460	MNEERR 011041	OPR1B 010141
I\$PROT= 000040	L\$LADP 002026 G	L10042 036256	MNOCLR 006307	OPR10 010003
I\$PTAB= 000041	L\$LAST 040764 G	L10043 036154	MNOINT 006240	OPR11 010051
I\$PWR = 000041	L\$LOAD 002100 G	L10044 036254	MOPER 005414	OPR12 010102
I\$RPT = 000041	L\$LUN 002074 G	L10045 036730	MOPERR 010766	OPR12A 010121
I\$SEG = 000041	L\$MREV 002050 G	L10046 036710	MORCE 003014	OPR2 007616
I\$SETU= 000041	L\$NAME 002000 G	L10047 037150	MOUTIN 005455	OPR3 007651
I\$SFT = 000041	L\$PRIO 002042 G	L10050 037062	MPNAM 006137	OPR6 007665
I\$SRV = 000041	L\$PROT 014404 G	L10051 037430	MQUALS= 003760	OPR7 007720
I\$SUB = 000041	L\$PRT 002112 G	L10052 037406	MREAD 005250	OPR8 007747
I\$TST = 000041	L\$REPP 002062 G	L10053 040076	MREADH 005261	OPR9 007766
JJJ 002300	L\$REV 002010 G	L10054 040040	MRESKO 005665	OUTINS= 000040
J\$JMP = 000167	L\$SOFT 040552 G	L10055 040372	MREVSK 005543	OSAPTS= 000000
LAB 014754	L\$SPC 002056 G	L10056 040336	MRLFAL 011150	OSAU = 000000

O\$BGMR= 000000	RELDWT= 040000	SVCTAG= 000000	T\$TSTM= 177777	T16.1 040154
O\$BGNS= 000001	RESE3 011300	SVCTST= 000001	T\$TSTS= 000001	T172\$ 037522
O\$DU = 000001	RESE4 011304	SWAPHD 022044	T\$SAUT= 010016	T1765\$ 040076
O\$ERRT= 000000	RESE5 011311	S\$LSYM= 010000	T\$SCLE= 010017	T1865\$ 040372
O\$GNSW= 000001	RESE6 011316	TBLSTR 003024	T\$SDU = 010020	T187\$ 040154
O\$POIN= 000001	RESPAR 003062	IBT 002544	T\$SHAR= 010057	T2 025456 G
O\$SETU= 000000	RESTAR 014772	TCERR 010363	T\$SHW = 010012	T25TBL 002430
PART1 = 000001 G	RESTBL 002320	TCLK 014564	T\$SINI= 010015	T25TB2 002456
PASCNT 003152	REVSKO= 001000	TEMPO 003116	T\$SMG= 010011	T3 025664 G
PASNEW 015030	REVSKS= 000200	TEMP1 003120	T\$SPRO= 010014	T33TBL 002504
PASNUM 003360	RLBA = 000002	TEMP2 003122	T\$SOF= 010010	T365\$ 030342
PATTBL 002360	RLBAS 003026	TEMP3 003124	T\$SRV= 010023	T4 030344 G
PAT1 004764	RLCS = 000000	TEMP4 003126	T\$SUB= 010056	T4.1 030364
PAT10 005240	RLCSR = 000000	TEMP5 003130	T\$SW = 010013	T465\$ 031420
PAT2 004766	RLDA = 000004	TEMP6 003132	T\$STES= 010055	T5 031532 G
PAT3 005026	RLDRV 003032	TEMP7 003134	T.BA 003046	T5.1 032024
PAT4 005066	RLMP = 000006	TEMP8 003136	T.CS 003044	T504\$ 032056
PAT5 005126	RLVEC 003030	TOSLOW= 000001	T.DA 003050	T6 032136 G
PAT6 005134	RORWOP= 020000	TRPFLG 003366	T.DRIV 002276	T6.1 032650
PAT7 005174	RPTOP 023716	TRPHAN 016436 G	T.MP 003052	T7 033410 G
PAT8 005176	RPTREM 024712	TSTCLK 014556	T.STAT 003060	T8 034302 G
PAT9 005236	RPTRES 024504	TSTINT 017030	TOSERR 006530	T9 034370 G
PC1.K 014412	RSTRT 014710	TSTLAB 006365	T09ERR J06543	UAM = 000200 G
PNT = 001000 G	SAMSK = 000077	TSTZ 030364	T1 025176 G	ULOAD = 000010
POSHDO 022624	SBSFIL 003374	TY. JR = 000006	T10 035024 G	UNDTST 010151
POSHSE 022620	SECWD 010220	T\$ARGC= 000002	T10ERR 006553	UNXERR 006350
POSHW1 022612	SEEK = 000106	T\$CODE= 004052	T10.1 035056	VCMRST 006327
PRI = 002000 G	SEEKOP= 010000	T\$ERRN= 000311	T104\$ 035056	VCSTAT= 001000
PRIOR = 000004	SELQ 040606	T\$EXCP= 000000	T11 035502 G	VECMG 040472
PRI00 = 000000 G	SEQMES 010253	T\$FLAG= 000040	T11.1 036002	VECT = 000002
PRI01 = 000040 G	SE?DON 015056	T\$GMAN= 000000	T11.2 036156	WAITIN 016636
PRI02 = 000100 G	SFTPRM 040550 G	T\$HILI= 000377	T115\$ 036156	WCMSK = 017777
PRI03 = 000140 G	SGNWD 010207	T\$LAST= 000001	T12 036260 G	WCRNG = 160000
PRI04 = 000200 G	SIMSEK 021152	T\$LOLI= 000000	T12ERR 006573	WDESTA= 100000
PRI05 = 000240 G	SPDERR 006430	T\$LSYM= 010600	T12.1 036306	WGESTA= 002000
PRI06 = 000300 G	SPDSTA= 004000	T\$LTNO= 000020	T124\$ 036306	WLSTAT= 020000
PRI07 = 000340 G	SPTCOD 014324 G	T\$NEST= 177777	T13 036732 G	WRTSWI 003022
PSETNM 003362	SSINDX 003002	T\$NSO = 000000	T13ERR 006605	WTDATA= 000112
PWCON 015300	STAMES 010276	T\$NS1 = 000005	T13.1 036756	XRDHD 022140
PWRFLG 003370	STAMSK= 000007	T\$NS2 = 000002	T134\$ 036756	XRDHDC 022130
P2T01E 006645	STATE2 011323	T\$PTNU= 000000	T14 037152 G	XRDHDG 022144
P2T02E 006645	STATE3 011333	T\$SAVL= 177777	T14ERR 006621	XSEEK 020362
RDALHD 023350	STATE5 011343	T\$SEGL= 177777	T14.1 037174	XSEEKT 020352
RDDATA= 000114	STOSTA= 010000	T\$SUBN= 000001	T15 037432 G	XSEEK1 020366
RDHEAD= 000110	SUBSTK 002404	T\$TAGL= 177777	T15.1 037522	X\$ALWA= 000000
RDMOHR= 000116	SVCBGL= 000001	T\$TAGN= 010061	T153\$ 037174	X\$FALS= 000040
RDYCHK 021462	SVCGBL= 000000	T\$TEMP= 000000	T16 040100 G	X\$OFFS= 000400
RDYWAI 022650	SVCINS= 000000	T\$TEST= 000020	T16ERR 006635	X\$TRUE= 000020
READRL 016604	SVCSUB= 000001			

. ABS. 040764 000  
 000000 001  
 ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 30464 WORDS ( 119 PAGES)  
 DYNAMIC MEMORY AVAILABLE FOR 71 PAGES  
 .CZRLID/C=SVC33.SRC/P:1,CZRLID.P11









DONE	6-147#	7-746*	8-38	8-61	8-133*	8-140	8-210*	8-222	8-310*	8-317	8-371*	8-376	8-595*	8-606
	9-420*	9-427	9-466*	9-469	9-776*	9-786	9-789*	9-801						
DRDYS	2-89#	7-648	7-697	7-719	8-104	8-149	8-491	8-502	8-608	8-616	8-764	8-777	9-241	9-273
	9-511	9-632	9-647	9-696	9-713	9-846	9-863							
DRMSG	9-:76	9-:94#												
DRSB	2-12#	9-:76	9-:76	9-:76										
DRSELT	2-26#	7-541	9-353	9-403										
DRSET	3-12#	8-79	8-95	8-144	8-:87	9-423	9-460	9-780						
DRTYPE	9-:78	9-:92#												
DRVCNT	6-180#	7-553*	7-578	7-582*	7-585*									
DRVERR	2-78#	8-97	8-114	8-168	9-233	9-563								
DRVNAM	6-388#	7-421	7-655	7-690	7-701	8-15	8-<48	9-17	9-60	9-98	9-102	9-276	9-284	9-336
	9-355	9-516	9-527	9-761	9-772									
DSESTA	3-28#	9-478												
DSMSK	2-85#													
DSPCOD	7-461#													
ESEND	1-8#													
ESLOAD	1-8#	1-17												
EF.CON	2-5#	7-572												
EF.NEW	2-5#	7-575												
EF.PWR	2-5#	7-544												
EF.RES	2-5#	7-568												
EF.STA	2-5#	7-549												
ERHEAD	6-149#	8-:71	9-14*	9-57*	9-106*	9-141*	9-157*	9-186*	9-194*	9-204*	9-247*	9-269*	9-361*	9-406*
	9-506*	9-593*	9-615*	9-677*	9-828*	9-901*	9-950*	9-:18*	9-:01*					
ERLIM	2-20#	9-<11	9-<11	9-<11										
ERLIMQ	9-<11	9-<23#												
ERLIMW	7-456#	8-10	8-14											
ERR1	7-189#	8-120	8-157	8-176	8-321	8-384	8-635	8-905	9-474	9-942				
ERR10	7-415#	9-985	9-991											
ERR2	7-203#	9-72	9-77	9-570	9-668	9-735	9-885	9-927						
ERR3	7-217#	8-771	9-29	9-33	9-37	9-143	9-481	9-547	9-666	9-733	9-883	9-925		
ERR4	7-231#	9-173	9-557	9-606	9-634	9-698	9-848							
ERR5	7-246#	8-498	8-612	8-623	8-783	9-162	9-184	9-201	9-231	9-236	9-250	9-653	9-720	9-870
ERR6	7-261#	8-170	8-326	8-509	8-627	8-787	8-910	9-579	9-658	9-724	9-875	9-925		
ERR7	7-366#	9-41	9-124	9-138	9-192	9-214	9-226	9-306	9-318	9-330	9-644	9-710	9-859	9-:71
	9-:78													
ERR8	7-380#	9-:48												
ERR9	7-402#													
ERRCNT	6-240#	7-555	7-560	7-581	8-510*	8-788*								
ERRPOI	6-239#	7-192*	7-204*	7-218*	7-232*	7-247*	7-264*	7-367*	7-381*	7-403*	7-419*	7-560*	7-581*	7-586*
	8-10													
ERRSWI	6-151#	8-94*	8-177*	8-190	8-192	8-251*	8-322*	8-327*	8-333	8-335	8-351*	8-385*	8-391	8-393
	8-487*	8-511*	8-517	8-519	8-585*	8-618*	8-636*	8-649	8-651	8-759*	8-789*	8-795	8-797	8-828
	8-830	8-876*	8-906*	8-911*	8-923	8-925	9-332*	9-350*	9-382*	9-487*	9-:82*	9-:50*		
ERRVEC	6-230#	7-661*	7-681	7-704	7-713	7-726								
EVL	2-5#													
EXTOS	9-352	9-354	9-394#											
FSAU	1-8#													
FSAUTO	1-8#	7-679	7-705											
F\$BGN	1-8#	1-16	1-18	2-3	3-37	5-4	6-353	6-361	7-132	7-141	7-189	7-203	7-217	7-231
	7-246	7-261	7-366	7-380	7-402	7-415	7-429	7-432	7-441	7-443	7-459	7-461	7-468	7-471
	7-480	7-481	7-667	7-679	7-710	7-711	7-729	7-733	7-739	7-752	7-758	8-4	8-<68	9-1
	9-7	9-45	9-51	9-79	9-84	9-91	9-125	9-144	9-163	9-174	9-185	9-193	9-202	9-215
	9-227	9-232	9-237	9-254	9-259	9-266	9-268	9-268	9-307	9-319	9-334	9-344	9-349	9-379
	9-379	9-384	9-395	9-400	9-405	9-455	9-455	9-475	9-482	9-488	9-495	9-500	9-505	9-548



FMT16	7-119#	8-:67												
FMT17	7-120#	7-335												
FMT18	7-121#													
FMT19	7-122#													
FMT2	7-106#	8-<00	8-<04											
FMT20	7-123#													
FMT21	7-124#													
FMT22	7-125#	8-<11												
FMT23	7-126#													
FMT24	7-127#	7-654	7-688	7-699										
FMT25	7-128#	8-14												
FMT26	7-129#													
FMT27	7-130#	7-358												
FMT28	7-131#	7-302												
FMT3	7-107#	7-657	7-691	7-702	8-16									
FMT4	7-108#	8-:71	9-389											
FMT5	7-109#	7-421	7-655	7-690	7-701	8-15	8-<48	9-761						
FMT6	7-110#	8-<50												
FMT7	7-111#	8-<52												
FMT8	7-112#	8-<51												
FMT9	7-113#	7-530	8-:66	9-432	9-489	9-763	9-765	9-767	9-770	9-813				
FMTOP1	7-101#	9-17	9-60	9-98	9-102	9-276	9-284	9-336	9-355	9-516	9-527	9-772		
FMTOP2	7-102#	9-449												
FMTOP3	7-103#	9-373												
FOLWRT	2-51#	2-61												
FPMWD	7-38#	8-<08												
FWDSKO	2-55#	2-61												
FWDSKS	2-53#	2-61												
GSCNTO	1-8#													
G\$DELM	1-8#	7-281	7-651	7-721	8-99	8-116	8-139	8-151	8-159	8-316	8-380	8-493	8-504	8-605
	8-768	8-779	9-468	9-:74										
G\$DISP	1-8#													
G\$EXCP	1-8#													
G\$HILI	1-8#													
G\$LOLI	1-8#													
G\$NO	1-8#	9-20	9-62	9-118	9-278	9-303	9-339	9-358	9-376	9-391	9-451	9-491	9-529	9-815
G\$OFFS	1-3#	9-20	9-62	9-118	9-278	9-303	9-339	9-358	9-376	9-391	9-451	9-491	9-529	9-815
	9-:70	9-:72	9-:74	9-:76	9-:78	9-:80	9-<05	9-<07	9-<09	9-<11				
G\$OF\$SI	1-8#	9-20	9-62	9-118	9-278	9-303	9-339	9-358	9-376	9-391	9-451	9-491	9-529	9-815
	9-:70	9-:72	9-:74	9-:76	9-:78	9-:80	9-<05	9-<07	9-<09	9-<11				
G\$PRMA	1-8#	9-:72	9-:74											
G\$PRMD	1-8#	9-:76	9-:80	9-<11										
G\$PRML	1-8#	9-20	9-62	9-118	9-278	9-303	9-339	9-358	9-376	9-391	9-451	9-491	9-529	9-815
	9-:70	9-:78	9-<05	9-<07	9-<09									
G\$RADA	1-8#													
G\$RADB	1-8#													
G\$RADD	1-8#	9-<11												
G\$RADL	1-8#	9-20	9-62	9-118	9-278	9-303	9-339	9-358	9-376	9-391	9-451	9-491	9-529	9-815
	9-:70	9-:78	9-<05	9-<07	9-<09									
G\$RADO	1-8#	9-:72	9-:74	9-:76	9-:80									
G\$XFER	1-8#													
G\$YES	1-8#	9-:70	9-:72	9-:74	9-:76	9-:78	9-:80	9-<05	9-<07	9-<09	9-<11			
GBND	5-33#	7-606*	7-614*											
GDRSTA	8-202#	9-477	9-:66	9-:75										
GETPOS	8-254	8-807#	9-:34	9-:13	9-:38									
GETSTA	3-11#	7-277	8-79	8-82	8-129	8-208	9-423	9-460	9-780					

GLBDAT	5-4#													
GLBEQA	2-3#													
GLBERR	7-141#													
GLBSUB	8-4#													
GLBTXT	6-361#													
GSTAT	8-84#	8-102	8-154	8-160	8-489	8-500	8-614	8-762	8-775	9-630	9-637	9-694	9-703	9-844
	9-852													
GSTATC	8-81#	9-108	9-123	9-151	9-165	9-176	9-188	9-197	9-206	9-218	9-239	9-289	9-310	9-322
	9-363	9-380	9-519	9-533										
GSTATG	8-80	8-83	8-86#											
GSTATR	8-78#	9-24	9-42	9-67	9-93	9-271	9-509	9-596	9-622	9-688	9-809	9-838	9-911	9-959
	9-:21	9-:04												
GSTER1	6-409#	9-406	9-474											
GTSTAT	2-35#	8-136	8-214	9-421	9-459	9-779								
HADONE	6-148#	7-562*	9-755	9-759*										
HAMES1	7-11#	9-763												
HAMES2	7-12#	9-765												
HAMES3	7-13#	9-767												
HAMES4	7-14#	9-770												
HCESTA	3-34#	8-108												
HRCRER	2-83#	7-299	7-306											
HDALIG	2-27#	7-541	9-751											
HDCYL	5-36#	7-609*	7-617*											
HDHSEL	3-20#	9-976	9-979											
HDMOVF	7-8#													
HDR40	2-60#	8-878	8-<01											
HDRCMP	2-45#	9-969												
HDSEC	3-19#	9-997	9-:00											
HDSEL	3-8#	8-306	8-369	8-889	9-790	9-794								
HDWD	7-35#	8-<08	8-<11	8-<50										
HDWRD1	6-168#	7-387	8-734	8-817	9-935	9-938*	9-939							
HDWRD2	6-169#	8-629												
HDWRD3	6-170#													
HEAD	2-19#													
HEADLM	2-28#	8-527	8-536											
HEADW	7-455#	8-529												
HICYL	2-29#	7-621												
HILIM	2-18#													
HILIMW	7-454#	7-623*												
HLMTW	5-30#	7-605*	7-613*	7-623	8-257	8-259	8-261	8-283	9-:46	9-:25				
HNFERR	2-81#	7-297												
HOE	2-5#													
HOSTAT	3-25#	8-106	9-199	9-540										
HPTCOD	7-432#													
HRDPRM	9-:67#													
HRDWTS	9-1#													
HSMK	3-3#													
HSSTAT	3-27#	8-738	9-553	9-602										
ISAU	1-8#													
ISAUTO	1-8#	7-679#	7-705#											
ISCLN	1-8#	7-711#	7-727#											
ISDU	1-8#	7-729#	7-731#											
ISHRD	9-:68#	9-:82#												
ISINIT	1-8#	7-481#	7-665#											
ISMOD	1-8#	1-16	1-16#	1-18	1-1#	2-3	2-3#	3-37	3-37#	5-4	5-4#	6-353	6-353#	6-361
	6-361#	7-132	7-132#	7-141	7-141#	7-429	7-429#	7-432	7-432#	7-441	7-441#	7-443	7-443#	7-459

	7-4>9#	7-461	7-461#	7-468	7-468#	7-480	7-480#	7-667	7-667#	7-710	7-710#	7-733	7-733#	8-4
	8-4#	8-<68	8-<68#	9-1	9-1#	9-;64	9-;64#	9-;67	9-;67#	9-;98	9-;98#	9-<02	9-<02#	9-<27
	9-<27#													
ISMSG	1-8#	7-189#	7-201#	7-203#	7-215#	7-217#	7-229#	7-231#	7-244#	7-246#	7-259#	7-261#	7-364#	7-366#
	7-378#	7-380#	7-400#	7-402#	7-414#	7-415#	7-428#							
ISPROT	1-8#	7-471#												
ISPTAB	1-8#													
ISPR	1-8#													
ISRPT	1-8#													
ISSEG	1-8#	9-7	9-51	9-84	9-259	9-268	9-349	9-379	9-400	9-455	9-500	9-592	9-614	9-676
	9-686	9-750	9-774	9-807	9-827	9-836	9-900	9-909	9-949	9-957	9-;16	9-;33	9-;00	9-;12
ISSETU	1-8#													
ISSFT	9-<03#	9-<13#												
ISSRV	1-8#	7-739#	7-748#	7-752#	7-754#	7-758#	7-760#							
ISSUB	1-8#	9-7	9-51	9-84	9-259	9-268	9-268#	9-307	9-319	9-334	9-334#	9-334#	9-349	9-379
	9-379#	9-384	9-384#	9-384#	9-400	9-455	9-455#	9-475	9-482	9-488	9-488#	9-488#	9-500	9-592
	9-614	9-676	9-686	9-686#	9-699	9-711	9-721	9-725	9-734	9-739	9-739#	9-739#	9-750	9-774
	9-774#	9-805	9-805#	9-805#	9-807	9-807#	9-820	9-820#	9-827	9-836	9-836#	9-836#	9-849	9-860
	9-871	9-876	9-884	9-889	9-889#	9-889#	9-900	9-909	9-909#	9-926	9-931	9-931#	9-931#	9-949
	9-957	9-957#	9-;05	9-;05#	9-;05#	9-;16	9-;33	9-;33#	9-;83	9-;83#	9-;83#	9-;00	9-;12	9-;12#
	9-;51	9-;51#	9-;51#											
ISTST	1-8#	9-7	9-7#	9-45	9-45#	9-45#	9-51	9-51#	9-79	9-79#	9-79#	9-84	9-84#	9-91
	9-125	9-144	9-163	9-174	9-185	9-193	9-202	9-215	9-227	9-232	9-237	9-254	9-254#	9-254#
	9-259	9-259#	9-266	9-268	9-344	9-344#	9-344#	9-349	9-349#	9-379	9-395	9-395#	9-395#	9-400
	9-400#	9-405	9-455	9-495	9-495#	9-495#	9-500	9-500#	9-505	9-548	9-558	9-580	9-587	9-587#
	9-587#	9-592	9-592#	9-609	9-609#	9-609#	9-614	9-614#	9-635	9-645	9-654	9-659	9-667	9-671
	9-671#	9-671#	9-676	9-676#	9-686	9-745	9-745#	9-745#	9-750	9-750#	9-758	9-774	9-807	9-822
	9-822#	9-822#	9-827	9-827#	9-836	9-895	9-895#	9-895#	9-900	9-900#	9-909	9-944	9-944#	9-944#
	9-949	9-949#	9-957	9-;11	9-;11#	9-;11#	9-;16	9-;16#	9-;33	9-;84	9-;95	9-;95#	9-;95#	9-;00
	9-;00#	9-;12	9-;52	9-;63	9-;63#	9-;63#								
IBE	2-5#													
IBUFF	6-254#	7-344	8-879	9-971										
IDU	2-5#													
IER	2-5#													
INITCO	7-480#													
INITST	6-410#	9-506	9-593											
INOUTS	2-49#	2-61												
INTEBL	2-87#													
INTHLR	7-624	7-739#												
ISR	2-5#													
IXE	2-5#													
JSJMP	1-8#													
JJJ	5-29#													
LSACP	1-17#													
LSAPT	1-17#													
LSAUT	1-17#													
LSAUTO	1-17	7-679#												
LSCCP	1-17#													
LSCLEA	1-17	7-711#												
LSCO	1-17#													
L\$DEPO	1-17#													
L\$DESC	1-17	1-20#												
L\$DESP	1-17#													
L\$DEVP	1-17#													
L\$DISP	1-17	7-463#												
L\$DLY	1-17#	7-281	7-651	7-721	8-50	8-99	8-116	8-139	8-151	8-159	8-316	8-380	8-493	8-504

























SVCSUB	9-<11	9-<11	9-<11	9-<11	9-<13	9-<13	9-<29	9-<29	9-<29	9-<29	9-<29	9-<29	9-<29	9-<29
SVCTAG	1-8#	1-10#	9-268	9-379	9-455	9-686	9-774	9-807	9-836	9-909	9-957	9-:33	9-:12	7-244
	1-8#	1-13#	7-201	7-201	7-201	7-215	7-215	7-215	7-229	7-229	7-229	7-244	7-244	7-244
	7-259	7-259	7-259	7-364	7-364	7-364	7-378	7-378	7-378	7-400	7-400	7-400	7-414	7-414
	7-414	7-428	7-428	7-428	7-440	7-440	7-440	7-458	7-458	7-458	7-665	7-665	7-665	7-705
	7-705	7-705	7-727	7-727	7-727	7-731	7-731	7-731	7-748	7-748	7-748	7-754	7-754	7-754
	7-760	7-760	7-760	9-20	9-20	9-20	9-45	9-45	9-45	9-62	9-62	9-62	9-79	9-79
	9-79	9-118	9-118	9-118	9-254	9-254	9-254	9-278	9-278	9-278	9-303	9-303	9-303	9-334
	9-334	9-334	9-339	9-339	9-339	9-344	9-344	9-344	9-358	9-358	9-358	9-376	9-376	9-376
	9-384	9-384	9-384	9-391	9-391	9-391	9-395	9-395	9-395	9-451	9-451	9-451	9-488	9-488
	9-488	9-491	9-491	9-491	9-495	9-495	9-495	9-529	9-529	9-529	9-587	9-587	9-587	9-609
	9-609	9-609	9-671	9-671	9-671	9-739	9-739	9-739	9-745	9-745	9-745	9-805	9-805	9-805
	9-815	9-815	9-815	9-820	9-820	9-820	9-822	9-822	9-822	9-889	9-889	9-889	9-895	9-895
	9-895	9-931	9-931	9-931	9-944	9-944	9-944	9-:05	9-:05	9-:05	9-:11	9-:11	9-:11	9-:83
	9-:83	9-:83	9-:95	9-:95	9-:95	9-:51	9-:51	9-:51	9-:63	9-:63	9-:63	9-:82	9-:82	9-:82
	9-<13	9-<13	9-<13											
SVCTST	1-8#	1-9#	9-7	9-51	9-84	9-259	9-349	9-400	9-500	9-592	9-614	9-676	9-750	9-827
	9-900	9-949	9-:16	9-:00										
SWAPHD	8-536#	9-:90	9-:58											
TSSAUT	7-679#	7-705												
TSSCLE	7-711#	7-727												
TSSDU	7-729#	7-731												
TSSHAR	9-:68	9-:68#	9-:82											
TSSHW	7-433	7-433#	7-440											
TSSINI	7-481#	7-665												
TSSMSG	7-189#	7-201	7-203#	7-215	7-217#	7-229	7-231#	7-244	7-246#	7-259	7-261#	7-364	7-366#	7-378
	7-380#	7-400	7-402#	7-414	7-415#	7-428								
	7-471#													
TSSPRO	9-<03	9-<03#	9-<13											
TSSSRV	7-739#	7-748	7-752#	7-754	7-758#	7-760								
TSSSUB	9-268#	9-307	9-319	9-334	9-379#	9-384	9-455#	9-475	9-482	9-488	9-686#	9-699	9-711	9-721
	9-725	9-734	9-739	9-774#	9-805	9-807#	9-820	9-836#	9-849	9-860	9-871	9-876	9-884	9-889
	9-909#	9-926	9-931	9-957#	9-:05	9-:33#	9-:83	9-:12#	9-:51					
TSSSW	7-444	7-444#	7-458											
TSSTES	9-7#	9-45	9-51#	9-79	9-84#	9-91	9-125	9-144	9-163	9-174	9-185	9-193	9-202	9-215
	9-227	9-232	9-237	9-254	9-259#	9-266	9-344	9-349#	9-395	9-400#	9-405	9-495	9-500#	9-505
	9-548	9-558	9-580	9-587	9-592#	9-609	9-614#	9-635	9-645	9-654	9-659	9-667	9-671	9-676#
	9-745	9-750#	9-758	9-822	9-827#	9-895	9-900#	9-944	9-949#	9-:11	9-:16#	9-:84	9-:95	9-:00#
	9-:52	9-:63												
TSARGC	1-17	1-17	1-17	1-17	1-17	1-17	1-17	1-17	1-17	1-17	1-17	1-17#	1-17#	1-17#
	1-17#	1-17#	1-17#	7-302	7-302	7-302	7-302	7-302	7-302#	7-302#	7-302#	7-302#	7-302#	7-335
	7-335	7-335	7-335#	7-335#	7-335#	7-349	7-349	7-349	7-349	7-349	7-349	7-349	7-349	7-349#
	7-349#	7-349#	7-349#	7-349#	7-349#	7-349#	7-358	7-358	7-358	7-358	7-358	7-358	7-358	7-358#
	7-358#	7-358#	7-358#	7-421	7-421	7-421	7-421	7-421	7-421	7-421#	7-421#	7-421#	7-421#	7-421#
	7-422	7-422	7-422	7-422	7-422	7-422	7-422	7-422	7-422	7-422#	7-422#	7-422#	7-422#	7-422#
	7-422#	7-422#	7-422#	7-424	7-424	7-424	7-424	7-424	7-424	7-424	7-424	7-424#	7-424#	7-424#
	7-424#	7-424#	7-424#	7-424#	7-654	7-654	7-654	7-654#	7-654#	7-655	7-655	7-655	7-655	7-655
	7-655	7-655#	7-655#	7-655#	7-655#	7-655#	7-657	7-657	7-657#	7-688	7-688	7-688	7-688#	7-688#
	7-690	7-690	7-690	7-690	7-690	7-690	7-690#	7-690#	7-690#	7-690#	7-690#	7-691	7-691	7-691#
	7-699	7-699	7-699	7-699#	7-699#	7-701	7-701	7-701	7-701	7-701	7-701	7-701#	7-701#	7-701#
	7-701#	7-701#	7-702	7-702	7-702#	8-14	8-14	8-14	8-14	8-14	8-14#	8-14#	8-14#	8-15
	8-15	8-15	8-15	8-15	8-15#	8-15#	8-15#	8-15#	8-15#	8-16	8-16	8-16#	8-16#	8-15
	8-:66	8-:66#	8-:66#	8-:67	8-:67	8-:67	8-:67#	8-:67#	8-:67#	8-:71	8-:71	8-:71	8-:71	8-:66
	8-:71#	8-:84	8-:84	8-:84	8-:84	8-:84	8-:84#	8-:84#	8-:84#	8-:71	8-:71	8-:71	8-:71#	8-:71#
	8-<04	8-<04	8-<04#	8-<04#	8-<04#	8-<08	8-<08	8-<08	8-<08	8-<00	8-<00	8-<00	8-<00#	8-<04
	8-<08#	8-<08#	8-<08#	8-<08#	8-<08#	8-<08#	8-<08#	8-<08#	8-<08#	8-<08	8-<08	8-<08	8-<08	8-<08
										8-<11	8-<11	8-<11	8-<11	8-<11





T\$SUBN	1-8# 9-455 9-807 9-:16#	9-7# 9-455# 9-807# 9-:33	9-51# 9-500# 9-827# 9-:33	9-84# 9-592# 9-836 9-:33#	9-259# 9-614# 9-836 9-:00#	9-268 9-676# 9-836# 9-:12	9-268 9-686 9-90G# 9-:12	9-268# 9-686 9-909 9-:12#	9-349# 9-686# 9-909 9-:12#	9-379 9-750# 9-909# 9-:12#	9-379 9-774 9-949# 9-:12#	9-379# 9-774 9-957 9-:12#	9-400# 9-774# 9-957 9-:12#	9-455 9-807 9-957# 9-:12#	
T\$TAGL	1-8#	7-189	7-189	7-189#	7-203	7-203	7-203#	7-217	7-217	7-217#	7-231	7-231	7-231#	7-246	
T\$TAGN	1-8# 7-246 7-415 7-481# 7-752 9-259 9-400# 9-676 9-827 9-949# 9-:12	7-246# 7-415 7-679 7-752# 9-259 9-455 9-676# 9-827 9-957 9-:12#	7-261 7-415# 7-679 7-758 9-259# 9-455# 9-686 9-827# 9-957 9-:68	7-261# 7-433 7-679# 7-758 9-268 9-455# 9-686 9-836 9-957# 9-:68	7-261# 7-433 7-679# 7-758 9-268 9-455# 9-686 9-836 9-957# 9-:68#	7-203 7-366 7-433# 7-711 7-758# 9-268 9-500 9-686# 9-836 9-:16	7-203 7-366 7-433# 7-711 7-758# 9-268 9-500 9-686# 9-836 9-:16	7-203# 7-366 7-444 7-711# 9-7 9-349 9-500 9-592 9-750 9-900 9-:16#	7-217 7-366# 7-444 7-729 9-7 9-349 9-500 9-592 9-750 9-900 9-:33	7-217# 7-380 7-444# 7-729 9-51 9-349# 9-592 9-774 9-900# 9-:33#	7-217# 7-380 7-471 7-729# 9-51 9-379 9-592# 9-774 9-909 9-:00	7-231 7-380# 7-471 7-739 9-51# 9-84 9-379# 9-614 9-807 9-909# 9-:00	7-231 7-402 7-471# 7-739 9-84 9-379# 9-614 9-807 9-909# 9-:00	7-231# 7-402 7-481 7-739# 9-84 9-400 9-614# 9-807 9-949 9-:00#	7-246 7-402# 7-481 7-752 9-84# 9-400 9-676 9-807# 9-949 9-:12
T\$TEMP	1-18 7-244 7-429 7-463 7-463 7-463# 7-463# 7-727# 9-20 9-79# 9-163# 9-232# 9-303 9-339# 9-376# 9-405# 9-491 9-529# 9-645# 9-721# 9-815 9-871# 9-944# 9-:52# 9-:72# 9-:76# 9-:82# 9-:07# 9-:13#	1-18# 7-244# 7-429# 7-463 7-463 7-463# 7-463# 7-731 9-20 9-91 9-174 9-237 9-303 9-339# 9-376# 9-451 9-491 9-548 9-654 9-725 9-815 9-876 9-944# 9-:05 9-:63 9-:72# 9-:78 9-:98 9-:09 9-:27	3-37 7-259 7-440 7-463 7-463 7-463# 7-463# 7-731# 9-20# 9-91# 9-174# 9-237# 9-303# 9-339# 9-376# 9-451 9-491# 9-548# 9-654# 9-725# 9-815# 9-876# 9-944# 9-:05# 9-:63# 9-:72# 9-:78 9-:98# 9-:09# 9-:27#	3-37# 7-259# 7-440# 7-463 7-463 7-463# 7-468 7-733# 9-20# 9-118 9-185 9-254 9-303# 9-344 9-384 9-451 9-491# 9-558 9-659 9-734 9-815# 9-884 9-944# 9-:11 9-:64 9-:74 9-:78# 9-:05 9-:09#	6-353 7-364 7-441 7-463 7-463 7-463# 7-468# 7-733# 9-20# 9-118 9-185# 9-254# 9-303# 9-344# 9-384# 9-451# 9-491# 9-558# 9-667 9-734# 9-815# 9-884# 9-944# 9-:11# 9-:64# 9-:74 9-:78# 9-:05# 9-:09#	6-353# 7-378 7-441# 7-463 7-463 7-463# 7-468# 7-748 9-45 9-118 9-193 9-266 9-307 9-358 9-391 9-451# 9-495 9-580 9-667# 9-739 9-820 9-889 9-944# 9-:83 9-:70 9-:74# 9-:78# 9-:05# 9-:09#	7-132 7-378# 7-458 7-463 7-463 7-463# 7-468# 7-748 9-45# 9-118# 9-193# 9-266# 9-307# 9-358 9-391 9-451# 9-495 9-580# 9-671 9-745 9-822 9-895 9-949 9-:84 9-:70# 9-:74# 9-:80 9-:05# 9-:11	7-132# 7-378# 7-458# 7-463 7-463 7-463# 7-468# 7-754 9-62 9-118# 9-202 9-278 9-319 9-358# 9-391# 9-475 9-505 9-587 9-671# 9-745# 9-822# 9-895# 9-949# 9-:84# 9-:70# 9-:74# 9-:80 9-:05# 9-:11	7-201 7-400 7-459 7-463 7-463 7-463# 7-463# 7-665 7-754 9-62 9-118# 9-202# 9-278 9-319# 9-358# 9-391# 9-482 9-529 9-609 9-699 9-758 9-849 9-895# 9-926 9-957 9-:84# 9-:70# 9-:76 9-:80 9-:07# 9-:11	7-201# 7-400# 7-459# 7-463 7-463 7-463# 7-463# 7-667# 7-760 9-62 9-125 9-215 9-278# 9-334 9-358# 9-391# 9-482# 9-529 9-609# 9-699# 9-758# 9-849# 9-895# 9-926# 9-957# 9-:95 9-:70# 9-:76 9-:80# 9-:07# 9-:11#	7-215 7-414 7-463 7-463 7-463# 7-463# 7-705 8-:68 9-62# 9-144 9-215# 9-278# 9-334# 9-358# 9-391# 9-488 9-529 9-635 9-711 9-758# 9-805 9-849# 9-895# 9-926# 9-957# 9-:51 9-:72 9-:76# 9-:80# 9-:07# 9-:11#	7-229 7-428 7-463 7-463 7-463# 7-463# 7-705# 8-:68# 9-62# 9-144# 9-227 9-278# 9-339 9-376 9-395# 9-488# 9-529# 9-635# 9-711# 9-758# 9-805# 9-860# 9-931 9-957# 9-:51# 9-:72 9-:76# 9-:80# 9-:07# 9-:11#	7-229# 7-428# 7-463 7-463 7-463# 7-463# 7-727 8-:68# 9-79 9-163# 9-232# 9-303 9-376 9-405# 9-491 9-529# 9-645# 9-721# 9-815 9-871# 9-944# 9-:52# 9-:72# 9-:76# 9-:82# 9-:07# 9-:13#		
T\$TEST	1-8# 9-349 9-614 9-827# 9-:00	9-7 9-349 9-614 9-836 9-:00	9-7 9-349# 9-614# 9-900 9-:00#	9-7# 9-379 9-676 9-900 9-:12	9-51 9-400 9-676 9-900# 9-:29	9-51 9-400 9-676# 9-909 9-:29	9-51# 9-455 9-686 9-949 9-:29	9-84 9-500 9-750 9-949 9-:29	9-84 9-500 9-750# 9-949# 9-:29	9-84# 9-500# 9-750# 9-957 9-:16	9-259 9-500# 9-774 9-:16	9-259 9-592 9-807 9-:16	9-259# 9-592 9-827 9-:16#	9-268 9-592# 9-827 9-:33	
T\$TSTM	1-8# 7-421 7-631 7-701 8-15 8-498	7-201 7-422 7-651 7-702 8-17 8-504	7-215 7-424 7-654 7-703 8-17 8-509	7-229 7-428 7-655 7-704 8-18 8-612	7-244 7-538 7-657 7-705 8-99 8-623	7-259 7-539 7-658 7-713 8-116 8-627	7-302 7-544 7-659 7-715 8-120 8-635	7-335 7-549 7-665 7-721 8-151 8-761	7-349 7-568 7-681 7-722 8-157 8-771	7-358 7-572 7-688 7-726 8-170 8-774	7-364 7-575 7-690 7-731 8-176 8-779	7-378 7-589 7-691 7-731 8-321 8-783	7-400 7-624 7-692 8-12 8-326 8-787	7-414 7-625 7-699 8-14 8-384 8-905	





WGESTA	3-30#					
WLSTAT	3-33#	9-34	9-74	9-567	9-792	9-811
WRTSWI	6-153#					
WTDATA	2-38#					
X\$ALWA	1-8#					
X\$FALS	1-8#					
X\$OFFS	1-8#					
X\$TRUE	1-8#					
XRDHD	8-575#	8-815				
XRDHDC	8-573#	9-919				
XRDHDG	8-574	8-576#				
XSEEK	8-241#	9-:64	9-:33			
XSEEK1	8-240	8-242#				
XSEEKT	8-239#					





GPRMD	9-:76	9-:80	9-<11												
GPRML	9-20	9-20#	9-62	9-62#	9-118	9-118#	9-278	9-278#	9-303	9-303#	9-339	9-339#	9-358	9-358#	
	9-376	9-376#	9-391	9-391#	9-451	9-451#	9-491	9-491#	9-529	9-529#	9-815	9-815#	9-:70	9-:78	
	9-<05	9-<07	9-<09												
HEADER	1-17														
INLOOP	8-12	9-:38	9-;17												
LASTAD	9-<29														
MSBYTE	1-17	1-17	1-17	1-17#											
MSCHEC	9-91	9-91#	9-125	9-125#	9-144	9-144#	9-163	9-163#	9-174	9-174#	9-185	9-185#	9-193	9-193#	
	9-202	9-202#	9-215	9-215#	9-227	9-227#	9-232	9-232#	9-237	9-237#	9-266	9-266#	9-307	9-307#	
	9-319	9-319#	9-405	9-405#	9-475	9-475#	9-482	9-482#	9-505	9-505#	9-548	9-548#	9-558	9-558#	
	9-580	9-580#	9-635	9-635#	9-645	9-645#	9-654	9-654#	9-659	9-659#	9-667	9-667#	9-699	9-699#	
	9-711	9-711#	9-721	9-721#	9-725	9-725#	9-734	9-734#	9-758	9-758#	9-849	9-849#	9-860	9-860#	
	9-871	9-871#	9-876	9-876#	9-884	9-884#	9-926	9-926#							
MSCNTO	9-20	9-20#	9-62	9-62#	9-118	9-118#	9-278	9-278#	9-303	9-303#	9-339	9-339#	9-358	9-358#	
	9-376	9-376#	9-391	9-391#	9-451	9-451#	9-491	9-491#	9-529	9-529#	9-815	9-815#	9-:70	9-:70#	
	9-:72	9-:72#	9-:74	9-:74#	9-:76	9-:76#	9-:78	9-:78#	9-:80	9-:80#	9-<05	9-<05#	9-<07	9-<07#	
	9-<09	9-<09#	9-<11	9-<11#											
MSCOUN	7-302	7-302	7-302	7-302#	7-335	7-335	7-335#	7-349	7-349	7-349	7-349	7-349	7-349	7-349	
	7-358	7-358	7-358	7-358#	7-421	7-421	7-421	7-421	7-421	7-421#	7-422	7-422	7-422	7-422	
	7-422	7-422	7-422	7-422#	7-424	7-424	7-424	7-424	7-424	7-424#	7-654	7-654#	7-655	7-655	
	7-655	7-655	7-655	7-655#	7-657	7-657#	7-688	7-688#	7-690	7-690	7-690	7-690	7-690#	7-691	
	7-691#	7-699	7-699#	7-701	7-701	7-701	7-701	7-701#	7-702	7-702#	8-14	8-14	8-14#	8-15	
	8-15	8-15	8-15	8-15#	8-16	8-16#	8-:66	8-:66#	8-:67	8-:67#	8-:71	8-:71	8-:71#	8-:84	
	8-:84	8-:84#	8-<00	8-<00#	8-<04	8-<04#	8-<08	8-<08#	8-<08	8-<08	8-<08	8-<08	8-<08	8-<08	
	8-<08#	8-<11	8-<11	8-<11#	8-<11	8-<11	8-<11	8-<11#	8-<25	8-<25	8-<25#	8-<34	8-<34#	8-<34#	
	8-<35	8-<35	8-<35#	8-<38	8-<38	8-<38#	8-<48	8-<48#	8-<48	8-<48	8-<48#	8-<50	8-<50#	8-<50	
	8-<50	8-<50	8-<50#	8-<51	8-<51	8-<51	8-<51	8-<51#	8-<51	8-<51#	8-<52	8-<52	8-<52	8-<52	
	8-<52	8-<52	8-<52#	9-17	9-17	9-17	9-17	9-17#	9-17	9-17#	9-17#	9-60	9-60	9-60	
	9-60	9-60	9-60	9-60#	9-98	9-98	9-98	9-98#	9-98	9-98	9-98	9-102	9-102	9-102	
	9-102	9-102	9-102	9-102#	9-276	9-276	9-276	9-276#	9-276	9-276	9-276	9-284	9-284	9-284	
	9-284	9-284	9-284	9-284#	9-336	9-336	9-336	9-336#	9-336	9-336	9-336#	9-355	9-355	9-355	
	9-355	9-355	9-355	9-355#	9-373	9-373	9-373	9-373#	9-373	9-373#	9-389	9-389#	9-432	9-432#	
	9-449	9-449	9-449	9-449#	9-489	9-489	9-516	9-516#	9-516	9-516	9-516	9-516	9-516#	9-516#	
	9-527	9-527	9-527	9-527#	9-527	9-527#	9-761	9-761#	9-761	9-761	9-761#	9-763	9-763#	9-763#	
	9-765	9-765#	9-767	9-767#	9-770	9-770#	9-772	9-772	9-772	9-772	9-772	9-772	9-772#	9-813	
	9-813#														
MSDATA	1-17	1-17	1-17	1-17	1-17	1-17	1-17	1-17	1-17	1-17	1-17	1-17	1-17	1-17	
	1-17	1-17	1-17	1-17	1-17	1-17	1-17	1-17	1-17	1-17	1-17	1-17	1-17	1-17	
	1-17	1-17	1-17	1-17	1-17	1-17	1-17	1-17	1-17	1-17#	1-17#	1-20	1-20#	1-22	
	1-22#														
MSDECR	1-18	1-18#	3-37	3-37#	6-353	6-353#	7-132	7-132#	7-201	7-201#	7-215	7-215#	7-229	7-229#	
	7-244	7-244#	7-259	7-259#	7-364	7-364#	7-378	7-378#	7-400	7-400#	7-414	7-414#	7-428	7-428#	
	7-429	7-429#	7-440	7-440#	7-441	7-441#	7-458	7-458#	7-459	7-459#	7-468	7-468#	7-475	7-475#	
	7-665	7-665#	7-667	7-667#	7-705	7-705#	7-727	7-727#	7-731	7-731#	7-733	7-733#	7-748	7-748#	
	7-754	7-754#	7-760	7-760#	8-<68	8-<68#	9-45	9-45#	9-79	9-79#	9-254	9-254#	9-334	9-334#	
	9-344	9-344#	9-384	9-384#	9-395	9-395#	9-488	9-488#	9-495	9-495#	9-587	9-587#	9-609	9-609#	
	9-671	9-671#	9-739	9-739#	9-745	9-745#	9-805	9-805#	9-820	9-820#	9-822	9-822#	9-889	9-889#	
	9-895	9-895#	9-931	9-931#	9-944	9-944#	9-:05	9-:05#	9-:11	9-:11#	9-:83	9-:83#	9-:95	9-:95#	
	9-:51	9-:51#	9-:63	9-:63#	9-:64	9-:64#	9-:82	9-:82#	9-:98	9-:98#	9-<13	9-<13#	9-<27	9-<27#	
MSDEFA	9-20	9-20#	9-62	9-62#	9-118	9-118#	9-278	9-278#	9-303	9-303#	9-339	9-339#	9-358	9-358#	
	9-376	9-376#	9-391	9-391#	9-451	9-451#	9-491	9-491#	9-529	9-529#	9-815	9-815#	9-:70	9-:70#	
	9-:72	9-:72#	9-:74	9-:74#	9-:76	9-:76#	9-:78	9-:78#	9-:80	9-:80#	9-<05	9-<05#	9-<07	9-<07#	
	9-<09	9-<09#	9-<11	9-<11#											
MSENDE	1-18#	3-37#	6-353#	7-132#	7-201#	7-215#	7-229#	7-244#	7-259#	7-364#	7-378#	7-400#	7-414#	7-428#	
	7-429#	7-440#	7-441#	7-458#	7-459#	7-468#	7-665#	7-667#	7-705#	7-727#	7-731#	7-733#	7-748#	7-754#	

	7-760#	8-<68#	9-45#	9-79#	9-254#	9-334#	9-344#	9-384#	9-395#	9-488#	9-495#	9-587#	9-609#	9-671#
	9-739#	9-745#	9-805#	9-820#	9-822#	9-889#	9-895#	9-931#	9-944#	9-:05#	9-:11#	9-:83#	9-:95#	9-:51#
MSERRI	9-:63#	9-:64#	9-:82#	9-:98#	9-:13#	9-:27#								
	8-120	8-120#	8-157	8-157#	8-170	8-170#	8-176	8-176#	8-321	8-321#	8-326	8-326#	8-384	8-384#
	8-498	8-498#	8-509	8-509#	8-612	8-612#	8-623	8-623#	8-627	8-627#	8-635	8-635#	8-771	8-771#
	8-783	8-783#	8-787	8-787#	8-905	8-905#	8-910	8-910#	9-29	9-29#	9-33	9-33#	9-37	9-37#
	9-41	9-41#	9-72	9-72#	9-77	9-77#	9-124	9-124#	9-138	9-138#	9-143	9-143#	9-162	9-162#
	9-173	9-173#	9-184	9-184#	9-192	9-192#	9-201	9-201#	9-214	9-214#	9-226	9-226#	9-231	9-231#
	9-236	9-236#	9-250	9-250#	9-306	9-306#	9-318	9-318#	9-330	9-330#	9-474	9-474#	9-481	9-481#
	9-547	9-547#	9-557	9-557#	9-570	9-570#	9-579	9-579#	9-606	9-606#	9-634	9-634#	9-644	9-644#
	9-653	9-653#	9-658	9-658#	9-666	9-666#	9-668	9-668#	9-698	9-698#	9-710	9-710#	9-720	9-720#
	9-724	9-724#	9-733	9-733#	9-735	9-735#	9-848	9-848#	9-859	9-859#	9-870	9-870#	9-875	9-875#
	9-883	9-883#	9-885	9-885#	9-925	9-925#	9-927	9-927#	9-942	9-942#	9-985	9-985#	9-991	9-991#
	9-:71	9-:71#	9-:78	9-:78#	9-:48	9-:48#								
MSESCA	9-:84	9-:84#	9-:52	9-:52#										
MSES	9-:84#	9-:52#												
MSEXCP	9-:72	9-:72#	9-:72#	9-:74	9-:74	9-:74#	9-:76	9-:76	9-:76#	9-:80	9-:80	9-:80#	9-<11	9-<11
	9-<11#													
MSEXIT	9-91	9-91#	9-125	9-125#	9-144	9-144#	9-163	9-163#	9-174	9-174#	9-185	9-185#	9-193	9-193#
	9-202	9-202#	9-215	9-215#	9-227	9-227#	9-232	9-232#	9-237	9-237#	9-266	9-266#	9-307	9-307#
	9-319	9-319#	9-405	9-405#	9-475	9-475#	9-482	9-482#	9-505	9-505#	9-548	9-548#	9-558	9-558#
	9-580	9-580#	9-635	9-635#	9-645	9-645#	9-654	9-654#	9-657	9-657#	9-667	9-667#	9-699	9-699#
	9-711	9-711#	9-721	9-721#	9-725	9-725#	9-734	9-734#	9-758	9-758#	9-849	9-849#	9-860	9-860#
	9-871	9-871#	9-876	9-876#	9-884	9-884#	9-926	9-926#						
MSEXSE	9-91#	9-125#	9-144#	9-163#	9-174#	9-185#	9-193#	9-202#	9-215#	9-227#	9-232#	9-237#	9-266#	9-307#
	9-319#	9-405#	9-475#	9-482#	9-505#	9-548#	9-558#	9-580#	9-635#	9-645#	9-654#	9-659#	9-667#	9-699#
	9-711#	9-721#	9-725#	9-734#	9-758#	9-849#	9-860#	9-871#	9-876#	9-884#	9-926#			
MSEX TJ	9-91#	9-125#	9-144#	9-163#	9-174#	9-185#	9-193#	9-202#	9-215#	9-227#	9-232#	9-237#	9-266#	9-307#
	9-319#	9-405#	9-475#	9-482#	9-505#	9-548#	9-558#	9-580#	9-635#	9-645#	9-654#	9-659#	9-667#	9-699#
	9-711#	9-721#	9-725#	9-734#	9-758#	9-849#	9-860#	9-871#	9-876#	9-884#	9-926#			
M\$GEN	1-16	1-16#	1-17	1-17	1-17	1-17	1-17	1-17	1-17	1-17	1-17	1-17	1-17	1-17
	1-17	1-17	1-17	1-17	1-17	1-17	1-17	1-17	1-17	1-17	1-17	1-17	1-17	1-17
	1-17	1-17	1-17	1-17	1-17	1-17	1-17	1-17	1-17	1-17	1-17	1-17	1-17	1-17
	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#
	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#
	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#
	5-4	5-4#	6-361	6-361#	7-141	7-141#	7-189	7-189#	7-201	7-201#	7-203	7-203#	7-215	7-215#
	7-217	7-217#	7-229	7-229#	7-231	7-231#	7-244	7-244#	7-246	7-246#	7-259	7-259#	7-261	7-261#
	7-364	7-364#	7-366	7-366#	7-378	7-378#	7-380	7-380#	7-400	7-400#	7-402	7-402#	7-414	7-414#
	7-415	7-415#	7-428	7-428#	7-432	7-432#	7-433	7-433#	7-440	7-440#	7-443	7-443#	7-444	7-444#
	7-444#	7-444#	7-458	7-458#	7-461	7-461#	7-463	7-463#	7-471	7-471#	7-480	7-480#	7-481	7-481#
	7-665	7-665#	7-679	7-679#	7-705	7-705#	7-710	7-710#	7-711	7-711#	7-727	7-727#	7-729	7-729#
	7-731	7-731#	7-739	7-739#	7-748	7-748#	7-752	7-752#	7-754	7-754#	7-758	7-758#	7-760	7-760#
	8-4	8-4#	9-1	9-1#	9-7	9-7#	9-20	9-20#	9-45	9-45#	9-51	9-51#	9-62	9-62#
	9-79	9-79#	9-84	9-84#	9-118	9-118#	9-254	9-254#	9-259	9-259#	9-268	9-268#	9-278	9-278#
	9-303	9-303#	9-334	9-334#	9-339	9-339#	9-344	9-344#	9-349	9-349#	9-358	9-358#	9-376	9-376#
	9-379	9-379#	9-384	9-384#	9-391	9-391#	9-395	9-395#	9-400	9-400#	9-451	9-451#	9-455	9-455#
	9-488	9-488#	9-491	9-491#	9-495	9-495#	9-500	9-500#	9-529	9-529#	9-587	9-587#	9-592	9-592#
	9-609	9-609#	9-614	9-614#	9-671	9-671#	9-676	9-676#	9-686	9-686#	9-739	9-739#	9-745	9-745#
	9-750	9-750#	9-774	9-774#	9-805	9-805#	9-807	9-807#	9-815	9-815#	9-820	9-820#	9-822	9-822#
	9-827	9-827#	9-836	9-836#	9-889	9-889#	9-895	9-895#	9-900	9-900#	9-909	9-909#	9-931	9-931#
	9-944	9-944#	9-949	9-949#	9-957	9-957#	9-:05	9-:05#	9-:11	9-:11#	9-:16	9-:16#	9-:33	9-:33#
	9-:83	9-:83#	9-:95	9-:95#	9-:00	9-:00#	9-:12	9-:12#	9-:51	9-:51#	9-:63	9-:63#	9-:67	9-:67#
	9-:68	9-:68#	9-:82	9-:82#	9-<02	9-<02#	9-<03	9-<03#	9-<13	9-<13#	9-<29	9-<29#		
M\$GENB	9-20	9-20#	9-62	9-62#	9-118	9-118#	9-278	9-278#	9-303	9-303#	9-339	9-339#	9-358	9-358#
	9-376	9-376#	9-391	9-391#	9-451	9-451#	9-491	9-491#	9-529	9-529#	9-815	9-815#		

MSGETS	1-18	1-18#	3-37	3-37#	6-353	6-353#	7-132	7-132#	7-201	7-201#	7-215	7-215#	7-229	7-229#
	7-244	7-244#	7-259	7-259#	7-364	7-364#	7-378	7-378#	7-400	7-400#	7-414	7-414#	7-428	7-428#
	7-429	7-429#	7-440	7-440#	7-441	7-441#	7-458	7-458#	7-459	7-459#	7-468	7-468#	7-475	7-475#
	7-665	7-665#	7-667	7-667#	7-705	7-705#	7-727	7-727#	7-731	7-731#	7-733	7-733#	7-748	7-748#
	7-754	7-754#	7-760	7-760#	8-<68	8-<68#	9-45	9-45#	9-79	9-79#	9-254	9-254#	9-334	9-334#
	9-344	9-344#	9-384	9-384#	9-395	9-395#	9-488	9-488#	9-495	9-495#	9-587	9-587#	9-609	9-609#
	9-671	9-671#	9-739	9-739#	9-745	9-745#	9-805	9-805#	9-820	9-820#	9-822	9-822#	9-889	9-889#
	9-895	9-895#	9-931	9-931#	9-944	9-944#	9-:05	9-:05#	9-:11	9-:11#	9-:83	9-:83#	9-:95	9-:95#
	9-:51	9-:51#	9-:63	9-:63#	9-:64	9-:64#	9-:82	9-:82#	9-:98	9-:98#	9-:13	9-:13#	9-:27	9-:27#
MSGETT	9-91#	9-125#	9-144#	9-163#	9-174#	9-185#	9-193#	9-202#	9-215#	9-227#	9-232#	9-237#	9-266#	9-307#
	9-319#	9-405#	9-475#	9-482#	9-505#	9-548#	9-558#	9-580#	9-635#	9-645#	9-654#	9-659#	9-667#	9-699#
	9-711#	9-721#	9-725#	9-734#	9-758#	9-849#	9-860#	9-871#	9-876#	9-884#	9-926#	9-:84#	9-:52#	
MSGNGB	1-16	1-16#	1-17	1-17#	1-17	1-17#	1-17	1-17#	1-17	1-17#	1-17	1-17#	1-17	1-17#
	1-17	1-17#	1-17	1-17#	1-17	1-17#	1-17	1-17#	1-17	1-17#	1-17	1-17#	1-17	1-17#
	1-17	1-17#	1-17	1-17#	1-17	1-17#	1-17	1-17#	1-17	1-17#	1-17	1-17#	1-17	1-17#
	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#
	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#
	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#
	5-4	5-4#	6-361	6-361#	7-141	7-141#	7-189	7-189#	7-203	7-203#	7-217	7-217#	7-231	7-231#
	7-246	7-246#	7-261	7-261#	7-366	7-366#	7-380	7-380#	7-402	7-402#	7-415	7-415#	7-432	7-432#
	7-433	7-433#	7-433#	7-443	7-443#	7-444	7-444#	7-461	7-461#	7-463	7-463#	7-471	7-471#	7-471#
	7-480	7-480#	7-481	7-481#	7-679	7-679#	7-710	7-710#	7-71:	7-711#	7-729	7-729#	7-739	7-739#
	7-752	7-752#	7-758	7-758#	8-4	8-4#	9-1	9-1#	9-:67	9-:67#	9-:68	9-:68#	9-:02	9-:02#
	9-:03	9-:03#	9-:29	9-:29#										
MSGNIN	1-17	1-17#	1-17	1-17#	1-17	1-17#	1-17	1-17#	1-17	1-17#	1-17	1-17#	1-17	1-17#
	1-17	1-17#	1-17	1-17#	1-17	1-17#	1-17	1-17#	1-17	1-17#	1-17	1-17#	1-17	1-17#
	1-17	1-17#	1-17	1-17#	1-17	1-17#	1-17	1-17#	1-17	1-17#	1-17	1-17#	1-17	1-17#
	1-17	1-17#	1-17	1-17#	1-17	1-17#	1-17	1-17#	1-17	1-17#	1-17	1-17#	1-17	1-17#
	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#
	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#
	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#	1-17#
	7-201	7-201#	7-215	7-215#	7-229	7-229#	7-244	7-244#	7-259	7-259#	7-281	7-281#	7-281	7-281#
	7-281	7-281#	7-281	7-281#	7-302	7-302#	7-302	7-302#	7-302	7-302#	7-302	7-302#	7-302	7-302#
	7-302#	7-302#	7-302#	7-302#	7-302#	7-302#	7-335	7-335#	7-335	7-335#	7-335	7-335#	7-335	7-335#
	7-335#	7-335#	7-335#	7-335#	7-335#	7-335#	7-349	7-349#	7-349	7-349#	7-349	7-349#	7-349	7-349#
	7-349	7-349#	7-349#	7-349#	7-349#	7-349#	7-349#	7-349#	7-349#	7-349#	7-349#	7-349#	7-349#	7-349#
	7-358	7-358#	7-358	7-358#	7-358	7-358#	7-358	7-358#	7-358#	7-358#	7-358#	7-358#	7-358#	7-358#
	7-358#	7-364	7-364#	7-378	7-378#	7-400	7-400#	7-414	7-414#	7-421	7-421#	7-421	7-421#	7-421
	7-421	7-421#	7-421	7-421#	7-421	7-421#	7-421#	7-421#	7-421#	7-421#	7-421#	7-421#	7-421#	7-421#
	7-422	7-422#	7-422	7-422#	7-422	7-422#	7-422	7-422#	7-422	7-422#	7-422	7-422#	7-422	7-422#
	7-422#	7-422#	7-422#	7-422#	7-422#	7-422#	7-422#	7-422#	7-422#	7-422#	7-422#	7-422#	7-422#	7-422#
	7-424	7-424#	7-424	7-424#	7-424	7-424#	7-424	7-424#	7-424	7-424#	7-424	7-424#	7-424	7-424#
	7-424#	7-428	7-428#	7-433	7-433#	7-444	7-444#	7-463	7-463#	7-463	7-463#	7-463	7-463#	7-463
	7-463	7-463#	7-463	7-463#	7-463	7-463#	7-463	7-463#	7-463	7-463#	7-463	7-463#	7-463	7-463#
	7-463#	7-463#	7-463#	7-463#	7-463#	7-463#	7-463#	7-463#	7-463#	7-463#	7-463#	7-463#	7-463#	7-463#
	7-538	7-538#	7-538#	7-539	7-539#	7-540	7-540#	7-544	7-544#	7-544	7-544#	7-544	7-544#	7-544
	7-549	7-549#	7-549#	7-550	7-550#	7-568	7-568#	7-568	7-568#	7-568	7-568#	7-568	7-568#	7-568
	7-572#	7-573	7-573#	7-575	7-575#	7-575	7-575#	7-576	7-576#	7-589	7-589#	7-589	7-589#	7-589
	7-590	7-590#	7-624	7-624#	7-624	7-624#	7-624	7-624#	7-624	7-624#	7-624	7-624#	7-624	7-624#
	7-625	7-625#	7-625#	7-625#	7-631	7-631#	7-632	7-632#	7-651	7-651#	7-651	7-651#	7-651	7-651#
	7-651	7-651#	7-651	7-651#	7-651	7-651#	7-654	7-654#	7-654	7-654#	7-654	7-654#	7-654	7-654#
	7-654#	7-654#	7-655	7-655#	7-655	7-655#	7-655	7-655#	7-655	7-655#	7-655	7-655#	7-655	7-655#
	7-655#	7-655#	7-655#	7-655#	7-655#	7-655#	7-655#	7-655#	7-655#	7-655#	7-655#	7-655#	7-655#	7-655#
	7-657#	7-658	7-658#	7-658#	7-658	7-658#	7-659	7-659#	7-665	7-665#	7-681	7-681#	7-681	7-681#
	7-681	7-681#	7-681#	7-681#	7-681#	7-681#	7-681#	7-688	7-688#	7-688	7-688#	7-688	7-688#	7-688
	7-688#	7-688#	7-688#	7-688#	7-690	7-690#	7-690	7-690#	7-690	7-690#	7-690	7-690#	7-690	7-690#





9-516	9-516	9-516	9-516	9-516	9-516	9-516	9-516	9-516	9-516	9-516	9-516#	9-516#	9-516#
9-516#	9-516#	9-516#	9-516#	9-516#	9-516#	9-516#	9-516#	9-521	9-521#	9-527	9-527#	9-527#	9-527#
9-527	9-527	9-527	9-527	9-527	9-527	9-527	9-527	9-527#	9-527#	9-527#	9-527#	9-527#	9-527#
9-527#	9-527#	9-527#	9-529	9-529	9-529	9-529	9-529	9-529#	9-529#	9-529#	9-529#	9-529#	9-529#
9-535#	9-547	9-547	9-547	9-547	9-547#	9-547#	9-547#	9-547#	9-547#	9-547#	9-547#	9-547#	9-548#
9-557	9-557	9-557	9-557	9-557#	9-557#	9-557#	9-557#	9-557#	9-557#	9-558	9-558#	9-558#	9-558#
9-570	9-570	9-570	9-570#	9-570#	9-570#	9-570#	9-570#	9-570#	9-579	9-579	9-579#	9-579#	9-579#
9-579#	9-579#	9-579#	9-580	9-580	9-580#	9-580#	9-580#	9-587	9-587#	9-606	9-606	9-606#	9-606#
9-606#	9-606#	9-606#	9-606#	9-609	9-609#	9-609#	9-634	9-634	9-634	9-634	9-634#	9-634#	9-634#
9-634#	9-635	9-635	9-635#	9-635#	9-639	9-639#	9-644	9-644	9-644	9-644	9-644#	9-644#	9-644#
9-644#	9-644#	9-645	9-645	9-645#	9-645#	9-645#	9-651	9-651	9-651	9-651	9-651#	9-651#	9-651#
9-651#	9-651#	9-651#	9-651#	9-653	9-653	9-653	9-653	9-653#	9-653#	9-653#	9-653#	9-653#	9-654
9-654	9-654#	9-654#	9-658	9-658	9-658	9-658	9-658#	9-658#	9-658#	9-658#	9-658#	9-658#	9-658#
9-659#	9-659#	9-666	9-666	9-666	9-666	9-666#	9-666#	9-666#	9-666#	9-666#	9-666#	9-666#	9-666#
9-667#	9-668	9-668	9-668	9-668	9-668#	9-668#	9-668#	9-668#	9-668#	9-668#	9-668#	9-668#	9-668#
9-698	9-698	9-698	9-698	9-698#	9-698#	9-698#	9-698#	9-698#	9-699	9-699	9-699#	9-699#	9-705
9-705#	9-710	9-710	9-710	9-710	9-710#	9-710#	9-710#	9-710#	9-710#	9-710#	9-710#	9-710#	9-711#
9-717	9-717	9-717	9-717	9-717	9-717#	9-717#	9-717#	9-717#	9-717#	9-717#	9-717#	9-717#	9-711#
9-720	9-720	9-720#	9-720#	9-720#	9-720#	9-720#	9-721	9-721	9-721	9-721#	9-721#	9-721#	9-720
9-724	9-724#	9-724#	9-724#	9-724#	9-724#	9-724#	9-725	9-725#	9-725#	9-725#	9-725#	9-725#	9-724
9-733#	9-733#	9-733#	9-733#	9-733#	9-733#	9-733#	9-734	9-734#	9-734#	9-735	9-735	9-735#	9-733
9-735#	9-735#	9-735#	9-735#	9-735#	9-739	9-739#	9-745	9-745#	9-745#	9-758	9-758#	9-758#	9-735#
9-761	9-761	9-761	9-761	9-761	9-761	9-761	9-761	9-761#	9-761#	9-761#	9-761#	9-761#	9-761#
9-761#	9-761#	9-763	9-763	9-763	9-763	9-763	9-763	9-763#	9-763#	9-763#	9-763#	9-763#	9-765
9-765	9-765	9-765	9-765	9-765	9-765#	9-765#	9-765#	9-765#	9-765#	9-765#	9-765#	9-765#	9-767
9-767	9-767	9-767#	9-767#	9-767#	9-767#	9-767#	9-770	9-770	9-770	9-770	9-770#	9-770#	9-770#
9-770#	9-770#	9-770#	9-770#	9-772	9-772	9-772#	9-772	9-772#	9-772#	9-772#	9-772#	9-772#	9-772#
9-772	9-772	9-772#	9-772#	9-772#	9-772#	9-772#	9-772#	9-772#	9-772#	9-772#	9-772#	9-772#	9-772#
9-799	9-799#	9-805	9-805#	9-807	9-807#	9-813	9-813	9-813	9-813	9-813	9-813#	9-813#	9-813#
9-813#	9-813#	9-813#	9-815	9-815	9-815	9-815	9-815	9-815#	9-815#	9-815#	9-815#	9-815#	9-820
9-820#	9-822	9-822#	9-836	9-836#	9-848	9-848	9-848	9-848#	9-848#	9-848#	9-848#	9-848#	9-820#
9-849	9-849	9-849#	9-849#	9-854	9-854#	9-859	9-859	9-859	9-859#	9-859#	9-859#	9-859#	9-859#
9-859#	9-860	9-860	9-860#	9-860#	9-867	9-867	9-867	9-867	9-867#	9-867#	9-867#	9-867#	9-867#
9-867#	9-867#	9-867#	9-870	9-870	9-870	9-870	9-870#	9-870#	9-870#	9-870#	9-870#	9-870#	9-871
9-871#	9-871#	9-875	9-875	9-875	9-875#	9-875#	9-875#	9-875#	9-875#	9-875#	9-875#	9-875#	9-876#
9-876#	9-883	9-883	9-883	9-883	9-883#	9-883#	9-883#	9-883#	9-883#	9-884	9-884#	9-884#	9-884#
9-885	9-885	9-885	9-885	9-885#	9-885#	9-885#	9-885#	9-885#	9-889	9-889#	9-889#	9-889#	9-899
9-909#	9-925	9-925	9-925	9-925	9-925#	9-925#	9-925#	9-925#	9-925#	9-926	9-926#	9-926#	9-909
9-927	9-927	9-927	9-927	9-927#	9-927#	9-927#	9-927#	9-927#	9-931	9-931#	9-931#	9-931#	9-926#
9-942	9-942#	9-942#	9-942#	9-942#	9-942#	9-944	9-944#	9-957	9-957	9-957	9-957#	9-957#	9-942
9-985#	9-985#	9-985#	9-985#	9-985#	9-985#	9-991	9-991	9-991	9-991	9-991#	9-991#	9-991#	9-985
9-:05	9-:05#	9-:11	9-:11#	9-:33	9-:33#	9-:36	9-:36#	9-:38	9-:38#	9-:39	9-:39#	9-:39#	9-991#
9-:71	9-:71	9-:71	9-:71	9-:71#	9-:71#	9-:71#	9-:71#	9-:71#	9-:71#	9-:74	9-:74	9-:74	9-:63
9-:74	9-:74	9-:74	9-:74	9-:74#	9-:74#	9-:74#	9-:74#	9-:74#	9-:74#	9-:74	9-:74	9-:74	9-:63#
9-:78#	9-:83	9-:83#	9-:84	9-:84	9-:84	9-:84#	9-:84#	9-:84#	9-:84#	9-:78	9-:78#	9-:78#	9-:74
9-:17#	9-:18	9-:18#	9-:48	9-:48	9-:48	9-:48	9-:48#	9-:48#	9-:48#	9-:78	9-:78#	9-:78#	9-:78#
9-:52	9-:52	9-:52#	9-:63	9-:63	9-:63#	9-:63#	9-:68	9-:68#	9-:68#	9-:12	9-:12#	9-:12#	9-:15
9-:72	9-:72	9-:72#	9-:74	9-:74	9-:74	9-:74	9-:74#	9-:74#	9-:74#	9-:15	9-:15#	9-:15#	9-:17#
9-:78	9-:78	9-:78	9-:80	9-:80	9-:80	9-:80	9-:80	9-:80	9-:80	9-:48#	9-:48#	9-:48#	9-:51
9-<05	9-<05	9-<05	9-<05#	9-<07	9-<07	9-<07	9-<07#	9-<07#	9-<07#	9-:70	9-:70#	9-:70#	9-:72
9-<11	9-<11	9-<11	9-<11#	9-<13	9-<13#	9-<29	9-<29	9-<29	9-<29	9-:76	9-:76	9-:76	9-:72
MSGNLS	9-20	9-62	9-62#	9-118	9-118	9-278	9-278#	9-303	9-303#	9-:80#	9-:80#	9-:80#	9-:76#
	9-376	9-391	9-391#	9-451	9-451#	9-491	9-491#	9-529	9-529#	9-:82	9-:82#	9-:82#	9-<03#
MSGNSU	9-268	9-379	9-379#	9-455	9-455#	9-686	9-686#	9-774	9-774#	9-<09	9-<09#	9-<09#	9-<03#
	9-909	9-957	9-957#	9-:33	9-:33#	9-:12	9-:12#			9-<29#	9-<29#	9-<29#	9-<11
MSGNTA	7-201	7-215	7-215#	7-229	7-229#	7-244	7-244#	7-259	7-259#	9-339	9-339#	9-358	9-358#
										9-815	9-815#	9-836	9-836#
										9-807	9-807#	9-836	9-836#
										7-364	7-364#	7-378	7-378#

	7-400#	7-414#	7-414#	7-428#	7-428#	7-440#	7-440#	7-458#	7-458#	7-665	7-665#	7-705	7-705#
	7-727	7-731#	7-731#	7-748	7-748#	7-754	7-754#	7-760	7-760#	9-45	9-45#	9-79	9-79#
	9-254	9-334#	9-334#	9-344	9-344#	9-384	9-384#	9-395	9-395#	9-488	9-488#	9-495	9-495#
	9-587	9-609#	9-609#	9-671	9-671#	9-739	9-739#	9-745	9-745#	9-805	9-805#	9-820	9-820#
	9-822	9-889#	9-889#	9-895	9-895#	9-931	9-931#	9-944	9-944#	9-:05	9-:05#	9-:11	9-:11#
	9-:83	9-:95#	9-:95#	9-:51	9-:51#	9-:63	9-:63#	9-:82	9-:82#	9-:13	9-:13#		
MSGNTE	9-7	9-51#	9-51#	9-84	9-84#	9-259	9-259#	9-349	9-349#	9-400	9-400#	9-500	9-500#
	9-572	9-614#	9-614#	9-676	9-676#	9-750	9-750#	9-827	9-827#	9-900	9-900#	9-949	9-949#
	9-:16	9-:00	9-:00#										
M\$HAPT	1-17												
M\$HNAP	1-17												
M\$INCR	1-16	2-3	2-3#	5-4	5-4#	6-361	6-361#	7-141	7-141#	7-189	7-189	7-189#	7-189#
	7-201#	7-203	7-203#	7-203#	7-215#	7-217	7-217#	7-217#	7-217#	7-229#	7-231	7-231	7-231#
	7-231#	7-244#	7-246	7-246#	7-246#	7-259#	7-261	7-261#	7-261#	7-261#	7-302#	7-335#	7-349#
	7-358#	7-364#	7-366	7-366#	7-366#	7-378#	7-380	7-380#	7-380#	7-380#	7-400#	7-402	7-402
	7-402#	7-402#	7-414#	7-415	7-415#	7-415#	7-421#	7-422#	7-424#	7-428#	7-432	7-432#	7-433
	7-433	7-433#	7-443	7-443#	7-444	7-444#	7-444#	7-444#	7-461	7-461#	7-471	7-471#	7-471#
	7-471#	7-480#	7-480#	7-481	7-481#	7-481#	7-538#	7-539#	7-544#	7-549#	7-568#	7-572#	7-575#
	7-589#	7-624#	7-625#	7-631#	7-651#	7-654#	7-655#	7-657#	7-658#	7-659#	7-665#	7-679	7-679#
	7-679#	7-681#	7-688#	7-690#	7-691#	7-692#	7-699#	7-701#	7-702#	7-703#	7-704#	7-705#	7-710
	7-711	7-711	7-711#	7-711#	7-713#	7-715#	7-721#	7-722#	7-726#	7-727#	7-729	7-729#	7-729#
	7-731#	7-739	7-739#	7-739#	7-739#	7-752	7-752#	7-752#	7-752#	7-758	7-758#	7-758#	8-4
	8-4#	8-12#	8-14#	8-15#	8-16#	8-17#	8-18#	8-99#	8-116#	8-120#	8-151#	8-157#	8-176#
	8-321#	8-326#	8-384#	8-498#	8-504#	8-509#	8-612#	8-623#	8-627#	8-635#	8-761#	8-771#	8-774#
	8-783#	8-787#	8-905#	8-910#	8-:66#	8-:67#	8-:71#	8-:84#	8-:10#	8-:04#	8-:08#	8-:11#	8-:25#
	8-<35#	8-<38#	8-<48#	8-<50#	8-<51#	8-<52#	9-1	9-1#	9-7	9-7	9-7	9-7#	9-7#
	9-17#	9-20	9-20#	9-20#	9-29#	9-33#	9-37#	9-41#	9-45#	9-51	9-51	9-51#	9-51#
	9-51#	9-60#	9-62	9-62#	9-62#	9-72#	9-77#	9-79#	9-84	9-84	9-84	9-84#	9-84#
	9-91#	9-98#	9-102#	9-114#	9-118	9-118#	9-118#	9-124#	9-125#	9-136#	9-138#	9-143#	9-147#
	9-162#	9-163#	9-171#	9-173#	9-174#	9-182#	9-184#	9-185#	9-192#	9-193#	9-201#	9-202#	9-212#
	9-215#	9-224#	9-226#	9-227#	9-231#	9-232#	9-236#	9-237#	9-245#	9-250#	9-254#	9-259	9-259
	9-259#	9-259#	9-259#	9-266#	9-268	9-268	9-268	9-268#	9-268#	9-268#	9-276#	9-278	9-278#
	9-284#	9-300#	9-303	9-303#	9-303#	9-306#	9-307#	9-316#	9-318#	9-319#	9-328#	9-330#	9-336#
	9-339	9-339#	9-339#	9-344#	9-349	9-349	9-349	9-349#	9-349#	9-349#	9-355#	9-358	9-358#
	9-373#	9-376	9-376#	9-376#	9-379	9-379	9-379	9-379#	9-379#	9-379#	9-384#	9-389#	9-391#
	9-391#	9-395#	9-400	9-400	9-400	9-400#	9-400#	9-400#	9-405#	9-415#	9-426#	9-432#	9-449#
	9-451#	9-451#	9-455	9-455	9-455	9-455#	9-455#	9-455#	9-458#	9-474#	9-475#	9-476#	9-481#
	9-488#	9-489#	9-491	9-491#	9-491#	9-495#	9-500	9-500	9-500	9-500#	9-500#	9-500#	9-505#
	9-516#	9-521#	9-527#	9-529	9-529#	9-529#	9-535#	9-547#	9-548#	9-557#	9-558#	9-570#	9-579#
	9-587#	9-592	9-592	9-592	9-592#	9-592#	9-592#	9-606#	9-609#	9-614	9-614	9-614	9-614#
	9-614#	9-634#	9-635#	9-639#	9-644#	9-645#	9-651#	9-653#	9-654#	9-658#	9-659#	9-666#	9-667#
	9-671#	9-676	9-676	9-676	9-676#	9-676#	9-676#	9-686	9-686	9-686	9-686#	9-686#	9-686#
	9-699#	9-705#	9-710#	9-711#	9-717#	9-720#	9-721#	9-724#	9-725#	9-733#	9-734#	9-735#	9-739#
	9-750	9-750	9-750	9-750#	9-750#	9-750#	9-758#	9-761#	9-763#	9-765#	9-767#	9-770#	9-772#
	9-774	9-774	9-774#	9-774#	9-774#	9-799#	9-805#	9-807	9-807	9-807	9-807#	9-807#	9-813#
	9-815	9-815#	9-815#	9-820#	9-822#	9-827	9-827	9-827	9-827#	9-827#	9-827#	9-836	9-836
	9-836#	9-836#	9-836#	9-848#	9-849#	9-854#	9-859#	9-860#	9-867#	9-870#	9-871#	9-875#	9-876#
	9-884#	9-885#	9-889#	9-895#	9-900	9-900	9-900	9-900#	9-900#	9-900#	9-909	9-909	9-909#
	9-909#	9-909#	9-925#	9-926#	9-927#	9-931#	9-942#	9-944#	9-949	9-949	9-949	9-949#	9-949#
	9-957	9-957	9-957#	9-957#	9-957#	9-957#	9-985#	9-991#	9-:05#	9-:11#	9-:16	9-:16	9-:16#
	9-:16#	9-:16#	9-:33	9-:33	9-:33	9-:33#	9-:33#	9-:33#	9-:36#	9-:38#	9-:63#	9-:71#	9-:74#
	9-:83#	9-:84#	9-:95#	9-:00	9-:00	9-:00	9-:00#	9-:00#	9-:00#	9-:12	9-:12	9-:12	9-:12#
	9-:12#	9-:15#	9-:17#	9-:48#	9-:51#	9-:52#	9-:63#	9-:67	9-:67#	9-:68	9-:68	9-:68#	9-:68#
	9-<02#	9-<03	9-<03	9-<03#	9-<03#								
M\$LDRO	7-538	7-544#	7-544#	7-549	7-549#	7-568	7-568#	7-572	7-572#	7-575	7-575#	7-589	7-589#
	7-625	7-625#	7-658	7-658#	7-692	7-692#	7-703	7-704	7-704#	7-715	7-715#	7-722	7-722#



	7-726	7-726#	8-17	8-17#	9-415	9-415#								
MSMCHI	1-8	1-8#												
MSMLLO	1-8	1-8#												
MSPOP	1-18	1-18#	3-37	3-37#	6-353	6-353#	7-132	7-132#	7-201	7-201#	7-215	7-215#	7-229	7-229#
	7-244	7-244#	7-259	7-259#	7-364	7-364#	7-378	7-378#	7-400	7-400#	7-414	7-414#	7-428	7-428#
	7-429	7-429#	7-440	7-440#	7-441	7-441#	7-458	7-458#	7-459	7-459#	7-468	7-468#	7-475	7-475#
	7-665	7-665#	7-667	7-667#	7-705	7-705#	7-727	7-727#	7-731	7-731#	7-733	7-733#	7-748	7-748#
	7-754	7-754#	7-760	7-760#	8-68	8-68#	9-45	9-45#	9-79	9-79#	9-254	9-254#	9-334	9-334#
	9-344	9-344#	9-384	9-384#	9-395	9-395#	9-488	9-488#	9-495	9-495#	9-587	9-587#	9-609	9-609#
	9-671	9-671#	9-739	9-739#	9-745	9-745#	9-805	9-805#	9-820	9-820#	9-822	9-822#	9-889	9-889#
	9-895	9-895#	9-931	9-931#	9-944	9-944#	9-:05	9-:05#	9-:11	9-:11#	9-:83	9-:83#	9-:95	9-:95#
	9-:51	9-:51#	9-:63	9-:63#	9-:64	9-:64#	9-:82	9-:82#	9-:98	9-:98#	9-<13	9-<13#	9-<27	9-<27#
MSPRIN	7-302	7-302#	7-335	7-335#	7-349	7-349#	7-358	7-358#	7-421	7-421#	7-422	7-422#	7-424	7-424#
	7-654	7-654#	7-655	7-655#	7-657	7-657#	7-688	7-688#	7-690	7-690#	7-691	7-691#	7-699	7-699#
	7-701	7-701#	7-702	7-702#	8-14	8-14#	8-15	8-15#	8-16	8-16#	8-:66	8-:66#	8-:67	8-:67#
	8-:71	8-:71#	8-:84	8-:84#	8-<00	8-<00#	8-<04	8-<04#	8-<08	8-<08#	8-<11	8-<11#	8-<25	8-<25#
	8-<34	8-<34#	8-<35	8-<35#	8-<38	8-<38#	8-<48	8-<48#	8-<50	8-<50#	8-<51	8-<51#	8-<52	8-<52#
	9-17	9-17#	9-60	9-60#	9-98	9-98#	9-102	9-102#	9-276	9-276#	9-284	9-284#	9-336	9-336#
	9-355	9-355#	9-373	9-373#	9-389	9-389#	9-432	9-432#	9-449	9-449#	9-489	9-489#	9-516	9-516#
	9-527	9-527#	9-761	9-761#	9-763	9-763#	9-765	9-765#	9-767	9-767#	9-770	9-770#	9-772	9-772#
	9-813	9-813#												
MSPUSH	1-16	1-16#	2-3	2-3#	5-4	5-4#	6-361	6-361#	7-141	7-141#	7-189	7-189#	7-203	7-203#
	7-217	7-217#	7-231	7-231#	7-246	7-246#	7-261	7-261#	7-366	7-366#	7-380	7-380#	7-402	7-402#
	7-415	7-415#	7-432	7-432#	7-433	7-433#	7-443	7-443#	7-444	7-444#	7-461	7-461#	7-471	7-471#
	7-480	7-480#	7-481	7-481#	7-679	7-679#	7-710	7-710#	7-711	7-711#	7-729	7-729#	7-739	7-739#
	7-752	7-752#	7-758	7-758#	8-4	8-4#	9-1	9-1#	9-7	9-7#	9-51	9-51#	9-84	9-84#
	9-259	9-259#	9-268	9-268#	9-349	9-349#	9-379	9-379#	9-400	9-400#	9-455	9-455#	9-500	9-500#
	9-592	9-592#	9-614	9-614#	9-676	9-676#	9-686	9-686#	9-750	9-750#	9-774	9-774#	9-807	9-807#
	9-827	9-827#	9-836	9-836#	9-900	9-900#	9-909	9-909#	9-949	9-949#	9-957	9-957#	9-:16	9-:16#
	9-:33	9-:33#	9-:00	9-:00#	9-:12	9-:12#	9-:67	9-:67#	9-:68	9-:68#	9-<02	9-<02#	9-<03	9-<03#
MSPUT	7-302	7-302	7-302	7-302	7-302	7-302#	7-335	7-335	7-335	7-335	7-335#	7-349	7-349	7-349
	7-349	7-349	7-349	7-349	7-349	7-349#	7-358	7-358	7-358	7-358	7-358#	7-358	7-358#	7-421
	7-421	7-421	7-421	7-421	7-421	7-421#	7-422	7-422	7-422	7-422	7-422#	7-422	7-422	7-422
	7-422	7-422#	7-424	7-424	7-424	7-424#	7-424	7-424	7-424	7-424	7-424#	7-624	7-624	7-624
	7-624	7-624#	7-654	7-654	7-654	7-654#	7-655	7-655	7-655	7-655	7-655#	7-655	7-655#	7-657
	7-657	7-657#	7-681	7-681	7-681	7-681#	7-688	7-688	7-688	7-688	7-688#	7-690	7-690	7-690
	7-690	7-690	7-690	7-690#	7-691	7-691#	7-699	7-699	7-699	7-699	7-699#	7-701	7-701	7-701
	7-701	7-701	7-701	7-701#	7-702	7-702#	7-713	7-713	7-713	7-713	7-713#	7-713	7-713#	8-14
	8-14	8-14	8-14#	8-15	8-15	8-15#	8-15	8-15	8-15	8-15	8-15#	8-16	8-16#	8-:66
	8-:66	8-:66	8-:66#	8-:67	8-:67	8-:67#	8-:67	8-:67	8-:67	8-:67	8-:67#	8-:71	8-:71#	8-:84
	8-:84	8-:84	8-:84#	8-<00	8-<00	8-<00#	8-<00	8-<00	8-<00	8-<00	8-<00#	8-<04	8-<04#	8-<08
	8-<08	8-<08	8-<08	8-<08	8-<08	8-<08#	8-<08	8-<08	8-<08	8-<08	8-<08#	8-<11	8-<11#	8-<11
	8-<11	8-<11	8-<11#	8-<25	8-<25	8-<25#	8-<25	8-<25	8-<25	8-<25	8-<25#	8-<34	8-<34#	8-<35
	8-<35	8-<35	8-<35#	8-<38	8-<38	8-<38#	8-<38	8-<38	8-<38	8-<38	8-<38#	8-<48	8-<48#	8-<48
	8-<48	8-<48#	8-<50	8-<50	8-<50	8-<50#	8-<50	8-<50	8-<50	8-<50	8-<50#	8-<51	8-<51#	8-<51
	8-<51	8-<51	8-<51	8-<51	8-<51#	8-<52	8-<52	8-<52	8-<52	8-<52	8-<52#	8-<52	8-<52	8-<52
	8-<52#	9-17	9-17	9-17	9-17	9-17#	9-17	9-17	9-17	9-17	9-17#	9-60	9-60	9-60
	9-60	9-60	9-60	9-60	9-60#	9-98	9-98	9-98	9-98	9-98	9-98#	9-98	9-98	9-98#
	9-102	9-102	9-102	9-102	9-102	9-102#	9-102	9-102	9-102	9-102	9-102#	9-114	9-114	9-114#
	9-136	9-136	9-136	9-136	9-136#	9-147	9-147	9-147	9-147	9-147	9-147#	9-171	9-171	9-171#
	9-171#	9-182	9-182	9-182	9-182	9-182#	9-212	9-212	9-212	9-212	9-212#	9-224	9-224	9-224#
	9-224	9-224#	9-245	9-245	9-245	9-245#	9-245	9-245	9-245	9-245#	9-276	9-276	9-276	9-276#
	9-276	9-276#	9-284	9-284	9-284	9-284#	9-284	9-284	9-284	9-284	9-284#	9-300	9-300	9-300#
	9-300	9-300#	9-316	9-316	9-316	9-316#	9-328	9-328	9-328	9-328	9-328#	9-328	9-328#	9-336
	9-336	9-336#	9-336	9-336	9-336	9-336#	9-355	9-355	9-355	9-355	9-355#	9-355	9-355#	9-355
	9-355	9-355#	9-373	9-373	9-373	9-373#	9-373	9-373	9-373	9-373#	9-389	9-389	9-389	9-389#



	9-516	9-516	9-516#	9-516#	9-516#	9-516#	9-516#	9-516#	9-516#	9-516#	9-527	9-527	9-527	9-527	
	9-527	9-527	9-527	9-527	9-527#	9-527#	9-527#	9-527#	9-527#	9-527#	9-527#	9-527#	9-527#	9-651	9-651
	9-651	9-651	9-651#	9-651#	9-651#	9-651#	9-717	9-717	9-717	9-717	9-717#	9-717#	9-717#	9-717#	9-717#
	9-761	9-761	9-761	9-761	9-761	9-761	9-761#	9-761#	9-761#	9-761#	9-761#	9-761#	9-761#	9-763	9-763
	9-763	9-763#	9-763#	9-763#	9-765	9-765	9-765	9-765#	9-765#	9-765#	9-765	9-767	9-767	9-767	9-767#
	9-767#	9-767#	9-770	9-770	9-770	9-770#	9-770#	9-770#	9-770#	9-772	9-772	9-772	9-772	9-772	9-772
	9-772	9-772	9-772#	9-772#	9-772#	9-772#	9-772#	9-772#	9-772#	9-772#	9-772#	9-772#	9-772#	9-772	9-772
	9-813#	9-813#	9-867	9-867	9-867	9-867	9-867#	9-867#	9-867#	9-867#	9-813	9-813	9-813	9-813	9-813#
MSRADI	9-20	9-20#	9-62	9-62#	9-118	9-118#	9-278	9-278#	9-303	9-303#	9-339	9-339#	9-358	9-358#	9-358#
	9-376	9-376#	9-391	9-391#	9-451	9-451#	9-491	9-491#	9-529	9-529#	9-815	9-815#	9-70	9-70#	9-70#
	9-;72	9-;72#	9-;74	9-;74#	9-;76	9-;76#	9-;78	9-;78#	9-;80	9-;80#	9-<05	9-<05#	9-<07	9-<07#	9-<07#
	9-<09	9-<09#	9-<11	9-<11#											
MSRNRO	7-589	7-589#	9-415	9-415#											
M\$SETS	1-16	1-16#	2-3	2-3#	5-4	5-4#	6-361	6-361#	7-141	7-141#	7-189	7-189#	7-203	7-203#	7-203#
	7-217	7-217#	7-231	7-231#	7-246	7-246#	7-261	7-261#	7-366	7-366#	7-380	7-380#	7-402	7-402#	7-402#
	7-415	7-415#	7-432	7-432#	7-433	7-433#	7-443	7-443#	7-444	7-444#	7-461	7-461#	7-471	7-471#	7-471#
	7-480	7-480#	7-481	7-481#	7-679	7-679#	7-710	7-710#	7-711	7-711#	7-729	7-729#	7-739	7-739#	7-739#
	7-752	7-752#	7-758	7-758#	8-4	8-4#	9-1	9-1#	9-7	9-7#	9-51	9-51#	9-84	9-84#	9-84#
	9-259	9-259#	9-268	9-268#	9-349	9-349#	9-379	9-379#	9-400	9-400#	9-455	9-455#	9-500	9-500#	9-500#
	9-592	9-592#	9-614	9-614#	9-676	9-676#	9-686	9-686#	9-750	9-750#	9-774	9-774#	9-807	9-807#	9-807#
	9-827	9-827#	9-836	9-836#	9-900	9-900#	9-909	9-909#	9-949	9-949#	9-957	9-957#	9-;16	9-;16#	9-;16#
	9-;33	9-;33#	9-;00	9-;00#	9-;12	9-;12#	9-;67	9-;67#	9-;60	9-;68#	9-<02	9-<02#	9-<03	9-<03#	9-<03#
M\$SVC	7-201	7-201#	7-215	7-215#	7-229	7-229#	7-244	7-244#	7-259	7-259#	7-302	7-302#	7-335	7-335#	7-335#
	7-349	7-349#	7-358	7-358#	7-364	7-364#	7-378	7-378#	7-400	7-400#	7-414	7-414#	7-421	7-421#	7-421#
	7-422	7-422#	7-424	7-424#	7-428	7-428#	7-538	7-538#	7-539	7-539#	7-544	7-544#	7-549	7-549#	7-549#
	7-568	7-568#	7-572	7-572#	7-575	7-575#	7-589	7-589#	7-624	7-624#	7-625	7-625#	7-631	7-631#	7-631#
	7-651	7-651#	7-654	7-654#	7-655	7-655#	7-657	7-657#	7-658	7-658#	7-659	7-659#	7-665	7-665#	7-665#
	7-681	7-681#	7-688	7-688#	7-690	7-690#	7-691	7-691#	7-692	7-692#	7-699	7-699#	7-701	7-701#	7-701#
	7-702	7-702#	7-703	7-703#	7-704	7-704#	7-705	7-705#	7-713	7-713#	7-715	7-715#	7-721	7-721#	7-721#
	7-722	7-722#	7-726	7-726#	7-727	7-727#	7-731	7-731#	8-12	8-12#	8-14	8-14#	8-15	8-15#	8-15#
	8-16	8-16#	8-17	8-17#	8-18	8-18#	8-99	8-99#	8-116	8-116#	8-120	8-120#	8-151	8-151#	8-151#
	8-170	8-176	8-321	8-326	8-384	8-498	8-504	8-504#	8-509	8-612	8-623	8-627	8-635	8-761	8-761
	8-761#	8-771	8-774	8-774#	8-779	8-779#	8-783	8-787	8-905	8-910	8-;66	8-;66#	8-;67	8-;67#	8-;67#
	8-;71	8-;71#	8-;84	8-;84#	8-<00	8-<00#	8-<04	8-<04#	8-<08	8-<08#	8-<11	8-<11#	8-<25	8-<25#	8-<25#
	8-<34	8-<34#	8-<35	8-<35#	8-<38	8-<38#	8-<48	8-<48#	8-<50	8-<50#	8-<51	8-<51#	8-<52	8-<52#	8-<52#
	9-17	9-17#	9-20	9-20#	9-29	9-33	9-37	9-41	9-45	9-45#	9-60	9-60#	9-62	9-62#	9-62#
	9-72	9-77	9-79	9-79#	9-91	9-91#	9-98	9-98#	9-102	9-102#	9-114	9-114#	9-118	9-118#	9-118#
	9-124	9-125	9-125#	9-136	9-136#	9-138	9-143	9-144	9-144#	9-147	9-147#	9-162	9-162#	9-163	9-163#
	9-171	9-171#	9-173	9-174	9-174#	9-182	9-182#	9-184	9-185	9-185#	9-192	9-192#	9-193	9-193#	9-201
	9-202	9-202#	9-212	9-212#	9-214	9-215	9-215#	9-224	9-224#	9-226	9-227	9-227#	9-231	9-232	9-232
	9-232#	9-236	9-237	9-237#	9-245	9-245#	9-250	9-254	9-254#	9-266	9-266#	9-268	9-268#	9-276	9-276
	9-276#	9-278	9-278#	9-284	9-284#	9-300	9-300#	9-303	9-303#	9-306	9-307	9-307#	9-316	9-316#	9-316#
	9-318	9-319	9-319#	9-328	9-328#	9-330	9-334	9-334#	9-336	9-336#	9-339	9-339#	9-344	9-344#	9-344#
	9-355	9-355#	9-358	9-358#	9-373	9-373#	9-376	9-376#	9-379	9-379#	9-384	9-384#	9-389	9-389#	9-389#
	9-391	9-391#	9-395	9-395#	9-405	9-405#	9-415	9-415#	9-426	9-426#	9-432	9-432#	9-449	9-449#	9-449#
	9-451	9-451#	9-455	9-455#	9-458	9-458#	9-474	9-475	9-475#	9-476	9-476#	9-481	9-482	9-482#	9-482#
	9-488	9-488#	9-489	9-489#	9-491	9-491#	9-495	9-495#	9-505	9-505#	9-508	9-508#	9-516	9-516#	9-516#
	9-521	9-521#	9-527	9-527#	9-529	9-529#	9-535	9-535#	9-547	9-548	9-548#	9-557	9-558	9-558#	9-558#
	9-570	9-579	9-580	9-580#	9-587	9-587#	9-606	9-609	9-609#	9-634	9-635	9-635#	9-639	9-639#	9-639#
	9-644	9-645	9-645#	9-651	9-651#	9-653	9-654	9-654#	9-658	9-659	9-659#	9-666	9-667	9-667#	9-667#
	9-668	9-671	9-671#	9-686	9-686#	9-698	9-699	9-699#	9-705	9-705#	9-710	9-711	9-711#	9-717	9-717
	9-717#	9-720	9-721	9-721#	9-724	9-725	9-725#	9-733	9-734	9-734#	9-735	9-739	9-739#	9-745	9-745
	9-745#	9-758	9-758#	9-761	9-761#	9-763	9-763#	9-765	9-765#	9-767	9-767#	9-770	9-770#	9-772	9-772
	9-772#	9-774	9-774#	9-799	9-799#	9-805	9-805#	9-807	9-807#	9-813	9-813#	9-815	9-815#	9-820	9-820
	9-820#	9-822	9-822#	9-836	9-836#	9-848	9-849	9-849#	9-854	9-854#	9-859	9-860	9-860#	9-867	9-867
	9-867#	9-870	9-871	9-871#	9-875	9-876	9-876#	9-883	9-884	9-884#	9-885	9-889	9-889#	9-895	9-895

	9-895#	9-909	9-909#	9-925	9-926	9-926#	9-927	9-931	9-931#	9-942	9-944	9-944#	9-957	9-957#
	9-985	9-991	9-:05	9-:05#	9-:11	9-:11#	9-:33	9-:33#	9-:36	9-:36#	9-:38	9-:38#	9-:63	9-:63#
	9-:71	9-:74	9-:74#	9-:78	9-:83	9-:83#	9-:84	9-:84#	9-:95	9-:95#	9-:12	9-:12#	9-:15	9-:15#
MSTLAB	9-:17	9-:17#	9-:48	9-:51	9-:51#	9-:52	9-:52#	9-:63	9-:63#					
	7-201#	7-215#	7-229#	7-244#	7-259#	7-302#	7-335#	7-349#	7-358#	7-364#	7-378#	7-400#	7-414#	7-421#
	7-422#	7-424#	7-428#	7-538#	7-539#	7-544#	7-549#	7-568#	7-572#	7-575#	7-589#	7-624#	7-625#	7-631#
	7-651#	7-654#	7-655#	7-657#	7-658#	7-659#	7-665#	7-681#	7-688#	7-690#	7-691#	7-692#	7-699#	7-701#
	7-702#	7-703#	7-704#	7-705#	7-713#	7-715#	7-721#	7-722#	7-726#	7-727#	7-731#	8-12#	8-14#	8-15#
	8-16#	8-17#	8-18#	8-99#	8-116#	8-120#	8-151#	8-157#	8-170#	8-176#	8-321#	8-326#	8-384#	8-498#
	8-504#	8-509#	8-612#	8-623#	8-627#	8-635#	8-761#	8-771#	8-774#	8-779#	8-783#	8-787#	8-905#	8-910#
	8-:66#	8-:67#	8-:71#	8-:84#	8-:00#	8-:04#	8-:08#	8-:11#	8-:25#	8-:34#	8-:35#	8-:38#	8-:48#	8-:50#
	8-:51#	8-:52#	9-17#	9-20#	9-29#	9-33#	9-37#	9-41#	9-45#	9-60#	9-62#	9-72#	9-77#	9-79#
	9-91#	9-98#	9-102#	9-114#	9-118#	9-124#	9-125#	9-136#	9-138#	9-143#	9-144#	9-147#	9-162#	9-163#
	9-171#	9-173#	9-174#	9-182#	9-184#	9-185#	9-192#	9-193#	9-201#	9-202#	9-212#	9-214#	9-215#	9-224#
	9-226#	9-227#	9-231#	9-232#	9-236#	9-237#	9-245#	9-250#	9-254#	9-266#	9-268#	9-276#	9-278#	9-284#
	9-300#	9-303#	9-306#	9-307#	9-316#	9-318#	9-319#	9-328#	9-330#	9-334#	9-336#	9-339#	9-344#	9-355#
	9-358#	9-373#	9-376#	9-379#	9-384#	9-389#	9-391#	9-395#	9-405#	9-415#	9-426#	9-432#	9-449#	9-451#
	9-455#	9-458#	9-474#	9-475#	9-476#	9-481#	9-482#	9-488#	9-489#	9-491#	9-495#	9-505#	9-508#	9-516#
	9-521#	9-527#	9-529#	9-535#	9-547#	9-548#	9-557#	9-558#	9-570#	9-579#	9-580#	9-587#	9-606#	9-609#
	9-634#	9-635#	9-639#	9-644#	9-645#	9-651#	9-653#	9-654#	9-658#	9-659#	9-666#	9-667#	9-668#	9-671#
	9-686#	9-698#	9-699#	9-705#	9-710#	9-711#	9-717#	9-720#	9-721#	9-724#	9-725#	9-733#	9-734#	9-735#
	9-739#	9-745#	9-758#	9-761#	9-763#	9-765#	9-767#	9-770#	9-772#	9-774#	9-799#	9-805#	9-807#	9-813#
	9-815#	9-820#	9-822#	9-836#	9-848#	9-849#	9-854#	9-859#	9-860#	9-867#	9-870#	9-871#	9-875#	9-876#
	9-883#	9-884#	9-885#	9-889#	9-895#	9-909#	9-925#	9-926#	9-927#	9-931#	9-942#	9-944#	9-957#	9-985#
	9-991#	9-:05#	9-:11#	9-:33#	9-:36#	9-:38#	9-:63#	9-:71#	9-:74#	9-:78#	9-:83#	9-:84#	9-:95#	9-:12#
MSTSTL	9-:15#	9-:17#	9-:48#	9-:51#	9-:52#	9-:63#								
	7-201	7-201#	7-215	7-215#	7-229	7-229#	7-244	7-244#	7-259	7-259#	7-302	7-302#	7-335	7-335#
	7-349	7-349#	7-358	7-358#	7-364	7-364#	7-378	7-378#	7-400	7-400#	7-414	7-414#	7-421	7-421#
	7-422	7-422#	7-424	7-424#	7-428	7-428#	7-538	7-538#	7-539	7-539#	7-544	7-544#	7-549	7-549#
	7-568	7-568#	7-572	7-572#	7-575	7-575#	7-589	7-589#	7-624	7-624#	7-625	7-625#	7-631	7-631#
	7-651	7-651#	7-654	7-654#	7-655	7-655#	7-657	7-657#	7-658	7-658#	7-659	7-659#	7-665	7-665#
	7-681	7-681#	7-688	7-688#	7-690	7-690#	7-691	7-691#	7-692	7-692#	7-699	7-699#	7-701	7-701#
	7-702	7-702#	7-703	7-703#	7-704	7-704#	7-705	7-705#	7-713	7-713#	7-715	7-715#	7-721	7-721#
	7-722	7-722#	7-726	7-726#	7-727	7-727#	7-731	7-731#	8-12	8-12#	8-14	8-14#	8-15	8-15#
	8-16	8-16#	8-17	8-17#	8-18	8-18#	8-99	8-99#	8-116	8-116#	8-120	8-120#	8-120#	8-151
	8-151#	8-157	8-157#	8-157#	8-170	8-170#	8-170#	8-176	8-176#	8-176#	8-176#	8-321	8-321#	8-326
	8-326#	8-326#	8-384	8-384#	8-384#	8-498	8-498#	8-498#	8-504	8-504#	8-509	8-509#	8-509#	8-612
	8-612#	8-612#	8-623	8-623#	8-623#	8-627	8-627#	8-627#	8-635	8-635#	8-635#	8-635#	8-761	8-771
	8-771#	8-771#	8-774	8-774#	8-779	8-779#	8-783	8-783#	8-783#	8-787	8-787#	8-787#	8-905	8-905#
	8-905#	8-910	8-910#	8-910#	8-:66	8-:66#	8-:67	8-:67#	8-:71	8-:71#	8-:84	8-:84#	8-:00	8-:00#
	8-:04	8-:04#	8-:08	8-:08#	8-:11	8-:11#	8-:25	8-:25#	8-:34	8-:34#	8-:35	8-:35#	8-:38	8-:38#
	8-:48	8-:48#	8-:50	8-:50#	8-:51	8-:51#	8-:52	8-:52#	9-17	9-17#	9-20	9-20#	9-29	9-29#
	9-29#	9-33	9-33#	9-33#	9-37	9-37#	9-37#	9-37#	9-41	9-41#	9-41#	9-45	9-45#	9-60#
	9-62	9-62#	9-72	9-72#	9-72#	9-77	9-77#	9-77#	9-79	9-79#	9-91	9-91#	9-98	9-98#
	9-102	9-102#	9-114	9-114#	9-118	9-118#	9-124	9-124#	9-124#	9-125	9-125#	9-136	9-136#	9-138
	9-138#	9-138#	9-143	9-143#	9-143#	9-144	9-144#	9-147	9-147#	9-162	9-162#	9-162#	9-163	9-163#
	9-171	9-171#	9-173	9-173#	9-173#	9-174	9-174#	9-182	9-182#	9-184	9-184#	9-184#	9-185	9-185#
	9-192	9-192#	9-192#	9-193	9-193#	9-201	9-201#	9-201#	9-202	9-202#	9-212	9-212#	9-214	9-214#
	9-214#	9-215	9-215#	9-224	9-224#	9-226	9-226#	9-226#	9-227	9-227#	9-231	9-231#	9-231#	9-232
	9-232#	9-236	9-236#	9-236#	9-237	9-237#	9-245	9-245#	9-250	9-250#	9-254	9-254#	9-266	9-266#
	9-266#	9-268	9-268#	9-276	9-276#	9-278	9-278#	9-284	9-284#	9-300	9-300#	9-303	9-303#	9-306
	9-306#	9-306#	9-307	9-307#	9-316	9-316#	9-318	9-318#	9-318#	9-319	9-319#	9-328	9-328#	9-330
	9-330#	9-330#	9-334	9-334#	9-336	9-336#	9-339	9-339#	9-344	9-344#	9-355	9-355#	9-358	9-358#
	9-373	9-373#	9-376	9-376#	9-379	9-379#	9-384	9-384#	9-389	9-389#	9-391	9-391#	9-395	9-395#
	9-405	9-405#	9-415	9-415#	9-426	9-426#	9-432	9-432#	9-449	9-449#	9-451	9-451#	9-455	9-455#
	9-458	9-458#	9-474	9-474#	9-474#	9-475	9-475#	9-476	9-476#	9-481	9-481#	9-481#	9-482	9-482#

	9-488	9-488#	9-489	9-489#	9-491	9-491#	9-495	9-495#	9-505	9-505#	9-508	9-508#	9-516	9-516#
	9-521	9-521#	9-527	9-527#	9-529	9-529#	9-535	9-535#	9-547	9-547#	9-548	9-548#	9-548#	9-557
	9-557#	9-557#	9-558	9-558#	9-570	9-570#	9-570#	9-579	9-579#	9-579#	9-580	9-580#	9-587	9-587#
	9-606	9-606#	9-606#	9-609	9-609#	9-634	9-634#	9-634#	9-635	9-635#	9-639	9-639#	9-644	9-644#
	9-644#	9-645	9-645#	9-651	9-651#	9-653	9-653#	9-653#	9-654	9-654#	9-658	9-658#	9-658#	9-659
	9-659#	9-666	9-666#	9-666#	9-667	9-667#	9-668	9-668#	9-668#	9-671	9-671#	9-686	9-686#	9-698
	9-698#	9-698#	9-699	9-699#	9-705	9-705#	9-710	9-710#	9-710#	9-711	9-711#	9-717	9-717#	9-720
	9-720#	9-720#	9-721	9-721#	9-724	9-724#	9-724#	9-725	9-725#	9-733	9-733#	9-733#	9-734	9-734#
	9-735	9-735#	9-735#	9-739	9-739#	9-745	9-745#	9-758	9-758#	9-761	9-761#	9-763	9-763#	9-765
	9-765#	9-767	9-767#	9-770	9-770#	9-772	9-772#	9-774	9-774#	9-799	9-799#	9-805	9-805#	9-807
	9-807#	9-813	9-813#	9-815	9-815#	9-820	9-820#	9-822	9-822#	9-836	9-836#	9-848	9-848#	9-848#
	9-849	9-849#	9-854	9-854#	9-859	9-859#	9-859#	9-860	9-860#	9-867	9-867#	9-870	9-870#	9-870#
	9-871	9-871#	9-875	9-875#	9-875#	9-876	9-876#	9-883	9-883#	9-883#	9-884	9-884#	9-885	9-885#
	9-885#	9-889	9-889#	9-895	9-895#	9-909	9-909#	9-925	9-925#	9-925#	9-926	9-926#	9-927	9-927#
	9-927#	9-931	9-931#	9-942	9-942#	9-942#	9-944	9-944#	9-957	9-957#	9-985	9-985#	9-985#	9-991
	9-991#	9-991#	9-:05	9-:05#	9-:11	9-:11#	9-:33	9-:33#	9-:36	9-:36#	9-:38	9-:38#	9-:63	9-:63#
	9-:71	9-:71#	9-:71#	9-:74	9-:74#	9-:74#	9-:78	9-:78#	9-:83	9-:83#	9-:84	9-:84#	9-:95	9-:95#
	9-:12	9-:12#	9-:15	9-:15#	9-:17	9-:17#	9-:48	9-:48#	9-:48#	9-:51	9-:51#	9-:52	9-:52#	9-:63
	9-:63#													
MSWORD	1-17	1-17#	7-463	7-463	7-463	7-463	7-463	7-463	7-463	7-463	7-463	7-463	7-463	7-463
	7-463	7-463	7-463	7-463	7-463	7-463#	8-120	8-120	8-120	8-120#	8-157	8-157	8-157	8-157#
	8-170	8-170	8-170	8-170#	8-176	8-176	8-176	8-176#	8-327	8-327	8-321	8-321#	8-326	8-326#
	8-326	8-326#	8-384	8-384	8-384	8-384#	8-498	8-498	8-498	8-498#	8-509	8-509	8-509	8-509#
	8-612	8-612	8-612	8-612#	8-623	8-623	8-623	8-623#	8-627	8-627	8-627	8-627#	8-635	8-635#
	8-635	8-635#	8-771	8-771	8-771	8-771#	8-783	8-783	8-783	8-783#	8-787	8-787	8-787	8-787#
	8-905	8-905	8-905	8-905#	8-910	8-910	8-910	8-910#	9-20	9-20	9-20#	9-20#	9-29	9-29#
	9-29	9-29#	9-33	9-33	9-33	9-33#	9-37	9-37	9-37	9-37#	9-41	9-41	9-41	9-41#
	9-62	9-62	9-62#	9-62#	9-72	9-72	9-72	9-72#	9-77	9-77	9-77	9-77#	9-91	9-91#
	9-118	9-118#	9-118#	9-124	9-124	9-124	9-124#	9-125#	9-138	9-138	9-138	9-138#	9-143	9-143#
	9-143	9-143#	9-144#	9-162	9-162	9-162	9-162#	9-163#	9-173	9-173	9-173	9-173#	9-174#	9-184
	9-184	9-184	9-184#	9-185#	9-192	9-192	9-192	9-192#	9-193#	9-201	9-201	9-201	9-201#	9-202#
	9-214	9-214	9-214	9-214#	9-215#	9-226	9-226	9-226	9-226#	9-227#	9-231	9-231	9-231	9-231#
	9-232#	9-236	9-236	9-236	9-236#	9-237#	9-250	9-250	9-250	9-250#	9-266#	9-278	9-278	9-278#
	9-278#	9-303	9-303	9-303#	9-303#	9-306	9-306	9-306	9-306#	9-307#	9-318	9-318	9-318	9-318#
	9-319#	9-330	9-330	9-330	9-330#	9-339	9-339	9-339#	9-339#	9-358	9-358	9-358#	9-358#	9-376
	9-376	9-376#	9-376#	9-391	9-391	9-391#	9-391#	9-405#	9-451	9-451	9-451#	9-451#	9-474	9-474#
	9-474	9-474#	9-475#	9-481	9-481	9-481	9-481#	9-482#	9-491	9-491	9-491#	9-491#	9-505#	9-529
	9-529	9-529#	9-529#	9-547	9-547	9-547	9-547#	9-548#	9-557	9-557	9-557	9-557#	9-558#	9-570
	9-570	9-570	9-570#	9-579	9-579	9-579	9-579#	9-580#	9-606	9-606	9-606	9-606#	9-634	9-634#
	9-634	9-634#	9-635#	9-644	9-644	9-644	9-644#	9-645#	9-653	9-653	9-653	9-653#	9-654#	9-658
	9-658	9-658	9-658#	9-659#	9-666	9-666	9-666	9-666#	9-667#	9-668	9-668	9-668	9-668#	9-698
	9-698	9-698	9-698#	9-699#	9-710	9-710	9-710	9-710#	9-711#	9-720	9-720	9-720	9-720#	9-721#
	9-724	9-724	9-724	9-724#	9-725#	9-733	9-733	9-733	9-733#	9-734#	9-735	9-735	9-735	9-735#
	9-758#	9-815	9-815	9-815#	9-815#	9-848	9-848	9-848	9-848#	9-849#	9-859	9-859	9-859	9-859#
	9-860#	9-870	9-870	9-870	9-870#	9-871#	9-875	9-875	9-875	9-875#	9-876#	9-883	9-883	9-883
	9-883#	9-884#	9-885	9-885	9-885	9-885#	9-925	9-925	9-925	9-925#	9-926#	9-927	9-927	9-927
	9-927#	9-942	9-942	9-942	9-942#	9-985	9-985	9-985	9-985#	9-991	9-991	9-991	9-991#	9-:71
	9-:71	9-:71	9-:71#	9-:78	9-:78	9-:78	9-:78	9-:78#	9-:48	9-:48	9-:48#	9-:70	9-:70#	9-:72
	9-:72#	9-:74	9-:74#	9-:76	9-:76	9-:76#	9-:78	9-:78#	9-:80	9-:80#	9-:80#	9-:80#	9-:80#	9-:80#
	9-<09#	9-<11	9-<11#	9-<29	9-<29									
MANUAL	7-539	7-631												
POINTE	1-14													
PRINTB	7-302	7-335	7-349	7-358	7-421	7-422	7-424	8-:66	8-:67	8-:71	8-:84	8-<00	8-<04	8-<08
	8-<11	8-<25	8-<34	8-<35	8-<38	8-<48	8-<50	8-<51	8-<52					
PRINTF	7-654	7-655	7-657	7-688	7-690	7-691	7-699	7-701	7-702	8-14	8-15	8-16	9-17	9-60
	9-98	9-102	9-276	9-284	9-336	9-355	9-373	9-389	9-432	9-449	9-489	9-516	9-527	9-761

	9-763	9-765	9-767	9-770	9-772	9-813								
REDEF	7-544	7-549	7-568	7-572	7-575									
SETPRI	7-538	7-625	7-715											
SETVEC	7-624	7-681	7-713	9-114	9-136	9-147	9-171	9-182	9-212	9-224	9-245	9-300	9-316	9-328
	9-426	9-458	9-476	9-508	9-651	9-717	9-867							
SVC	1-6#	1-8												
TIMDLV	4-28#	9-114	9-136	9-147	9-171	9-182	9-212	9-224	9-245	9-300	9-316	9-328	9-426	9-458
	9-476	9-508	9-651	9-717	9-867									
WAITMS	4-7#	7-651	7-721	8-99	8-116	8-151	8-504	8-779	9-:74					
WAITUS	4-21#	7-281	8-139	8-100	8-316	8-380	8-493	8-605	8-768	9-468				
XFER	9-91#	9-125#	9-144#	9-165#	9-174#	9-185#	9-193#	9-202#	9-215#	9-227#	9-232#	9-237#	9-266#	9-307#
	9-319#	9-405#	9-475#	9-482#	9-505#	9-548#	9-558#	9-580#	9-635#	9-645#	9-654#	9-659#	9-667#	9-699#
	9-711#	9-721#	9-725#	9-734#	9-758#	9-849#	9-860#	9-871#	9-876#	9-884#	9-926#			