

.REM 3

## IDENTIFICATION

PRODUCT CODE: AC F843B-MC  
PRODUCT NAME: CZRLNBO RL01/02 DRIVE TEST 3  
DATE CREATED: 5-JAN-79  
REVISED: 21-JAN-83  
MAINTAINER: DIAGNOSTIC ENGINEERING  
AUTHORS: D. DEKNIS, C. CAMPBELL

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

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

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

COPYRIGHT (C) 1979,1983 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 RUN TIME
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
3.1.2	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

## GENERAL INFORMATION

## 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 AT ADDRESS 200, CONTROL GOES FIRST TO THE SUPERVISOR PORTION, WHICH WILL ASK CERTAIN "HARD CORE" QUESTIONS ABOUT THE ENVIRONMENT. THEN IT WILL ENTER COMMAND MODE, INDICATED BY A PROMPT CHARACTER (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 9K 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 RL01/02 DISK DRIVES RL11/RLV11 CONTROLLERS (4 DRIVES PER CONTROLLER). THE ENTIRE PROGRAM IS RUN ON THE FIRST DRIVE BEFORE STARTING ON THE SECOND. THE PROGRAM STARTS BY TESTING THE SIMPLEST FUNCTIONS FIRST USING THE LOGIC TESTED IN EARLIER TESTS TO TEST MORE COMPLEX FUNCTIONS.

THIS PROGRAM FIRST TESTS THE RL01/02 SEEK TIMING. DATA TRANSFERS ARE DONE AFTER THE SEEK TIMING TEST. THE FIRST DATA TRANSFER IS READING OF THE BAD SECTOR FILES WHICH ARE STORED AND USED LATER TO PREVENT TESTING ON BAD SECTORS. FOLLOWING DATA READ AND WRITE TESTING, THE PROGRAM TESTS FOR OVERWRITE PROBLEMS AND ADJACENT CYLINDER INTERFERENCE.

THE WRITE LOCK DATA PROTECTION TEST IS PERFORMED IF MANUAL INTERVENTION IS REQUESTED.

## 1.1.3      DIAGNOSTIC RUN TIME

THIS DIAGNOSTIC TAKES 4 MINUTES TO RUN THE FIRST PASS AND 28.5 MINUTES FOR THE SECOND PASS.

## 1.2        SYSTEM REQUIREMENTS

### 1.2.1 HARDWARE REQUIREMENTS

-----

- PDP-11/LSI-11 PROCESSOR WITH 16K OR MORE OF MEMORY
- CONSOLF DEVICE (LA30,LA36,VT50,ETC.)
- 1 OR 2 RL11/RLV11 CONTROLLER(S) WITH:
  - 1 - 8 RLO1 DRIVES WITH RLO1K CARTRIDGES CONTAINING A 'BAD SECTOR FILE'
  - 1 - 8 RLO2 DRIVES WITH RLO2K CARTRIDGES CONTAINING A 'BAD SECTOR FILE'
- KW11-P CLOCK (REQUIRED TO PERFORM TESTS 1 AND 4)
- LINE PRINTER (OPTIONAL)

### 1.2.2 SOFTWARE REQUIREMENTS

-----

CZRLJBO RLO1/02 DRIVE TEST PART 2 (FORMERLY CZRLDBO)

### 1.3 RELATED DOCUMENTS AND STANDARDS

-----

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

### 1.4 DIAGNOSTIC HIERARCHY PREREQUISITES

-----

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

CVRLABO	RLV11 RLO1 DISKLESS TEST (RLV11 ONLY)
CZRLGBO	RL11/RLV11 RLO1/02 CONTROLLER TEST (PART 1)
CZRLMBO	RL11/RLV11 RLO1/02 CONTROLLER TEST (PART 2)
CZRLIBO	RLO1/02 DRIVE TEST (PART 1)

### 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:

```
CHMDKAO 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:H0E". 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 H0E 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 OCCURRED. 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:H0E=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 H0E 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.

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 CZRLNB	O
DRS LOADED	D
DIAG. RUN-TIME SERVICES REV. D APR-79	D
CZRLN-B O	D
CZRLN TESTS SEEK AND ROTATIONAL TIMING & WRITE & READ DATA	D
UNIT IS RL01, RL02	D
DR>STA/PASS:1/FLAGS:HOE	D.O
# UNITS (O) ? 2	D.O
UNIT 0	D
RL11 (L) Y ?	D.O
BUS ADDRESS (O) 174400 ?	D.O
VECTOR (O) 160 ?	D.O
DRIVE (O) 0 ?	D.O
DRIVE TYPE = RL01 (L) Y ?	D.O
BR LEVEL (O) 5 ?	D.O
UNIT 1	D
RL11 (L) Y ?	D.O
BUS ADDRESS (O) 174400 ?	D.O
VECTOR (O) 160 ?	D.O
DRIVE (O) 0 ? 1	D.O
DRIVE TYPE = RL01 (L) ? N	D.O (N=RL02)
BR LEVEL (O) 5 ?	D.O
CHANGE SW (L) ? Y	D.O
USE ALL CYL (L) N ?	D.O
USE ALL SECT (L) N ?	D.O
DO MANUAL INTERVENTION TEST (L) N ?	D.O
LOW SEEK LIMIT (L) N ?	D.O
UPPER SEEK LIMIT (L) N ?	D.O
USE ONLY ONE SURF (L) N ?	D.O
INPUT ERROR LIMIT (D) 20 ?	D.O
DATA CMP ERR LMT (D) 10 ?	D.O
CZRLN HRD ERR 00004 TST 003 SUB 002 PC:004130 ERR HLT	
DR>PRO/FLAGS:IER:LOE:HOE=0	D.O

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

```

↑C                0
DR>CON/FLAGS:MOE:IER:LOE=0      D,0
CHANGE SW (L) ? N              D,0
CZRLN EOP 1                    0
↑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.

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> OR
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 SOFTWARE SWITCH REGISTER 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	START PRINT DISPLAY FLAGS ZFLAGS EXIT

4. AN ERROR WAS ENCOUNTERED WITH THE MOE FLAG SET SET

START  
RESTART  
CONTINUE  
PROCEED  
PRINT  
DISPLAY  
FLAGS  
ZFLAGS  
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 MOE 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:

- MOE 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.

\*\*\*\*\*  
 CONTINUE)/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

\*\*\*\*\*  
 PROCEED)/FLAGS:<FLAG-LIST>  
 \*\*\*\*\*

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

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

1. UNSPECIFIED FLAG SETTINGS ARE UNCHANGED

\*\*\*\*  
 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.

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 (0) ? 8

UNIT 0  
 RL11 (L) Y ?  
 BUS ADDRESS (0) 174400 ?  
 VECTOR (0) 160 ?  
 DRIVE (0) 0 ? 0-3  
 DRIVE TYPE = RLO1 (L) Y ?  
 BR LEVEL (0) 5 ?

UNIT 4  
 RL11 (L) Y ?  
 BUS ADDRESS (0) 174400 ? 175400  
 VECTOR (0) 160 ? 164  
 DRIVE (0) 0 ? 0-3  
 DRIVE TYPE = RLO1 (L) Y ? N  
 BR LEVEL (0) 5 ?

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

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

## 2.5 HARDWARE 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 RL11 CONTROLLER.

BUS ADDRESS (0) 174400?

ANSWER WITH THE BUS ADDRESS OF THE CONTROLLER.

VECTOR (0) 160?

ANSWER WITH THE INTERRUPT VECTOR OF THE CONTROLLER

DRIVE (0) 0?

ANSWER WITH THE DRIVE(S) CONNECTED TO THE CONTROLLER

DRIVE TYPE = RLO1 (L) ?

ANSWER NO (N) IF DRIVE IS AN RLO2

BR LEVEL (0) 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.

USE ALL CYLINDERS (N)?

IF "YES", THOSE TESTS THAT NORMALLY USE A SELECTED SET OF CYLINDERS WILL TEST EVERY CYLINDER ON THE CARTRIDGE.

USE ALL SECTORS (N)?

IF "YES", THOSE TESTS THAT NORMALLY USE A SINGLE SECTOR TO TEST A GIVEN OPERATION (SUCH AS SEEK DESTINATION) WILL READ AND VERIFY EVERY SECTOR HEADER.

EXECUTE MANUAL INTERVENTION TESTS (N)?

IF "YES", SEEK TIMING, ROTATIONAL TIMING, AND WRITE LOCK ERROR AND DATA PROTECTION TESTS ARE EXECUTED. THE ONLY TEST THAT ACTUALLY REQUIRES MANUAL INTERVENTION IS THE WRITE LOCK TEST AND THAT TEST WILL BYPASS AUTOMATICALLY AFTER WAITING 30 SECONDS FOR WRITE LOCK TO BE SET.

LOWER SEEK LIMIT (N)?

IF "YES", THE NEXT PARAMETER IS REQUESTED.

ENTER VALUE (DECIMAL) (0)?

THIS LIMIT IS IMPOSED ON ALL SEEK OPERATIONS SUCH THAT TESTING IS NOT DONE BELOW THAT LIMIT. IN ADDITION, SETTING THIS LIMIT (OR THE UPPER LIMIT, SEE BELOW) CAUSES THE FORWARD AND REVERSE OSCILLATING SEEK TESTS TO PERFORM DIFFERENTLY (SEE TEST DESCRIPTION). TESTS THAT REQUIRE ACCESS TO A SPECIFIC CYLINDER THAT FALLS BELOW THE SPECIFIED LIMIT WILL IGNORE THE LIMIT (SEE WRITE/READ TEST PART 1).

UPPER SEEK LIMIT (N)?

IF "YES", AN UPPER CYLINDER LIMIT IS IMPOSED IN THE SAME MANNER AS THE LOWER SEEK LIMIT. A "YES" RESPONSE WILL CAUSE THE FOLLOWING PARAMETER REQUEST.

ENTER VALUE (DECIMAL) (255)?

USE ONLY ONE SURFACE (N)?

IF "YES", THE NEXT PARAMETER IS REQUESTED.

SPECIFY SURFACE (0 OR 1) (DECIMAL) (0)?

WHICHEVER SURFACE IS SPECIFIED IS THE ONLY SURFACE TESTED IN THE ENTIRE PROGRAM. ANY TEST THAT IS DESIGNED TO TEST THE OTHER SURFACE IS AUTOMATICALLY BYPASSED. THE PROGRAM DOES NOT PRINT ANY INDICATION THAT A TEST IS BYPASSED IN THIS CASE.

SPECIFY ERROR LIMIT (DECIMAL) (20)?

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

DATA COMPARE ERROR LIMIT (DECIMAL) (20)?

THIS PARAMETER SPECIFIES THE NUMBER OF DATA COMPARE ERRORS THAT WILL BE LISTED FOR A GIVEN COMPARE OPERATION. AFTER THE LIMIT IS REACHED, THE DATA ERRORS ARE NOT PRINTED BUT THE COMPARE CONTINUES UNTIL THE END OF THE DATA FIELD. A TOTAL IS REPORTED AT THE END OF THE COMPARE.

### 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

-----

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

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

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

READ DATA WITH DATA COMPARE IS AN ERROR THAT WAS DETECTED AS BAD DATA IN THE BUFFER AFTER

A READ DATA OPERATION. WHEN THIS OPERATION IS REPORTED IT INDICATES THE ACTUAL READ DATA OPERATION COMPLETED WITH NO DETECTED ERRORS BUT THE DATA WAS WRONG.

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

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

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

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

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

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

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

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

THE QUALIFIER "ADJ CYL WRITTEN AFTER FWD SK" AND "ADJ CYL WRITTEN AFTER REV SK" WILL BE REPORTED ONLY IN THE ADJACENT CYL-

INDER INTERFERENCE TEST. THESE QUALIFIERS ARE USED WHEN THE ERROR OCCURS ON THE CYLINDER UNDER TEST AND DEFINE THE DIRECTION THE HEADS WERE MOVED WHEN THE ADJACENT CYLINDER WAS WRITTEN.

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

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

### 3.1.2 SPECIFIC RESULT MESSAGES

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

THE GENERAL FORMAT FOR THE RESULT LINE IS:

RESULT:(VAR 1) IS (VAR 2) SB (VAR 3) (OPTIONAL QUALIFIER)  
WHERE VARIABLE 1 CAN BE ONE OF THE FOLLOWING:

CONT ERR	(CONTROLLER ERROR)
DRV ERR	(DRIVE ERROR)
NON-EXSTNT MEM	(NON-EXISTANT MEMORY)
HDR CRC	(HEADER CRC ERROR)
DATA CRC	
HDR NOT FND	(HEADER NOT FOUND)
DATA LATE	
HDR NOT FND/HDR CRC/OPI	(ALL 3 BITS SET)
DRV RDY	(DRIVE READY)
SELECTED HEAD	
VOL CHK	(VOLUME CHECK)
COVER OPEN	
BRUSH HME	(BRUSH HOME)
WRT LCK	(WRITE LOCK)
HDS OUT	(HEADER OUT)
DRV SEL ERR	(DRIVE SELECT ERROR)
DRV STATE	(DRIVE STATE)
SPIN TIMEOUT	(SPINDLE TIMEOUT SPD ERROR)
WRT GAT ERR	(WRITE GATE ERROR)
SEEK TIMEOUT	(SKTO ERROR)
CUR HEAD ERR	(CURRENT IN HEAD ERROR)
WRT DAT ERR	(WRITE DATA ERROR)

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

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

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

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

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

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

"INTERRUPT 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.

## "OPI SET-NO DRIVE RESPONSE"

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

## "NO INTERRUPT ON CMND COMPLETE"

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

## "ERR DID NOT CLEAR"

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

## 'DRV ERR IS NOT CLEARED

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

## "UNEXPECTED ERR"

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

## "BAD SEC FILE FMT ERR"

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

## 3.1.3 OTHER MESSAGES

OTHER INFORMATION IS REPORTED UNDER VARIOUS CIRCUMSTANCES. THESE ARE:

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

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

"ERROR LIMIT EXCEEDED-UNIT DROPPED"

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

MOST ERROR REPORTS HAVE THE FOLLOWING FORMAT.

```

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

```

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

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

THE SUBTEST NUMBER IS UNIQUE IN THIS PROGRAM IN THAT IT DOES NOT REFER TO A PHYSICAL SUBTEST WITHIN A GIVEN TEST. RATHER IT REFLECTS THE NUMBER OF TIMES A SUBTEST HAS BEEN EXECUTED WITHIN A TEST. CONSEQUENTLY, ON A TEST THAT TESTS AN INCREMENTAL TYPE OF OPERATION (SUCH A INCREMENTAL SEEKS, READ ALL HEADERS FROM BOTH SURFACES, ETC.) THE SUBTEST WILL BE DESCRIPTIVE OF WHERE IN THE TEST THE ERROR OCCURRED.

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

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

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

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

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

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

HNF/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.

### 3.2 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 EXISTENT MEMORY ERROR  
 BIT 12 - HEADER NOT FOUND (WITH BIT 10 SET)  
           - DATA LATE (WITH BIT 10 CLEAR)  
 BIT 11 - HEADER CRC (WITH BIT 10 SET)  
           DATA CRC (WITH BIT 10 CLEAR)

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

BIT 0 - DRIVE READY

RLBA BUS ADDRESS REGISTER (XXXXX2)

-----

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

RLDA - DISK ADDRESS REGISTER (XXXXX4)

-----

FOR READ/WRITE FUNCTIONS

-----

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

FOR SEEK FUNCTION

-----

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

FOR GET STATUS FUNCTION

-----

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

## RLMP MULTIPURPOSE REGISTER

## FOR READ/WRITE FUNCTION

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

## FOR READ HEADER FUNCTION

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

## FOR GET STATUS FUNCTION

## HAS DRIVE STATUS

BIT 15 - WRITE DATA ERROR  
 BIT 14 - CURRENT HEAD ERROR (CHE)  
 BIT 13 - WRITE LOCK STATUS (WL)  
 BIT 12 - SEEK TIME OUT (SKTO)  
 BIT 11 SPIN ERROR (SPE)  
 BIT 10 WRITE GATE ERROR (WGE)  
 BIT 9 - VOLUME CHECK (VC)  
 BIT 8 - DRIVE SELECT ERROR (DSE)  
 BIT 7 - DRIVE TYPE IS RLO2 IF SET  
 BIT 6 SURFACE (0=UPPER, 1=LOWER)  
 BIT 5 - COVER OPEN  
 BIT 4 - HEADS HOME  
 BIT 3 - BRUSHES HOME  
 BIT 2-0 -STATE BITS  
 0 - LOAD STATE  
 1 SPIN UP  
 2 BRUSH CYCLE  
 3 - LOAD HEADS  
 4 SEEK - TRACK COUNTING  
 5 - SEEK - LINEAR MODE  
 6 - UNLOAD HEADS  
 7 - SPIN DOWN

## 6.0 TEST SUMMARIES

## TEST 1 SEEK TIMING

\*\*\*\*\*

(P-CLOCK IS REQUIRED TO PERFORM THIS TEST.)

POSITION HEADS AT CYLINDER 0.

DO 64 SEEKS FROM 0 TO 1 AND 1 TO 0, MEASURING THE SEEK TIME FOR EACH SEEK. AVERAGE THE SEEK TIMES (FORWARD AND REVERSE INDEPENDENTLY) AND REPORT.

REPEAT ABOVE SEEKING BETWEEN CYLINDER 127 TO 128 AND 254 TO 255 FOR RL01 AND 255 TO 256 AND 256 TO 511 FOR RL02.

REPEAT ABOVE SEEKING BETWEEN CYLINDER 0 TO 127 AND 128 TO 256 FOR RL01 AND CYLINDER 0 TO 256 AND 256 TO 511 FOR RL02.

REPEAT ABOVE SEEKING BETWEEN CYLINDER 0 AND 255 FOR RL01 AND 0 TO 511 FOR RL02.

THE SEEK TIMES WILL BE REPORTED AS SHOWN BELOW. THE TIME MEASURED IS FROM START OF SEEK COMMAND UNTIL INTERRUPT IS RECEIVED.

	INNER	MIDDLE	OUTER	MAX TIME
1 CYL FWD	X	X	X	X
1 CYL REV	X	X	X	X
MID CYL FWD	X		X	X
MID CYL REV	X		X	X
MAX CYL FWD		X		X
MAX CYL REV		X		X

THE X INDICATES WHERE TIME WILL BE REPORTED.

#### TEST 2 BASIC READ DATA TEST

\*\*\*\*\*

POSITION HEADS AT MAX CYLINDER.

DO READ DATA, HEAD 1. CHECK FOR ANY ERRORS AND REPORT. IF ERROR, READ SECTOR 1 THROUGH 19 UNTIL NO ERROR ON READ. REPORT ALL ERRORS BUT DO NOT INCREMENT ERROR COUNT. IF NONE CAN BE READ, SUCCESSFULLY, REPORT THAT FACTORY BAD SECTOR FILE CANNOT BE READ, INCREMENT ERROR COUNT AND PROCEED WITH READ OF SECTOR 20.

ON SECTOR WITH NO CRC ERROR, VERIFY DATA FORMAT (WORD 0 AND 1 ARE NOT 0, WORD 2 AND 3 ARE 0, LOCATE FIRST WORD OF ALL ONE'S AND THAT WORD TO WORD 127 ARE ALL ONE'S.) STORE BAD SECTOR DATA.

READ DATA, HEAD ONE, SECTOR 20. CHECK FOR ANY ERRORS AND

REPORT. IF ERROR, READ SECTOR 21 THROUGH 39 UNTIL NO ERROR ON READ. REPORT ALL ERRORS BUT DO NOT INCREMENT ERROR COUNT. IF NONE CAN BE READ SUCCESSFULLY, REPORT THAT SOFTWARE BAD SECTOR FILFS CANNOT BE READ, INCREMENT ERROR COUNT AND EXIT TEST.

ON SECTOR WITH NO CRC ERROR, VERIFY DATA AS ABOVE. STORE BAD SECTOR DATA.

NOTE: IF SURFACE 0 IS SELECTED THIS TEST WILL BE BYPASSED.

TEST 3 WRITE/READ DATA TEST (PART 1)

\*\*\*\*\*

POSITION HEADS AT CYLINDER 0

WRITE PATTERN 1 ON HEAD 0, SECTOR 0. CHECK FOR ANY ERROR.

READ HEAD 0, SECTOR 0. CHECK FOR CRC ERROR. COMPARE DATA.

REPEAT FOR OTHER DATA PATTERNS (2 THROUGH 8).

CHECK IF CYLINDER 0, TRACK 1, SECTOR 0 IS LISTED IN BAD SECTOR DATA. IF NOT, REPEAT ABOVE TEST AT CYLINDER 0, TRACK 1, SECTOR 0. IF IT IS LISTED AS BAD, LOCATE FIRST SECTOR 0, TRACK 1 THAT IS GOOD AND DO ABOVE TESTS.

NOTE: CYLINDER LIMITS ARE IGNORED, TESTING IS DONE AT CYLINDER 0. HOWEVER, CHOOSING A SINGLE SURFACE WILL LIMIT TESTING TO THAT SURFACE.

TEST 4 ROTATIONAL TIMING TEST

\*\*\*\*\*

(P CLOCK IS REQUIRED TO PERFORM THIS TEST.)

POSITION HEADS TO CYLINDER 0.

DO WRITE DATA TO CYLINDER 0, HEAD 0, SECTOR 0. WAIT FOR INTERRUPT.

DO WRITE DATA TO CYLINDER 0, HEAD 0, SECTOR 0. START TIMING. WHEN INTERRUPT OCCURS, STOP TIMING. RESULT IS SPINDLE ROTATION TIME.

REPEAT TEST 64 TIMES. REPORT THE AVERAGE AS SPINDLE ROTATION TIME. THE TIME REPORTED IS IN 100'S OR MICROSECONDS.

TEST 5 WRITE/READ TEST (PART 2)

\*\*\*\*\*

CC IS CURRENT CYLINDER SELECTED FROM SET.  
LET SELECTED CYLINDER SET BE AS DEFINED IN PARAGRAPH 4.3.

SEEK FORWARD TO CC. WRITE PATTERNS 1 THROUGH 8 REPEATED 5 TIMES ON HEAD 0. READ/COMPARE ALL DATA.

SEEK REVERSE TO "LOLIMIT". SEEK FORWARD TO CC. READ/COMPARE ALL DATA. SEEK FORWARD TO "HILIMIT". SEEK REVERSE TO CC. READ/COMPARE ALL DATA. REWRITE DATA PATTERNS 1 THROUGH 8 REPEATED 5 TIMES ON HEAD 0. READ COMPARE ALL DATA.

SEEK FORWARD TO "HILIMIT". SEEK REVERSE TO CC. READ/COMPARE ALL DATA. SEEK REVERSE TO "LOLIMIT". SEEK FORWARD TO CC.



READ/COMPARE ALL DATA.

REPEAT ABOVE TEST FOR HEAD 1.

REPEAT ABOVE TESTS FOR ALL CYLINDERS IN SELECTED CYLINDER SET.

NOTE 1: IF ANY OF THE SECTORS IN THE SELECTED CYLINDER SET ARE LISTED AS BAD, THAT SECTOR WILL BE BYPASSED.

NOTE 2: IF THE "USE ALL CYLINDERS" PARAMETER IS SPECIFIED AS "Y", THE TEST WILL INCLUDE ALL CYLINDERS IN THE SELECTED PARAMETER SET.

NOTE 3: IN THE FIRST PASS OF THE PROGRAM THIS TEST IS EXECUTED ON ONLY 6 OF THE CYLINDERS LISTED IN THE CYLINDER SET. THOSE USED WILL BE EVERY 8TH ENTRY IN THE TABLE. ON THE SECOND AND SUBSEQUENT PASSES ALL ENTRIES IN THE SELECTED CYLINDER SET ARE USED.

NOTE 4: TESTING WILL BE DONE BETWEEN UPPER AND LOWER LIMITS. CYLINDERS IN THE CYLINDER SET BEYOND THESE LIMITS WILL NOT BE TESTED. CHOOSING A SINGLE SURFACE WILL LIMIT TESTING TO THAT SURFACE.

#### TEST 6 WRITE LOCK ERROR AND DATA PROTECTION TEST

\*\*\*\*\*

DO WRITE DATA PATTERN 0 AT SECTOR 0. READ DATA AND VERIFY.

ASK OPERATOR TO WRITE LOCK DRIVE. DO GET STATUS LOOP UNTIL WRITE LOCK IS SET. IF NOT SET IN 30 SECONDS, ABORT THE TEST.

WHEN WRITE LOCK IS SET, DO WRITE DATA PATTERN 1 AT SECTOR 0. REPORT FAILURE IF DRIVE ERROR DOES NOT SET OR IF ANY OTHER ERROR SETS. CLEAR ERROR AND READ DATA AT SECTOR 0. CHECK THAT DATA HAS NOT BEEN DISTURBED.

REQUEST OPERATOR TO RESET WRITE LOCK. DO GET STATUS LOOP UNTIL WRITE LOCK IS RESET. IF NOT RESET IN 30 SECONDS, REPEAT THE REQUEST.

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

#### TEST 7 ADJACENT CYLINDER INTERFERENCE TEST

\*\*\*\*\*

CC IS CURRENT CYLINDER SELECTED FROM SET  
LET SELECTED CYLINDER SET BE AS DEFINED IN PARAGRAPH 4.3.  
DATA PATTERN IS 155555.

SEEK FORWARD TO CYLINDER CC. WRITE PATTERN ON TRACK 0. ALL SECTORS. READ/COMPARE DATA.

SEEK FORWARD TO "HILIMIT". SEEK REVERSE TO CC-1. WRITE PATTERN. SEEK FORWARD TO "HILIMIT". SEEK REVERSE TO CC. WRITE PATTERN. (THIS HAS BRACKETED ORIGINAL WRITE WITH WRITES IN ADJACENT CYLINDERS. NOTE ADJACENT CYLINDERS WERE WRITTEN AFTER HEADS CAME ON CYLINDER IN REVERSE DIRECTION WHICH IS OPPOSITE OF CENTER CYLINDER.)

SEEK REVERSE TO "LOLIMIT". SEEK FORWARD TO CC. READ/COMPARE DATA FROM ALL SECTORS. ANY ERRORS (READ OR COMPARE) ARE ATTRIBUTED TO ADJACENT CYLINDER INTERFERENCE.

SEEK FORWARD TO "HILIMIT". SEEK REVERSE TO CC. WRITE DATA PATTERN. SEEK REVERSE TO "LOLIMIT". SEEK FORWARD TO CC-1. WRITE PATTERN. SEEK REVERSE TO "LOLIMIT". SEEK FORWARD TO CC-1. WRITE PATTERN. SEEK FORWARD TO "HILIMIT". SEEK REVERSE TO CC. READ/COMPARE DATA IN ALL SECTORS. ANY ERRORS (READ OR COMPARE) ARE ATTRIBUTED TO ADJACENT CYLINDER INTERFERENCE.

REPEAT ABOVE TESTS ON HEAD 1.

NOTE 1: IF ANY SECTOR ON A SELECTED CYLINDER IS LISTED BAD, THAT SECTOR WILL BE BYPASSED.

NOTE 2: IF THE "USE ALL CYLINDERS" PARAMETER IS SPECIFIED AS "Y", THE TEST WILL INCLUDE ALL CYLINDERS (EXCEPT 0 AND MAX CYL) IN THE SELECTED PARAMETER SET.

NOTE 3: IN THE FIRST PASS OF THE PROGRAM THIS TEST IS EXECUTED ON ONLY 3 OF THE CYLINDERS LISTED IN THE CYLINDER SET. THOSE USED WILL BE THE FIRST, TWENTYFIRST, AND FORTYFIRST ENTRIES IN THE TABLE. ON SECOND AND SUBSEQUENT PASSES EVERY FOURTH CYLINDER SET ENTRY WILL BE TESTED.

NOTE 4: TESTING WILL BE DONE BETWEEN UPPER AND LOWER LIMITS. CYLINDERS IN THE CYLINDER SET BEYOND THESE LIMITS WILL NOT BE TESTED. CHOOSING A SINGLE SURFACE WILL LIMIT TESTING TO THAT SURFACE.

#### TEST 8 OVERWRITE TEST

\*\*\*\*\*

CC IS CURRENT CYLINDER SELECTED FROM SET  
 SELECTED CYLINDER SET DEFINED IN PARAGRAPH 4.3.  
 PATTERN A = 125252  
 PATTERN B = 000000

SEEK FORWARD TO CC. WRITE DATA OF PATTERN A IN ALL SECTORS, HEAD 0. READ/COMPARE DATA.

SEEK FORWARD TO "HILIMIT", SEEK REVERSE TO CC. WRITE PATTERN B. SEEK REVERSE TO "LOLIMIT", SEEK FORWARD TO CC. READ/COMPARE DATA.

SEEK FORWARD TO "HILIMIT", SEEK REVERSE TO CC. WRITE DATA PATTERN A. READ/COMPARE DATA. SEEK REVERSE TO "LOLIMIT", SEEK FORWARD TO CC. WRITE PATTERN B. SEEK FORWARD TO "HILIMIT" SEEK REVERSE TO CC. READ/COMPARE DATA.

ANY FAILURES (READ OR COMPARE) ARE ATTRIBUTED TO OVERWRITE PROBLEM.

REPEAT ABOVE TESTS ON HEAD 1.

NOTE 1: IF ANY SECTOR ON A SELECTED CYLINDER IS LISTED AS BAD, THAT SECTOR WILL BE BYPASSED.

NOTE 2: IF THE "USE ALL CYLINDERS" PARAMETER IS SPECIFIED AS "Y", THE TEST WILL INCLUDE ALL CYLINDERS IN THE SELECTED PARAMETER SET.

NOTE 3: IN THE FIRST PASS OF THE PROGRAM THIS TEST IS EXECUTED ON ONLY 3 OF THE CYLINDERS LISTED IN THE CYLINDER SET. THOSE USED WILL BE THE FIRST, TWENTYFIRST, AND FORTYFIRST ENTRIES IN THE TABLE. ON SECOND AND SUBSEQUENT PASSES EVERY FOURTH CYLINDER SET ENTRY WILL BE TESTED.

NOTE 4: TESTING WILL BE DONE BETWEEN UPPER AND LOWER LIMITS. CYLINDERS IN THE CYLINDER SET BEYOND THESE LIMITS WILL NOT BE TESTED. CHOOSING A SINGLE SURFACE WILL LIMIT TESTING TO THAT SURFACE.

3	2	MACRO DEFINITIONS
4	32	GLOBAL DATA SECTION
4	166	GLOBAL DATA SECTION
4	587	GLOBAL MESSAGES
5-	1	ERROR MESSAGES
6-	1	INITIALIZATION SECTION
7-	2	AUTO DROP SECTION
8-	2	CLEANUP CODE SECTION
9-	1	GLOBAL SUBROUTINES
12-	5	*TEST 1            **SEEK TIMING
13	1	*TEST 2            **BASIC READ DATA (BAD SECTOR FILE)
14-	1	*TEST 3            **WRITE/READ DATA (PART 1)
15-	1	*TEST 4            **ROTATIONAL TIMING
16-	1	*TEST 5            **WRITE/READ DATA (PART 2)
17-	1	*TEST 6            **WRITE LOCK ERROR AND DATA PROTECTION
18-	1	*TEST 7            **ADJACENT CYLINDER INTERFERENCE
19-	1	*TEST 8            **OVERWRITE
20-	1	PARAMETER CODING

1			
2		000001	PART2**1
3	000000		.ENABLE ABS
4			.ENABLE AMA
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			
15			

```
1
2
3      .SBTTL  MACRO DEFINITIONS
4
5      .MACRO  WAITUS  ARG          ;MACRO MICRO SEC WAIT
6              MOV    ARG,XDELAY  ;SAVE ARGUMENT
7              JSR    PC,TIME      ;CALL TIMING ROUTINE
8
9      .ENDM
10
11     .MACRO  WAITMS  ARG          ;MACRO MILLI-SEC WAIT
12             MOV    ARG,YDELAY  ;SAVE ARGUMENT
13             JSR    PC,XTIME    ;CALL TIMING ROUTINE
14
15     .ENDM
16
17     .MACRO  ABORTWAIT          ;MACRO CLEAR UNELAPSED TIME
18             MOV    XDELAY,TEMPO ;SAVE MICRO-SEC RUN TIME
19             MOV    YDELAY,TEMP  ;SAVE MILLI SEC RUN TIME
20             CLR    XDELAY       ;ABORT MICRO-SEC WAIT
21             CLR    YDELAY       ;ABORT MILLI-SEC WAIT
22
23     .ENDM
24
25     .MACRO  GETTIM  ARG          ;MACRO GET ELAPSED TIME
26             MOV    @#CLKCTR,ARG ;STORE CLOCK COUNTER CONTENTS
27             CLR    @#CLKCSR     ;EVENT FINISHED, STOP CLOCK
28
29     .ENDM
30
31     .MACRO  STCLK          ;MACRO START P-CLOCK
32             CLR    @#CLKCSB     ;CLEAR CLOCK COUNT SET BUFFER
33             CLR    @#CLKCTR     ;CLEAR CLOCK COUNTER
34             MOV    #23,@#CLKCSR ;INITIALIZE CLOCK FOR COUNT UP MODE,
                                ;/10 KHZ RATE, AND START CLOCK
35
36     .ENDM
```

```

1
2
3
4
5 002000          POINTER BGNSW,BGNSFT,BGNDU
6
7 002000          BGNMOD MDHEDR
8 002000          HEADER CZRLN,B,0,30000,0
   002000          103      .ASCII /C/
   002001          132      .ASCII /Z/
   002002          122      .ASCII /R/
   002003          114      .ASCII /L/
   002004          116      .ASCII /N/
   002005          000      .BYTE 0
   002006          000      .BYTE 0
   002007          000      .BYTE 0
   002010          102      .ASCII /B/
   002011          060      .ASCII /O/
   002012          000000    .WORD 0
   002014          030000    .WORD 30000
   002016          036620    .WORD L$HARD
   002020          036774    .WORD L$SOFT
   002022          014102    .WORD L$HW
   002024          014120    .WORD L$SW
   002026          037400    .WORD L$LAST
   002030          000000    .WORD 0
   002032          000000    .WORD 0
   002034          000000    .WORD 0
   002036          000000    .WORD 0
   002040          014136    .WORD L$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          002216    .WORD L$DVTYP
   002062          000000    .WORD 0
   002064          000000    .WORD 0
   002066          000000    .WORD 0
   002070          000000    .WORD 0
   002072          015616    .WORD L$DU
   002074          000000    .WORD 0
   002076          002122    .WORD L$DESC
   002100          104035    EMT E$LOAD
   002102          000000    .WORD 0
   002104          014156    .WORD L$INIT
   002106          015470    .WORD L$CLEAN
   002110          015132    .WORD L$AUTO
   002112          014072    .WORD L$PROT
   002114          000000    .WORD 0
   002116          000000    .WORD 0
   002120          000000    .WORD 0
9 002122          ENDMOD
10 002122         DESCRIPT <CZRLN TESTS SEEK & ROTATIONAL TIMING AND WRITE & READ DATA>

```

	002122	103	132	122	.ASCIZ /CZRLN TESTS SEEK & ROTATIONAL TIMING AND WRITE & READ DATA/
	002125	114	116	040	
	002130	124	105	123	
	002133	124	123	040	
	002136	123	105	105	
	002141	113	040	046	
	002144	040	122	117	
	002147	124	101	124	
	002152	111	117	116	
	002155	101	114	040	
	002160	124	111	115	
	002163	111	116	107	
	002166	040	101	116	
	002171	104	040	127	
	002174	122	111	124	
	002177	105	040	046	
	002202	040	122	105	
	002205	101	104	040	
	002210	104	101	124	
	002213	101	000		
11	002216				.EVEN
	002216	122	114	060	DEV TYP <RL01,RL02>
	002221	061	054	122	.ASCIZ /RL01,RL02/
	002224	114	060	062	
	002227	000			
12					.EVEN
13					;COPYRIGHT (C) 1979,1983
14					;THIS SOFTWARE IS FURNISHED UNDER LICENSE FOR USE ONLY
15					;ON A SINGLE COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH
16					;THE INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS
17					;SOFTWARE, OR ANY COPIES THEREOF, MAY NOT BE PROVIDED
18					;OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT
19					;FOR USE ON SUCH SYSTEM, AND TO ONE WHO AGREES TO THESE
20					;LICENSE TERMS. TITLE TO OWNERSHIP OF THE SOFTWARE SHALL
21					;AT ALL TIMES REMAIN IN DEC.
22					;
23					;THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE
24					;WITHOUT NOTICE AND SHALL NOT BE CONSTRUED AS A COMMITMENT
25					;BY DIGITAL EQUIPMENT CORPORATION.
26					;
27					;DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY
28					;OF ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
29					
30					
31					
32					.SBTTL GLOBAL DATA SECTION
33					
34	002230				BGNMOD GLBEQAT
35					
36	002230				EQUALS
					;
					; BIT DEFINITIONS
					;
	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
37          ;
38          000000      CSR      =0          ;BUS ADDRESS
39          000002      VECT     =2          ;VECTOR ADDRESS
40          000004      PRIOR    =4          ;PRIORITY
41          000006      TYPDR    =6          ;DRIVE TYPE
42          000010      DRSB     =10         ;DRIVE SELECT BIT
43          000012      CNT      =12         ;CONTROLLER TYPE
44
45          ;
46          000000      MISWI    =0          ;SOFTWARE PARAMETERS SWITCHES
47          000002      LOLIM    =2          ;CYLINDER LOWER LIMIT
48          000004      HILIM    =4          ;CYLINDER HIGH LIMIT
49          000006      HEAD     =6          ;SELECTED HEAD FOR RUNNING TESTS
50          000010      ERLIM    =10         ;ERROR LIMIT
51          000012      DCLIM    =12         ;DATA COMPARE ERROR LIMIT
52
53          ;
54          000001      ALLCYL   =BIT00     ;USE ALL CYLINDERS
55          000002      ALLSEC   =BIT01     ;USE ALL SECTORS
56          000004      DRSELT   =BIT02     ;EXECUTE DRIVE SELECT TEST
57          000010      HDALIGN   =BIT03     ;EXECUTE HEAD ALIGNMENT TEST
58          010000      HEADLM   =BIT12     ;HEAD LIMIT SPECIFIED FLAG
59          020000      HICYL    =BIT13     ;HI LIMIT SPECIFIED FLAG
60          040000      LOCYL    =BIT14     ;LO LIMIT SPECIFIED
61          100000      MITEST    =BIT15     ;EXECUTE MANUAL INTERVENTION TESTS
62
63          ;
64          000102      CKDATA    =102        ;WRITE CHECK
65          000104      GTSTAT    =104        ;GET STATUS
66          000106      SEEK      =106        ;SEEK
67          000110      RDHEAD    =110        ;READ HEADER
68          000112      WTDATA    =112        ;WRITE DATA
69          000114      RDDATA    =114        ;READ DATA
70          000116      RDNQHR    =116        ;READ DATA, IGNORE HEADERS
71          000100      NOOP      =100        ;NO OPERATION
72
73          ;
74          007777      COMPOP    =7777       ;COMPOSITE OPERATION FLAGS
75          000002      HDRCMP    =BIT01     ;HEADER COMPARE OPERATION
76          000001      DATACMP   =BIT00     ;DATA COMPARE OPERATION
77          000004      CYLUP     =BIT02     ;CYCLE UP OPERATION
78          000010      ULOAD     =BIT03     ;UNLOAD OPERATION
79          000020      INOUTS    =BIT04     ;IN-OUT SEEK OPERATION
80          000040      OUTINS    =BIT05     ;OUT-IN SEEK OPERATION
81          000100      FOLWRT    =BIT06     ;FOLLOWING WRITE OPERATION
82          000200      REVSKS    =BIT07     ;REV SEEK SEQ (ADJ INTERFERENCE)
83          000400      FWDSKS    =BIT08     ;FWD SEEK SEQ (ADJ INTERFERENCE)
84          001000      REVSKO    =BIT09     ;REV SEEK SEQ (OVERWRITE)
85          002000      FWDSKO    =BIT10     ;FWD SEEK SEQ (OVERWRITE)
86          004000      BADADD    =BIT11     ;BAD DISK ADDRESS
87          010000      SEEKOP    =BIT12     ;SEEK OPERATION

```

```

88      020000      RORWOP  =BIT13      ;READ OR WRITE OPERATION
89      040000      RELDWT  =BIT14      ;RELOAD WAIT
90      100000      HDR40   =BIT15      ;40 HEADER OPERATION
91      003760      MQUALS  =OUTINS!INOUTS!FOLWRT!REVSKS!FWDSKS!RLVSKO!FWDSKO
92                                     ;MESSAGE QUALIFIER BITS
93
94      ;           ;           ;           ;           ;           ;
95      000001      ;           ;           ;           ;           ;           ;
96      000002      ;           ;           ;           ;           ;           ;
97      000004      ;           ;           ;           ;           ;           ;
98      000010      ;           ;           ;           ;           ;           ;
99
100     000000      RLCS    =0           ;CONTROL AND STATUS REGISTER
101     000002      RLBA    =2           ;BUS ADDRESS REGISTER
102     000004      RLDA    =4           ;DISK ADDRESS REGISTER
103     000006      RLMP    =6           ;MULTI-PURPOSE REGISTER
104
105     ;           ;           ;           ;           ;           ;
106     000000      ;           ;           ;           ;           ;           ;
107     100000      ;           ;           ;           ;           ;           ;
108     040000      ;           ;           ;           ;           ;           ;
109     020000      ;           ;           ;           ;           ;           ;
110     010000      ;           ;           ;           ;           ;           ;
111     010000      ;           ;           ;           ;           ;           ;
112     004000      ;           ;           ;           ;           ;           ;
113     004000      ;           ;           ;           ;           ;           ;
114     002000      ;           ;           ;           ;           ;           ;
115     001400      ;           ;           ;           ;           ;           ;
116     000200      ;           ;           ;           ;           ;           ;
117     000100      ;           ;           ;           ;           ;           ;
118     000060      ;           ;           ;           ;           ;           ;
119     000001      ;           ;           ;           ;           ;           ;
120
121     ;           ;           ;           ;           ;           ;
122     000077      ;           ;           ;           ;           ;           ;
123     000100      ;           ;           ;           ;           ;           ;
124
125     ;           ;           ;           ;           ;           ;
126     000001      ;           ;           ;           ;           ;           ;
127     000004      ;           ;           ;           ;           ;           ;
128     000020      ;           ;           ;           ;           ;           ;
129
130     ;           ;           ;           ;           ;           ;
131     000003      ;           ;           ;           ;           ;           ;
132     000010      ;           ;           ;           ;           ;           ;
133
134     ;           ;           ;           ;           ;           ;
135     017777      ;           ;           ;           ;           ;           ;
136     160000      ;           ;           ;           ;           ;           ;
137
138     ;           ;           ;           ;           ;           ;
139     000077      ;           ;           ;           ;           ;           ;
140     000100      ;           ;           ;           ;           ;           ;
141
142     ;           ;           ;           ;           ;           ;
143     000007      ;           ;           ;           ;           ;           ;
144     000010      ;           ;           ;           ;           ;           ;
    
```

```

145          000020          HOSTAT =20          ;HEADS OUT STATUS
146          000040          COSTAT =40          ;COVER OPEN STATUS
147          000100          HSSTAT =100         ;HEAD SELECT STATUS
148          000400          DSESTAT =400        ;DRIVE SELECT ERROR STATUS
149          001000          VCSTAT =1000       ;VOLUME CHECK STATUS
150          002000          WGESTAT =2000      ;WRITE GATE ERROR STATUS
151          004000          SPDSTAT =4000      ;SPIN ERROR STATUS
152          010000          STOSTAT =10000     ;SEEK TIMEOUT ERROR STATUS
153          020000          WLSTAT =20000     ;WRITE LOCK STATUS
154          040000          HCESTAT =40000    ;HEAD CURRENT ERROR STATUS
155          100000          WDESTAT =100000   ;WRITE DATA ERROR STATUS
156
157          ; P-CLOCK REGISTERS
158          172540          CLKCSR =172540     ;CLOCK CONTROL AND STATUS REGISTER
159          172542          CLKCSB =172542     ;CLOCK COUNT SET BUFFER
160          172544          CLKCTR =172544     ;CLOCK COUNTER
161
162 002230          ENDMOD
163
164
165
166          .SBTTL GLOBAL DATA SECTION
167
168 002230          BGNMOD GLBDAT
169
170          ; TABLE OF OPERATION MESSAGES
171
172 002230 000000          OPMSG: .WORD 0          ;FILLER
173 002232 005375          .WORD MWRCHK        ;MESSAGE FOR WRITE CHECK
174 002234 005420          .WORD MGTSTA        ;
175 002236 005350          .WORD MSEEK         ;
176 002240 005365          .WORD MREADH       ;
177 002242 005406          .WORD MWRITE       ;
178 002244 005354          .WORD MREAD        ;
179 002246 005503          .WORD MWRSET       ;
180 002250 005432          .WORD MDATCP       ;
181 002252 005451          .WORD MHDRCP       ;
182 002254 005550          .WORD MCYLUP       ;
183 002256 005537          .WORD MLOAD        ;
184 002260 005577          .WORD MINOUT       ;
185 002262 005560          .WORD MOUTIN       ;
186 002264 005620          .WORD MFOLWRT      ;
187 002266 005640          .WORD MREVSK       ;
188 002270 005671          .WORD MFWSK        ;
189 002272 005756          .WORD MRESKO       ;
190 002274 005722          .WORD MFWSKO       ;
191 002276 006012          .WORD MBADAD       ;
192 002300 005467          .WORD MAOMDR       ;
193 002302 000000          T.DRIVE: .WORD 0          ;
194 002304 000000          JJJ: .WORD 0          ;
195 002306 000000          HLMTW: .WORD 0         ;
196 002310 000000          CLRBYT: .WORD 0        ;
197 002312 000000          NXTHL: .WORD 0        ;
198 002314 000000          GBND: .WORD 0         ;
199 002316 000000          CAMSK: .WORD 0         ;
200 002320 000000          DIRMSK: .WORD 0        ;
201 002322 000000          MDCYL: .WORD 0         ;

```

```

202
203
204 002324 010333
205 002326 010444
206 002330 010662
207 002332 010634
208 002334 010617
209 002336 010607
210 002340 010714
211 002342 000000
212 002344 010572
213 002346 010554
214 002350 000000
215 002352 010540
216 002354 010505
217 002356 010523
218 002360 000000
219 002362 010455
220
221
222 002364 005072
223 002366 005074
224 002370 005134
225 002372 005174
226 002374 005234
227 002376 005242
228 002400 005302
229 002402 005304
230 002404 005344
231 002406 005346
232
233
234
235 002410 000000
236 002412 000000
237 002414 000000
238 002416 000000
239 002420 000000
240 002422 000000
241 002424 000000
242 002426 000000
243 002430 000000
244 002432 000000
245
246
247 002434 000002
248 002436 000006
249 002440 000011
250 002442 000014
251 002444 000021
252 002446 000026
253 002450 000033
254 002452 000042
255 002454 000051
256 002456 000200
257 002460 000377
258

; RESTBL: TABLE OF RESULT NAME MESSAGE ADDRESSES
; .WORD MCERR ;CONTROLLER ERROR
; .WORD MDRERR ;DRIVE ERROR
; .WORD MNEERR ;NON-EXISTANT MEMORY ERROR
; .WORD MFLERR ;HEADER NOT FOUND-DATA LATE
; .WORD MWDERR ;HEADER OR DATA ERROR
; .WORD MOPERR ;OPERATION INCOMPLETE
; .WORD MNDRST ;NO DRIVE STATUS AVAILABLE
; .WORD 0
; .WORD MWDERR ;WRITE DATA ERROR
; .WORD MHCERR ;HEAD CURRENT ERROR
; .WORD 0
; .WORD MSTERR ;SEEK TIMEOUT ERROR
; .WORD MSPERR ;SPINDLE ERROR
; .WORD MWGERR ;WRITE GATE ERROR
; .WORD 0
; .WORD MDSERR ;DRIVE SELECT ERROR

; PATTBL: PATTERN TABLE
; .WORD PAT1
; .WORD PAT2
; .WORD PAT3
; .WORD PAT4
; .WORD PAT5
; .WORD PAT6
; .WORD PAT7
; .WORD PAT8
; .WORD PAT9
; .WORD PAT10

; SUBSTK: SUBROUTINE CALLING STACK
; .WORD 0 ;STACK IS 12 WORDS LONG
; .WORD 0
; .WORD 0
; .WORD 0
; .WORD 0
; .WORD 0
; .WORD 0
; .WORD 0
; .WORD 0
; .WORD 0

; RLO1 TABLE OF CYLINDERS
; T25TBL: .WORD 2 ;TABLE OF DIFFERENCES
; .WORD 6
; .WORD 9.
; .WORD 12.
; .WORD 17.
; .WORD 22.
; .WORD 27.
; .WORD 34.
; .WORD 41.
; .WORD 128.
; .WORD 255.
    
```

259  
 260 002462 000004  
 261 002464 000014  
 262 002466 000022  
 263 002470 000030  
 264 002472 000042  
 265 002474 000054  
 266 002476 000066  
 267 002500 000104  
 268 002502 000122  
 269 002504 000400  
 270 002506 000777  
 271  
 272  
 273  
 274 002510  
 275 002550  
 276  
 277  
 278 002610 002  
 279 002611 007  
 280 002612 016  
 281 002613 024  
 282 002614 033  
 283 002615 041  
 284 002616 046  
 285 002617 055  
 286 002620 064  
 287 002621 072  
 288 002622 101  
 289 002623 110  
 290 002624 115  
 291 002625 124  
 292 002626 133  
 293 002627 141  
 294 002630 146  
 295 002631 154  
 296 002632 161  
 297 002633 170  
 298 002634 177  
 299 002635 206  
 300 002636 213  
 301 002637 222  
 302 002640 230  
 303 002641 235  
 304 002642 244  
 305 002643 252  
 306 002644 261  
 307 002645 270  
 308 002646 275  
 309 002647 303  
 310 002650 312  
 311 002651 317  
 312 002652 326  
 313 002653 334  
 314 002654 343  
 315 002655 352

;RLO2 TABLE OF CYLINDERS  
 T25TB2: .WORD 4  
 .WORD 12.  
 .WORD 18.  
 .WORD 24.  
 .WORD 34.  
 .WORD 44.  
 .WORD 54.  
 .WORD 68.  
 .WORD 82.  
 .WORD 256.  
 .WORD 511.

; TABLE TO BE USED TO BUILD AND STORE THE CYLINDERS

T33TBL: .BLKW 16.  
 TBT: .BLKW 16.

CYLTBL: .BYTE 2 ;TABLE OF DEFAULT CYLINDERS  
 .BYTE 7.  
 .BYTE 14.  
 .BYTE 20.  
 .BYTE 27.  
 .BYTE 33.  
 .BYTE 38.  
 .BYTE 45.  
 .BYTE 52.  
 .BYTE 58.  
 .BYTE 65.  
 .BYTE 72.  
 .BYTE 77.  
 .BYTE 84.  
 .BYTE 91.  
 .BYTE 97.  
 .BYTE 102.  
 .BYTE 108.  
 .BYTE 113.  
 .BYTE 120.  
 .BYTE 127.  
 .BYTE 134.  
 .BYTE 139.  
 .BYTE 146.  
 .BYTE 152.  
 .BYTE 157.  
 .BYTE 164.  
 .BYTE 170.  
 .BYTE 177.  
 .BYTE 184.  
 .BYTE 189.  
 .BYTE 195.  
 .BYTE 202.  
 .BYTE 207.  
 .BYTE 214.  
 .BYTE 220.  
 .BYTE 227.  
 .BYTE 234.

316	002656	361	.BYTE	241.	
317	002657	367	.BYTE	247.	
318	002660	375	.BYTE	253.	
319	002661	000	.BYTE	0	
320	002662	000401	.WORD	257.	
321	002664	000406	.WORD	262.	
322	002666	000415	.WORD	269.	
323	002670	000423	.WORD	275.	
324	002672	000432	.WORD	282.	
325	002674	000445	.WORD	293.	
326	002676	000454	.WORD	300.	
327	002700	000463	.WORD	307.	
328	002702	000471	.WORD	313.	
329	002704	000500	.WORD	320.	
330	002706	000507	.WORD	327.	
331	002710	000514	.WORD	332.	
332	002712	000523	.WORD	339.	
333	002714	000532	.WORD	346.	
334	002716	000540	.WORD	352.	
335	002720	000545	.WORD	357.	
336	002722	000553	.WORD	363.	
337	002724	000560	.WORD	368.	
338	002726	000567	.WORD	375.	
339	002730	000576	.WORD	382.	
340	002732	000605	.WORD	389.	
341	002734	000612	.WORD	394.	
342	002736	000621	.WORD	401.	
343	002740	000627	.WORD	407.	
344	002742	000634	.WORD	412.	
345	002744	000643	.WORD	419.	
346	002746	000651	.WORD	425.	
347	002750	000660	.WORD	432.	
348	002752	000667	.WORD	439.	
349	002754	000674	.WORD	444.	
350	002756	000702	.WORD	450.	
351	002760	000711	.WORD	457.	
352	002762	000716	.WORD	462.	
353	002764	000725	.WORD	469.	
354	002766	000733	.WORD	475.	
355	002770	000742	.WORD	482.	
356	002772	000751	.WORD	489.	
357	002774	000760	.WORD	496.	
358	002776	000766	.WORD	502.	
359	003000	000774	.WORD	508.	
360	003002	000774	.WORD	508.	
361	003004	000000	.WORD	0	
362	003006	000000	.WORD	0	;SUBROUTINE STACK INDEX POINTER
363					
364					
365	003010	000000	; OPERATIONAL FLAGS		
366	003012	000000	OPFLAG: .WORD	0	;OPERATION FLAGS
367	003014	000000	DONE: .WORD	0	;OPERATION COMPLETE FLAG
368	003016	000000	HADONE: .WORD	0	;HEAD ALIGNMENT DONE FLAG
369	003020	000000	ERHEAD: .WORD	0	;ADDRESS OF ERROR HEADER
370	003022	000000	MORECE: .WORD	0	;MORE THAN 1 COMPARE ERROR
371	003024	000000	ERRSWI: .WORD	0	;ERROR RETURN SWITCH
372	003026	000000	BSFLAG: .WORD	0	;BAD SECTOR FLAGS
			WRTSWI: .WORD	0	;WRITE SWITCH

373 003030 000000 TBLSTR: .WORD 0 ;TABLE STORAGE  
 374  
 375 003032 000000 RLBAS: .WORD 0 ;RL11 BASE ADDRESS  
 376 003034 000000 RLVEC: .WORD 0 ;RL11 VECTOR ADDRESS  
 377 003036 000000 RLDRV: .WORD 0 ;DRIVE NUMBER UNDER TEST  
 378  
 379 003040 000000 L.CS: .WORD 0 ;CONTROLLER REGISTER STORAGE  
 380 003042 000000 L.BA: .WORD 0 ;BEFORE OPERATION  
 381 003044 000000 L.DA: .WORD 0  
 382 003046 000000 L.MP: .WORD 0  
 383 003050 000000 T.CS: .WORD 0 ;CONTROLLER REGISTER STORAGE  
 384 003052 000000 T.BA: .WORD 0 ; AFTER OPERATION  
 385 003054 000000 T.DA: .WORD 0  
 386 003056 T.MP:  
 387 003056 000000 HDWRD1: .WORD 0 ;HEADER WORD STORAGE  
 388 003060 000000 HDWRD2: .WORD 0  
 389 003062 000000 HDWRD3: .WORD 0  
 390  
 391 003064 000000 T.STAT: .WORD 0 ;DRIVE STATE STORAGE  
 392  
 393 003066 000000 RESPARM: .WORD 0 ;PARAM BLOCK FOR REASON REPORT  
 394 003070 000000 .WORD 0  
 395 003072 000000 .WORD 0  
 396 003074 000000 .WORD 0  
 397 003076 000000 .WORD 0  
 398  
 399 003100 000000 DRVCNT: .WORD 0 ;DRIVE COUNT FOR DRIVES UNDER TEST  
 400 003102 000000 DIFAUG: .WORD 0 ;DIFFERENCE AUGMENT FOR SEEK  
 401 003104 000000 OLDCYL: .WORD 0 ;OLD CYLINDER  
 402 003106 000000 NEWCYL: .WORD 0 ;NEW CYLINDER  
 403 003110 000000 CURCYL: .WORD 0 ;CURRENT CYLINDER  
 404 003112 000000 DESDIF: .WORD 0 ;DESIRED DIFFERENCE  
 405 003114 000000 DESSGN: .WORD 0 ;DESIRED SIGN  
 406 003116 000000 DESHD: .WORD 0 ;DESIRED HEAD  
 407 003120 000000 DESSEC: .WORD 0 ;DESIRED SECTOR  
 408 003122 000000 TEMPO: .WORD 0 ;TEMPORARY STORAGE  
 409 003124 000000 TEMP1: .WORD 0 ;TEMPORARY STORAGE  
 410 003126 000000 TEMP2: .WORD 0 ;TEMPORARY STORAGE  
 411 003130 000000 TEMP3: .WORD 0 ;TEMPORARY STORAGE  
 412 003132 000000 TEMP4: .WORD 0 ;TEMPORARY STORAGE  
 413 003134 000000 TEMP5: .WORD 0 ;TEMPORARY STORAGE  
 414 003136 000000 TEMP6: .WORD 0 ;TEMPORARY STORAGE  
 415 003140 000000 TEMP7: .WORD 0 ;TEMPORARY STORAGE  
 416 003142 000000 TEMP8: .WORD 0 ;TEMPORARY STORAGE  
 418 ; TIMER STORAGE  
 419 003144 000000 OFIN: .WORD 0 ;ONE CYLINDER FORWARD INNER  
 420 003146 000000 OFINU: .WORD 0 ; UPPER  
 421 003150 000000 OFMID: .WORD 0 ;ONE CYLINDER FORWARD MIDDLE  
 422 003152 000000 OFMIDU: .WORD 0 ; UPPER  
 423 003154 000000 OFOUT: .WORD 0 ;ONE CYLINDER FORWARD OUTER  
 424 003156 000000 OFOUTU: .WORD 0 ; UPPER  
 425 003160 000000 ORIN: .WORD 0 ;ONE CYLINDER REVERSE INNER  
 426 003162 000000 ORINU: .WORD 0 ; UPPER  
 427 003164 000000 ORMID: .WORD 0 ;ONE CYLINDER REVERSE MIDDLE  
 428 003166 000000 ORMIDU: .WORD 0 ; UPPER  
 429 003170 000000 OROUT: .WORD 0 ;ONE CYLINDER REVERSE OUTER  
 430 003172 000000 OROUTU: .WORD 0 ; UPPER



171

431	003174	000000	HFIN:	.WORD	0	;128 CYLINDER FORWARD INNER
432	003176	000000	HFINU:	.WORD	0	; UPPER
433	003200	000000	HFOUT:	.WORD	0	;128 CYLINDER FORWARD OUTER
434	003202	000000	HFOUTU:	.WORD	0	; UPPER
435	003204	000000	HRIN:	.WORD	0	;128 CYLINDER REVERSE INNER
436	003206	000000	HRINU:	.WORD	0	; UPPER
437	003210	000000	HROUT:	.WORD	0	;128 CYLINDER REVERSE OUTER
438	003212	000000	HROUTU:	.WORD	0	; UPPER
439	003214	000000	AFMID:	.WORD	0	;256 CYLINDER FORWARD
440	003216	000000	AFMIDU:	.WORD	0	; UPPER
441	003220	000000	ARMID:	.WORD	0	;256 CYLINDER REVERSE
442	003222	000000	ARMIDU:	.WORD	0	; UPPER
443						
444	003224	000226	EXOCYL:	.WORD	150.	;EXPECTED TIME ONE CYLINDER
445	003226	001046	EXHCYL:	.WORD	550.	;EXPECTED TIME 128 CYLINDER
446	003230	001750	EXACYL:	.WORD	1000.	;EXPECTED TIME 256 CYLINDER
447	003232	000372	EXROT:	.WORD	250.	;EXPECTED ROTATION TIME
449	003234	000004	ERRVEC:	.WORD	4	;ERROR VECTOR
450						
451						
452	003236	000000	PASCNT:	.WORD	0	;PASS COUNTER (LOCAL TO A TEST)
453	003240	000000	COUNT:	.WORD	0	;A COUNTER (LOCAL TO A TEST)
454	003242	000000	ERRPOINT:	.WORD	0	;ERROR POINTER
455	003244		ERRCNT:	.BLKW	64.	;ERROR COUNTER FOR PROGRAM
456	003444	000000	PASNUM:	.WORD	0	;PASS NUMBER FOR PROGRAM
457	003446	000000	PSETNM:	.WORD	0	;COUNTER FOR PARAMETER SET NUMBER IN JSE
458	003450	000	LOCERR:	.BYTE	0	;LOCAL ERROR COUNTER
459	003451	000	NOERCT:	.BYTE	0	;INHIBIT ERROR COUNTING FLAG
460	003452	000000	TRPFLG:	.WORD	0	;HARDWARE TRAP OCCURANCE
461	003454	000000	PWRFLG:	.WORD	0	;POWER FAILURE OCCURANCE
462	003456	000000	XDELAY:	.WORD	0	
463	003460	000000	YDELAY:	.WORD	0	
464	003462	000000	MININC:	.WORD	0	
465	003464	000000	TEMP:	.WORD	0	
466	003466	000000	TIM.US:	.WORD	0	
467	003470	000000	TAG:	.WORD	0	
468	003472	000000	MAJINC:	.WORD	0	
469	003474	000000	CLKFLG:	.WORD	0	;FLAG INDICATING PRESENCE OF A P CLOCK
470	003476	000000	CLKADR:	.WORD	0	;POINTER TO DIAGNOSTIC MONITOR CLOCK TABLE
471						
472						
473						
474	003500	000000	BSFVAL:	.WORD	0	;BAD SECTORS FILES VALID FLAG
475						
476	003502		SBSFIL:	.BLKW	76	;SOFTWARE BAD SECTOR FILE
477	003676		FBSFIL:	.BLKW	76	;FACTORY BAD SECTOR FILE
478						
479	004072		IBUFF:	.BLKW	200	;INPUT BUFFER
480	004472		OBUFF:	.BLKW	200	;OUTPUT BUFFER
481						
482	005072	000000	PAT1:	.WORD	0	;PATTERN 1 (ALL ZEROS)
483	005074	177772	PAT2:	.WORD	177772	
484	005076	177777		.WORD	177777	
485	005100	177777		.WORD	177777	
486	005102	052525		.WORD	052525	
487	005104	052525		.WORD	052525	
488	005106	052525		.WORD	052525	

489	005110	177777		.WORD	177777
490	005112	177777		.WORD	177777
491	005114	052525		.WORD	052525
492	005116	052525		.WORD	052525
493	005120	177777		.WORD	177777
494	005122	052525		.WORD	052525
495	005124	177252		.WORD	177252
496	005126	177252		.WORD	177252
497	005130	172765		.WORD	172765
498	005132	172765		.WORD	172765
499					
500	005134	000003	PAT3:	.WORD	000003
501	005136	000000		.WORD	000000
502	005140	000000		.WORD	000000
503	005142	177777		.WORD	177777
504	005144	177777		.WORD	177777
505	005146	177777		.WORD	177777
506	005150	000000		.WORD	000000
507	005152	000000		.WORD	000000
508	005154	177777		.WORD	177777
509	005156	177777		.WORD	177777
510	005160	000000		.WORD	000000
511	005162	177777		.WORD	177777
512	005164	000000		.WORD	000000
513	005166	177777		.WORD	177777
514	005170	000000		.WORD	000000
515	005172	177777		.WORD	177777
516					
517	005174	025252	PAT4:	.WORD	025252
518	005176	052525		.WORD	052525
519	005200	052525		.WORD	052525
520	005202	125252		.WORD	125252
521	005204	125252		.WORD	125252
522	005206	125252		.WORD	125252
523	005210	052525		.WORD	052525
524	005212	052525		.WORD	052525
525	005214	125252		.WORD	125252
526	005216	125252		.WORD	125252
527	005220	052525		.WORD	052525
528	005222	125252		.WORD	125252
529	005224	052525		.WORD	052525
530	005226	125252		.WORD	125252
531	005230	052525		.WORD	052525
532	005232	125252		.WORD	125252
533					
534	005234	155555	PAT5:	.WORD	155555
535	005236	133333		.WORD	133333
536	005240	066666		.WORD	066666
537					
538	005242	121105	PAT6:	.WORD	121105
539	005244	150442		.WORD	150442
540	005246	064221		.WORD	064221
541	005250	132110		.WORD	132110
542	005252	055044		.WORD	055044
543	005254	026442		.WORD	026442
544	005256	013211		.WORD	013211
545	005260	105504		.WORD	105504

546	005262	042642			.WORD	042642
547	005264	021321			.WORD	021321
548	005266	110550			.WORD	110550
549	005270	044264			.WORD	044264
550	005272	022132			.WORD	022132
551	005274	011055			.WORD	011055
552	005276	104426			.WORD	104426
553	005300	042213			.WORD	042213
554						
555	005302	177777		PAT7:	.WORD	177777
556						
557	005304	045513		PAT8:	.WORD	045513
558	005306	122645			.WORD	122645
559	005310	151322			.WORD	151322
560	005312	064551			.WORD	064551
561	005314	132264			.WORD	132264
562	005316	055132			.WORD	055132
563	005320	026455			.WORD	026455
564	005322	113226			.WORD	113226
565	005324	045513			.WORD	045513
566	005326	122645			.WORD	122645
567	005330	151322			.WORD	151322
568	005332	064551			.WORD	064551
569	005334	132264			.WORD	132264
570	005336	055132			.WORD	055132
571	005340	026455			.WORD	026455
572	005342	113226			.WORD	113226
573						
574	005344	125252		PAT9:	.WORD	125252
575						
576	005346	155555		PAT10:	.WORD	155555
577						
578	005350			ENDMOD		
579						
580						
581						
585						
586						
587						
588				.SBTTL	GLOBAL MESSAGES	
589	005350			BGNMOD	GLBTXT	
590						
591	005350	123	113	040	MSEEK: .ASCIZ	/SK /
592	005354	122	104	040	MREAD: .ASCIZ	/RD DATA /
593	005365	122	104	040	MREADM: .ASCIZ	/RD HDR /
594	005375	127	122	124	MWRCHK: .ASCIZ	/WRT CHCK/
595	005406	127	122	124	MWRITE: .ASCIZ	/WRT DATA /
596	005420	107	105	124	MGTSTA: .ASCIZ	/GET STAT /
597	005432	127	111	124	MSTATCP: .ASCIZ	/WITH DATA CMP /
598	005451	127	111	124	MHDRCP: .ASCIZ	/WITH HDR CMP /
599	005467	106	117	122	M40HDR: .ASCIZ	/FOR 40 HDRS/
600	005503	127	111	124	MWRSET: .ASCIZ	/WITH RESET /
601	005517	117	120	105	MOPER: .ASCIZ	/OPER: /
602	005526	122	105	123	MRESLT: .ASCIZ	/RESULT: /
603	005537	125	116	114	MULOAD: .ASCIZ	/UNLD DRV/
604	005550	114	104	040	MCYLUP: .ASCIZ	/LD DRV /
605	005560	106	117	114	MOUTIN: .ASCIZ	/FOL 0 TO CC SK/

606	005577	106	117	114	MINOUT: .ASCIZ	/FOL 255 TO CC SK/
607	005620	106	117	114	MFOLWRT: .ASCIZ	/FOL WRT (NO SK)/
608	005640	101	104	112	MREVSX: .ASCIZ	/ADJ CYL WRITTN AFT REV SK/
609	005671	101	104	112	MFWDSK: .ASCIZ	/ADJ CYL WRITTN AFT FWD SK/
610	005722	123	113	040	MFWSKO: .ASCIZ	/SK FWD,WRT - SK REV,OVERWRT/
611	005756	123	113	040	MRESKO: .ASCIZ	/SK REV,WRT SK FWD,OVERWRT/
612	006012	117	116	040	MBADAD: .ASCIZ	/ON BAD SEC FILES/
613	006033	103	101	116	MBADSF: .ASCIZ	/CAN'T GET BAD SEC FILES/
614	006063	102	101	104	MFMTER: .ASCIZ	/BAD SEC FILE FMT ERR/
615	006110	124	117	040	MTMBS: .ASCIZ	/TO MANY BAD SEC /
616	006131	102	125	123	BASADD: .ASCIZ	/BUS ADD=/
617	006142	104	122	126	DRVNAM: .ASCIZ	/DRV=/
618	006147	116	117	040	DRVNAV: .ASCIZ	/NO DRV FOR TST/
619	006166	104	122	126	NOPIR: .ASCIZ	/DRV DID NOT REC'R FROM PWR FAIL/
620	006226	122	114	103	CSNAM: .ASCIZ	/RLCS/
621	006233	122	114	102	BANAM: .ASCIZ	/RLBA/
622	006240	122	114	104	DANAM: .ASCIZ	/RLDA/
623	006245	122	114	115	MPNAM: .ASCIZ	/RLMP/
624	006252	117	120	040	LAB1: .ASCIZ	/OP INIT = /
625	006265	117	120	040	LAB2: .ASCIZ	/OP DONE = /
626	006300	127	117	122	MWORD: .ASCIZ	/WORD /
627	006306	111	116	124	MTOSLOW: .ASCIZ	/INTRPT TOO LATE/
628	006326	116	117	040	MORRES: .ASCIZ	/NO DRV RSPNSE/
629	006344	116	117	040	MNOINT: .ASCIZ	/NO INTRPT ON CMND COMPLETE/
630	006377	103	116	124	MCONHNG: .ASCIZ	/CNTLR HUNG /
631	006413	105	122	122	MNOCLR: .ASCIZ	/ERR DID NOT CLR/
632	006433	126	117	114	VCNRST: .ASCIZ	/VOL CHK NOT RSET/
633	006454	125	116	130	UNXERR: .ASCIZ	/UNXPCTED ERR/
634	006471	040	124	105	TSTLAB: .ASCIZ	/ TEST /
652	006477	117	125	124	P2T03E: .ASCIZ	/OUT GRD BAND /
653	006515	111	116	103	P2T04E: .ASCIZ	/INC SK FWD HD 0/
654	006535	111	116	103	P2T05E: .ASCIZ	/INC SK REV HD 0/
655	006555	111	116	103	P2T06E: .ASCIZ	/INC SK FWD HD 1/
656	006575	111	116	116	P2T07E: .ASCIZ	/INN GRD BAND /
657	006613	111	116	103	P2T08E: .ASCIZ	/INC SK REV HD 1/
658	006633	123	113	000	P2T09E: .ASCIZ	/SK/
659	006636	106	127	104	P2T10E: .ASCIZ	/FWD OSC SK/
660	006651	122	105	126	P2T11E: .ASCIZ	/REV OSC SK/
661	006664	123	113	040	P2T12E: .ASCIZ	/SK TIMING/
662	006676	102	123	103	P2T13E: .ASCIZ	/BSC RD DATA/
663	006712	127	122	124	P2T14E: .ASCIZ	&WRT/RD DATA (P1)&
664	006733	123	120	111	P2T15E: .ASCIZ	/SPINDLE ROT TIMING/
665	006756	127	122	124	P2T16E: .ASCIZ	&WRT/RD DATA (P2)&
666	006777	127	122	124	P2T17E: .ASCIZ	/WRT LCK ERR AND DATA PROT/
667	007031	101	104	112	P2T18E: .ASCIZ	/ADJ CYL INTERFNCE/
668	007053	117	126	105	P2T19E: .ASCIZ	/OVERWRT/
669	007063	123	113	040	SKTMES: .ASCIZ	/SK TIMES /
670	007075	123	120	111	SRTMES: .ASCIZ	/SPINDLE ROT TIME /
671	007117	050	111	116	VALDES: .ASCIZ	/(IN 100'S OF U-SEC)/
672	007143	101	120	120	MAPROX: .ASCIZ	/APPROX /
673	007153	111	116	116	LABIN: .ASCIZ	/INNER/
674	007161	115	111	104	LABMID: .ASCIZ	/MIDDLE/
675	007170	117	125	124	LABOUT: .ASCIZ	/OUTER/
676	007176	115	101	130	LABEXP: .ASCIZ	/MAX TIME/
677	007207	061	040	103	LABOCF: .ASCIZ	/1 CYL FWD/
678	007221	061	040	103	LABOCR: .ASCIZ	/1 CYL REV/
679	007233	115	111	104	LABHCF: .ASCIZ	/MID CYL FWD/

680	007247	115	111	104	LABMCR:	.ASCIZ	/MID CYL REV/
681	007263	115	101	130	LABACF:	.ASCIZ	/MAX CYL FWD/
682	007277	115	101	130	LABACR:	.ASCIZ	/MAX CYL REV/
684	007313	110	104	123	HDMOVF:	.ASCIZ	/HDS FAILED TO MV IN 10 TRYS/
702	007347	122	105	123	OPR12:	.ASCIZ	/RESET WRT LCK /
703	007366	117	116	040	OPR1A:	.ASCIZ	/ON /
704	007372	117	116	040	OPR1B:	.ASCIZ	/ON DRV /
705	007402	125	116	104	UNDTST:	.ASCIZ	/UNDER TEST/
706	007415	123	105	124	OPR004:	.ASCIZ	/SET WRT LCK /
707	007432	104	111	106	DIFWD:	.ASCIZ	/DIFF /
708	007440	123	107	116	SGNWD:	.ASCIZ	/SGN /
709	007445	110	104	040	HOWD:	.ASCIZ	/NO /
710	007451	123	105	103	SECWD:	.ASCIZ	/SEC /
711	007456	103	131	114	CYLWD:	.ASCIZ	/CYL /
712	007463	106	122	117	FRMWD:	.ASCIZ	/FROM /
713	007471	040	102	131	BYPSSM:	.ASCIZ	/ BYPASSED /
714	007504	122	117	125	SEQMES:	.ASCIZ	/ROUTINE TRACE SEQ:/
715	007527	104	122	126	STAMES:	.ASCIZ	/DRV STAT/
716	007540	102	101	104	BSNSTR:	.ASCIZ	/BAD SEC FILES NOT STRD. ALL SEC ASSUMED OK./
717	007614	124	117	124	TCERR:	.ASCIZ	/TOTAL CMP ERRS: /
718	007635	104	122	111	NOCTLR:	.ASCIZ	/DRIVE DROPPED - NO CONTROLLER/
719	007673	104	122	111	NOTRDY:	.ASCIZ	/DRIVE DROPPED - DID NOT RESPOND WITH "READY"/
720	007750	124	105	123	NOTST1:	.ASCIZ	/TEST 1 CANNOT BE PERFORMED...P-CLOCK IS NOT AVAILABLE/
721	010036	122	105	123	NTST1A:	.ASCIZ	/RESOLUTION OF A P-CLOCK IS REQUIRED TO MEASURE SEEK TIME/<15><12>
722	010131	124	105	123	NOTST4:	.ASCIZ	/TEST 4 CANNOT BE PERFORMED...P CLOCK IS NOT AVAILABLE/
723	010217	122	105	123	NTST4A:	.ASCIZ	/RESOLUTION OF A P CLOCK IS REQUIRED TO MEASURE ROTATIONAL TIMING/<15><11>
724							
725							
726							
727	010322	104	122	126	MDRDY:	.ASCIZ	/DRV RDY /
728	010333	103	117	116	MCERR:	.ASCIZ	/CONT ERR /
729	010345	110	104	122	MMCRC:	.ASCIZ	/MDR CRC/
730	010355	104	101	124	MDCRC:	.ASCIZ	/DATA CRC/
731	010366	110	104	122	MNF:	.ASCIZ	/MDR NOT FND/
732	010402	104	101	124	MOLT:	.ASCIZ	/DATA LATE/
733	010414	110	104	122	MFCRC:	.ASCIZ	EMDR NOT FND/MDR CRC/OPIE
734	010444	104	122	126	MDRERR:	.ASCIZ	/DRV ERR /
743	010455	104	122	126	MDSERR:	.ASCIZ	/DRV SEL ERR /
744	010472	104	122	126	MDRVST:	.ASCIZ	/DRV STATE /
745	010505	123	120	111	MSPERR:	.ASCIZ	/SPIN TIMEOUT /
746	010523	127	122	124	MWGERR:	.ASCIZ	/WRT GAT ERR /
747	010540	123	113	040	MSTERR:	.ASCIZ	/SK TIMEOUT /
748	010554	110	105	101	MHCERR:	.ASCIZ	/HEAD CUR ERR /
749	010572	127	122	124	MHDERR:	.ASCIZ	/WRT DAT ERR /
750	010607	117	120	122	MOPERR:	.ASCIZ	/OPR-INC/
751	010617	110	104	122	MHDERR:	.ASCIZ	EMDR/DAT ERR E
752	010634	110	104	122	MFLERR:	.ASCIZ	EMDR NOT FND/DAT LATE E
753	010662	116	117	116	MNEERR:	.ASCIZ	/NON-EXISTENT MEMORY /
754	010707	103	131	114	MCYLOC:	.ASCIZ	/CYL /
755	010714	103	101	116	MMDRST:	.ASCIZ	/CAN'T GET DRV STAT/
756	010737	125	116	113	MUNDEF:	.ASCIZ	/UNKN DRV STATE-NO RDY,NO ERR,HDS OUT/
757	011004	106	101	111	MRLFAL:	.ASCIZ	/FAIL TO RELD HDS AFTER ERR CLR/
758	011043	127	122	124	MWRTAB:	.ASCIZ	/WRT ABRTD/
759	011055	040	117	126	MEXERS:	.ASCIZ	/ OVR ERR LIMIT - UNIT DRPPD /
760	011112	040	105	122	MERRS:	.ASCIZ	/ ERR/
761	011117	207	377	377	BELL:	.ASCIZ	<207><377><377>
762							

```

~63
764 011123      111      123      040      I      RESULT SETTINGS
765 011127      040      123      102      RESE3: .ASCIZ /IS /
766             040      123      102      RESF4: .ASCIZ /SB /
~67
768 011134      040      111      116      I      RESULT CONDITIONS
769 011141      040      117      106      RESE5: .ASCIZ /IN /
770 011146      123      124      101      RESE6: .ASCIZ /OF /
771 011156      123      124      101      STATE2: .ASCIZ /STATE 2/
772 011166      123      124      101      STATE3: .ASCIZ /STATE 3/
776 011176      061      123      124      STATE5: .ASCIZ /STATE 5/
777 011207      065      060      060      C1C1S: .ASCIZ /1ST 3 MS/
778 011215      103      131      103      C500MS: .ASCIZ /500MS/
779 011224      104      101      124      CCYLUP: .ASCIZ /CYC UP/
780 011235      065      040      123      CAFDT: .ASCIZ /DATA XFR/
781             065      040      123      CSSEC: .ASCIZ /5 SEC/
782 011243      045      116      045      FMTOP1: .ASCIZ /#N#T#N#T#T#06#S#T#01#N/
783 011272      045      116      045      FMTOP2: .ASCIZ /#N#T#01#S1#T#01#N/
784 011314      045      116      045      FMTOP3: .ASCIZ /#N#T#01#S1#T#T#N/
785 011335      045      124      045      FMT1: .ASCIZ /#T#T/
786 011342      045      116      045      FMT1.1: .ASCIZ /#N#T#T/
787 011351      045      124      000      FMT2: .ASCIZ /#T/
788 011354      045      116      000      FMT3: .ASCIZ /#N/
789 011357      045      116      045      FMT4: .ASCIZ /#N#T#T#N/
790 011370      045      116      045      FMT5: .ASCIZ /#N#T#06#S1#T#01/
791 011410      045      116      045      FMT6: .ASCIZ /#N#S11#T#S4#T#S4#T#S4#T#S4#T#S2#T/
792 011452      045      116      045      FMT7: .ASCIZ /#N#T#06#S2#06#S2#06#S2#06#S3#03#S2#01#N/
793 011522      045      116      045      FMT8: .ASCIZ /#N#T#06#S2#06#S2#06#S2#06/
794 011554      045      116      045      FMT9: .ASCIZ /#N#T/
795 011561      045      124      C35      FMT11: .ASCIZ /#T#01/
796 011567      045      124      045      FMT12: .ASCIZ /#T#03/
797 011575      045      116      045      FMT13: .ASCIZ /#N#S11#T#03#S1#T#03#S1#T#01#S1#T#01/
798 011641      045      116      045      FMT14: .ASCIZ /#N#T#T#03#S1#T#06#S1#T#06/
799 011673      045      116      045      FMT15: .ASCIZ /#N#S11#T#03#S1#T#06#S1#T#06/
800 011727      045      116      045      FMT16: .ASCIZ /#N#S5#06/
801 011740      045      123      061      FMT17: .ASCIZ /#S10#T#N#S11#06#N/
802 011762      045      116      045      FMT18: .ASCIZ /#N#S15#T#S5#T#S4#T#S5#T#N/
803 012014      045      124      045      FMT19: .ASCIZ /#T#S4#D6#S4#D6#S4#D6#S4#D6#N/
804 012051      045      124      045      FMT20: .ASCIZ /#T#S2#D6#S14#D6#S4#D6#N/
805 012101      045      124      045      FMT21: .ASCIZ /#T#S12#D6#S14#D6#N/
806 012124      045      116      045      FMT22: .ASCIZ /#N#S11#T#03#S1#T#01#S1#T#02/
807 012160      045      124      045      FMT23: .ASCIZ /#T#T#T#01#N/
808 012174      045      116      045      FMT24: .ASCIZ /#N#T/
809 012201      045      116      045      FMT25: .ASCIZ /#N#D2#T/
810 012211      045      116      045      FMT26: .ASCIZ /#N#S1#T#D4#T#T#D3#N/
811 012235      045      116      045      FMT27: .ASCIZ /#N#T#D3#T#D3#N/
812 012254      045      116      045      FMT28: .ASCIZ /#N#T#T#T/
813
814 012265
815
820
                                ENDMOD
    
```

```

1          .SBTTL  ERROR MESSAGES
2 012266   BGNMOD  GLBERR
3          :      ERR1  R3 POINTS TO RESULT MESSAGE
4          :      :      RESULT: (R3)
5
6          :      ERR2  R5 POINTS TO RESULT NAME
7          :      :      RESULT: (R3) IS 1 SB 0
8
9          :      ERR3  R3 POINTS TO RESULT NAME
10         :      :      RESULT: (R3) IS 0 SB 1
11
12         :      ERR4  R3 POINTS TO RESULT NAME
13         :      :      R4 POINTS TO RESULT CONDITIONS
14         :      :      RESULT: (R3) IS 1 SB 0 (R4)
15
16         :      ERR5  R3 POINTS TO RESULT NAME
17         :      :      R4 POINTS TO RESULT CONDITJONS
18         :      :      RESULT: (R3) IS 0 SB 1 (R4)
19
20         :      ERR6  RESULT ROUTINE DETERMINES WHICH ERROR(S) ARE SET AND
21         :      :      REPORTS ALL
22         :      :      RESULT: "ERROR" IS 1 SB 0
23
24         :      ERR7  DRIVE STATE ERROR REPORT
25         :      :      R3 CONTAINS EXPECTED STATE
26         :      :      T. STAT CONTAINS BAD STATE
27         :      :      RESULT: DRIVE STATE IS (T. STAT) SB (R3)
28
29         :      ERR8  HEAD POSITIONING ERROR REPORT
30         :      :      NEWCYL CONTAINS EXPECTED CYLINDER
31         :      :      HDWRD1 CONTAINS BAD CYLINDER
32         :      :      RESULT: CYLINDER IS (HDWRD1) SB (NEWCYL)
33
34         :      ERR9  UTILITY RESULT REPORT
35         :      :      R3 POINTS TO RESULT NAME
36         :      :      R4 POINTS TO VALUE 1
37         :      :      R5 POINTS TO VALUE 2
38         :      :      RESULT: (R3-NAME) IS (R4-VALUE 1) SB (R5 VALUE 2)
39
40         :      ERR10 COMPARE ERROR REPORT
41         :      :      R3 CONTAINS THE BAD WORD NUMBER
42         :      :      R4 POINTS TO BAD WORD
43         :      :      R5 POINTS TO GOOD WORD
44         :      :      RESULT: WORD (R3) IS (R4) SB (R5)
45
46
47 012266   BGNMSG  ERR1  NOERCT          ;TEST IF ERROR COUNTING INHIBITED
48 012266   105737  003451  TSTB          ;YES - SKIP
49 012272   001002          BNE          ;ELSE BUMP ERROR COUNT
50 012274   005277  170742  INC          ;STORE R1
51 012300   010146          14:  MOV      R1,-(SP) ;REPORT OPERATION
52 012302   004737  025060  JSR      PC,RPTOP ;SET PARAM NUMBER
53 012306   012721  000001  MOV      #1,(R1)+ ;INSERT MESSAGE ADDRESS POINTER
54 012312   010321          MOV      R3,(R1)+ ;REPORT RESULTS
55 012314   004737  025646  JSR      PC,RPTRES ;REPORT REMAINDER
56 012320   004737  026054  JSR      PC,RPTREM ;RESTORE R1
57 012324   012601          MOV      (SP)+,R1
  
```

58	012326	004737	016230		JSR	PC,CKERLM	;	GO CHECK IF ERROR COUNT EXCEEDED
59	012332			ENDMSG				
	012332			L10000:				
	012332	104423			TRAP	C#MSG		
60								
61	012334			BGNMSG	ERR2			
62	012334	005277	170702		INC	BERRPOINT	;	BUMP ERROR COUNT
63	012340	010146			MOV	R1, (SP)	;	STORE R1
64	012342	004737	025060		JSR	PC,RPTOP	;	REPORT OPERATION
65	012346	012721	000003		MOV	#3,(R1)	;	SET PARAM NUMBER
66	012352	010321			MOV	R3,(R1)	;	INSERT NAME ADD POINTER
67	012354	012721	000001		MOV	#1,(R1)	;	SET IS VALUE
68	012360	005021			CLR	(R1)	;	SET SB VALUE
69	012362	004737	025646		JSR	PC,RPTRES	;	REPORT RESULTS
70	012366	004737	026054		JSR	PC,RPTREM	;	REPORT REMAINDER
71	012372	012601			MOV	(SP),R1	;	RESTORE R1
72	012374	004737	016230		JSR	PC,CKERLM	;	GO CHECK IF ERROR COUNT EXCEEDED
73	012400			ENDMSG				
	012400			L10001:				
	012400	104423			TRAP	C#MSG		
74								
75	012402			BGNMSG	ERR3			
76	012402	005277	170634		INC	BERRPOINT	;	BUMP ERROR COUNT
77	012406	010146			MOV	R1, -(SP)	;	STORE R1
78	012410	004737	025060		JSR	PC,RPTOP	;	REPORT OPERATION
79	012414	012721	000003		MOV	#3,(R1)	;	SET PARAM NUMBER
80	012420	010321			MOV	R3,(R1)	;	INSERT NAME ADD POINTER
81	012422	005021			CLR	(R1)	;	SET IS VALUE
82	012424	012721	000001		MOV	#1,(R1)	;	SET SB VALUE
83	012430	004737	025646		JSR	PC,RPTRES	;	REPORT RESULTS
84	012434	004737	026054		JSR	PC,RPTREM	;	REPORT REMAINDER
85	012440	012601			MOV	(SP),R1	;	RESTORE R1
86	012442	004737	016230		JSR	PC,CKERLM	;	GO CHECK IF ERROR COUNT EXCEEDED
87	012446			ENDMSG				
	012446			L10002:				
	012446	104423			TRAP	C#MSG		
88								
89	012450			BGNMSG	ERR4			
90	012450	005277	170566		INC	BERRPOINT	;	BUMP ERROR COUNT
91	012454	010146			MOV	R1, -(SP)	;	STORE R1
92	012456	004737	025060		JSR	PC,RPTOP	;	REPORT OPERATION
93	012462	012721	000004		MOV	#4,(R1)	;	SET PARAM NUMBER
94	012466	010321			MOV	R3,(R1)	;	INSERT NAME ADD POINTER
95	012470	012721	000001		MOV	#1,(R1)	;	SET IS VALUE
96	012474	005021			CLR	(R1)	;	SET SB VALUE
97	012476	010411			MOV	R4,(R1)	;	INSERT ADD OF CONDITION POINTER
98	012500	004737	025646		JSR	PC,RPTRES	;	REPORT RESULTS
99	012504	004737	026054		JSR	PC,RPTREM	;	REPORT REMAINDER
100	012510	012601			MOV	(SP),R1	;	RESTORE R1
101	012512	004737	016230		JSR	PC,CKERLM	;	GO CHECK IF ERROR COUNT EXCEEDED
102	012516			ENDMSG				
	012516			L10003:				
	012516	104423			TRAP	C#MSG		
103								
104	012520			BGNMSG	ERR5			
105	012520	005277	170516		INC	BERRPOINT	;	BUMP ERROR COUNT
106	012524	010146			MOV	R1, (SP)	;	STORE R1



107	012526	004737	025060		JSR	PC,RPTOP		;REPORT OPERATION
108	012532	012721	000004		MOV	#4,(R1)		;SET PAPAM NUMBER
109	012536	010321			MOV	R3,(R1)		;INSERT NAME ADD POINTER
110	012540	005021			CLR	(R1)		;SET IS VALUE
111	012542	012721	000001		MOV	#1,(R1)		;SET SB VALUE
112	012546	010411			MOV	R4,(R1)		;INSERT ADD OF CONDITION POINTER
113	012550	004737	025646		JSR	PC,RPTRES		;REPORT RESULTS
114	012554	004737	026054		JSR	PC,RPTREM		;REPORT REMAINDER
115	012560	012601			MOV	(SP),R1		;RESTORE R1
116	012562	004737	016230		JSR	PC,CKERLM		;GO CHECK IF ERROR COUNT EXCFEDED
117	012566							
	012566				ENDMSG			
	012566				L10004:			
	012566	104423			TRAP	C#MSG		
118								
119	012570				BGNMSG	ERR6		
120	012570	105737	003451		TSTB	NOERCT		;TEST IF ERROR COUNTING INHIBITED
121	012574	001002			BNE	17#		;YES - SKIP
122	012576	005277	170440		INC	BERRPOINT		;ELSE BUMP ERROR COUNT
123	012602	010146			MOV	R1,-(SP)		;STORE R1
124	012604	010346			MOV	R3,-(SP)		;STORE R3
125	012606	010446			MOV	R4,-(SP)		;STORE R4
126	012610	010546			MOV	R5,-(SP)		;STORE R5
127	012612	004737	025060		JSR	PC,RPTOP		;REPORT OPERATION
128	012616	012721	000003		MOV	#3,(R1)		;SET PARAM NUMBER
129	012622	012761	000001	000002	MOV	#1,2(R1)		;INSERT IS VALUE
130	012630	005037	003130		CLR	,EMP3		;CLEAR FOR STATUS STORAGE
131	012634	013703	003050		MOV	T,CS,R3		;GET T.CS
132	012640	042703	177761		BIC	#177761,R3		;AND CLEAR ALL BUT FUNCTION
133	012644	022703	000004		CMF	#4,R3		;CHECK IF IT WAS GET STATUS
134	012650	001434			BEQ	1#		;YES - STATUS IS IN T.MP. SKIP
135	012652	012762	000003	000004	MOV	#GETSTAT,RLDA(R2)		;ELSE DO GET STATUS
136	012660	012703	000004		MOV	#4,R3		
137	012664	053703	003036		BIS	RLDRV,R3		
138	012670	010362	000000		MOV	R3,RLCS(R2)		
139	012674				WAITUS	#10.		;WAIT FOR CONTROLLER READY
140	012706	032762	000200	000000	BIT	#CRDYSK,RLCS(R2)		;TEST IF READY
141	012714	001003			BNE	10#		;YES - SKIP
142	012716	012703	001000		MOV	#BIT9,R3		;ELSE SET NO DRIVE STATUS BIT
143	012722	000413			BR	2#		;IN MESSAGE WORD AND SKIP
144	012724	016203	000006		MOV	RLMP(R2),R3		;STORE STATUS FOR REPORT
145	012730	010337	003130		MOV	R3,TEMP3		
146	012734	113703	003131		MOVB	TEMP3+1,R3		;GET ERROR BITS IN PROPER POSITION
147	012740	000402			BR	13#		
148	012742	113703	003057		MOVB	T,MP+1,R3		;GET ERROR BITS FROM MP REG
149	012746	042703	177442		BIC	#177442,R3		;CLEAR UNUSED BITS
150	012752	013704	003050		MOV	T,CS,R4		;GET ERROR BITS FROM CS REG
151	012756	042704	001777		BIC	#1777,R4		;CLEAR UNUSED BITS
152	012762	050403			BIS	R4,R3		;MAKE ONE WORD OF POSSIBLE ERRORS
153	012764	032703	002000		BIT	#OPIERR,R3		;TEST IF OPI SET
154	012770	001442			BEQ	115#		;NO - SKIP
155	012772	032703	010000		BIT	#HNFERR,R3		;TEST IF HDR NOT FOUND ERROR
156	012776	001026			BNE	107#		;YES - SKIP
157	013000	032703	004000		BIT	#MCRERR,R3		;TEST IF HDR CRC ERR
158	013004	001020			BNE	105#		;YES - SKIP
159	013006	012704	0 0607		MOV	#MOPERR,R4		;SET OPI ALONE MESSAGE
160	013012				PRINTB	#FMT28,#MRESLT,R4,#MERRS		;REPORT ERROR
	013012	012746	011112		MOV	#MERRS,-(SP)		

65

	013016	010446			MOV	R4,-(SP)	
	013020	012746	005526		MOV	#MRSLT,-(SP)	
	013024	012746	012254		MOV	#FMT28,-(SP)	
	013030	012746	000004		MOV	#4,-(SP)	
	013034	010600			MOV	SP,R0	
	013036	104414			TRAP	C#PNTB	
	013040	062706	000012		ADD	#12,SP	
161	013044	000430			BR	120#	;SKIP
162	013046	012704	010345	105#:	MOV	#MRCR,R4	;HDR CRC MESSAGE
163	013052	000757			BR	100#	
164	013054	032703	004000	107#:	BIT	#HCRCERR,R3	;TEST IF HCRC WITH HDR NOT FND
165	013060	001003			BNE	109#	;YES - SKIP
166	013062	012704	010366		MOV	#MNF,R4	;MESSAGE HEADER NOT FOUND
167	013066	000751			BR	100#	
168	013070	012774	010414	109#:	MOV	#MFCRC,R4	;MNF AND HCRC MESSAGE
169	013074	000746			BR	100#	;SKIP
170	013076	032703	004000	115#:	BIT	#DCKERR,R3	;TEST IF DATA CHECK SET, NOT OPI
171	013102	001403			BEQ	118#	;NO - SKIP
172	013104	012704	010355		MOV	#MDCRC,R4	;SET MESSAGE DATA CHECK
173	013110	000740			BR	100#	;SKIP
174	013112	032703	010000	118#:	BIT	#DLTERR,R3	;TEST IF DATA LATE ERROR
175	013116	001403			BEQ	120#	;NO - SKIP
176	013120	012704	010402		MOV	#DLT,R4	;SET MESSAGE DATA LATE
177	013124	000732			BR	100#	;SKIP
178	013126	012705	100000	120#:	MOV	#BIT15,R5	;SET BIT POINTER FOR TEST
179	013132	005004			CLR	R4	;CLEAR R4 FOR TABLE COUNT
180	013134	030503		3#:	BIT	R5,R3	;TEST IF BIT IS SET
181	013136	001005			BNE	6#	;YES - SKIP TO REPORT
182	013140	005724		4#:	TST	(R4).	;ELSE BUMP TABLE POINTER
183	013142	000241			CLC		;CLEAR CARRY
184	013144	006005			ROR	R5	;SHIFT BIT POINTER TO NEXT BIT
185	013146	001372			BNE	3#	;LOOP IF NOT 0
186	013150	000405			BR	7#	;ELSE REPORT REMAINDER
187	013152	016411	002324	6#:	MOV	RESTBL(R4),(R1)	;INSERT NAME ADDRESS
188	013156	004737	025646		JSR	PC,RPTRES	;REPORT RESULTS
189	013162	000766			BR	4#	;GET NEXT BIT
190	013164	004737	026054	7#:	JSR	PC,RPTREM	;REPORT REMAINDER
191	013170	005737	003130		TST	TEMP3	;TEST IF ANY NEW STATUS
192	013174	001414			BEQ	15#	;NO - SKIP
193	013176				PRINTB	#FMT17,#STAMES,TEMP3	
	013176	013746	003130		MOV	TEMP3,-(SP)	
	013202	012746	007527		MOV	#STAMES,-(SP)	
	013206	012746	011740		MOV	#FMT17,-(SP)	
	013212	012746	000003		MOV	#3,-(SP)	
	013216	010600			MOV	SP,R0	
	013220	104414			TRAP	C#PNTB	
	013222	062706	000010		ADD	#10,SP	
194	013226	032737	004000	003050	15#:	BIT	#DCKERR,T.CS
195	013234	001453			BEQ	25#	;NO - SKIP
196	013236	032737	002000	003050	BIT	#OPIERR,T.CS	;TEST IF OPI SET
197	013244	0C1047			BNE	25#	;YES - SKIP
198	013246	005037	003020		CLR	MORECE	;CLEAR COMPARE ERROR COUNT
199	013252	012701	000200		MOV	#128,R1	;SET COMPARE LENGTH
200	013256	012703	000001		MOV	#1,R3	;SET WORD COUNT
201	013262	012705	004472		MOV	#OBUFF,R5	;SET GOOD WORD POINTER
202	013266	012704	004072		MOV	#IBUFF,R4	;SET TEST WORD POINTER
203	013272	021514		18#:	CMP	(R5),(R4)	;CHECK WORD

```

204 013274 001427          BEQ      19$          ;GOOD SKIP
205 013276 023727 003020 000012    CMP      MORECE,#10.  ;TEST IF COMPARE LIMIT REACHED
206 013304 003021          BGT      20$          ;YES SKIP
207 013306          PRINTB  @FMT15,@MWORD,R3,@RESE3,(R4),@RESE4,(R5)
      013306 011546          MOV      (R5),-(SP)
      013310 012746 011127          MOV      @RESE4,-(SP)
      013314 011446          MOV      (R4),(SP)
      013316 012746 011123          MOV      @RESE3,-(SP)
      013322 010346          MOV      R3,-(SP)
      013324 012746 006300          MOV      @MWORD,-(SP)
      013330 012746 011673          MOV      @FMT15,-(SP)
      013334 012746 000007          MOV      #7,-(SP)
      013340 010600          MOV      SP,R0
      013342 104414          TRAP    C$PNTB
      013344 062706 000020          ADD      #20,SP
208 013350 005237 003020          20$:   TNC      MORECE          ;BUMP ERROR COUNTER
209 013354 022524          19$:   CMP      (R5)+,(R4)+  ;BUMP POINTERS
210 013356 005203          INC      R3          ;BUMP COUNTER
211 013360 005301          DEC      R1          ;DEC LENGTH COUNT
212 013362 001343          BNE     18$          ;LOOP IF NOT DONE
213 013364 005737 003020          25$:   TST      MORECE          ;TEST IF ANY COMPARE ERRORS
214 013370 001421          BEQ     27$          ;NO - SKIP
215 013372 012701 000200          MOV      #128,R1      ;SET COMPARE LENGTH
216 013376          PRINTB  @FMT27,@TCERR,MORECE,@RESE6,R1
      013376 010146          MOV      R1,-(SP)
      013400 012746 011141          MOV      @RESE6,-(SP)
      013404 013746 003020          MOV      MORECE,-(SP)
      013410 012746 007614          MOV      @TCERR,-(SP)
      013414 012746 012235          MOV      @FMT27,-(SP)
      013420 012746 000005          MOV      #5,-(SP)
      013424 010600          MOV      SP,R0
      013426 104414          TRAP    C$PNTB
      013430 062706 000014          ADD      #14,SP
217 013434 012605          27$:   MOV      (SP)+,R5      ;RESTORE R5, 4, 3, 1
218 013436 012604          MOV      (SP)+,R4
219 013440 012603          MOV      (SP)+,R3
220 013442 012601          MOV      (SP)+,R1
221 013444 004737 016230          JSR     PC,CKERLM    ;GO CHECK IF ERROR COUNT EXCEEDED
222 013450          ENDMMSG
      013450          L10005:
      013450 104423          TRAP    C$MSG
223
224 013452          BGNMSG  ERR7
225 013452 005277 167564          INC      $ERRPOINT   ;BUMP ERROR COUNT
226 013456 010146          MOV      R1,-(SP)   ;STORE R1
227 013460 004737 025060          JSR     PC,RPTOP    ;REPORT OPERATION
228 013464 012721 000003          MOV      #3,(R1)+  ;SET PARAM NUMBER
229 013470 012721 010472          MOV      @MDRVST,(R1)+ ;INSERT NAME ADD POINTER
230 013474 013721 003064          MOV      T,STAT,(R1)+ ;INSERT IS VALUE
231 013500 010311          MOV      R3,(R1) ;INSERT SB VALUE
232 013502 004737 025646          JSR     PC,RPTRES   ;REPORT RESULTS
233 013506 004737 026054          JSR     PC,RPTREM   ;REPORT REMAINDER
234 013512 012601          MOV      (SP)+,R1   ;RESTORE R1
235 013514 004737 016230          JSR     PC,CKERLM  ;GO CHECK IF ERROR COUNT EXCEEDED
236 013520          ENDMMSG
      013520          L10006:
      013520 104423          TRAP    C$MSG
    
```

```

237
238 013522          BGNMSG  ERR8
239 013522 005277 167514      INC      @ERRPOINT      ;BUMP ERROR COUNT
240 013526 010146          MOV      R1,-(SP)      ;STORE R1
241 013530 010346          MOV      R3,-(SP)      ;STORE R3
242 013532 004737 025060      JSR      PC,RPTOP      ;REPORT OPERATION
243 013536 012721 000003      MOV      @3,(R1)+      ;SET PARAM NUMBER
244 013542 012721 010707      MOV      @MCYLOC,(R1)+ ;INSERT NAME ADD POINTER
245 013546 013711 003056      MOV      @HWRD1,(R1)   ;GET HEADER WORD
246 013552 012703 000007      MOV      @7,R3        ;SET SHIFT COUNT
247 013556 000241          3$:      CLC
248 013560 006011          ROR      (R1)          ;ALIGN CHAR FOR PRINTING
249 013562 005303          DEC      R3           ; AS IS VALUE
250 013564 001374          BNE      3$
251 013566 005721          TST      (R1)+        ;BUMP PARAM POINTER
252 013570 013711 003106      MOV      @NEWCYL,(R1)  ;INSERT SB VALUE
253 013574 004737 025646      JSR      PC,RPTRES     ;REPORT RESULTS
254 013600 004737 026054      JSR      PC,RPTREM     ;REPORT REMAINDER
255 013604 012603          MOV      (SP)+,R3      ;RESTORE R3
256 013606 012601          MOV      (SP)+,R1      ;RESTORE R1
257 013610 004737 016230      JSR      PC,CKERLM    ;GO CHECK IF ERROR COUNT EXCEEDED
258 013614          ENDMSG
      013614          L10007:
      013614 104423          TRAP    C#MSG
259
260 013616          BGNMSG  ERR9
261 013616 005277 167420      INC      @ERRPOINT      ;BUMP ERROR COUNT
262 013622 010146          MOV      R1,-(SP)      ;STORE R1
263 013624 004737 025060      JSR      PC,RPTOP      ;REPORT OPERATION
264 013630 012721 000003      MOV      @3,(R1)+      ;SET PARAM NUMBER
265 013634 010321          MOV      R3,(R1)+      ;INSERT NAME ADD POINTER
266 013636 010421          MOV      R4,(R1)+      ;SET IS VALUE
267 013640 010521          MOV      R5,(R1)+      ;SET SB VALUE
268 013642 004737 025646      JSR      PC,RPTRES     ;REPORT RESULTS
269 013646 004737 026054      JSR      PC,RPTREM     ;REPORT REMAINDER
270 013652 012601          MOV      (SP)+,R1      ;RESTORE R1
271 013654 004737 016230      JSR      PC,CKERLM    ;GO CHECK IF ERROR COUNT EXCEEDED
272 013660          ENDMSG
      013660          L10010:
      013660 104423          TRAP    C#MSG
273 013662          BGNMSG  ERR10
274 013662 010146          MOV      R1,-(SP)      ;STORE R1
275 013664 005737 003020      TST      @MORECE      ;TEST IF 2ND BAD LINE
276 013670 001051          BNE      3$           ;YES - SKIP
277 013672 005277 167344      INC      @ERRPOINT      ;BUMP ERROR COUNT
278 013676 004737 025060      JSR      PC,RPTOP      ;REPORT OPERATION
279 013702          PRINTB @FMT5,@BASADD,RLBAS,@DRVNM,<@,RLDRV+1> ;REPORT ID
      013702 005046          CLR      -(SP)
      013704 153716 003037      BISB    RLDRV+1,(SP)
      013710 012746 006142      MOV      @DRVNM,-(SP)
      013714 013746 003032      MOV      RLBAS,-(SP)
      013720 012746 006131      MOV      @BASADD,-(SP)
      013724 012746 011370      MOV      @FMT5,-(SP)
      013730 012746 000005      MOV      @5,-(SP)
      013734 010600          MOV      SP,R0
      013736 104414          TRAP    C#PNTB
      013740 062706 000014      ADD     @14,SP
    
```

```

280 013744          PRINTB  @FMT14,@MRSLT,@MWORD,R3,@RESE3,(R4),@RESE4,(R5)
    013744 011546    MOV      (R5),-(SP)
    013746 012746 011127  MOV      @RESE4,-(SP)
    013752 011446    MOV      (R4),-(SP)
    013754 012746 011123  MOV      @RESE3,-(SP)
    013760 010346    MOV      R3,-(SP)
    013762 012746 006300  MOV      @MWORD,-(SP)
    013766 012746 005526  MOV      @MRSLT,-(SP)
    013772 012746 011641  MOV      @FMT14,-(SP)
    013776 012746 000010  MOV      @10,-(SP)
    014002 010600    MOV      SP,R0
    014004 104414    TRAP    C#PNTB
    014006 062706 000022  ADD      @22,SP
281 014012 000421    BR      4#
282 014014          3#: PRINTB  @FMT15,@MWORD,R3,@RESE3,(R4),@RESE4,(R5);REPORT DATA
    014014 011546    MOV      (R5),-(SP)
    014016 012746 011127  MOV      @RESE4,-(SP)
    014022 011446    MOV      (R4),-(SP)
    014024 012746 011123  MOV      @RESE3,-(SP)
    014030 010346    MOV      R3,-(SP)
    014032 012746 006300  MOV      @MWORD,-(SP)
    014036 012746 011673  MOV      @FMT15,-(SP)
    014042 012746 000007  MOV      @7,-(SP)
    014046 010600    MOV      SP,R0
    014050 104414    TRAP    C#PNTB
    014052 062706 000020  ADD      @20,SP
283 014056 005237 003020  4#: INC      MORECE          ;INC COMPARE ERROR COUNT
284 014062 012601          MOV      (SP)+,R1          ;RESTORE R1
285 014064 004737 016230  JSR      PC,CKERLM        ;GO CHECK IF ERROR COUNT EXCEEDED
286 014070          ENDMMSG
    014070          L10011: TRAP    C#MSG
    014070 104423          ENDMOD
287 014072          ;LOAD PROTECTION TABLE
288          BGNP#OT
289          .WORD    0          ;OFFSET OF CSR IN P-TABLE
290 014072          .WORD    -1         ;NOT A MASS-BUS DRIVE
291 014072 000000          .WORD    DRSB         ;OFFSET OF DRIVE IN P-TABLE
292 014074 177777          .EVEN
293 014076 000010          BGNMOD  HPTCODE
294 014100          BGNHW
295          .WORD    L10013-L#HW/2
296          .WORD    174400   ;CSR BASE ADDRESS DEFAULT
297          .WORD    160      ;VECTOR DEFAULT
298 014100 000006          .WORD    240         ;PRIORITY DEFAULT
299 014100          .WORD    1          ;TYPE OF DRIVE
300 014102 174400          .WORD    0          ;DRIVE NUMBER DEFAULT
301 014104 000160          .WORD    1          ;RL11 CONTROLLER
302 014106 000240          ENDMHW
303 014110 000001          L10013: ENDMOD
304 014112 000000
305 014114 000001          BGNMOD  SPTCODE
306 014116
307 014116
308
309 014116

```

310 014116  
 014116 000006  
 311 014120 000000  
 312  
 313  
 314  
 315  
 316  
 317  
 318  
 319 014122 000000  
 320 014124 000377  
 321 014126 000000  
 322 014130 000024  
 323 014132 000012  
 324 014134  
 014134  
 325 014134  
 326  
 327 014134  
 332 014134  
 014134 000010  
 014136 026340  
 014140 030276  
 014142 031014  
 014144 031230  
 014146 032062  
 014150 033172  
 014152 034210  
 014154 035424  
 334 014156  
 335  
 336

BGNSW  
 MISWIW: .WORD L10014-L#SW/2  
 0  
 LOLIMW: .WORD 0  
 HILIMW: .WORD 255.  
 HEADW: .WORD 0  
 ERLIMW: .WORD 20.  
 DCLIMW: .WORD 10.  
 ENDSW  
 L10014:  
 ENDMOD  
 BGNMOD DSPCODE  
 DISPATCH 8  
 .WORD 8  
 .WORD T1  
 .WORD T2  
 .WORD T3  
 .WORD T4  
 .WORD T5  
 .WORD T6  
 .WORD T7  
 .WORD T8  
 ENDMOD

;BIT 0 = USE ALL CYLINDERS  
 ;BIT 1 = USE ALL SECTORS  
 ;BIT 2 = EXECUTE DRIVE SELECT TEST  
 ;BIT 3 = EXECUTE HEAD ALIGNMENT  
 ;BIT 12 = MFAD SELECT SUPPLIED FLAG  
 ;BIT 13 = HILIMIT SPECIFIED FLAG  
 ;BIT 14 = LO LIMIT SPECIFIED FLAG  
 ;BIT 15 = DO MANUAL INTERVENTION

;ERROR LIMIT  
 ;COMPARE ERROR LIMIT

```

1          .SBTTL  INITIALIZATION SECTION
2
3 014156   BGNMOD  INITCODE
4 014156   BGNINIT
5
6          ;CHECK FOR PRESENCE OF A P-CLOCK
7 014156   005037   003474   CLR      CLKFLG      ;CLEAR CLOCK FLAG
8 014162   012700   000120   CLOCK    P,CLKADR   ;P-CLOCK?
          014166   104462   MOV      @'P,RO
          014170   010037   003476   TRAP    C:CLCK
9 014174   014174   103002   MOV      RO,CLKADR
          014174   103002   BNCOMPLETE 1$      ;BRANCH IF NO P CLOCK
10 014176   005237   003474   BCC     1$
11 014202   012700   000340   1$:    INC      CLKFLG      ;INDICATE PRESENCE OF A P CLOCK
          014202   012700   000340   SETPRI  @340        ;SET PRIORITY TO 7 TO INHIBIT ALL INTERRUPTS
          014206   104441   MOV      @340,RO
          014210   104433   TRAP    C:SPRI
12 014210   014210   104433   BRESET          ;FOR LSI-11 CPU'S
          014212   104450   TRAP    C:RESET
13 014212   014212   104450   MANUAL          ;CHECK IF MANUAL INTERVENTION ALLOWED
          014214   103403   TRAP    C:MANI
14 014214   042737   100014   014120   BCOMPLETE 2$      ;YES - SKIP
          014216   042737   100014   014120   BIC      @MITEST!DRSELT!HDALIGN,MISWIW ;CLEAR ALL MANUAL
15 014224   005037   003006   2$:    CLR      SSINDX      ;CLEAR SUBROUTINE STACK INDEX
          014230   012700   000034   READEF  @EF.PWR     ;POWER FAILURE
          014234   104447   MOV      @EF.PWR,RO
          014236   103005   TRAP    C:REFG
16 014236   013737   002012   003454   BNCOMPLETE 4$      ;NO. GO CHECK NEW PASS
          014240   000137   014660   MOV      L:UNIT,PWRFLG ;SET POWER FAIL FLAG
          014246   000137   014660   JMP     PWCON       ;GO SERVICE POWER FAIL
17 014252   012700   000040   4$:    READEF  @EF.START   ;CHECK IF START
          014256   104447   MOV      @EF.START,RO
          014260   103034   TRAP    C:REF
18 014260   013737   002012   BNCOMPLETE RESTART ;NO - SKIP
          014260   103034   BCC     RESTART
19
20          ; ON START INITIALIZE TO START AT FIRST DRIVE, CLEAR INTERNAL
21          ; PASS COUNT, AND ERROR COUNT.
22
23 014262   013737   002012   003100   MOV      L:UNIT,DRVCNT ;SET UP UNIT COUNT
24 014270   005037   003444   RSTRT: CLR      PASNUM   ;CLEAR PASS NUMBER
25 014274   012700   003244   MOV      @ERRCNT,RO
26 014300   012701   000100   MOV      @64.,R1      ;GET A COUNT
27 014304   005020   177777   1$:    CLR      (RO)+      ;CLEAR AN ERROR COUNTER STORAGE AREA
          014306   005301   DEC     R1
          014310   001375   BNE     1$          ;LOOP TILL ALL CLEARED
          014312   012737   003242   003242   MOV      @ERRCNT-2,ERRPOINT ;INIT ERROR POINTER
          014320   012737   177777   003446   MOV      @-1,PSETNM   ;SET PARAM SELECT TO INITIAL VALUE
          014326   012737   177777   003014   MOV      @-1,HADONE   ;PRESET HEAD ALIGN DONE FLAG
28 014334   032737   0-10000  014120   LAB:    BIT      @LOCYL,MISWIW ;TEST IF LO LIMIT SET
          014342   001002   BNE     5$          ;YES - SKIP
          014344   005037   014122   CLR      LOLIMW     ;ELSE CLEAR LO LIMIT
          014350   006432   5$:    BR      SETDON
          014352   RESTART:

```

```

43 014352          READEF  @EF.RESTART      ;CHECK IF RESTART
    014352 012700 000037  MOV      @EF.RESTART,RO
    014356 104447          TRAP      C$REFG
44 014360          BCOMPLETE  RSTRT      ;NO SKIP
    014360 103743          BCS      RSTRT
CONTINUE:
46 014362          READEF  @EF.CONTINUE    ;TEST IF CONTINUE
    014362 012700 000036  MOV      @EF.CONTINUE,RO
    014366 104447          TRAP      C$REFG
47 014370          BCOMPLETE  PWCON
    014370 103533          BCS      PWCON
48          : ON CONTINUE PICK UP UNIT LAST UNDER TEST
49 014372          READEF  @EF.NEW        ;CHECK IF STARTING NEW PASS
    014372 012700 000035  MOV      @EF.NEW,RO
    014376 104447          TRAP      C$REFG
50 014400          BCOMPLETE  PASNEW
    014400 103403          BCS      PASNEW
51 014402          NXPAS:
52 014402 005737 003100  TST      DRVCNT      ;TEST IF ALL UNITS CHECKED
53 014406 001013          BNE      SETDON      ;NO - SKIP
54 014410 005237 003444  PASNEW: INC      PASNUM      ;ELSE BUMP PASS COUNT
55 014414 012737 003242 003242  MOV      @ERRCNT-2,ERRPOINT ;INIT ERROR POINTER
56 014422 013737 002012 003100  MOV      L$UNIT,DRVCNT    ;GET ALL DRIVES
57 014430 012737 177777 003446  MOV      @-1,PSETNM      ;SET PARAM SELECT TO INITIAL
58 014436 005237 003446  SETDON: INC      PSETNM      ;NEXT SET OF PARAMETERS
59 014442 005337 003100          DEC      DRVCNT      ;DOWN COUNT DRIVE TOTAL
60 014446 062737 000002 003242  ADD      @2,ERRPOINT      ;UPDATE THE ERROR POINTER
61 014454 013700 003446          MOV      PSETNM,RO
62 014460 012702 003032          MOV      @RLBAS,R2
63 014464          GPHARD  RO,R1
    014464 104442          TRAP      C$GPHRD
    014466 010001          MOV      RO,R1
64 014470          BCOMPLETE  7$      ;SKIP IF GOOD PARAM
    014470 103406          BCS      7$
65 014472 005737 003454          TST      PWRFLG      ;RECENT POWER FAILURE
66 014476 001741          BEQ      NXPAS      ;NO
67 014500 005337 003454          DEC      PWRFLG      ;ACCOUNT FOR DRIVE
68 014504 000736          BR      NXPAS
69 014506 012122          7$: MOV      (R1)+,(R2)+    ;STORE PARAMETERS CSR
70 014510 012122          MOV      (R1)+,(R2)+    ;
71 014512 005721          TST      (R1)+      ;BUMP PAST PRIORITY
72 014514 012137 002302          MOV      (R1)+,T.DRIVE
73 014520 012122          MOV      (R1)+,(R2)+
74 014522 022737 000001 002302  CMP      @1,T.DRIVE
75 014530 001426          BEQ      65$
76 014532 012737 000776 002312  MOV      @510.,NXTHL
77 014540 012737 000777 002306  MOV      @511.,HLMTW
78 014546 012737 001000 002314  MOV      @512.,GBND
79 014554 012737 177600 002316  MOV      @177600,CAMSK
80 014562 012737 177600 002320  MOV      @177600,DIRMSK
81 014570 012737 177600 002322  MOV      @177600,HDCYL
82 014576 012737 177000 002310  MOV      @177000,CLRBYT
83 014604 000425          BR      PWCON
84
85 014606 012737 000377 002306 65$: MOV      @255.,HLMTW
86 014614 012737 000400 002314  MOV      @256.,GBND
87 014622 012737 077600 002316  MOV      @77600,CAMSK

```



```

88 014630 012737 077600 002320      MOV      #077600,DIRMSK
89 014636 012737 077600 002322      MOV      #077600,HDCYL
90 014644 012737 000376 002312      MOV      #254.,NXTHL
91 014652 012737 177400 002310      MOV      #177400,CLRBYT
92
93 014660 032737 020000 014120  PWCON:  BIT      #HICYL,MISWIW
94 014666 001003      BNE      1$
95 014670 013737 002306 014124      MOV      HLMTW,HILIMW
96 014676      1$:  SETVEC  RLVEC,#INTHLR,#340      ;SET UP VECTOR
   014676 012746 000340      MOV      #340,-(SP)
   014702 012746 016150      MOV      #INTHLR,-(SP)
   014706 013746 003034      MOV      RLVEC,-(SP)
   014712 012746 000003      MOV      #3,-(SP)
   C14716 104437      TRAP     C$SVEC
   014720 062706 000010      ADD      #10,SP
97 014724      SETPRI  #0      ;SET PRIORITY
   014724 012700 000000      MOV      #0,R0
   014730 104441      TRAP     C$SPRI
98 014732 013702 003032      MOV      RLBAS,R2      ;SET RL11 BASE ADDRESS POINTER
109      ;CHECK IF POWER FAILURE WAIT IS NEEDED
110
111 014736 005737 003454      TST      PWRFLG      ;NEEDED???
112 014742 001472      BEQ      8$      ;NO, SKIP
113
114 014744 013705 003036      MOV      RLDRV,R5      ;DRIVE SELECT
115 014750 052705 000200      BIS      #CRD,MSK,R5      ;SET CRDY
116 014754 010562 000000      MOV      R5,RLCS(R2)      ;SELECT DRIVE
117 014760 012701 000170      MOV      #120.,R1      ;INITIALIZE WAIT COUNT
118 014764 032762 000001 000000  9$:  BIT      #DRDYMSK,RLCS(R2)      ;DRIVE UP YET?
119 014772 001056      BNE      8$      ;YES START TEST
120
121 014774      WAITMS  #10.      ;WAIT A SECOND
122 015006 005301      DEC      R1      ;SIXTY GONE BY
123 015010 001365      BNE      9$      ;NO
124 015012      PRINTF  #FMT24,#NOPWR
   015012 012746 006166      MOV      #NOPWR,-(SP)
   015016 012746 012174      MOV      #FMT24,-(SP)
   015022 012746 000002      MOV      #2,-(SP)
   015026 010600      MOV      SP,R0
   015030 104417      TRAP     C$PNTF
   015032 062706 000006      ADD      #6,SP
125 015036      PRINTF  #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
   015036 005046      CLR      -(SP)
   015040 153716      BISB    RLDRV+1,(SP)
   015044 012746 006142      MOV      #DRVNAM,-(SP)
   015050 013746 003032      MOV      RLBAS,-(SP)
   015054 012746 006131      MOV      #BASADD,-(SP)
   015060 012746 011370      MOV      #FMT5,-(SP)
   015064 012746 000005      MOV      #5,-(SP)
   015070 010600      MOV      SP,R0
   015072 104417      TRAP     C$PNTF
   015074 062706 000014      ADD      #14,SP
126 015100      PRINTF  #FMT3
   015100 012746 011354      MOV      #FMT3,-(SP)
   015104 012746 000001      MOV      #1,-(SP)
   015110 010600      MOV      SP,R0
   015112 104417      TRAP     C$PNTF

```

```
127 015114 062706 000004      ADD    #4,SP
      015120      DODU    PSETNM      ;DROP DRIVE
      015120 013700 003446      MOV    PSETNM,RO
      015124 104451      TRAP   C%DODU
128 015126      DOCLN
      015126 104444      TRAP   C%DCLN
129 015130      8$:
130
131 015130      ENDINIT
      015130  L10015
      015130 104411      TRAP   C$INIT
132 015132      ENDMOD
133
```

C6,

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28

.SBTTL AUTO DROP SECTION

;THE AUTO DROP SECTION IS INVOKED BY THE DIAGNOSTIC SUPERVISOR WHENEVER THE  
 ;"ADR" FLAG IS SET BY THE OPERATOR. IT IS EXECUTED AFTER THE INITIALIZATION  
 ;CODE AND CHECKS THE DRIVE TO DETERMINE IF IT IS READY TO RECEIVE A COMMAND.  
 ;IF THE DRIVE IS NOT READY IT IS DROPPED FROM THE TEST CYCLE AND THE NEXT  
 ;DRIVE IS ACCESSED. IF THE DRIVE IS READY THE HARDWARE TESTS ARE PERFORMED  
 ;AFTER WHICH THE NEXT DRIVE IS ACCESSED.

```

BGNAUTO
      CLR      TRPFLG          ;CLEAR TRAP FLAG
      SETVEC  ERRVEC, @TRPHAN, @340 ;SET UP TRAP VECTOR TO DETECT
      MOV     @340, -(SP)
      MOV     @TRPHAN, -(SP)
      MOV     ERRVEC, -(SP)
      MOV     @3, -(SP)
      TRAP   C$SVEC
      ADD    @10, SP

      ;/NON EXISTENT CONTROLLER
      MOV     RLBAS, R2          ;GET RL11 BASE ADDRESS
      TST    RLCS(R2)          ;ACCESS DRIVE CONTROLLER ADDRESS
      TST    TRPFLG           ;DID TRAP OCCUR?
      BEQ    1$              ;BRANCH TO CHECK DRIVE IF TRAP DID NOT OCCUR
      PRINTF @FMT24, @NOCTLR ;ELSE, PRINT MSG. "DRIVE DROPPED NO CONTROLLER"
      MOV     @NUCTLR, -(SP)
      MOV     @FMT24, -(SP)
      MOV     @2, -(SP)
      MOV     SP, R0
      TRAP   C$PNTF
      ADD    @6, SP
      PRINTF @FMT5, @BASADD, RLBAS, @DRVNAM, <B, RLDRV+1,
      CLR    -(SP)
      BISB   RLDRV+1, (SP)
      MOV    @DRVNAM, -(SP)
      MOV    RLBAS, (SP)
      MOV    @BASADD, -(SP)
      MOV    @FMT5, -(SP)
      MOV    @5, -(SP)
      MOV    SP, R0
      TRAP   C$PNTF
      ADD    @14, SP

      ;PRINT DRIVE INFORMATION
      PRINTF @FMT3
      MOV    @FMT3, -(SP)
      MOV    @1, -(SP)
      MOV    SP, R0
      TRAP   C$PNTF
      ADD    @4, SP

      DODU   PSETNM           ;DO DROP UNIT ON DRIVE
      MOV    PSETNM, R0
      TRAP   C$DODU
      BR     2$              ;BRANCH TO EXIT
      1$:   MOV    RLDRV, R5   ;ELSE, GET DRIVE NUMBER
      BIS    @CRDYMSK, R5    ;SET CONTROLLER READY
      MOV    R5, RLCS(R2)   ;LOAD IN THE DRIVE NUMBER
  
```

015132  
015132 005037 003452  
015136 012746 000340  
C15142 012746 016142  
015146 013746 003234  
015152 012746 000003  
015156 104437  
015160 062706 000010  
015164 013702 003032  
015170 005762 000000  
015174 005737 003452  
015200 001447  
015202 012746 007635  
015206 012746 012174  
015212 012746 000002  
015216 010600  
015220 104417  
015222 062706 000006  
015226 005046  
015230 153716 003037  
015234 012746 006142  
015240 013746 003032  
015244 012746 006131  
015250 012746 011370  
015254 012746 000005  
015260 010600  
015262 104417  
015264 062706 000014  
015270 012746 011354  
015274 012746 000001  
015300 010600  
015302 104417  
015304 062706 000004  
015310 013700 003446  
015314 104451  
015316 000460  
015320 013705 003036  
015324 052705 000200  
015330 010562 000000

```

29 015334 032762 000001 000000      BIT      @DRDYMSK,RLCS(R2)          ;IS DRIVE READY?
30 015342 001046                      BNE      2$                       ;BRANCH TO PERFORM TESTS IF DRIVE IS READY
31 015344                      PRINTF   @FMT24,@NOTRDY          ;PRINT MSG. "DRIVE DROPPED DID NOT RESPOND
    015344 012746 007673      MOV      @NOTRDY,-(SP)
    015350 012746 012174      MOV      @FMT24,-(SP)
    015354 012746 000002      MOV      @2,(SP)
    015360 010600      MOV      SP,RO
    015362 104417      TRAP    C$PNTF
    015364 062706 000006      ADD      @6,SP
32                                ;/WITH READY'"
33 015370      PRINTF   @FMT5,@BASADD,RLBAS,@DRVNAM,<8,RLDRV+1>
    015370 005046      CLR      (SP)
    015372 153716 003037      BISB    RLDRV+1,(SP)
    015376 012746 006142      MOV      @DRVNAM,-(SP)
    015402 013746 003032      MOV      RLBAS,-(SP)
    015406 012746 006131      MOV      @BASADD,-(SP)
    015412 012746 011370      MOV      @FMT5,-(SP)
    015416 012746 000005      MOV      @5,-(SP)
    015422 010600      MOV      SP,RO
    015424 104417      TRAP    C$PNTF
    015426 062706 000014      ADD      @14,SP
34                                ;PRINT DRIVE INFORMATION
35 015432      PRINTF   @FMT3
    015432 012746 011354      MOV      @FMT3,-(SP)
    015436 012746 000001      MOV      @1,-(SP)
    015442 010600      MOV      SP,RO
    015444 104417      TRAP    C$PNTF
    015446 062706 000004      ADD      @4,SP
36 015452      DODU     PSETNM          ;DO DROP UNIT ON DRIVE
    015452 013700 003446      MOV      PSETNM,RO
    015456 104451      TRAP    C$DODU
37 015460      2$:      CLRVEC   ERRVEC          ;RELEASE ERROR VECTOR
    015460 013700 003234      MOV      ERRVEC,RO
    015464 104436      TRAP    C$CVEC
38 015466      ENDAUTO
    015466      L10016:
    015466 104461      TRAP    C$AUTO
39
    
```

{ 6 }

```

1
2
3
4 015470
5 015470
6
7 015470
   015470 012746 000340
   015474 012746 016142
   015500 013746 003234
   015504 012746 000003
   015510 104437
   015512 062706 000010
8
9 015516
   015516 012700 000007
   015522 104441
10 015524 032762 000200 000000 2$:
11 015532 001407
12 015534 053762 003036 000000
13 015542 032762 000001 000000
14 015550 001005
15 015552
16 015564
   015564 013700 003034
   015570 104436
17 015572 005737 003454
18 015576 001402
19 015600 005337 003454
20 015604
   015604 013700 003234
   015610 104436
21 015612
   015612 104433
22
23 015614
   015614
   015614 104412
24
25 015616
26 015616 000240
27 015620
   015620
   015620 104453
28
29 015622
30

```

.SBITL CLEANUP CODE SECTION

BGNMOD CLNCODE  
 BGNCLN

SETVEC ERRVEC,@TRPHAN,@340  
 MOV @340,-(SP)  
 MOV @TRPHAN,-(SP)  
 MOV ERRVEC,(SP)  
 MOV @3,-(SP)  
 TRAP C\$SVEC  
 ADD @10,SP

SETPRI @7 ;SET PRIORITY TO 7

MOV @7,R0  
 TRAP C\$SPRI  
 BIT @CRDYMSK,RLCS(R2) ;TEST IF CONTROLLER READY  
 BEQ 3\$ ;NO LOOP UNTIL READY  
 BIS RLDRV,RLCS(R2) ;SET DRIVE NUMBER  
 BIT @DRDYMSK,RLCS(R2) ;TEST IF DRIVE BUSY  
 BNE 5\$ ;NO - SKIP  
 WAITMS @3 ;WAIT 300 MS  
 CLRVEC RLVEC ;RELEASE VEC

MOV RLVEC,R0  
 TRAP C\$CVEC  
 TST PWRFLG ;PWR FAIL SET  
 BEQ 7\$ ;NO

7\$: CLRVEC ERRVEC  
 MOV ERRVEC,R0  
 TRAP C\$CVEC  
 ERESET ;TAKE CARE OF LSI-11  
 TRAP C\$RESET

ENDCLN  
 L10017:  
 TRAP C\$CLEAN

BGN DU  
 NOP  
 ENDDU  
 L10020:  
 TRAP C\$DU

ENDMOD

16

```

1          .SBTTL  GLOBAL SUBROUTINES
2
3 015622    BGNMOD  GLBSUB
4
5
6 015622    012737  000160  002116  TIME:  MOV    #160,L#DLY      ;GET OUTER DELAY LOOP
7 015630    005237  003466                INC    TIM.US        ;US-WAIT ROUTINE INDICATOR
8 015634    013737  003456  003462    MOV    XDELAY,MININC ;SAVE ORIGINAL US WAIT
9 015642    005437  003456                NEG    XDELAY        ;GET NEGATIVE OF FACTOR
10 015646    READBUS ;Q BUS?
11 015646    104407    TRAP   C#RDBU
12 015650    BCOMPLETE 2# ;BRANCH - IF YES
13 015650    103420    BCS    2#
14 015652    012727  000001    1# :  DELAY  1.          ;WAIT
15 015656    000000    MOV    #1.,(PC)+
16 015660    013727  002116    .WORD  0
17 015664    000000    MOV    L#DLY,(PC)+
18 015666    005367  177772    .WORD  0
19 015672    001375    DEC    -6(PC)
20 015674    005367  177756    BNE    -.4
21 015700    001367    DEC    -22(PC)
22 015702    005237  003456    BNE    -.20
23 015706    002761    INC    XDELAY
24 015710    000422    BLT   1#          ;WAIT FACTOR EXPIRED?
25 015712    012737  000065  002116  2# :  MOV    #65,L#DLY      ;BRANCH - IF NO
26 015720    012727  000001    3# :  DELAY  1.          ;GET TIME
27 015724    000000    MOV    #1.,(PC)+ ;GET OUTER DELAY LOOP
28 015726    013727  002116    .WORD  0          ;WAIT WITH RESPECT TO FONZ BUS
29 015732    000000    MOV    L#DLY,(PC)+
30 015734    005367  177772    .WORD  0
31 015740    001375    DEC    -6(PC)
32 015742    005367  177756    BNE    -.4
33 015746    001367    DEC    -22(PC)
34 015750    005237  003456    BNE    -.20
35 015754    002761    INC    XDELAY
36 015756    063737  003462  003122  4# :  BLT   3#          ;WAIT FACTOR EXPIRED?
37 015764    000207    ADD   MININC,TEMPO ;BRANCH - IF NO
38                RTS    PC          ;GET TIME EXPIRED
39                ;RETURN
40
41
42
43
44 015766    012737  000160  002116  XTIME: MOV    #160,L#DLY      ;GET OUTER DELAY LOOP
45 015774    005037  003466                CLR    TIM.US        ;MS. WAIT INDICATOR
46 016000    013737  003460  003472    MOV    YDELAY,MAJINC ;SAVE ORIGINAL WAIT MS
47 016006    006337  003460                ASL   YDELAY        ;MULTIPLY BY FACTOR 4
48 016012    006337  003460                ASL   YDELAY
49 016016    005437  003460                NEG   YDELAY
50 016022    READBUS ;Q - BUS?
51 016022    104407    TRAP   C#RDBU
52 016024    BCOMPLETE 1# ;BRANCH - IF NO
53 016024    103023    BCC   1#
54 016026    012737  000150  002116  2# :  MOV    #150,L#DLY      ;GET OUTER DELAY LOOP
55 016034    012727  000020    DELAY  20          ;WAIT WITH RESPECT TO FONZ BUS
56 016034    000000    MOV    #20,(PC)+
57 016040    000000    .WORD  0
58 016042    013727  002116    MOV    L#DLY,(PC)+
59 016046    000000    .WORD  0
    
```

```

016050 005367 177772          DEC      6(PC)
016054 001375                BNE     -4
016056 005367 177756          DEC     -22(PC)
016062 001367                BNE     -20
34 016064 005237 003460      INC     YDELAY      ;WAIT FACTOR EXPIRED
35 016070 002761                BLT     2$          ;BRANCH - IF NO
36 016072 000417                BR      3$          ;GET TIME
37 016074                1$: DELAY 10          ;WAIT
016074 012727 000010          MOV     @10,(PC)+
016100 000000                .WORD  0
016102 013727 002116          MOV     L$DLY,(PC)+
016106 000000                .WORD  C
016110 005367 177772          DEC     -6(PC)
016114 001375                BNE     -4
016116 005367 177756          DEC     -22(PC)
016122 001367                BNE     -20
38 016124 005237 003460      INC     YDELAY      ;WAIT FACTOR EXPIRED?
39 016130 002761                BLT     1$          ;BRANCH - IF NO
40 016132 063737 003472 003464 3$: ADD     MAJINC,TEMP ;GET EXPIRED TIME
41 016140 000207                RTS     PC          ;RETURN
42
43
44
45 016142                BGNSRV
46
47                ;TRAP HANDLER INDICATES OCCURRENCE OF A TRAP.
48
49 016142 005237 003452      TRPHAN: INC     TRPFLG
50
51 016146                ENDSRV
016146 L10021:
016146 000002                RTI
r2
53 016150                BGNSRV
54
55                ;INTERRUPT HANDLER. ABORTS WAIT TIMER AND STORES RL11 REGISTERS.
56
57 016150                INTHLR:
58
59 016150 012237 003050          MOV     (R2)+,T.CS      ;STORE RL REGISTERS
60 016154 012237 003052          MOV     (R2)+,T.BA
61 016160 012237 003054          MOV     (R2)+,T.DA
62 016164 011237 003056          MOV     (R2),T.MP
63 016170 012737 177777 003012  MOV     @-1,DONE      ;SET DONE FLAG
64 016176 013702 003032          MOV     RLBAS,R2      ;RESTORE R2
65 016202                ABORTWAIT
66
67 016226                ENDSRV
016226 L10022:
016226 000002                RTI
68
    
```

```

1
2
3           ;           ERROR LIMIT CHECKING ROUTINE
4           ;           DROPS DRIVE IF ERROR LIMIT EXCEEDED
5
6 016230 027737 165006 014130 CKERLM: CMP     @ERRPOINT,ERLIMW           ;TEST IF ERROR LIMIT EXCEEDED
7 016236 002453                                BLT     1#                       ;NO - SKIP
8 016240                                INLOOP                                ;CHECK IF IN ERROR LOOP
9 016242 104420                                TRAP   C#INLP
016242 103451                                BCOMPLETE 1#                     ;YES - SKIP
10 016244                                BCS     1#
016244 012746 011055                                PRINTF  @FMT25,ERLIMW,@MEXERS
016250 013746 014130                                MOV     @MEXERS,-(SP)
016254 012746 012201                                MOV     ERLIMW,-(SP)
016260 012746 000003                                MOV     @FMT25,-(SP)
016264 010600                                MOV     @3,-(SP)
016266 104417                                MOV     SP,RO
016270 062706 000010                                TRAP   C#PNTF
11 016274                                ADD     @10,SP
016274 005046                                PRINTF  @FMT5,@BASADD,RLBAS,@DRVNAM,<B,RLDRV+1>
016276 153716 003037                                CLR     -(SP)
016302 012746 006142                                BISE   RLDRV+1,(SP)
016306 013746 003032                                MOV     @DRVNAM,-(SP)
016312 012746 006131                                MOV     RLBAS,-(SP)
016316 012746 011370                                MOV     @BASADD,-(SP)
016322 012746 000005                                MOV     @FMT5,-(SP)
016326 010600                                MOV     @5,-(SP)
016330 104417                                MOV     SP,RO
016332 062706 000014                                TRAP   C#PNTF
12 016336                                ADD     @14,SP
016336 012746 011354                                PRINTF  @FMT3
016342 012746 000001                                MOV     @FMT3,-(SP)
016346 010600                                MOV     @1,-(SP)
016350 104417                                MOV     SP,RO
016352 062706 000004                                TRAP   C#PNTF
13 016356                                ADD     @4,SP
016356 013700 003446                                DODU   PSETNM                     ;DROP DRIVE
016362 104451                                MOV     PSETNM,RO
14 016364                                TRAP   C#DODU
016364 104444                                DOCLN                                ;GO TO CLEAN UP
15 016366 000207                                TRAP   C#DOCLN
16
17
18 016370 016237 000000 003050           ; READ AND STORE ALL RL11 REGISTERS
19 016376 016237 000002 003052 READRL: MOV     RLCSR(R2),T.CS      ;GET CS REG
20 016404 016237 000004 003054           MOV     RLBA(R2),T.BA      ;GET BUS ADDRESS REG
21 016412 016237 000006 003056           MOV     RLDA(R2),T.DA      ;GET DISK ADDRESS
22 016420 000207                                MOV     RLMP(R2),T.MP      ;GET MULTI-PURPOSE REG
23
24
25 016422 011646                                RTS     PC                     ;RETURN
26 016424 005066                                ; WAIT FOR CONTROLLER TIMEOUT TO FORCE INTERRUPT ROUTINE
27 016430 032762 000200 000000 WAITIN: MOV     (SP),-(SP)        ;MAKE ROOM FOR ERROR POINTER
28 016436 001420                                CLR     2(SP)                 ;CLEAR FOR POINTER
29 016440 004737 016370                                BIT     @CRDYMSK,RLCSR(R2)    ;TEST IF CONTROLLER READY
30 016444 005737 003012                                BEQ     4#                     ;NO - SKIP TO WAIT
                                           JSR     PC,READRL             ;READ ALL RL REGS
                                           TST     DONE                  ;TEST IF INTERRUPT OCCURRED
    
```





```

88 016772 001444      BEQ      5#      ;NO - SKIP
89 016774 052737 040000 003010    BIS      @RELDWT,OPFLAG ;ELSE SET WAIT FLAG
90 017002 000440      BR        5#      ;SKIP TO CLEAR
91 017004 032737 040000 003050 51# : BIT      @DRVERR,T.CS  ;TEST IF DRIVE ERROR NOW
92 017012 001034      BNF      5#      ;YES - SKIP TO CLEAR
93 017014      WAITMS #1    ;WAIT FOR DRIVE TO GET ERROR, RDY, OR HEADS OUT
94 017026 005301      DEC      R1      ;DEC WAIT COUNTER
95 017030 001342      BNE      50#     ;IF NOT DONE, LOOP
96 017032 012703 010737    MOV      @MUNDEF,R3  ;MESSAGE FOR UNDEFINED STATE
97 017036      ERRHRD 10001.,,ERR1
   017036 104456      TRAP    C#ERHRD
   017040 023421      .WORD  10001
   017042 000000      .WORD  0
   017044 012266      .WORD  ERR1
98 017046 000565      BR        14#     ;EXIT
99 017050 005737 003132      11# : TST      TEMP4    ;TEST IF SAVE REGISTERS
100 017054 001013      BNE      5#      ;NO SKIP
101 017056 012701 000004    MOV      @4,R1      ;SET SAVE COUNT
102 017062 012703 003050    MOV      @L.MP+2,R3 ;SET ADDRESS OF FIRST SAVE
103 017066 014346      8# : MOV      -(R3),-(SP) ;PUT REG ON STACK
104 017070 005301      DEC      R1      ;DEC COUNT
105 017072 001375      BNE      8#      ;LOOP UNTIL ALL SAVED
106 017074 012737 000003 003044    MOV      @GETSTAT,L.DA ;SET FOR GET STATUS
107 017102 000403      BR        6#      ;SKIP
108 017104 013737 003132 003044 5# : MOV      TEMP4,L.DA ;INSERT PRESET FOR STATUS
109 017112      6# :
110 017112 005037 003012      CLR      DONE     ;CLEAR INTERRUPT FLAG
111 017116 013737 003036 003040    MOV      RLDIV,L.CS ;SET UP TO GET STATUS
112 017124 042737 002000 003040    BIC      @BIT10,L.CS ;CLEAR FOR DRIVE 4 - 7 SPEC'D
113 017132 052737 000104 003040    BIS      @GTSTAT,L.CS
114 017140 013762 003044 000004    MOV      L.DA,RLDA(R2) ;LOAD RL REGS
115 017146 013762 003040 000000    MOV      L.CS,RLCSR(R2) ;LOAD CS REG
116 017154      WAITUS #1    ;WAIT 100 US FOR INTERRUPT
117 017166 005737 003012      TST      DONE     ;CHECK IF INTERRUPT OCCURRED
118 017172 001504      BEQ      1#      ;NO - SKIP
119 017174 013737 003056 003064 4# : MOV      T.MP,T.STAT ;STORE MP REGISTER
120 017202 042737 177770 003064    BIC      @C<STAMSK>,T.STAT ;CLEAR ALL BUT STATE
121 017210 032737 000010 003044    BIT      @DRSET,L.DA ;TEST IF RESET WAS SPECIFIED
122 017216 001503      BEQ      3#      ;NO - SKIP TO EXIT
123 017220 032737 040000 003010    BIT      @RELDWT,OPFLAG ;TEST IF RELOAD WAIT FLAG SET
124 017226 001427      BEQ      12#     ;NO - SKIP
125 017230 012701 001130      MOV      @600.,R1  ;SET WAIT COUNT FOR 60 SECONDS
126 017234 032762 000001 000000 13# : BIT      @RDYMSK,RLCS(R2) ;TEST IF DRIVE NOW READY
127 017242 001021      BNE      12#     ;YES - SKIP
128 017244      WAITMS #1    ;CALL WAIT
129 017256 005301      DEC      R1      ;DEC COUNT
130 017260 001365      BNE      13#     ;LOOP IF NOT 0
131 017262 004737 016626      JSR      PC,GSTAT  ;GET DRIVE STATUS
132 017266 017426      3#      ;ERROR RETURN
133 017270 012703 011004      MOV      @MRLFAL,R3 ;SET RESULT MESSAGE POINTER
134 017274      ERRHRD 10003.,,ERR1
   017274 104456      TRAP    C#ERHRD
   017276 023423      .WORD  10003
   017300 000000      .WORD  0
   017302 012266      .WORD  ERR1
135 017304 000446      BR        14#     ;GO TO EXIT
136 017306      12# : WAITUS #10. ;WAIT FOR 1MS

```

```

137 017320 004737 016626          JSR    PC,GSTAT          ;GET DRIVE STATUS
138 017324 017426                   3$
139 017326 032737 100000 003050    BIT    @ANYERR,T.CS      ;TEST IF ANY ERROR
140 017334 001434                   3$          ;NO - SKIP
141 017336 032737 001000 003056    BIT    @VCSTAT,T.MP     ;CHECK IF VOLUME CHECK RESET
142 017344 001403                   7$          ;YES SKIP
143 017346 012703 006433          MOV    @VCNRST,R3       ;SET REASON POINTER
144 017352 000417                   2$          ;EXIT
145 017354 032737 040000 003050  7$:   BIT    @DRVERR,T.CS     ;CHECK IF DRIVE ERROR
146 017362 001405                   9$          ;NO - SKIP
147 017364          ERRHRD    10004...ERR6
    017364 104456          TRAP    C$ERRHRD
    017366 023424          .WORD   10004
    017370 000000          .WORD   0
    017372 012570          .WORD   ERR6
148 017374 000412                   BR      14$          ;EXIT
149 017376 012703 006454          9$:   MOV    @UNXERR,R3     ;SET REASON POINTER
150 017402 000403                   BR      2$          ;EXIT
151 017404 004737 016422          1$:   JSR    PC,WAITIN     ;WAIT FOR INTERRUPT
152 017410 012603                   MOV    (SP)+,R3       ;STORE REASON POINTER FOR RETURN
153 017412          2$:   ERRHRD    10002...ERR1
    017412 104456          TRAP    C$ERRHRD
    017414 023422          .WORD   10002
    017416 000000          .WORD   0
    017420 012266          .WORD   ERR1
154 017422 005037 003022          14$:  CLR    ERRSWI          ;CLEAR FOR ERROR RETURN
155 017426 005737 003132          3$:   TST    TEMP4         ;TEST IF REGISTERS WERE SAVED
156 017432 001007                   BNE    22$          ;NO - SKIP
157 017434 012703 003040          MOV    @L.CS,R3       ;SET POINTER TO RESTORE
158 017440 012701 000004          MOV    @4,R1          ;SET REGISTER COUNT
159 017444 012623          20$:  MOV    (SP)+,(R3)+    ;RESTORE REG
160 017446 005301                   DEC    R1             ;DEC COUNT
161 017450 001375                   BNE    20$          ;LOOP UNTIL ALL ARE RESTORED
162 017452 162737 000002 003006  22$:  SUB    @2,SSINDEX     ;REMOVE ENTRY FROM SUBROUT STACK
163 017460 012601                   MOV    (SP)+,R1       ;RESTORE R1
164 017462 012600                   MOV    (SP)+,R0       ;RESTORE R0
165 017464 012603                   MOV    (SP)+,R3       ;RESTORE R3
166 017466 012637 003132          MOV    (SP)+,TEMP4    ;RESTORE TEMP4
167 017472 005737 003022          TST    ERRSWI         ;TEST IF ERROR RETURN
168 017476 001403                   BEQ    99$          ;YES - SKIP
169 017500 063716 003022          ADD    ERRSWI,(SP)    ;ADD IN ERROR RETURN
170 017504 000207                   RTS    PC
171 017506 017616 000000          99$:  MOV    @8(SP),(SP)    ;SET ERROR RETURN ADDRESS
172 017512 000207                   RTS    PC
173
174
175          ; SEEK ROUTINE
176 017514 012737 177777 003124  XSEEK: MOV    #-1,TEMP1    ;SET SPECIAL TIMING SEEK FLAG
177 017522 000402                   BR      XSEEK1
178 017524 005037 003124          XSEEK: CLR    TEMP1     ;CLEAR SPECIAL SEEK FOR TIMING FLAG
179 017530 010346                   XSEEK1: MOV    R3,-(SP)  ;STORE R3
180 017532 013703 003006          MOV    SSINDEX,R3     ;GET SUBROUTINE INDEX
181 017536 005723                   TST    (R3)+          ;BUMP IT FOR NEXT ENTRY
182 017540 016663 000002 002410  MOV    2(SP),SUBSTK(R3) ;INSERT THIS CALL
183 017546 162763 000004 002410  SUB    @4,SUBSTK(R3)  ;ADJUST IT TO CALLING LOCATION
184 017554 010337 003006          MOV    R3,SSINDEX    ;STORE IT BACK
185 017560 010046                   MOV    R0,-(SP)
    
```

186	017562	010146			MOV	R1, (SP)	
187	017564	010546			MOV	R5, (SP)	;STORE RFG
188	017566	012737	000002	003022	MOV	#2,ERRSWI	;SET FOR NO ERROR RETURN
189	017574	005037	003102		CLR	DIFAUG	;CLEAR DIFFERENCE AUGMENT (FOR SEEKING
190							; PAST GUARD BAND)
191	017600	004737	022704		JSR	PC,GETPOS	;GET PRESENT POSITION
192	017604	020236			65#		
193	017606	013737	003110	003104	MOV	CURCYL,OLDCYL	;MOVE CURRENT TO OLD CYLINDER
194	017614	023737	003106	002306	CMP	NEWCYL,HLMTW	;TEST IF NEW IS GREATER THAN 255
195	017622	003427			BLE	3#	;NO - SKIP
196	017624	163737	002306	003106	SUB	HLMTW,NEWCYL	;ELSE SUBTRACT 255.
197	017632	013737	003106	003102	MOV	NEWCYL,DIFAUG	;STORE DIFFERENCE AS AUGMENT
198	017640	013737	002306	003106	MOV	HLMTW,NEWCYL	;SET NEWCYL AS 255.
199	017646	022737	000001	002302	CMP	#1,T.DRIVE	
200	017654	001424			BEQ	6#	
201	017656	162737	000001	003106	SUB	#1,NEWCYL	
202	017664	012737	000001	003114	MOV	#1,DESSGN	
203	017672	012737	000001	003112	MOV	#1,DESDIF	
204	017700	000451			BR	18#	
205	017702	005737	003106		3#:	TST	NEWCYL ;TEST IF NEWCYL HAS NEGATIVE VALUE
206	017706	100007			BPL	6#	;NO - SKIP
207	017710	005437	003106		NEG	NEWCYL	;ELSE MAKE IT POSITIVE
208	017714	013737	003106	003102	MOV	NEWCYL,DIFAUG	;AND STORE IT AS AUGMENT
209	017722	005037	003106		CLR	NEWCYL	;AND SET NEWCYL TO 0
210	017726	013705	003110		6#:	MOV	CURCYL,R5 ;COMPUTE DIFFERENCE AND NEW CYLINDER
211	017732	163705	003106		SUB	NEWCYL,R5	;SUB NEWCYL FROM CURCYL
212	017736	100005			BPL	13#	;IF DIFF IS POSITIVE - SKIP(REV SEEK)
213	017740	012737	000001	003114	MOV	#1,DESSGN	;ELSE SET SIGN FOR FORWARD
214	017746	005405			NEG	R5	;MAKE DIFFERENCE POSITIVE
215	017750	000402			BR	14#	;SKIP
216	017752	005037	003114		13#:	CLR	DESSGN ;SET SIGN FOR REVERSE
217	017756	010537	003112		14#:	MOV	R5,DESDIF ;STORE DIFFERENCE
218	017762	005737	003102		TST	DIFAUG	;IS THERE A DIFFERENCE AUGMENT
219	017766	001416			BEQ	18#	;NO - SKIP
220	017770	023737	003106	002306	CMP	NEWCYL,HLMTW	;CHECK IF NEW CYL IS 255.
221	017776	001007			BNE	17#	;NO - SKIP
222	020000	012737	000001	003114	MOV	#1,DESSGN	;ELSE FORCE SIGN FOR FORWARD
223							;(INNER GUARD BAND)
224	020006	022737	000001	002302	CMP	#1,T.DRIVE	
225	020014	001003			BNE	18#	
226	020016	063737	003102	003112	17#:	ADD	DIFAUG,DESDIF
227	020024				18#:		
228	020024	012705	003040		MOV	#L,CS,R5	;GET L REG ADDRESS
229	020030	012715	000106		MOV	#SEEK,(R5)	;SET FOR SEEK
230	020034	053715	003036		BIS	RLDRV,(R5)	;INSERT DRIVE NUMBER
231	020040	042725	002000		BIC	#BIT10,(R5)+	;CLEAR IF DRIVE 4 - 7 SPEC'D
232	020044	005025			CLR	(R5)+	;CLEAR BUS ADDRESS
233	020046	013715	003112		MOV	DESDIF,(R5)	;LOAD DIFFERENCE
234	020052	012700	000007		MOV	#7,R0	;SET TO SHIFT DIFFERENCE
235	020056	006315			21#:	ASL	(R5)
236	020060	005300			DEC	R0	
237	020062	001375			BNE	21#	;LOOP UNTIL ALIGNED
238	020064	005737	003114		TST	DESSGN	;TEST SIGN
239	020070	001402			BEQ	23#	;SKIP IF 0
240	020072	052715	000004		BIS	#DIRBIT,(R5)	;ELSE INSERT SIGN
241	020076	005737	003116		23#:	TST	DESHD ;TEST IF HEAD 0
242	020102	001402			BEQ	25#	;YES - SKIP

116,

```

243 020104 052715 000020      BIS      #HDSEL,(R5)      ;ELSE SET HEAD BIT
244 020110 052725 000001      BIS      #MBSET0,(R5)+  ;INSERT MARKER BIT
245 020114 004737 020642      JSR      PC,RDYCHK      ;CHECK IF DRIVE READY
246 020120 020236              65$
247 020122 005037 003012      CLR      DONE           ;CLEAR INTERRUPT FLAG
248 020126 005737 003124      TST     TEMP1          ;CHECK IF SPECIAL SEEK FLAG SET
249 020132 001041              BNE     65$            ;YES - SKIP DO NOT START SEEK
250 020134 014562 000004      MOV     -(R5),RLDA(R2)  ;LOAD RL REGISTERS
251 020140 014562 000002      MOV     -(R5),RLBA(R2)
252 020144 014562 000000      MOV     -(R5),RLCS(R2)
253 020150              30$: WAITUS  #10.
254 020162 005737 003012      TST     DONE           ;TEST IF INTERRUPT DONE
255 020166 001012              BNE     32$            ;YES - SKIP
256 020170 004737 016422      JSR     PC,WAITIN      ;GO WAIT FOR INTERRUPT
257 020174 012603              MOV     (SP)+,R3       ;GET RESULT MESSAGE POINTER
258 020176              ERRHRD 10005...,ERR1
    020176 104456              TRAP   C$ERRHRD
    020200 023425              .WORD 10005
    020202 000000              .WORD 0
    020204 012266              .WORD ERR1
259 020206 005037 003022      CLR     ERRSWI         ;CLEAR FOR ERROR RETURN
260 020212 000411              BR      65$
261 020214 005737 003050      32$:  TST     T.CS      ;TEST IF ANY ERROR
262 020220 100006              BPL     65$            ;NO - SKIP
263 020222              ERRHRD 10006...,ERR6
    020222 104456              TRAP   C$ERRHRD
    020224 023426              .WORD 10006
    020226 000000              .WORD 0
    020230 012570              .WORD ERR6
264 020232 005037 003022      CLR     ERRSWI         ;CLEAR FOR ERROR RETURN
265 020236 162737 000002 003006 65$:  SUB     #2,SSINDX      ;REMOVE ENTRY FROM SUBROUT STACK
266 020244 012605              MOV     (SP)+,R5       ;RESTORE REGISTERS
267 020246 012601              MOV     (SP)+,R1
268 020250 012600              MOV     (SP)+,R0
269 020252 012603              MOV     (SP)+,R3
270 020254 005737 003022      TST     ERRSWI         ;TEST IF ERROR RETURN
271 020260 001403              BEQ     99$            ;YES - SKIP
272 020262 063716 003022      ADD     ERRSWI,(SP)    ;ADD IN ERROR RETURN
273 020266 000207              RTS     PC
274 020270 017616 000000      99$:  MOV     @ (SP),(SP) ;SET ERROR RETURN ADDRESS
275 020274 000207              RTS     PC
276
333
335      ; POSITION HEADS ROUTINE. POSITIONS HEADS USING 1 CYLINDER SEEKS
336      ; TO CYLINDER SPECIFIED IN R5 BY THE CALLING ROUTINE
337 020276 010346              POSHDS: MOV     R3,-(SP)    ;SAVE REGS
338 020300 013703 003006      MOV     SSINDX,R3      ;GET SUBROUTINE INDEX
339 020304 005723              TST     (R3),+        ;BUMP IT FOR NEXT ENTRY
340 020306 016663 000002 002410  MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
341 020314 162763 000004 002410  SUB     #4,SUBSTK(R3)  ;ADJUST IT TO CALLING LOCATION
342 020322 010337 003006      MOV     R3,SSINDX     ;STORE IT BACK
343 020326 010346              MOV     R3,-(SP)
344 020330 010446              MOV     R4,-(SP)
345 020332 012737 000002 003022  MOV     #2,ERRSWI     ;SET FOR NO ERROR RETURN
346 020340 004737 022704      JSR     PC,GETPOS     ;GET CURRENT POSITION
347 020344 020604              PH65$
348 020346 012704 000012      MOV     #10.,R4       ;SET RETRY COUNT
    
```

```

349 020352          BGNSEG
      020352 104404
350 020354          1$:  TRAP  C#BSEG          ;CHECK IF IN ERROR LOOP
      020354 104420          INLOOP
      020354          TRAP  C#INLP
351 020356          BNCOMPLETE 5$          ;NO - SKIP
      020356          BCC  5$
352 020360 004737 022704          JSR  PC,GETPOS          ;ELSE GET POSITION
353 020364 020602          60$
354 020366 023737 003110 003106          CMP  CURCYL,NEWCYL          ;CHECK IF AT INTENDED POSITION
355 020374 001017          BNE  8$          ;NO - SKIP
356 020376 004737 021202          JSR  PC,ONSWAP          ;SWAP OLDCYL AND NEWCYL
357 020402 000414          BR   8$          ;SKIP
358 020404 013737 003110 003104 5$:  MOV  CURCYL,OLDCYL          ;IN NOT LOOPING, STORE CURCYL AS OLDCYL
359 020412 023705 003110          CMP  CURCYL,R5          ;CHECK IF HDS AT FINAL POSITION
360 020416 001471          BEQ  60$          ;YES - GO TO EXIT
361 020420 003003          BGT  7$          ;IF CURCYL > FINAL POSITION - SKIP
362 020422 005237 003106          INC  NEWCYL          ;ELSE BUMP NEWCYL (MOVE HDS IN)
363 020426 000402          BR   8$          ;SKIP
364 020430 005337 003106          7$:  DEC  NEWCYL          ;DEC NEWCYL (MOVE HDS OUT)
365 020434 004737 017524          8$:  JSR  PC,XSEEK          ;DO SEEK
366 020440 020602          60$
367 020442 012701 005670          MOV  #3000.,R1          ;SET WAIT COUNT 300 MS
368 020446 004737 022420          JSR  PC,RDYWAIT          ;WAIT FOR DRIVE READY
369 020452 020602          60$
370 020454 005737 003050          TST  T.CS          ;TEST IF ANY ERROR
371 020460 100007          BPL  10$          ;NO - SKIP
372 020462          ERRHRD 10008.,,ERR6
      020462 104456          TRAP  C#ERHRD
      020464 023430          .WORD 10008
      020466 000000          .WORD 0
      020470 012570          .WORD ERR6
373 020472 005037 003022          CLR  ERRSWI          ;CLEAR FOR ERROR ERROR RETURN
374 020476 000441          BR   60$
375 020500 004737 022704          10$: JSR  PC,GETPOS          ;GET POSITION
376 020504 020602          60$
377 020506 023737 003110 003106          CMP  CURCYL,NEWCYL          ;CHECK IF ARRIVED AT DESIRED PLACE
378 020514 001003          BNE  15$          ;NO - SKIP
379 020516 012704 000012          14$: MOV  #10.,R4          ;ELSE INIT RETRY COUNT
380 020522 000714          BR   1$          ;GO DO NEXT SEEK
381 020524 005737 003114          15$: TST  DESSGN          ;TEST IF GOING IN
382 020530 001017          BNE  17$          ;YES - SKIP
383 020532 023737 003110 003106          CMP  CURCYL,NEWCYL          ;CHECK IF HEADS DID NOT MO - IN
384 020540 003366          BGT  14$          ;YES - SKIP
385 020542 005304          16$: DEC  R4          ;DEC RETRY COUNT
386 020544 001333          BNE  8$          ;DO ANOTHER SEEK IF NOT 0
387 020546 012703 007313          MOV  #HDMOVF,R3          ;ELSE SET RESULT MESSAGE POINTER
388 020552          ERRHRD 10009.,,ERR1
      020552 104456          TRAP  C#ERHRD
      020554 023431          .WORD 10009
      020556 000000          .WORD 0
      020560 012266          .WORD ERR1
389 020562 005037 003022          CLR  ERRSWI          ;CLEAR FOR ERROR ERROR RETURN
390 020566 000405          BR   60$
391 020570 023737 003110 003106 17$: CMP  CURCYL,NEWCYL          ;HDS SHOULD MOVE OUT, CHK THEY DID
392 020576 002747          BLT  14$          ;YES - SKIP
393 020600 000760          BR   16$          ;ELSE GO DEC AND RETRY
394 020602          20$:
    
```

137

```

395 020602          608:
396 020602          ENDSEG
      020602          100008:
      020602          104405
397 020604          162737 00G002 003006 PH658: TRAP      C8ESEG
398 020612          012604          SUB      #2,SSINDX      ;REMOVE ENTRY FROM SUBROUT STACK
399 020614          012600          MOV      (SP),R4      ;RESTORE REGISTERS
400 020616          012603          MOV      (SP),R0
401 020620          005737 003022          MOV      (SP),R3
402 020624          001403          TST     ERRSWI      ;TEST IF ERROR RETURN
403 020626          063716 003022          BEQ     998        ;YES - SKIP
404 020632          000207          ADD     ERRSWI,(SP) ;ADD IN ERROR RETURN
405 020634          017616 000000          998:   RTS     PC
406 020640          000207          MOV     B(SP),(SP) ;SET ERROR RETURN ADDRESS
407          RTS     PC
409          ;
410          ; DRIVE READY TEST ROUTINE. CHECKS DRIVE IS READY. IF NOT, WAIT
411 020642          010346          ;
412 020644          013703 003006          ; SOOMS FOR READY TO SET.
413 020650          005723          RDYCHK: MOV     R3,-(SP)      ;STORE REGS
414 020652          016663 000002 002410          MOV     SSINDX,R3    ;GET SUBROUTINE INDEX
415 020660          162763 000004 002410          TST     (R3)+        ;BUMP IT FOR NEXT ENTRY
416 020666          010337 003006          MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
417 020672          010046          SUB     #4,SUBSTK(R3) ;ADJUST IT TO CALLING LOCATION
418 020674          010146          MOV     R3,SSINDX    ;STORE IT BACK
419 020676          010446          MOV     R0,-(SP)
420 020700          012737 00G002 003022          MOV     R1,-(SP)
421 020706          012701 011610          MOV     R4,-(SP)
422 020712          004737 016626          18:   MOV     #2,ERRSWI    ;SET FOR NO ERROR RETURN
423 020716          021052          MOV     #5000,R1     ;SET WAIT COUNT
424 020720          032737 000001 003050          JSR     PC,GSTAT     ;GET DRIVE STATUS
425 020726          001053          48
426 020730          005301          BIT     @DRDYMSK,T.CS ;TEST IF DRIVE READY
427 020742          001362          BNE     58          ;YES - EXIT
428 020744          012703 010322          WAITUS #1
429 020746          012704 011207          DEC     R1          ;DEC WAIT COUNT
430 020752          012704 011207          18
431 020756          020756          MOV     @DRDY,R3     ;SET RESULT MESSAGE POINTER
      020760          023432          MOV     @C5GOM5,R4   ;SET CONDITION MESSAGE POINTER
      020762          000000          ERHRD  10010,,,ERR5
      020764          012520          TRAP   C8ERHRD
      020766          012701 000062          .WORD  10010
432 020766          012701 000062          .WORD  0
433 020772          004737 016626          28:   .WORD  0
434 020776          021052          .WORD  ERR5
435 021000          032737 000001 003050          MOV     #50,R1      ;SET WAIT COUNT FOR 5 SECONDS
436 021006          001007          JSR     PC,GSTAT     ;GET DRIVE STATUS
437 021010          005301          48
438 021022          001362          BIT     @DRDYMSK,T.CS ;TEST IF DRIVE READY
439 021024          001362          BNE     38          ;YES - SKIP
440 021026          032737 100000 003050          WAITMS #1          ;WAIT FOR 100MS
441 021034          001406          DEC     R1          ;DEC WAIT COUNTER
442 021036          021036          28
443 021040          023433          BNE     28          ;LOOP UNTIL TIME DONE
444 021042          000000          BIT     @ANYERR,T.CS ;TEST IF ANYERR SET
445 021044          012570          BEQ     48          ;NO - SKIP
      .WORD  10011,,,ERR6
      .WORD  10011
      .WORD  0
      .WORD  ERR6
    
```

```

443 021046 005337 003244          DEC      ERRCNT      ;REDUCE ERROR COUNT FOR DUAL ERRORS
444 021052 005037 003022          CLR      ERRSWI     ;CLEAR FOR ERROR RETURN
445 021056 162737 000002 003006 5#:  SUB      @2,SSINDX  ;REMOVE ENTRY FROM SUBROUT STACK
446 021064 012604          MOV      (SP)+,R4   ;RESTORE REGS
447 021066 012601          MOV      (SP)+,R1
448 021070 012600          MOV      (SP)+,R0
449 021072 012603          MOV      (SP)+,R3
450 021074 005737 003022          TST     ERRSWI     ;TEST IF ERROR RETURN
451 021100 001403          BEQ     99#        ;YES - SKIP
452 021102 063716 003022          ADD     ERRSWI,(SP) ;ADD IN ERROR RETURN
453 021106 000207          RTS     PC
454 021110 017616 000000          99#:  MOV     @ (SP),(SP) ;SET ERROR RETURN ADDRESS
455 021114 000207          RTS     PC
456
457
458 ;      CHOOSE HEAD ROUTINE. PICKS HEAD 0 UNLESS SPECIFIC HEAD IS
459 ;      SELECTED BY SOFTWARE PARAMETER.
459 021116 005037 003116          CHOSHD: CLR     DESHD   ;CLEAR TO HEAD 0
460 021122 032737 010000 014120  BIT     @HEADLM,MISWIW ;TEST IF HEAD SPECIFIED
461 021130 001403          BEQ     1#        ;NO - SKIP
462 021132 013737 014126 003116  MOV     HEADW,DESHD ;INSERT SPECIFIED HEAD
463 021140 000207          1#:  RTS     PC
464
465 ;      SWAP HEAD ROUTINE. CHANGES SELECTED HEAD TO HEAD 1
466 ;      UNLESS HEAD 0 SPECIFICALLY SELECTED BY SOFTWARE PARAMETER.
467 021142 032737 010000 014120  SWAPHD: BIT     @HEADLM,MISWIW ;TEST IF HEAD SPECIFIED
468 021150 001011          BNE     2#        ;YES - TAKE ABORT EXIT
469 021152 005737 003116          TST     DESHD     ;TEST IF HEAD ONE USED
470 021156 001006          BNE     2#        ;YES - TAKE ABORT EXIT
471 021160 012737 000001 003116  MOV     @1,DESHD   ;ELSE SET FOR HEAD ONE
472 021166 062716 000002          ADD     @2,(SP)   ;BUMP PAST ABORT RETURN
473 021172 000207          RTS     PC
474 021174 017616 000000          2#:  MOV     @ (SP),(SP) ;GET ABORT DESTINATION
475 021200 000207          3#:  RTS     PC
476
477 ;      SWAP OLD CYLINDER AND NEW CYLINDER ROUTINE.
478 021202 010046          ONSWAP: MOV     RO,-(SP)   ;STORE RO
479 021204 013700 003104          MOV     OLDCYL,RO ;MOVE OLD TO RO
480 021210 013737 003106 003104  MOV     NEWCYL,OLDCYL ;MOVE NEW TO OLD
481 021216 010037 003106          MOV     RO,NEWCYL ;PUT OLD IN NEW
482 021222 012600          MOV     (SP)+,RO  ;RESTORE RO
483 021224 000207          RTS     PC
484
485 ;      BAD SECTOR FILES VALID CHECK ROUTINE. CHECKS IF BAD SECTOR
486 ;      FILES HAVE BEEN READ AND STORED. IF NOT, REPORT AND FORCE
487 ;      FILES TO LOOK LIKE ALL SECTORS OK.
488 ;
489 021226 005737 003500          CKBSVD: TST     BSFVAL  ;TEST IF BAD SECTORS STORED
490 021232 001051          BNE     5#        ;YES - EXIT
491 021234          PRINTF  @FMT9,@BSNSTR ;REPORT
491 021234 012746 007540          MOV     @BSNSTR,-(SP)
491 021240 012746 011554          MOV     @FMT9,-(SP)
491 021244 012746 000002          MOV     @2,-(SP)
491 021250 010600          MOV     SP,RO
491 021252 104417          TRAP   C@PNTF
491 021254 062706 000006          ADD     @6,SP
492 021260          PRINTF  @FMT5,@BASADD,RLBAS,@DRVNAM,<B,RLDRV+1>
492 021260 005046          CLR     -(SP)
492 021262 153716 003037          BISB   RLDRV+1,(SP)
    
```



```

021266 012746 006142      MOV      #DRVNAM, -(SP)
021272 013746 003032      MOV      RLBAS, -(SP)
021276 012746 006131      MOV      #BASADD, -(SP)
021302 012746 011370      MOV      #FMT5, (SP)
021306 012746 000005      MOV      #5, -(SP)
021312 010600      MOV      SP, R0
021314 104417      TRAP     C:PNTF
021316 062706 000014      ADD      #14, SP
493 021322      PRINTF  #FMT3
021322 012746 011354      MOV      #FMT3, -(SP)
021326 012746 000001      MOV      #1, -(SP)
021332 010600      MOV      SP, R0
021334 104417      TRAP     C:PNTF
021336 062706 000004      ADD      #4, SP
494 021342 012737 177777 003502      MOV      #-1, SBSFIL      ;FORCE FILES TO NO ENTRIES
495 021350 012737 177777 003676      MOV      #-1, FBSFIL
496 021356 000207      RTS      PC
497
499
500 021360 012737 000001 003132      ; READ HEADERS ROUTINE.
XRDHDC: MOV      #1, TEMP4      ;SET FLAG TO BYPASS REG STORAGE
501 021366 000402      BR      XRDHDC      ;GO DO IT
502 021370 005037 003132      XRHDH: CLR     TEMP4      ;SET FLAG TO SAVE T. AND L. REGS
XRDHDC: MOV      R3, -(SP)      ;STORE REGISTERS
503 021374 010346      MOV      SSINDX, R3      ;GET SUBROUTINE INDEX
504 021376 013703 003006      TST     (R3)      ;BUMP IT FOR NEXT ENTRY
505 021402 005723      MOV      2(SP), SUBSTK(R3) ;INSERT THIS CALL
506 021404 016663 000002 002410      SUB      #4, SUBSTK(R3) ;ADJUST IT TO CALLING LOCATTON
507 021412 162763 000004 002410      MOV      R3, SSINDX      ;STORE IT BACK
508 021420 010337 003006      MOV      R0, -(SP)
509 021424 010046      MOV      R1, -(SP)
510 021426 010146      MOV      R4, -(SP)
511 021430 010446      MOV      #2, ERRSWI      ;SET FOR NO ERROR RETURN
512 021432 012737 000002 003022      TST     TEMP4      ;TEST IF REGISTERS TO BE SAVED
513 021440 005737 003132      BNE     2#      ;NO - SKIP
514 021444 001007      MOV      #L, MP+2, R3      ;SET POINTER FOR REGS
515 021446 012703 003050      MOV      #4, R1      ;SET COUNT
516 021452 012701 000004      1#: MOV      -(R3), -(SP)      ;SAVE REGISTER
517 021456 014346      DEC     R1      ;DEC COUNT
518 021460 005301      BNE     1#      ;LOOP UNTIL ALL ARE SAVED
519 021462 001375      JSR     PC, RDYCHK      ;CHECK DRIVE READY
520 021464 004737 020642      2#:
521 021470 021740      65#
522 021472 005037 003012      CLR     DONE      ;CLEAR INTERRUPT FLAG
523 021476 012701 003040      MOV      #L, CS, R1      ;GET ADDRESS OF LOAD REGS
524 021502 013711 003036      MOV      RLDIV, (R1)      ;LOAD DRIVE NUMBER
525 021506 042711 002000      BIC     #BIT10, (R1)      ;CLEAR FOR DRIVE 4 - 7 SPEC'D
526 021512 052721 000110      BIS     #RDHEAD, (R1)      ;INSERT COMMAND
527 021516 005021      CLR     (R1)      ;CLEAR BA
528 021520 005021      CLR     (R1)      ;CLEAR DA
529 021522 014162 000004      MOV      -(R1), RLDA(R2) ;LOAD RL11 REGS
530 021526 014162 000002      MOV      -(R1), RLBA(R2)
531 021532 014162 000000      MOV      -(R1), RLCSR(R2)
532 021536      3#: WAITUS  #10.      ;WAIT 1MS FOR INTERRUPT
533 021550 005737 003012      TST     DONE      ;TEST IN INTERRUPT FLAG SET
534 021554 001460      BEQ     14#      ;NO - SKIP
535 021556 032737 000001 003050      5#: BIT      #ORDYSK, T, CS      ;TEST IF DRIVE READY
536 021564 001035      BNE     10#      ;YES - SKIP
537 021566 012703 010322      MOV      #MORDY, R3      ;SET NO READY MESSAGE
    
```

```

533 021572 012704 011224      MOV      #CAFDT,R4      ;CONDITION OF AFTER DATA #FFH
539 021576                      ERRHRD  10017...ERR5
    021576 104456          TRAP   C#ERRHRD
    021600 023441          .WORD  1C017
    021602 000000          .WORD  0
    021604 012520          .WORD  ERR5
540 021606 012701 000062      MOV      #50.,R1      ;SET WAIT COUNT FOR 5 SECONDS
541 021612 004737 016626      4#:    JSR      PC,GSTAT  ;GET STATUS
542 021616 021734          60#
543 021620 032737 000001 003050  BIT      #DRDYMSK,T.CS ;TEST IF DRIVE HAS COME READY
544 021626 001403          BEQ     11#           ;NO - SKIP
545 021630 005037 003022      CLR     ERRSWI       ;CLEAR ERROR SWITCH
546 021634 000411          BR      10#         ;SKIP
547 021636 005301          11#:   DEC     R1        ;DEC WAIT COUNT
548 021640 001364          BNE     4#           ;LOOP UNTIL TIME DONE
549 021642 012704 011235      MOV      #CSSEC,R4    ;SET CONDITION AFTER 5 SECONDS
550 021646                      ERRHRD  10014...ERR5
    021646 104456          TRAP   C#ERRHRD
    021650 023436          .WORD  10014
    021652 000000          .WORD  0
    021654 012520          .WORD  ERR5
551 021656 000426          BR      60#
552 021660 005737 003050      10#:   TST     T.CS      ;CHECK FOR ANY ERRORS
553 021664 100005          BPL     12#         ;NO - SKIP
554 021666                      ERRHRD  10016...ERR6
    021666 104456          TRAP   C#ERRHRD
    021670 023440          .WORD  10016
    021672 000000          .WORD  0
    021674 012570          .WORD  ERR6
555 021676 000416          BR      60#
556 021700 012701 003060      12#:   MOV      #DWRD2,R1   ;GET POINTER
557 021704 016221 000006      MOV      RLMP(R2),(R1) ;STORE LAST TWO HEADER WORDS
558 021710 016221 000006      MOV      RLMP(R2),(R1)
559 021714 000411          BR      65#
560 021716 004737 016422      14#:   JSR      PC,WAITIN  ;WAIT FOR INTERRUPT
561 021722 012603          MOV      (SP)+,R3    ;GET RESULTS
562 021724                      ERRHRD  10015...ERR1
    021724 104456          TRAP   C#ERRHRD
    021726 023437          .WORD  10015
    021730 000000          .WORD  0
    021732 012266          .WORD  ERR1
563 021734 005037 003022      60#:   CLR     ERRSWI       ;CLEAR FOR ERROR ERROR RETURN
564 021740 005737 003132      65#:   TST     TEMP4      ;TEST IF REGISTERS WERE SAVED
565 021744 001007          BNE     22#         ;NO - SKIP
566 021746 012703 003040      MOV      #L.CS,R3    ;SET POINTER TO RESTORE REGS
567 021752 012701 000004      MOV      #4,R1       ;SET COUNT
568 021756 012623          20#:   MOV      (SP)+,(R3)  ;RESTORE REGISTER
569 021760 005301          DEC     R1           ;DEC COUNT
570 021762 001375          BNE     20#         ;LOOP UNTIL ALL ARE RESTORED
571 021764 162737 000002 003006  22#:   SUB     #2,SSINDEX  ;REMOVE ENTRY FROM SUBROUT STACK
572 021772 012604          MOV      (SP)+,R4    ;RESTORE REGS
573 021774 012601          MOV      (SP)+,R1
574 021776 012600          MOV      (SP)+,R0
575 022000 012603          MOV      (SP)+,R3
576 022002 005737 003022      TST     ERRSWI       ;TEST IF ERROR RETURN
577 022006 001403          BEQ     99#         ;YES - SKIP
578 022010 063716 003022      ADD     ERRSWI,(SP)  ;ADD IN ERROR RETURN
    
```

17

```

579 022014 000207          RTS      PC
580 022016 017615 000000 99$:  MOV      @ (SP), (SP)    ;SET ERROR RETURN ADDRESS
581 022022 000207          RTS      PC
582
584          ;
585          ; VERIFY HEADERS ROUTINE. COMPARES 40 HEADERS FOR CONTENT AND
586          ; SEQUENCE.
587 022024 010346          VEPHDR: MOV      R3, -(SP)      ;STORE REGS
588 022026 013703 003006  MOV      SSINDX, R3    ;GET SUBROUTINE INDEX
589 022032 005723          TST      (R3),          ;BUMP IT FOR NEXT ENTRY
590 022034 016663 000002 002410 MOV      2(SP), SUBSTK(R3) ;INSERT THIS CALL
591 022042 162763 000004 002410 SUB      @4, SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
592 022050 010337 003006  MOV      R3, SSINDX     ;STORE IT BACK
593 022054 010046          MOV      R0, (SP)
594 022056 010146          MOV      R1, -(SP)
595 022060 010446          MOV      R4, -(SP)
596 022062 010546          MOV      R5, -(SP)
597 022064 012737 000002 003022 MOV      @2, ERRSWI     ;SET FOR NO ERROR RETURN
598 022072 052737 000002 003010 BIS      @HDRCMP, OPFLAG ;SET HEADER COMPARE FLAG
599 022100 005037 003020  CLR      MORECE        ;CLEAR MORE ERRORS FLAG
600 022110 012704 004072  MOV      @IBUFF, R4     ;SET POINTER TO HEADERS
601 022114 012705 003122  MOV      @TEMPO, R5     ;SET POINTER TO WORK AREA
602 022116 005003          CLR      R3            ;CLEAR FOR WORD COUNTER
603 022118 011415          MOV      (R4), (R5)    ;MOVE HDR WORD TO WORK AREA
604 022120 011401          MOV      (R4), R1     ;PUT WORD IN REG 1
605 022122 042701 000177  BIC      @177, R1      ;CLEAR ALL BUT CYLINDER
606 022126 012700 000007  MOV      @7, R0        ;SET SHIFT COUNT
607 022132 006201 3$:   ASR      R1            ;SHIFT
608 022134 005300          DEC      R0           ;DEC
609 022136 001375          BNE      3$           ;LOOP
610 022140 020137 003106  CMP      R1, NEWCYL    ;CHECK IF CYLINDER PART GOOD
611 022144 001407          BEQ      4$           ;YES - SKIP
612 022146          ERHRD 10018,,,ERR10 ;REPORT ERROR
613 022146          TRAP  C$ERHRD
614 022150          .WORD 10018
615 022152          .WORD 0
616 022154          .WORD 0
617 022154          .WORD ERR10
618 022156 005037 003022  CLR      ERRSWI        ;CLEAR FOR ERROR ERROR RETURN
619 022162 000456          BR      65$
620 022164 012701 000050 4$:   MOV      @40, R1       ;SET HEADER COUNT
621 022170 042715 000100  BIC      @HDRSEL, (R5)  ;CLEAR HEAD SELECT AND 0 BIT
622 022174 005737 003116  TST      DESHD         ;ARE WE USING HD 0?
623 022200 001402          BEQ      5$           ;YES - SKIP
624 022202 052715 000100  BIS      @HDRSEL, (R5) ;INSERT HEAD BIT
625 022206 005065 000002 5$:   CLR      2(R5)        ;CLEAR 2ND WORD OF WORK AREA
626 022212 021524 6$:   CMP      (R5), (R4)    ;TEST FIRST WORD OK
627 022214 001410          BEQ      8$           ;YES - SKIP
628 022216 005744          TST      -(R4)       ;ELSE SET POINTER FOR ERROR
629 022220          ERHRD 10018,,,ERR10 ;REPORT
630 022220          TRAP  C$ERHRD
631 022222          .WORD 10018
632 022224          .WORD 0
633 022226          .WORD 0
634 022226          .WORD ERR10
635 022230 005037 003022  CLR      ERRSWI        ;CLEAR FOR ERROR RETURN
636 022234 005724          TST      (R4),
637 022236 005203 8$:   INC      R3           ;RESET POINTER
638 022240 005724          TST      (R4),
639 022242 001410          BEQ      12$          ;BUMP WORD COUNTER
640                                ;TEST 2ND WORD IS 0
641                                ;YES - SKIP
    
```

```

629 022244 022544      CMP      (R5),-(R4)      ;ADJUST POINTERS FOR REPORT
630 022246      ERRHRD   10018,,ERR10    ;REPORT
      022246      104456      TRAP     C:ERRHRD
      022250      023442      .WORD    10018
      022252      000000      .WORD    0
      022254      013662      .WORD    ERR10
631 022256      005037      003022      CLR      ERRSWI          ;CLEAR FOR FRROR RETURN
632 022262      024524      CMP      -(R5),(R4),    ;RESET POINTERS
633 022264      005724      12#:      TST      (R4),          ;BUMP PAST ECC WORD
634 022266      005203      INC      R3              ;BUMP WORD COUNTER
635 022270      005215      INC      (R5)            ;BUMP SECTOR OF EXPECTED HEADER
636 022272      011500      MOV      (R5),R0        ;MOVE EXPECTED HDR TO R0
637 022274      042700      177700      BIC      @+CHDSEC,R0    ;CLEAR ALL BUT SECTOR
638 022300      022700      000050      CMP      @40.,R0       ;TEST IF AT SECTOR 40
639 022304      001002      BNE     15#             ;NO - SKIP
640 022306      042715      000077      BIC      @HDSEC,(R5)   ;CLEAR SECTOR TO 0
641 022312      005203      15#:      INC      R3              ;BUMP HDR WORD COUNTER
642 022314      005301      DEC      R1              ;DEC HEADER COUNT
643 022316      001335      BNE     6#              ;LOOP IF NOT YET DONE
644 022320      162737      000002      003006      65#:      SUB      @2,SSINDEX    ;REMOVE ENTRY FROM SUBROUT STACK
645 022326      012605      MOV      (SP)+,R5      ;RESTORE REGISTERS
646 022330      012604      MOV      (SP)+,R4
647 022332      012601      MOV      (SP)+,R1
648 022334      012600      MOV      (SP)+,R0
649 022336      012603      MOV      (SP)+,R3
650 022340      005737      003022      TST      ERRSWI          ;TEST IF ERROR RETURN
651 022344      001403      BEQ     99#             ;YES - SKIP
652 022346      063716      003022      ADD      ERRSWI,(SP)   ;ADD IN ERROR RETURN
653 022352      000207      RTS     PC
654 022354      017616      000000      99#:      MOV      @B(SP),(SP)   ;SET ERROR RETURN ADDRESS
655 022360      000207      RTS     PC
656
657
658      ; POSITION HEAD BIT FROM HEADER OR MULTIPURPOSE REGISTER TO LSB.
659 022362      013705      003056      POSHW1: MOV      HDWRD1,R5    ;START FOR POSITION HD BIT IN WD 1
660 022366      000402      BR      POSHD0          ;SKIP
661 022370      013705      003056      POSHSB: MOV      T.MP,R5 ;START FOR POSITION HD BIT IN MP
662 022374      010146      POSHD0: MOV      R1,-(SP)  ;STORE R1
663 022376      042705      177677      BIC      @+CHSSTAT,R5  ;CLEAR ALL BUT HEAD SEL BIT
664 022402      012701      000006      MOV      @6,R1          ;SET SHIFT COUNT
665 022406      006205      1#:      ASR      R5              ;SHIFT FOR RIGHT JUSTIFY
666 022410      005301      DEC      R1
667 022412      001375      BNE     1#
668 022414      012601      MOV      (SP)+,R1      ;RESTORE R1
669 022416      000207      RTS     PC              ;RETURN
670
671      ; WAIT FOR READY ROUTINE. DURATION OF WAIT PASSED TO THE ROUTINE
672      ; FROM THE CALLING ROUTINE IN R1.
673 022420      010346      RDYWAIT: MOV      R3,-(SP)      ;STORE R3
674 022422      013703      003006      MOV      SSINDEX,R3    ;GET SUBROUTINE INDEX
675 022426      005723      TST      (R3),          ;BUMP IT FOR NEXT ENTRY
676 022430      016663      000002      002410      MOV      2(SP),SUBSTK(R3) ;INSERT THIS CALL
677 022436      162763      000004      002410      SUB      @4,SUBSTK(R3)  ;ADJUST IT TO CALLING LOCATION
678 022444      010337      003006      MOV      R3,SSINDEX    ;STORE IT BACK
679 022450      010046      MOV      R0,-(SP)
680 022452      010146      MOV      R1,-(SP)
681 022454      010446      MOV      R4,-(SP)
682 022456      012737      000002      003022      MOV      @2,ERRSWI     ;SET FOR NO ERROR RETURN
    
```

```

683 022464 004737 016626      5#: JSR    PC,GSTAT      ;GET DRIVE STATUS
684 022470 022640              10#:
685 022472 032737 000001 003050 BIT    #DRDYMSK,T.CS    ;CHECK IF READY
686 022500 001061              9#:
687 022502 005301              DEC    R1                ;DEC WAIT COUNT
688 022504 001406              BEQ    7#                ;SKIP IF 0
689 022506              W:ITUS #1
690 022520 000761              BR     5#
691 022522 012703 010322      7#: MOV    :MORDY,R3        ;SET NAME MESSAGE PTR
692 022526              ERRHRD 10020...ERR3    ;REPORT READY ERROR
        022526 104456          TRAP  C#ERRRD
        022530 023444          .WORD 10020
        022532 000000          .WORD 0
        022534 012402          .WORD ERR3
693 022536 012701 000062      MOV    #50,R1           ;SET WAIT COUNT FOR 5 SECONDS
694 022542 004737 016626      6#: JSR    PC,GSTAT      ;GET DRIVE STATUS
695 022546 022640              10#:
696 022550 032737 000001 003050 BIT    #DRDYMSK,T.CS    ;TEST IF DRIVE READY
697 022556 001016              BNE   8#                ;YES - SKIP
698 022560              WAITMS #1              ;WAIT 100 MS
699 022572 005301              DEC    R1                ;DEC WAIT COUNT
700 022574 001362              BNE   6#                ;LOOP UNTIL TIME DONE
701 022576 012704 011235      MOV    #C5SEC,R4        ;SET CONDITION AFTER 5 SECS
702 022602              ERRHRD 10021...ERR5
        022602 104456          TRAP  C#ERRRD
        022604 023445          .WORD 10021
        022606 000000          .WORD 0
        022610 012520          .WORD ERR5
703 022612 000410              BR     11#
704 022614 032737 100000 003050 8#: BIT    #ANYERR,T.CS    ;TEST IF ANY ERROR SET
705 022622 001406              BEQ    10#               ;NO - SKIP
706 022624              ERRHRD 10022...ERR6
        022624 104456          TRAP  C#ERRRD
        022626 023446          .WORD 10022
        022630 000000          .WORD 0
        022632 012570          .WORD ERR6
707 022634 005337 003244      11#: DEC    ERRCNT        ;DEC FOR DOUBLE ERROR REPORT
708 022640 005037 003022      10#: CLR    ERRSWI         ;CLEAR FOR ERROR RETURN
709 022644 162737 000002 003006 9#: SUB    #2,SSINDX        ;REMOVE ENTRY FROM SUBROUT STACK
710 022652 012604              MOV    (SP)+,R4         ;RESTORE REGISTERS
711 022654 012601              MOV    (SP)+,R1
712 022656 012600              MOV    (SP)+,R0
713 022660 012603              MOV    (SP)+,R3
714 022662 005737 003022      TST    ERRSWI           ;RESTORE R3
715 022666 001403              BEQ    99#              ;TEST IF ERROR RETURN
716 022670 063716 003022      ADD    ERRSWI,(SP)     ;YES - SKIP
717 022674 000207              RTS    PC                ;ADD IN ERROR RETURN
718 022676 017616 000000      99#: MOV    @ (SP),(SP)   ;SET ERROR RETURN ADDRESS
719 022702 000207              RTS    PC
720
721 ; GET POSITION ROUTINE. READS A HEADER FROM CURRENT CYLINDER
722 ; (WHERE IT IS PRESENTLY POSITIONED) AND STORES CYLINDER
723 ; NUMBER IN CURCYL.
724 022704 010346              GETPOS: MOV    R3,-(SP)   ;STORE REGISTERS
725 022706 013703 003006      MOV    SSINDX,R3        ;GET SUBROUTINE INDEX
726 022712 005723              TST    (R3)+            ;BUMP IT FOR NEXT ENTRY
727 022714 016663 000002 002410 MOV    2(SP),SUBSTK(R3) ;INSERT THIS CALL
    
```

```

728 022722 162763 000004 002410      SUB      #4,SUBSTK(R3)  ;ADJUST IT TO CALLING LOCATION
729 022730 010337 003006              MOV      R3,SSINDEX  ;STORE IT BACK
730 022734 010046              MOV      R0,(SP)
731 022736 010546              MOV      R5,-(SP)
732 022740 004737 021370      JSR      PC,XRDHD    ;DO READ HEADFR
733 022744 022774              65#
734 022746 013703 003056      MOV      HDWRD1,R3   ;GET HEADER WORD
735 022752 012705 000007      MOV      #7,R5      ;SET SHIFT COUNT
736 022756 006203              4# :    ASR      R3      ;SHIFT TO RIGHT JUSTIFY
737 022760 005305              DEC      R5
738 022762 001375              BNE     4#
739 022764 012703 177000      BIC     #177000,R3
740 022770 010337 003110      MOV      R3,CURCYL  ;STORE AS CURRENT CYLINDER
741 022774 162737 000002 003006 65# :    SUB      #2,SSINDEX ;REMOVE ENTRY FROM SUBROUT STACK
742 023002 012605              MOV      (SP)+,R5   ;RESTORE REGISTERS
743 023004 012600              MOV      (SP)+,R0
744 023006 012603              MOV      (SP)+,R3
745 023010 005737 003022      TST     ERRSWI     ;TEST IF ERROR RETURN
746 023014 001403              BEQ     99#        ;YES - SKIP
747 023016 063715 003022      ADD     ERRSWI,(SP) ;ADD IN ERROR RETURN
748 023022 000207              RTS     PC
749 023024 017616 000000      99# :    MOV      @ (SP),(SP) ;SET ERROR RETURN ADDRESS
750 023030 000207              RTS     PC
751
753
754 ;      VERIFY POSITION ROUTINE. READS A HEADER (USING GETPOS) AND
;      CHECKS HEADS ARE POSITIONED AT NEW CYLINDER (CURCYL = NEWCYL).
755 023032 010346              VERPOS: MOV      R3,-(SP)   ;STORE R3
756 023034 013703 003006      MOV      SSINDEX,R3 ;GET SUBROUTINE INDEX
757 023040 005723              TST     (R3)+      ;BUMP IT FOR NEXT ENTRY
758 023042 016663 000002 002410      MOV      2(SP),SUBSTK(R3) ;INSERT THIS CALL
759 023050 162763 000004 002410      SUB      #4,SUBSTK(R3) ;ADJUST IT TO CALLING LOCATION
760 023056 010337 003006      MOV      R3,SSINDEX ;STORE IT BACK
761
762 023062 012737 000002 003022      MOV      #2,ERRSWI  ;SET FOR NO ERROR RETURN
763 023070 004737 022704      JSR      PC,GETPOS  ;GET POSITION
764 023074 023122              65#
765 023076 023737 003106 003110      CMP     NEWCYL,CURCYL ;CHECK IF CURRENT CYL IS NEW CYL
766 023104 001406              BEQ     1#         ;YES - SKIP
767 023106              ERRHRD 10022,,ERR8
;      023106 104456      TRAP   C#ERRHRD
;      023110 023446      .WORD 10022
;      023112 000000      .WORD 0
;      023114 013522      .WORD ERR8
768 023116 005037 003022      CLR     ERRSWI     ;CLEAR FOR ERROR ERROR RETURN
769 023122              1# :
770 023122 162737 000002 003006 65# :    SUB      #2,SSINDEX ;REMOVE ENTRY FROM SUBROUT STACK
771 023130 012603              MOV      (SP)+,R3   ;RESTORE R3
772 023132 005737 003022      TST     ERRSWI     ;TEST IF ERROR RETURN
773 023136 001403              BEQ     99#        ;YES - SKIP
774 023140 063716 003022      ADD     ERRSWI,(SP) ;ADD IN ERROR RETURN
775 023144 000207              RTS     PC
776 023146 017616 000000      99# :    MOV      @ (SP),(SP) ;SET ERROR RETURN ADDRESS
777 023152 000207              RTS     PC
778
780 ;      READ ALL HEADERS ROUTINE. 40 HEADERS ARE READ AND STORED
781 ;      IN Ibuff.
782 023154 010346              RDALHD: MOV      R3,-(SP) ;STORE REGISTERS

```

```

783 023156 013703 003006      MOV      SSINDX,R3      ;GET SUBROUTINE INDEX
784 023162 005723              TST      (R3)+         ;BUMP IT FOR NEXT ENTRY
785 023164 016663 000002 002410  MOV      2(SP),SUBSTK(R3) ;INSERT THIS CALL
786 023172 162763 000004 002410  SUB      #4,SUBSTK(R3)  ;ADJUST IT TO CALLING LOCATION
787 023200 010337 003006      MOV      R3,SSINDX    ;STORE IT BACK
788 023204 010046              MOV      R0,-(SP)
789 023206 010146              MOV      R1,-(SP)
790 023210 010446              MOV      R4,-(SP)
791 023212 012737 000002 003022  MOV      #2,ERRSWI    ;SET FOR NO ERROR RETURN
792 023220 012701 000050              MOV      #40,R1      ;SET HEADER COUNT
793 023224 052737 100000 003010  BIS      #HDR40,OPFLAG ;SET 40 HDR OP FLAG
794 023232 012703 004072              MOV      #IBUFF,R3   ;SET POINTER TO STORE HDRS
795 023236 013704 003032              MOV      RLBAS,R4    ;GET BASE ADDRESS
796 023242 062704 000006              ADD      #RLMP,R4    ;MAKE IT POINT TO MP REG
797 023246 012737 000010 003040  MOV      #10,L.CS    ;LOAD FOR READ HEADER, NO INTERRUPT
798 023254 053737 003036 003040  BIS      RLDRV,L.CS  ;INSERT DRIVE NUMBER
799 023262 042737 002000 003040  BIC      #BIT10,L.CS ;CLEAR FOR DRIVE 4 - 7 SPEC'D
800 023270 005037 003042              CLR      L.BA        ;CLEAR BA
801 023274 005037 003044              CLR      L.DA        ;CLEAR DA
802 023300 005737 003116              TST      DESHD      ;TEST IF HEAD 0
803 023304 001403              BEQ      3$         ;YES - SKIP
804 023306 052737 000020 003044  BIS      #HSEL,L.DA  ;ELSE INSERT HEAD 0
805 023314 013762 003044 000004 3$:  MOV      L.DA,RLDA(R2) ;LOAD RLDA REG
806 023322 013762 003042 000002  MOV      L.BA,RLBA(R2) ;LOAD RLBA
807 023330 032762 000200 000000  BIT      #CRDYMSK,RLCS(R2) ;TEST IF CONTROLLER READY
808 023336 001003              BNE      6$         ;YES - SKIP
809 023340 004737 020642              JSR      PC,RDYCHK   ;ELSE CHECK READY
810 023344 023462              6$:
811 023346 013762 003040 000000 6$:  MOV      L.CS,RLCS(R2) ;LOAD RLCS REG
812 023354 012700 077777              MOV      #77777,R0  ;SET COUNT FOR WAIT
813 023360 032762 000200 000000 7$:  BIT      #CRDYMSK,RLCS(R2) ;CHECK THAT OPERATION COMPLETED
814 023366 001016              BNE      8$         ;YES - SKIP
815 023370 005300              DEC      R0         ;DEC COUNT
816 023372 001372              BNE      7$         ;SKIP IF NOT YET 0
817 023374 004737 016370              JSR      PC,READRL   ;ELSE GET ALL REGISTERS
818 023400 004737 016422              JSR      PC,WAITIN   ;ELSE WAIT FOR TIMEOUT
819 023404 012603              MOV      (SP)+,R3    ;GET RESULT MESSAGE POINTER
820 023406  ERRHRD 10025...ERR1
      023406  TRAP  C$ERRHRD
      023410  .WORD 10025
      023412  .WORD 0
      023414  .WORD ERR1
821 023416 005037 003022  CLR      ERRSWI    ;CLEAR FOR ERROR RETURN
822 023422 000417              BR      65$
823 023424 005737 003050 8$:  TST      T.CS      ;TEST FOR ANY ERRORS
824 023430 100007              BPL      12$       ;NO - SKIP
825 023432  ERRHRD 10026...ERR6
      023432  TRAP  C$ERRHRD
      023434  .WORD 10026
      023436  .WORD 0
      023440  .WORD ERR6
826 023442 005037 003022  CLR      ERRSWI    ;CLEAR FOR ERROR RETURN
827 023446 000405              BR      65$
828 023450 011423 12$:  MOV      (R4),(R3)+ ;STORE HEADER WORDS
829 023452 011423  MOV      (R4),(R3)+
830 023454 011423  MOV      (R4),(R3)+
831 023456 005301              DEC      R1         ;DEC HEADER COUNT
    
```

K7

```

832 023460 001332      BNE      6#
833 023462 162737 000002 003006 65# : SUB      @2,SSINDX ;REMOVE ENTRY FROM SUBROUT STACK
834 023470 012604      MOV      (SP)+,R4 ;RESTORE REGISTERS
835 023472 012601      MOV      (SP)+,R1
836 023474 012600      MOV      (SP)+,R0
837 023476 012603      MOV      (SP)+,R3
838 023500 005737 003022      TST      ERRSWI ;TEST IF ERPOP RETURN
839 023504 001403      BEQ      99# ;YES - SKIP
840 023506 063716 003022      ADD      ERRSWI,(SP) ;ADD IN ERROR RETURN
841 023512 000207      RTS
842 023514 017616 000000      99# : MOV      @ (SP),(SP) ;SET ERROR RETURN ADDRESS
843 023520 000207      RTS      PC
844
845
846
847 ; GENERATE DATA ROUTINE. PATTERN TO BE GENERATED IS GIVEN
848 ; IN THE WORD FOLLOWING THE CALL. 128 WORDS ARE GENERATED
849 ; IN OBUFF.
850 023522 010146      DATGEN: MOV      R1,-(SP) ;STORE REGISTERS
851 023524 010346      MOV      R3,-(SP)
852 023526 010446      MOV      R4,-(SP)
853 023530 012701 004472      MOV      @OBUFF,R1 ;SET POINTER TO OBUFF
854 023534 012504      MOV      (R5)+,R4 ;GET DATA PATTERN SELECTOR
855 023536 006304      ASL      R4 ;ADJUST IT FOR INDEXING
856 023540 016403 002364      MOV      PATTBL(R4),R3 ;GET ADDRESS OF PATTERN
857 023544 011321      MOV      (R3),(R1)+ ;MOVE FIRST PATTERN WORD
858 023546 001421      BEQ      5# ;SKIP IF PATTERN IS 0
859 023550 021327 177777      CMP      (R3),@-1 ;CHECK IF PATTERN IS ALL 1'S
860 023554 001416      BEQ      5# ;YES - SKIP
861 023556 020427 000010      CMP      R4,@8. ;TEST IF PATTERN 5
862 023562 001403      BEQ      3# ;YES - SKIP
863 023564 020427 000020      CMP      R4,@16. ;CHECK IF PATTERN 9 OR 10
864 023570 002413      BLT      6# ;NO - SKIP
865 023572 005723      3# : TST      (R3)+ ;BUMP SOURCE POINTER
866 023574 012321      MOV      (R3)+,(R1)+ ;MOVE TWO MORE WORDS FORM SOURCE
867 023576 012321      MOV      (R3)+,(R1)+
868 023600 012704 000015      MOV      @13.,R4 ;SET COUNT
869 023604 012703 004472      MOV      @OBUFF,R3 ;RESET POINTER
870 023610 000406      BR
871 023612 012703 004472      5# : MOV      @OBUFF,R3 ;ELSE SET OBUFF AS PATTERN SOURCE
872 023616 000401      BR      7# ;GO TO FILL
873 023620 005723      6# : TST      (R3)+ ;BUMP SOURCE POINTER
874 023622 012704 000017      7# : MOV      @15.,R4 ;SET MOVE COUNT
875 023626 012321      8# : MOV      (R3)+,(R1)+ ;MOVE 15 WORDS INTO BUFFER
876 023630 005304      DEC      R4
877 023632 001375      BNE      8#
878 023634 012703 004472      MOV      @OBUFF,R3 ;SET SOURCE TO TOP OF OBUFF
879 023640 012704 000160      MOV      @112.,R4 ;SET COUNT FOR REST OF BUFFER
880 023644 012321      10# : MOV      (R3)+,(R1)+ ;REPEAT PATTERN IN BUFFER
881 023646 005304      DEC      R4
882 023650 001375      BNE      10#
883 023652 012604      MOV      (SP)+,R4 ;RESTORE REGISTERS
884 023654 012603      MOV      (SP)+,R3
885 023656 012601      MOV      (SP)+,R1
886 023660 000205      RTS      R5 ;RETURN
887
888 ; DATA COMPARE ROUTINE. COMPARES THE CONTENTS OF IBUFF AND OBUFF.
889 ; ERROR REPORTING IS LIMITED BY SOFTWARE PARAMETER.
    
```



```

890 023662 010346          DATCOM: MOV      R3,(SP)          ;STORE R3
891 023664 013703 003006  MOV      SSINDY,R3      ;GET SUBROUTINE STACK INDEX
892 023670 005723          TST      (R3)+          ;BUMP INDEX TO NEXT ENTRY
893 023672 016663 000002 002410  MOV      2(SP),SUBSTK(R3) ;INSERT THIS CALL
894 023700 162763 000004 002410  SUB      @4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
895 023706 010337 003006          MOV      R3,SSINDX      ;STORE IT BACK
896 023712 010146          MOV      R1,-(SP)       ;STORE OTHER REGISTERS
897 023714 010446          MOV      R4,-(SP)
898 023716 010546          MOV      R5,-(SP)
899 023720 052737 000001 003010  BIS      @DATACMP,OPFLAG ;SET DATA COMPARE FLAG
900 023726 005037 003020          CLR      MORECE        ;CLEAR MORE ERROR FLAG
901 023732 012705 004472          MOV      @OBUFF,R5     ;SET POINTERS TO DATA FOR COMPARE
902 023736 012704 004072          MOV      @IBUFF,R4
903 023742 012703 000001          MOV      @1,R3         ;SET WORD COUNTER
904 023746 012701 000200          MOV      @128.,R1      ;SET COMPARE COUNT
905 023752 022425          5$:  CMP      (R4)+,(R5)+  ;COMPARE DATA
906 023754 001052          BNE     10$           ;ERROR - SKIP TO REPORT
907 023756 005203          7$:  INC      R3         ;BUMP WORD COUNT
908 023760 005301          DEC      R1           ;DEC COMPARE COUNT
909 023762 001373          BNE     5$           ;LOOP IF NOT 0
910 023764 042737 000001 003010  9$:  BIC      @DATACMP,OPFLAG ;CLEAR DATA COMPARE FLAG
911 023772 005737 003022          TST      ERRSWI        ;TEST IF ANY COMPARE ERRORS
912 023776 001021          BNE     15$          ;NO - SKIP
913 024000 012701 000200          MOV      @128.,R1      ;SET REPORT VALUE
914 024004          PRINTB @FMT27,@TCERR,MORECE,@RESE6,R1
      024004 010146          MOV      R1,-(SP)
      024006 012746 011141          MOV      @RESE6,-(SP)
      024012 013746 003020          MOV      MORECE,-(SP)
      024016 012746 007614          MOV      @TCERR,-(SP)
      024022 012746 012235          MOV      @FMT27,-(SP)
      024026 012746 000005          MOV      @5,-(SP)
      024032 010600          MOV      SP,R0
      024034 104414          TRAP    C$PNTB
      024036 062706 000014          ADD      @14,SP
915 024042 162737 000002 003006 15$:  SUB      @2,SSINDX      ;REMOVE ENTRY FROM SUBROUT STACK
916 024050 012605          MOV      (SP)+,R5     ;RESTORE REGS
917 024052 012604          MOV      (SP)+,R4
918 024054 012601          MOV      (SP)+,R1
919 024056 012603          MOV      (SP)+,R3
920 024060 005737 003022          TST      ERRSWI        ;TEST IF ERROR RETURN
921 024064 001403          BEQ     99$          ;YES - SKIP
922 024066 063716 003022          ADD      ERRSWI,(SP)  ;ADD IN ERROR RETURN
923 024072 000207          RTS     PC
924 024074 017616 000000          99$:  MOV      @8(SP),(SP)   ;SET ERROR RETURN ADDRESS
925 024100 000207          RTS     PC
926 024102 023737 003020 014132 10$:  CMP      MORECE,DCLIMW ;TEST IF COMPARE ERRORS LIMIT EXCEEDED
927 024110 002011          BGE     13$          ;YES - SKIP
928 024112 024445          CMP      -(R4)+,(R5)  ;SET PTRS BACK TO FRROR WORDS
929 024114          ERRHRD 10035.,ERR10  ;REPORT ERROR
      024114 104456          TRAP    C$ERHRD
      024116 023463          .WORD  10035
      024120 000000          .WORD  0
      024122 013662          .WORD  ERR10
930 024124 005037 003022          CLR      ERRSWI        ;CLEAR ERROR SWITCH
931 024130 022425          CMP      (R4)+,(R5)+  ;BUMP PTRS PAST ERROR WORD
932 024132 000711          BR      7$           ;DO NEXT COMPARE
933 024134 005237 003020          13$:  INC      MORECE        ;BUMP ERROR COUNTER

```

M7

934 024140 000706

BR 74

;DO NEXT COMPARE

```

1
2
3
4 024142 012737 177777 003124 XWRITT: MOV    0-1,TEMP1      ;SET SPECIAL WRITE FOR TIMING FLAG
5 024150 000402                BR      XWRIT1
6 024152 005037 003124        XWRITE: CLR    TEMP1          ;CLEAR SPECIAL WRITE FLAG
7 024156 012737 000112 003140 XWRIT1: MOV    0WTDATA,TEMP7    ;SET FOR WRITE
8 024164 023737 002306 003110        CMP    HLMTW,CURCYL          ;TEST IF CYLINDER 255 (BAD SEC)
9 024172 001006                BNE     1$                    ;NO - SKIP
10 024174 005737 003116        TST    DESHD                 ;TEST IF HEAD 1 (BAD SECTOR FILES)
11 024200 001403                BEQ     1$                    ;NO - SKIP
12 024202 052737 004000 003010        BIS    0BADADD,OPFLAG        ;SET BAD ADDRESS FLAG
13 024210 000403                1$:   BR      XREADG          ;SKIP TO EXECUTE
14 024212 012737 000114 003140 XREAD: MOV    0RDDATA,TEMP7    ;SET FOR READ
15 024220 010346        XREADG: MOV    R3,-(SP)        ;STORE R3
16 024222 013703 003006        MOV    SSINDEX,R3           ;SET SUBROUTINE INDEX
17 024226 005723                TST    (R3)+                 ;BUMP TO NEXT STACK ENTRY
18 024230 016663 000002 002410        MOV    2(SP),SUBSTK(R3)      ;INSERT THIS CALL
19 024236 162763 000004 002410        SUB    04,SUBSTK(R3)         ;ADJUST TO POINT TO CALL
20 024244 010337 003006        MOV    R3,SSINDEX           ;STORE IT BACK
21 024250 010046                MOV    R0,-(SP)
22 024252 010146                MOV    R1,-(SP)             ;STORE OTHER REGISTERS
23 024254 010446                MOV    R4,-(SP)
24 024256 004737 020642        JSR    PC,RDYCHK            ;CHECK IF DRIVE READY
25 024262 024650                65$
26 024264 012703 003040        MOV    0L,CS,R3             ;GET ADDRESS OF LOAD REGS
27 024270 013713 003140        MOV    TEMP7,(R3)           ;SET COMMAND
28 024274 053713 003036        BIS    RLDRV,(R3)           ;INSERT DRIVE NUMBER
29 024300 042713 002000        BIC    0BIT10,(R3)          ;CLEAR FOR DRIVE 4 - 7 SPEC'D
30 024304 032723 000004        BIT    0BIT2,(R3)+         ;TEST IF WRITE DATA
31 024310 001403                BEQ     3$                    ;YES - SKIP
32 024312 012723 004072        MOV    0IBUFF,(R3)+        ;ELSE SET BA FOR READ
33 024316 000402                BR      4$
34 024320 012723 004472        3$:   MOV    0OBUFF,(R3)+      ;SET BA FOR WRITE
35 024324 013713 003110        4$:   MOV    CURCYL,(R3)       ;GET CURRENT CYLINDER
36 024330 012704 000007        MOV    07,R4                ;ALIGN IT IN DA
37 024334 006313        5$:   ASL    (R3)
38 024336 005304                DEC    R4
39 024340 001375                BNE     5$
40 024342 005737 003116        TST    DESHD                ;TEST IF HEAD 0
41 024346 001402                BEQ     7$                    ;YES - SKIP
42 024350 052713 000100        BIS    0HSMK,(R3)          ;SET FOR HEAD 1
43 024354 053723 003120        7$:   BIS    DESSEC,(R3)+       ;INSERT DESIRED SECTOR
44 024360 012713 177600        MOV    0177600,(R3)        ;INSERT WORD COUNT
45 024364 005737 003124        TST    TEMP1                ;CHECK IF SPECIAL WRITE FOR TIMING
46 024370 001402                BEQ     8$                    ;NO - SKIP
47 024372 012713 177777        MOV    0177777,(R3)        ;ELSE SET FOR 1 WORD TRANSFER
48 024376 032737 004000 003010        8$:   BIT    0BADADD,OPFLAG      ;TEST IF BAD ADDRESS FLAG SET
49 024404 001414                BEQ     2$                    ;NO - SKIP
50 024406 042737 173777 003010        BIC    0+CBADADD,OPFLAG     ;CLEAR ALL BUT THIS FLAG
51 024414 012703 011043        MOV    0MWRTAB,R3          ;SET RESULT MESSAGE POINTER
52 024420                ERRHRD 10032,,ERR1
    024420 104456                TRAP   C$ERRHD
    024422 023460                .WORD 10032
    024424 000000                .WORD 0
    024426 012266                .WORD ERR1
53 024430 005037 003010        CLR    OPFLAG              ;CLEAR ALL FLAGS

```

```

54 024434 000503
55 024436 005037 003012 21: BR 641
56 024442 005737 003124 CLR DONE ;CLEAR INTERRUPT FLAG
57 024446 001100 TST TEMP1 ;CHECK IF SPECIAL WRITE FLAG SET
58 024450 011362 000006 BNE 651 ;YES DO NOT START WRITE
59 024454 014362 000004 MOV (R3),RLMP(R2) ;LOAD RL REGS
60 024460 014362 000002 MOV -(R3),RLD4(R2)
61 024464 014362 000000 MOV -(R3),RLBA(R2)
62 024470 101: MOV -(R3),RLCS(R2)
63 024502 005737 003012 WAITUS #3000. ;WAIT 300MS FOR INTERRUPT
64 024506 001010 TST DONE ;CHECK IF INTERRUPT
65 024510 004737 016422 BNE 141 ;YES SKIP
66 024514 012603 JSR PC,WAITIN ;WAIT FOR INTERRUPT
67 024516 012603 MOV (SP)+,R3 ;GET RESULT MESSAGE
    024516 104456 ERRHRD 10030,,,ERR1
    024520 023456 TRAP CERRHRD
    024522 000000 .WORD 10030
    024524 012266 .WORD 0
68 024526 000446 .WORD ERR1
69 024530 032737 000001 003050 141: BR 641
70 024536 001033 BIT #DRDYMSK,T.CS ;TEST IF DRIVE READY
71 024540 012703 010322 BNE 201 ;YES - SKIP
72 024544 012704 011224 MOV #DRDY,R3 ;SET RESULT MESSAGE
73 024550 012704 011224 MOV #CAFDT,R4 ;CONDITION AFTER DATA XFER
    024550 104456 ERRHRD 10032,,,ERR5
    024552 023460 TRAP CERRHRD
    024554 000000 .WORD 10032
    024556 012520 .WORD 0
74 024560 012701 000062 MOV #50.,R1 ;SET WAIT COUNT FOR 5 SECDS
75 024564 004737 016626 171: JSR PC,GSTAT ;GET DRIVE STATUS
76 024570 024644 641
77 024572 032737 000001 003050 BIT #DRDYMSK,T.CS ;TEST IF DRIVE READY NOW
78 024600 001012 BNE 201 ;YES SKIP
79 024602 005301 DEC R1 ;DEC WAIT COUNT
80 024604 001367 BNE 171 ;LOOP IF NOT TIME DONE
81 024606 012704 011235 MOV #C5SEC,R4 ;SET CONDITION 5 SECONDS
82 024612 012704 011235 ERRHRD 10033,,,ERR5
    024612 104456 TRAP CERRHRD
    024614 023461 .WORD 10033
    024616 000000 .WORD 0
    024620 012520 .WORD ERR5
83 024622 005037 003022 CLR ERRSWI ;CLEAR ERROR SWITCH
84 024626 005737 003050 201: TST T.CS ;CHECK IF ANY ERROR
85 024632 100006 BPL 651 ;NO SKIP
86 024634 100006 ERRHRD 10031,,,ERR6
    024634 104456 TRAP CERRHRD
    024636 023457 .WORD 10031
    024640 000000 .WORD 0
    024642 012570 .WORD ERR6
87 024644 005037 003022 641: CLR ERRSWI ;CLEAR ERROR SWITCH
88 024650 162737 000002 003006 651: SUB #2,SSINDEX ;REMOVE ENTRY FROM SUBROUT STACK
89 024656 012604 MOV (SP)+,R4 ;RESTORE REGISTERS
90 024660 012601 MOV (SP)+,R1
91 024662 012600 MOV (SP)+,R0
92 024664 012603 MOV (SP)+,R3
93 024666 005737 003022 TST ERRSWI ;TEST IF ERROR RETURN
94 024672 001403 BEQ 991 ;YES SKIP
    
```

CH

```

95 024674 063716 003022      ADD     ERRSMI,(SP)      ;ELSE ADD IN ERROR RETURN
96 024700 000207              RTS     PC
97 024702 017616 000000      99$:   MOV     @ (SP),(SP)      ;ADJUST FOR ERROR RETURN
98 024706 000207              RTS     PC
99
100
101      ;      BAD SECTOR CHECK ROUTINE. CHECKS IF SECTOR SPECIFIED IN CIRC'L,
102 024710 010046      BSCMK:  MOV     R0,(SP)      ;STORE REGISTERS
103 024712 010146      ;      DESHD, AND DESSEC IS LISTED AS BAD IN THE BAD SECTOR FILES.
104 024714 010346      MOV     R1,-(SP)
105 024716 005037 003024      MOV     R3,(SP)
106 024722 012703 003676      CLR     BSFLAG          ;CLEAR FLAG
107 024726 022713 177777      MOV     @FBSFIL,R3      ;GET POINTER TO FACTORY FILE
108 024732 001005              CMP     @-1,(R3)        ;CHECK IF ALL ONES
109 024734 012703 003502      BNE     4$              ;NO SKIP TO TEST
110 024740 022713 177777      2$:   MOV     @SBSFIL,R3      ;ELSE SET POINTER TO SOFTWARE FILE
111 024744 001431              CMP     @-1,(R3)        ;CHECK IF ALL ONES
112 024746 013700 003106      BEQ     20$             ;YES - EXIT
113 024752 012701 000007      4$:   MOV     NEWCYL,R0      ;BUILD HEADER OF ADDRESS IN QUESTION
114 024756 006300              MOV     @7,R1           ;POSITION CYLINDER
115 024760 005301              5$:   ASL     R0
116 024762 001375              DEC     R1
117 024764 005737 003116      BNE     5$
118 024770 001402              TST     DESHD           ;CHECK IF HEAD 0
119 024772 052700 000100      BEQ     7$              ;YES - SKIP
120 024776 053700 003120      BIS     @BIT6,R0        ;INSERT HEAD 1
121 025002 022300              7$:   BIS     DESSEC,R0      ;INSERT SECTOR
122 025004 001402              8$:   CMP     (R3),R0        ;CHECK THIS WORD IN FILE
123 025006 101005              BEQ     12$             ;YES - EXIT,ERROR
124 025010 000774              BMI     15$             ;EXIT NO ERROR
125 025012 012737 000001 003024 12$:  MOV     @1,BSFLAG       ;SET ERROR FLAG
126 025020 000403              BR     20$             ;GO TO EXIT
127 025022 020327 003676      15$:  CMP     R3,@FBSFIL      ;DONE BOTH FILES?
128 025026 003342              BGT     2$              ;NO GO DO SOFTWARE FILE
129 025030 012603              20$:  MOV     (SP),R3         ;ELSE RESTORE REGISTERS
130 025032 012601              MOV     (SP),R1
131 025034 012600              MOV     (SP),R0
132 025036 005737 003024      TST     BSFLAG          ;CHECK IF ERROR
133 025042 001003              BNE     99$             ;YES - SKIP
134 025044 062716 000002      ADD     @2,(SP)         ;ELSE BUMP ERROR RETURN
135 025050 000207              RTS     PC
136 025052 017616 000000      99$:  MOV     @ (SP),(SP)      ;SET FOR ERROR RETURN
137 025056 000207              RTS     PC
138
139
140      ;      REPORT OPERATION ROUTINE. PRINTS SUBROUTINE TRACE SEQUENCE AND
141      ;      OPERATION BEING PERFORMED PORTION OF ALL
142      ;      ERROR MESSAGES.
143 025060 010446      PPTOP: MOV     R4,-(SP)
144 025062 005737 003006      TST     SSINDEX        ;TEST SUBROUTINE INDEX 0
145 025066 001433              BEQ     1$              ;SKIP IF 0
146 025070 012704 000002      MOV     @2,R4          ;SET INDEXER TO FIRST ENTRY
147 025074              PRINTB @FMT9,@SEQMES   ;PRINT "SUBROUTINE CALL SEQ
148      MOV     @SEQMES,-(SP)
149      MOV     @FMT9,-(SP)
150      MOV     @2,-(SP)
151      MOV     SP,R0
152      TRAP  C$PNTB
    
```

Das

```

025114 062706 000006
148 025120 016446 002410      3:  ADD    #6,SP
    025120 012746 011727      PRINTB #FMT16,SUBSTK(R4)      ;PRINT CALLING LOCATION
    025124 012746 000002      MOV    SUBSTK(R4), (SP)
    025130 012746 000002      MOV    #FMT16, (SP)
    025134 010600      MOV    #2, (SP)
    025136 104414      MOV    SP,R0
    025140 062706 000006      TRAP  C#PNTB
    025144 062704 000002      ADD    #6,SP
149 025144 062704 000002      ADD    #2,R4      ;BUMP INDEX
150 025150 020437 003006      CMP    R4,SSINDX  ;CHECK IF ALL PRINTED
151 025154 003761      BLE    3:         ;LOOP IF NOT ALL PRINTED YET
152 025156      1:  PRINTB #FMT4,ERHEAD,#TSTLAB  ;PRINT ERROR HEADER
    025156 012746 006471      MOV    #TSTLAB,-(SP)
    025162 013746 003016      MOV    ERHEAD,-(SP)
    025166 012746 C11357      MOV    #FMT4,-(SP)
    025172 012746 000003      MOV    #3,-(SP)
    025176 010600      MOV    SP,R0
    025200 104414      TPAP  C#PNTB
    025202 062706 000010      ADD    #10,SP
153 025206 042737 030000 003010  BIC    #SEEKOP!RORWOP,OPFLAG  ;CLEAR SK & RD OR WRT FLAG
154 025214 013701 003040      MOV    L.CS,R1      ;GET COMMAND EXECUTED
155 025220 042701 177741      BIC    #177741,R1    ;STRIP ALL BUT FUNCTION COOL
156 025224 022701 000006      CMP    #6,R1      ;TEST IF SEEK OPERATION
157 025230 001003      BNE    2:         ;NO - SKIP
158 025232 052737 010000 003010  BIS    #SEEKOP,OPFLAG  ;ELSE SET SEEK FLAG
159 025240 022701 000012      2:  CMP    #12,R1      ;TEST IF WRITE
160 025244 001003      BNE    20:        ;NO - SKIP
161 025246 052737 020000 003010  BIS    #RORWOP,OPFLAG  ;SET RD OR WRT FLAG
162 025254 022701 000014      20:  CMP    #14,R1      ;TEST IF READ
163 025260 001003      BNE    22:        ;NO - SKIP
164 025262 052737 020000 003010  BIS    #RORWOP,OPFLAG  ;SET RD OR WRT FLAG
165 025270      22:  PRINTB #FMT1,#MOPER,OPMSGs(R1) ;PRINT OPERATION
    025270 016146 002230      MOV    OPMSGs(R1),-(SP)
    025274 012746 005517      MOV    #MOPER,-(SP)
    025300 012746 011335      MOV    #FMT1,-(SP)
    025304 012746 000003      MOV    #3,-(SP)
    025310 010600      MOV    SP,R0
    025312 104414      TRAP  C#PNTB
    025314 062706 000010      ADD    #10,SP
166 025320 020127 000004      CMP    R1,#4      ;CHECK IF GET STATUS
167 025324 001007      BNE    4:         ;NO - SKIP
168 025326 032737 000010 003044  BIT    #DRSET,L.DA  ;TEST IF RESET INCLUDED
169 025334 001403      BEQ    4:         ;NO - SKIP
170 025336 012701 000016      MOV    #16,R1     ;SET TO PRINT WITH RESET
171 025342 000436      BR     9:
172 025344 032737 007777 003010  4:  BIT    #COMPOP,OPFLAG  ;TEST IF ANY OTHER OPERATION
173 025352 001424      BEQ    8:         ;NO - SKIP
174 025354 013704 003010      MOV    OPFLAG,R4  ;SET UP TO DETERMINE WHICH ONE
175 025360 012701 000020      MOV    #20,R1     ;PRESET THE POINTER
176 025364 032704 000001      5:  BIT    #BIT00,R4  ;CHECK THE BIT
177 025370 001003      BNE    6:         ;IF SET - SKIP
178 025372 005721      TST   (R1)+      ;BUMP POINTER
179 025374 006204      ASR   R4
180 025376 000772      BR     5:
181 025400      6:  PRINTB #FMT2,OPMSGs(R1)
    025400 016146 002230      MOV    OPMSGs(R1),-(SP)
    025404 012746 011351      MOV    #FMT2,-(SP)
    
```

```

025410 012746 000002      MOV      #2,-(SP)
025414 010600      MOV      SP,RO
025416 104414      TRAP    C#PNTB
025420 062706 000006      ADD      #6,SP
182 025424 032737 100000 003010 8#: BIT      #HDR40,OPFLAG ;TEST IF 40 HEADER OPERATION
183 025432 001415      BEQ     10# ;NO SKIP
184 025434 012701 000050      MOV      #50,R1 ;ELSE PRINT IT
185 025440          9#: PRINTB  #FMT2,OPMSG5(R1)
025440 016146 002230      MOV      OPMSG5(R1),-(SP)
025444 012746 011351      MOV      #FMT2,-(SP)
025450 012746 000002      MOV      #2,-(SP)
025454 010600      MOV      SP,RO
025456 104414      TRAP    C#PNTB
025460 062706 000006      ADD      #6,SP
186 025464 000434      BR      15# ;SKIP
187 025466 032737 010000 003010 10#: BIT      #SEEKOP,OPFLAG ;TEST IF SEEK
188 025474 001430      BEQ     15# ;NO - SKIP
189 025476          PRINTB  #FMT13,#FRMWD,OLDCYL,#DIFWD,DESDIF,#SGNWD,DESSGN,#HWD,DESHD
025476 013746 003116      MOV      DESHD,-(SP)
025502 012746 007445      MOV      #HWD,-(SP)
025506 013746 003114      MOV      DESSGN,-(SP)
025512 012746 007440      MOV      #SGNWD,-(SP)
025516 013746 003112      MOV      DESDIF,-(SP)
025522 012746 007432      MOV      #DIFWD,-(SP)
025526 013746 003104      MOV      OLDCYL,-(SP)
025532 012746 007463      MOV      #FRMWD,-(SP)
025536 012746 011575      MOV      #FMT13,-(SP)
025542 012746 000011      MOV      #11,-(SP)
025546 010600      MOV      SP,RO
025550 104414      TRAP    C#PNTB
025552 062706 000024      ADU     #24,SP
190 025556 032737 020000 003010 15#: BIT      #RORWOP,OPFLAG ;TEST IF READ OR WRITE SET
191 025564 001424      BEQ     7# ;NO - SKIP
192 025566          PRINTB  #FMT22,#CYLWD,CURCYL,#HWD,DESHD,#SECWD,DESSEC
025566 013746 003120      MOV      DESSEC,-(SP)
025572 012746 007451      MOV      #SECWD,-(SP)
025576 013746 003116      MOV      DESHD,-(SP)
025602 012746 007445      MOV      #HWD,-(SP)
025606 013746 003110      MOV      CURCYL,-(SP)
025612 012746 007456      MOV      #CYLWD,-(SP)
025616 012746 012124      MOV      #FMT22,-(SP)
025622 012746 000007      MOV      #7,-(SP)
025626 010600      MOV      SP,RO
025630 104414      TRAP    C#PNTB
025632 062706 000020      ADD      #20,SP
193 025636 004737 026310 17#: JSR     PC,CLRPARM ;CLEAR PARAM TABLE
194 025642 012604      MOV      (SP)+,R4 ;RESTORE R4
195 025644 000207      RTS     PC
196
197 ;
198 ; REPORT REASON ROUTINE
199 025646 010146      ; PRINTS REASON PORTION FOR ALL ERROR REPORTS.
200 025650 010346      RPTRES: MOV      R1,-(SP) ;STORE R1
201 025652 010446      MOV      R3,-(SP) ;STORE R3
202 025654 012701 003066      MOV      R4,-(SP) ;STORE R4
203 025660 012103      MOV      #RESPARM,R1 ;GET START OF PARAM
204 025662          MOV      (R1)+,R3 ;GET NUMBER OF PARAM
          PRINTB  #FMT1.1,#MRSLT,(R1) ;PRINT NAME
    
```

025662	011146		MOV	(R1), (SP)	
025664	012746	005526	MOV	#MRSLT, -(SP)	
025670	012746	011342	MOV	#FMT1.1, -(SP)	
025674	012746	000003	MOV	#3, -(SP)	
025700	010600		MOV	SP, R0	
025702	104414		TRAP	C:PNTB	
025704	062706	000010	ADD	#10, SP	
205 025710	02:127	010714	CMP	(R1), #MNRST	;TEST IF MESSAGE IS NO DRV STATUS
206 025714	001453		BEQ	6:	;YES - SKIP REST OF REPORT
207 025716	012704	011561	MOV	#FMT11, R4	;PRISET FOR FORMAT 11
208 025722	022127	010707	CMP	(R1), #MCYLOC	;CHECK IF REPORTING CYLINDER LOC
209 025726	001002		BNE	3:	;NO - SKIP
210 025730	012704	011567	MOV	#FMT12, R4	;ELSE CHANGE TO FORMAT 12
211 025734	005303		DEC	R3	;DEC PARAM COUNT
212 025736	001442		BEQ	6:	;IF 0 - EXIT
213 025740			PRINTB	R4, #RESE3, (R1)	;REPORT IS VALUE
025740	012146		MOV	(R1), -(SP)	
025742	012746	011123	MOV	#RESE3, -(SP)	
025746	010446		MOV	R4, -(SP)	
025750	012746	000003	MOV	#3, -(SP)	
025754	010600		MOV	SP, R0	
025756	104414		TRAP	C:PNTB	
025760	062706	000010	ADD	#10, SP	
214 025764			PRINTB	R4, #RESE4, (R1)	;REPORT SB VALUE
025764	012146		MOV	(R1), -(SP)	
025766	012746	011127	MOV	#RESE4, -(SP)	
025772	010446		MOV	R4, -(SP)	
025774	012746	000003	MOV	#3, -(SP)	
026000	010600		MOV	SP, R0	
026002	104414		TRAP	C:PNTB	
026004	062706	000010	ADD	#10, SP	
215 026010	162703	000002	SUB	#2, R3	;DEC PARAM COUNT
216 026014	001413		BEQ	6:	;IF 0 EXIT
217 026016			PRINTB	#FMT1, #RESE5, (R1)	;REPORT CONDITION
026016	012146		MOV	(R1), -(SP)	
026020	012746	011134	MOV	#RESE5, -(SP)	
026024	012746	011335	MOV	#FMT1, -(SP)	
026030	012746	000003	MOV	#3, -(SP)	
026034	010600		MOV	SP, R0	
026036	104414		TRAP	C:PNTB	
026040	062706	000010	ADD	#10, SP	
218 026044	012604		MOV	(SP), R4	;RESTORE REGS
219 026046	012603		MOV	(SP), R3	
220 026050	012601		MOV	(SP), R1	
221 026052	000207		RTS	PC	;RETURN
222					
223					
224					
225 026054					
026054	005046		RPTREM: PRINTB	#FMT5, #BASADD, RLBAS, #DRVNAM, <B, RLDV*1>	
026056	153716	003037	CLR	-(SP)	
026062	012746	006142	BISB	RLDV*1, (SP)	
026066	013746	003032	MOV	#DRVNAM, -(SP)	
026072	012746	006131	MOV	RLBAS, -(SP)	
026076	012746	011370	MOV	#BASADD, -(SP)	
026102	012746	000005	MOV	#FMT5, -(SP)	
026106	010600		MOV	#5, -(SP)	
			MOV	SP, R0	



```

026110 104414
026112 062706 000014
226 026116
227 026116 012746 007445
026116 012746 007456
026122 012746 006245
026126 012746 006233
026132 012746 006240
026136 012746 006226
026142 012746 011410
026146 012746 000007
026152 010600
026156 104414
026160 062706 000020
228 026166
026166 013746 003046
026172 013746 003042
026176 013746 003044
026202 013746 003040
026206 012746 006252
026212 012746 011522
026216 012746 000006
026222 010600
026224 104414
026226 062706 000016
229 026232
026232 013746 003116
026236 013746 003110
026242 013746 003056
026246 013746 003052
026252 013746 003054
026256 013746 003050
026262 012746 006265
026266 012746 011452
026272 012746 000010
026276 010600
026300 104414
026302 062706 000022
230 026306 000207
231
232
233 026310 010546
234 026312 012701 003066
235 026316 012705 000005
236 026322 005021
237 026324 005305
238 026326 001375
239 026330 012701 003066
240 026334 012605
241 026336 000207
242
243 026340
244

TRAP C#PNTB
ADD #14,SP
REPORT RL11 REGISTERS
PRINTB #FMT6,#CSNAM,#DANAM,#BANAM,#MPNAM,#CYLWD,#HWDWD
MOV #HWDWD,(SP)
MOV #CYLWD,-(SP)
MOV #MPNAM,-(SP)
MOV #BANAM,-(SP)
MOV #DANAM,-(SP)
MOV #CSNAM,-(SP)
MOV #FMT6,-(SP)
MOV #7,-(SP)
MOV SP,RO
TRAP C#PNTB
ADD #20,SP
PRINTB #FMT8,#LAB1,L.CS,L.DA,L.BA,L.MP
MOV L.MP,-(SP)
MOV L.BA,-(SP)
MOV L.DA,-(SP)
MOV L.CS,-(SP)
MOV #LAB1,-(SP)
MOV #FMT8,-(SP)
MOV #6,-(SP)
MOV SP,RO
TRAP C#PNTB
ADD #16,SP
PRINTB #FMT7,#LAB2,T.CS,T.DA,T.BA,T.MP,CURCYL,DESMO
MOV DESMO,-(SP)
MOV CURCYL,-(SP)
MOV T.MP,-(SP)
MOV T.BA,-(SP)
MOV T.DA,-(SP)
MOV T.CS,-(SP)
MOV #LAB2,-(SP)
MOV #FMT7,-(SP)
MOV #10,-(SP)
MOV SP,RO
TRAP C#PNTB
ADD #22,SP
RTS PC

; 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

```

```

1          .TITLE  CZRLN80 RL01/02 DRIVE TEST 3
2
3 026340   BGNMOD  HRDWTST
4
5          .SBTTL  *TEST 1          **SEEK TIMING
6
7 026340   BGNSTST          ;TEST 1
8 026340   012737 006664 003016   MOV      #P2T12E,ERHEAD ;SET ERROR HEADER
9          ;CHECK FOR PRESENCE OF A P-CLOCK...BYPASS TEST IF NOT AVAILABLE
10 026346  005737 003474   TST      CLKFLG        ;P-CLOCK?
11 026352  001026          BNE      3$           ;BRANCH TO PERFORM TEST IF P-CLOCK IS PRESENT
12 026354          PRINTF  #FMT9,#NOTST1 ;ELSE, PRINT MSG. "TEST 1 CANNOT BE PERFORMED...
    026354  012746 007750   MOV      #NOTST1,-(SP)
    026360  012746 011554   MOV      #FMT9,-(SP)
    026364  012746 000002   MOV      #2,-(SP)
    026370  010600          MOV      SP,R0
    026372  104417          TRAP    C$PNTF
    026374  062706 000006   ADD      #6,SP
13
14 026400          PRINTF  #FMT9,#NTST1A ;/P-CLOCK IS NOT AVAILABLE"
    026400  012746 010036   MOV      #NTST1A,-(SP)
    026404  012746 011554   MOV      #FMT9,-(SP)
    026410  012746 000002   MOV      #2,-(SP)
    026414  010600          MOV      SP,R0
    026416  104417          TRAP    C$PNTF
    026420  062706 000006   ADD      #6,SP
15 026424  000137 030274   JMP      65$
16 026430  004737 016560   3$: JSR      PC,TSTINT ;EXIT TEST
17 026434  004737 016576   JSR      PC,GSTATR  ;INITIALIZE TEST
18 026440  030274          65$
19 026442  012700 003144   MOV      #OFIN,R0    ;GET ADDRESS OF 1ST TIME VALUE
20 026446  012701 000030   MOV      #24.,R1    ;SET COUNT FOR CLEAR
21 026452  005020          4$: CLR      (R0)+      ;CLEAR TIMER STORAGE
22 026454  005301          DEC      R1
23 026456  001375          BNE     4$
24 026460  005037 003236   CLR     PASCNT      ;CLEAR PASS COUNTER
25 026464  005037 003106   CLR     NEWCYL     ;POSITION HEADS AT 0
26 026470  004737 017524   JSR     PC,XSEEK   ;DO SEEK
27 026474  030274          65$
28 026476  012701 005670   MOV     #3000.,R1  ;SET WAIT FOR 300 MS
29 026502  004737 022420   JSR     PC,RDYWAIT ;WAIT FOR READY
30 026506  030274          65$
31 026510  004737 023032   JSR     PC,VERPOS  ;VERIFY POSITION
32 026514  030274          65$
33 026516  004737 021116   JSR     PC,CHOS$D  ;GO CHOSE HEAD
34 026522  012700 003154   MOV     #OFOUT,R0  ;SET PTRS FOR 1 CYL FWD OUTER TIMER
35 026526  012701 003156   MOV     #OFOUTU,R1
36 026532  012703 003170   MOV     #OROUT,R3
37 026536  012704 003172   MOV     #OROUTU,R4
38 026542  012737 000001 003106   MOV     #1,NEWCYL ;SET NEWCYL TO CYL 1
39 026550  012737 000200 003240   8$: MOV     #128.,COUNT ;SET COUNTER FOR SEEK LOOP
40 026556  012737 000110 003142   MOV     #RDHEAD,TEMP8 ;BUILD READ HEADER COMMAND
41 026564  053737 003036 003142   BIS     RLD$RV,TEMP8
42 026572  042737 002000 003142   BIC     #BIT10,TEMP8
43 026600  004737 017514          9$: JSR     PC,XSEEKT  ;DO SEEK BUILD BUT DO NOT START
44 026604  030274          65$

```

45	026606	013762	003044	000004	MOV	L.DA,RLDA(R2)	;LOAD RL REGISTERS
46	026614	013762	003040	000000	MOV	L.CS,RLCS(R2)	
47	026622	010046			MOV	RO, (SP)	;STORE RO
48	026624				WAITUS	#10.	;WAIT FOR INTERRUPT
49	026636	005737	003012		TST	DONE	;TEST IF INTERRUPT
50	026642	001011			BNE	17#	;YES SKIP
51	026644	004737	016422		JSR	PC,WAITIN	;WAIT FOR INTERRUPT
52	026650	012603			MOV	(SP),R3	;GET MESSAGE POINTER
53	026652				ERRHRD	1201,..,ERR1	
	026652	104456			TRAP	C1ERRHRD	
	026654	002261			.WORD	1201	
	026656	000000			.WORD	0	
	026660	012266			.WORD	ERR1	
54	026662	000137	030274		JMP	65#	
55	026666	005737	063050	17#:	TST	T.CS	;CHECK IF ANY ERRORS
56	026672	100006			BPL	14#	;NO - SKIP
57	026674				ERRHRD	1202,..,ERR6	
	026674	104456			TRAP	C1ERRHRD	
	026676	002262			.WORD	1202	
	026700	000000			.WORD	0	
	026702	012570			.WORD	ERR6	
58	026704	000137	030274		JMP	65#	
59	026710	005037	003012	14#:	CLR	DONE	;CLEAR INTERRUPT FLAG
60	026714				STCLK		;START P-CLOCK TO INITIATE MEASUREMENT
61							; /OF TIME INTERVAL
62	026732	013762	003142	000000	MOV	TEMP8,RLCS(R2)	;LOAD RL11 CONTROL AND STATUS REGISTER
63							; /TO INITIATE SEEK OPERATION
64	026740				WAITUS	#2000.	;WAIT FOR INTERRUPT
65	026752				GETTIM	R5	;GET ELAPSED TIME
66	026762	012600			MOV	(SP),RO	;RESTORE RO
67	026764	013737	003142	003040	MOV	TEMP8,L.CS	;SET IF ERROR TO REPORT
68	026772	004737	023032		JSR	PC,VERPOS	;VERIFY POSITION
69	026776	030274			65#		
70	027000	005737	003114		TST	DESSGN	;CHECK WHICH SEEK DIRECTION
71	027004	001403			BEQ	15#	;REVERSE - SKIP
72	027006	060510			ADD	R5,(R0)	;ADD TO FORWARD TOTAL
73	027010	005511			ADC	(R1)	;ADD IN OVERFLOW
74	027012	000402			BR	16#	;SKIP
75	027014	060513		15#:	ADD	R5,(R3)	;ADD TO REVERSE TOTAL
76	027016	005514			ADC	(R4)	;ADD IN OVERFLOW
77	027020	005337	003240	16#:	DEC	COUNT	;DEC SEEK COUNT
78	027024	001403			BEQ	18#	;SKIP IF 0
79	027026	004737	021202		JSR	PC,ONSWAP	;ELSE SWAP OLD AND NEW CYL
80	027032	000662			BR	9#	;REDO SEEK LOOP
81	027034	162710	000470	18#:	SUB	#312.,(R0)	;SUB CONSTANT FOR READ HEADER TIME
82	027040	162713	000470		SUB	#312.,(R3)	
83	027044	012705	000006		MOV	#6,R5	;SET SHIFT COUNT TO DIVIDE BY 64
84	027050	000241		10#:	CLC		;DIVIDE BOTH TOTALS BY 64
85	027052	006011			ROR	(R1)	
86	027054	006010			ROR	(R0)	
87	027056	000241			CLC		
88	027060	006014			ROR	(R4)	
89	027062	006013			ROR	(R3)	
90	027064	005305			DEC	R5	
91	027066	001370			BNE	10#	
92	027070	005237	003236		INC	PASCNT	;BUMP PASS COUNT
93	027074	022737	000001	003236	CMP	#1,PASCNT	;TEST IF PASS 1

94	027102	001051			BNE	24:			;NO SKIP
95	027104	012737	000177	003106	MOV		#127.,NEWCYL		;ELSE SET TO POSITION MDS TO 127
96	027112	022737	000001	002302	CMP		#1,T.DRIVE		;DRIVE = RLO1?
97	027120	001403			BEQ		101:		;YUP
98	027122	012737	000377	003106	MOV		#255.,NEWCYL		;NO - SET FOR A MID POS SEEK RLO2
99	027130	004737	017524		JSR	101:	PC,XSEEK		;DO SEEK
100	027134	030274			65:				
101	027136	012701	005670		MOV		#3000.,R1		;SET WAIT COUNT FOR 300 MS
102	027142	004737	022420		JSR		PC,RDYWAIT		;WAIT FOR READY
103	027146	030274			65:				
104	027150	004737	023032		JSR		PC,VERPOS		;VERIFY POSITION
105	027154	030274			65:				
106	027156	012700	003150		MOV		#0FMID,RO		;SET PTRS FOR TIMING 1 CYL SK AT 127
107	027162	012701	003152		MOV		#0FMIDU,R1		
108	027166	012703	003164		MOV		#0RMID,R3		
109	027172	012704	003166		MOV		#0RMIDU,R4		
110	027176	012737	000200	003106	MOV		#128.,NEWCYL		;SET NEWCYL TO 128
111	027204	022737	000001	002302	CMP		#1,T.DRIVE		;RLO1?
112	027212	001403			BEQ		102:		;YUP
113	027214	012737	000400	003106	MOV		#256.,NEWCYL		;SET FOR RLO2
114	027222	000137	026550		JMP	102:	8:		;DO SEEK LOOP
115	027226	022737	000002	003236	CMP	24:	#2,PASCNT		;TEST IF PASS 2
116	027234	001033			BNE		28:		;NO - SKIP
117	027236	013737	002312	003106	MOV		NXTHL,NEWCYL		;SET UP TO TIME 1 CYL SEEK AT INNER
118	027244	004737	017524		JSR		PC,XSEEK		; LIMIT
119	027250	030274			65:				
120	027252	012701	005670		MOV		#3000.,R1		;SET WAIT COUNT FOR 300 MS
121	027256	004737	022420		JSR		PC,RDYWAIT		;WAIT FOR READY
122	027262	030274			65:				
123	027264	004737	023032		JSR		PC,VERPOS		;VERIFY POSITION
124	027270	030274			65:				
125	027272	012700	003144		MOV		#0FIN,RO		;SET POINTERS
126	027276	012701	003146		MOV		#0FINU,R1		
127	027302	012703	003160		MOV		#0RIN,R3		
128	027306	012704	003162		MOV		#0RINU,R4		
129	027312	013737	002306	003106	MOV		HLMTW,NEWCYL		;LOAD NEW CYLINDER
130	027320	000137	026550		JMP		8:		;DO SEEK LOOP
131	027324	022737	000003	003236	CMP	28:	#3,PASCNT		;TEST IF PASS 3
132	027332	001040			BNE		32:		;NO - SKIP
133	027334	005037	003106		CLR		NEWCYL		;ELSE SET UP TO TIME 85/170 CYL SEEK
134	027340	004737	017524		JSR		PC,XSEEK		; AT OUTER LIMIT
135	027344	030274			65:				
136	027346	012701	005670		MOV		#3000.,R1		;SET WAIT COUNT FOR 300 MS
137	027352	004737	022420		JSR		PC,RDYWAIT		;WAIT FOR DRIVE READY
138	027356	030274			65:				
139	027360	004737	023032		JSR		PC,VERPOS		;VERIFY POSITION
140	027364	030274			65:				
141	027366	012700	003200		MOV		#0FOUT,RO		;SET POINTERS
142	027372	012701	003202		MOV		#0FOUTU,R1		
143	027376	012703	003210		MOV		#0ROUT,R3		
144	027402	012704	003202		MOV		#0ROUTU,R4		
145	027406	012737	000125	003106	MOV		#85.,NEWCYL		;LOAD NEWCYL FOR 85 CYL SEEK
146	027414	022737	000001	002302	CMP		#1,T.DRIVE		;RLO1?
147	027422	001505			BEQ		39:		;YUP
148	027424	012737	000252	003106	MOV		#170.,NEWCYL		;NO - SET FOR RLO2
149	027432	000501			BR		39:		
150	027434	022737	000004	003236	CMP	32:	#4,PASCNT		;TEST IF PASS 4

```

151 027442 001041          BNE      36$          ;NO - SKIP
152 027444 012737 000252 003106  MOV      @170.,NEWCYL ;ELSE SET UP TO TIME 85 CYL SEEK
153 027452 022737 000001 002302  CMP      @1.T.DRIVE   ;RLO1?
154 027460 001403          BEQ      321$        ;YES
155 027462 012737 000525 003106  MOV      @341.,NEWCYL ;NO - SET FOR RLO2
156 027470 004737 017524          JSR      PC,XSEEK    ; AT INNER LIMIT
157 027474 030274          65$
158 027476 012701 005670          MOV      @3000.,R1   ;SET WAIT COUNT FOR 300 MS
159 027502 004737 022420          JSR      PC,RDYWAIT  ;WAIT FOR READY
160 027506 030274          65$
161 027510 004737 023032          JSR      PC,VERPOS   ;VERIFY POSITION
162 027514 030274          65$
163 027516 012700 003174          MOV      @#FIN,R0    ;SET POINTERS
164 027522 012701 003176          MOV      @#FINU,R1
165 027526 012703 003204          MOV      @#RIN,R3
166 027532 012704 003206          MOV      @#RINU,R4
167 027536 013737 002306 003106  MOV      HLMTW,NEWCYL ;SET NEWCYL TO 255/511 FOR 85/170 CYL SEEK
168 027544 000434          BR       39$        ;DO TIMING LOOP
169 027546 022737 000005 003236 36$:  CMP      @5,PASCNT   ;TEST IF PASS 5
170 027554 001032          BNE      40$        ;NO - SKIP
171 027556 005037 003106          CLR      NEWCYL     ;ELSE SET UP TO TIME 256/512 CYL SEEK
172 027562 004737 017524          JSR      PC,XSEEK    ; OVER ALL SURFACE
173 027566 030274          65$
174 027570 012701 005670          MOV      @3000.,R1   ;SET WAIT COUNT FOR 300 MS
175 027574 004737 022420          JSR      PC,RDYWAIT  ;WAIT FOR DRIVE READY
176 027600 030274          65$
177 027602 004737 023032          JSR      PC,VERPOS   ;VERIFY POSITION
178 027606 030274          65$
179 027610 012700 003214          MOV      @#AFMID,R0  ;SET POINTERS
180 027614 012701 003215          MOV      @#AFMIDU,R1
181 027620 012703 003220          MOV      @#ARMID,R3
182 027624 012704 003222          MOV      @#ARMIDU,R4
183 027630 013737 002306 003106  MOV      HLMTW,NEWCYL ;SET NEWCYL
184 027636 000137 026550          JMP      @#
185 027642          39$: PRINTF   @#FMT1.1,@SKTMES,@VALDES
          40$: MOV      @VALDES,-(SP)
          MOV      @SKTMES,-(SP)
          MOV      @FMT1.1,-(SP)
          MOV      @3,-(SP)
          MOV      SP,R0
          TRAP    C#PNTF
          ADD     @10,SP
186 027672 PRINTF   @#FMT5,@BASADD,RLBAS,@DRVNAM,<B,RLDRV*1>
          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#PNTF
          ADD    @14,SP
187 027734 PRINTF   @#FMT18,@LABIN,@LABMID,@LABOUT,@LABEXP
          MOV    @LABEXP,-(SP)
          MOV    @LABOUT,-(SP)
          MOV    @LABMID,-(SP)

```

	027750	012746	007153	MOV	@LABIN, -(SP)
	027754	012746	011762	MOV	@FMT18, (SP)
	027760	012746	000005	MOV	@5, (SP)
	027764	010600		MOV	SP,RO
	027766	104417		TRAP	C#PNTF
	027770	062706	000014	ADD	@14, SP
188	027774			PRINTF	@FMT19, @LABOCF, OFIN, OFMID, OFOUT, EXOCYL
	027774	013746	003224	MOV	EXOCYL, -(SP)
	030000	013746	003154	MOV	OFOUT, -(SP)
	030004	013746	003150	MOV	OFMID, -(SP)
	030010	013746	003144	MOV	OFIN, -(SP)
	030014	012746	007207	MOV	@LABOCF, -(SP)
	030020	012746	012014	MOV	@FMT19, -(SP)
	030024	012746	000006	MOV	@6, -(SP)
	030030	010600		MOV	SP,RO
	030032	104417		TRAP	C#PNTF
	030034	062706	000016	ADD	@16, SP
189	030040			PRINTF	@FMT19, @LABOCR, ORIN, ORMID, OROUT, EXOCYL
	030040	013746	003224	MOV	EXOCYL, -(SP)
	030044	013746	003170	MOV	OROUT, -(SP)
	030050	013746	003164	MOV	ORMID, -(SP)
	030054	013746	003160	MOV	ORIN, -(SP)
	030060	012746	007221	MOV	@LABOCR, -(SP)
	030064	012746	012014	MOV	@FMT19, -(SP)
	030070	012746	000006	MOV	@6, -(SP)
	030074	010600		MOV	SP,RO
	030076	104417		TRAP	C#PNTF
	030100	062706	000016	ADD	@16, SP
190	030104			PRINTF	@FMT20, @LABHCF, HFIN, HFOUT, EXHCYL
	030104	013746	003226	MOV	EXHCYL, -(SP)
	030110	013746	003200	MOV	HFOUT, -(SP)
	030114	013746	003174	MOV	HFIN, -(SP)
	030120	012746	007233	MOV	@LABHCF, -(SP)
	030124	012746	012051	MOV	@FMT20, -(SP)
	030130	012746	000005	MOV	@5, -(SP)
	030134	010600		MOV	SP,RO
	030136	104417		TRAP	C#PNTF
	030140	062706	000014	ADD	@14, SP
191	030144			PRINTF	@FMT20, @LABHCR, HRIN, HROUT, EXHCYL
	030144	013746	003226	MOV	EXHCYL, -(SP)
	030150	013746	003210	MOV	HROUT, -(SP)
	030154	013746	003204	MOV	HRIN, -(SP)
	030160	012746	007247	MOV	@LABHCR, -(SP)
	030164	012746	012051	MOV	@FMT20, -(SP)
	030170	012746	000005	MOV	@5, -(SP)
	030174	010600		MOV	SP,RO
	030176	104417		TRAP	C#PNTF
	030200	062706	000014	ADD	@14, SP
192	030204			PRINTF	@FMT21, @LABACF, AFMID, EXACYL
	030204	013746	003230	MOV	EXACYL, -(SP)
	030210	013746	003214	MOV	AFMID, -(SP)
	030214	012746	007263	MOV	@LABACF, -(SP)
	030220	012746	012101	MOV	@FMT21, -(SP)
	030224	012746	000004	MOV	@4, -(SP)
	030230	010600		MOV	SP,RO
	030232	104417		TRAP	C#PNTF
	030234	062706	000012	ADD	@12, SP

193	030240			PRINTF	#FMT21,#LABACR,ARMID,EXAC rL
	030240	013746	003230	MOV	EXACYL,(SP)
	030244	013746	003220	MOV	ARMID,-(SP)
	030250	012746	007277	MOV	#LABACR,-(SP)
	030254	012746	012101	MOV	#FMT21,(SP)
	030260	012746	000004	MOV	#4,-(SP)
	030264	010600		MOV	SP,RO
	030266	104417		TRAP	C#PNTF
	030270	062706	000012	ADD	#12,SP
194	030274				
195	030274			65#:	
	030274			ENDTST	
	030274	104401		L10023:	
				TRAP	C#ETST

```

1          .SBTTL *TEST 2          **BASIC READ DATA (BAD SECTOR FILE)
2 030276   SGTST          ;TEST 2
3 030276   012737 006676 003016   MOV    #P2T13E,ERHEAD ;SET ERROR HEADER
4 030304   004737 01E560           JSR    PC,TSTINT    ;INITIALIZE TEST
5 030310   004737 016576           JSR    PC,GSTATR    ;CLEAR DRIVE
6 030314   030764           65#
7 030316   012737 000001 003116   MOV    #1,DESHD     ;SET TO HEAD 1
8 030324   032737 010000 014120   BIT    #HEADLM,MISWIW ;TEST IF HEAD SPEC'D
9 030332   001405           BEQ    2#           ;NO - SKIP
10 030334  005737 014126           TST   HEADW        ;TEST IF HEAD 0
11 030340  001002           BNE   2#           ;NO - SKIP
12 030342           EXIT    TST          ;ELSE EXIT TEST
13 030342   104432           TRAP  C#EXIT
14 030344   000446           .WORD L10024-
15 030346  013737 002306 003106 2# :  MOV    HLMTW,NEWCYL ;POSITION HEADS AT 255
16 030354  004737 017524           JSR    PC,XSEEK    ;DO SEEK
17 030360  030764           65#
18 030362  012701 005670           MOV    #3000.,R1   ;SET WAIT COUNT FOR 300 MS
19 030366  004737 022420           JSR    PC,RDYWAIT ;WAIT FOR INTERRUPT
20 030372  030764           65#
21 030374  004737 023032           JSR    PC,VERPOS   ;VERIFY POSITION
22 030400  030764           65#
23 030402  005037 003120           CLR   DESSEC       ;SET FOR SECTOR 0
24 030406  012737 003676 003134   MOV    #FBSFIL,TEMP5 ;SET TEMP STORAGE FOR FACTORY BS FILE
25 030414  012737 000020 003136   MOV    #16.,TEMP6  ;SET MAX SECTOR COUNT
26 030422  112737 000001 003451   MOV    #1,NOERCT   ;SET FOR NO ERROR COUNTING
27 030430  105037 003450           CLRB  LOCERR       ;CLEAR LOCAL ERROR COUNTER
28 030434  005037 003130           CLR   TEMP3        ;CLEAR ONES DETECTED FLAG
29 030440  013701 003134           MOV    TEMP5,R1    ;INIT POINTERS
30 030444  013700 003136           MOV    TEMP6,R0
31 030450  012703 004072           MOV    #IBUFF,R3
32 030454  012737 000002 003022   MOV    #2,ERRSWI   ;INIT ERROR SWITCH
33 030462  004737 024212           JSR    PC,XREAD    ;DO READ
34 030466  030640           39#
35 030470  005723           TST   (R3)+        ;TEST IF WORD 0 NOT NEG
36 030472  100516           BMI   45#          ;YES, BAD FMT ERROR
37 030474  005723           TST   (R3)+        ;ELSE TEST WORD 1 NOT NEG
38 030476  100514           BMI   45#          ;YES - BAD FMT ERROR REPORT
39 030500  005723           TST   (R3)+        ;TEST WORD 2 IS 0
40 030502  001112           BNE   45#          ;NO - SKIP TO FMT ERROR RPT
41 030504  005723           TST   (R3)+        ;TEST WORD 3 IS 0
42 030506  001110           BNE   45#          ;NO - SKIP TO FMT ERROR RPT
43 030510  021327 177777           CMP   (R3),#-1    ;TEST IF NEXT WORD IS ALL 1'S
44 030514  001004           BNE   10#          ;NO - SKIP
45 030516  012737 000001 003130   MOV    #1,TEMP3    ;ELSE SET 1'S DETECTED FLAG
46 030524  000403           BR   11#           ;SKIP
47 030526  005737 003130           TST   TEMP3        ;TEST IF ONES HAVE BEEN DETECTED
48 030532  001076           BNE   45#          ;YES - SKIP TO FMT ERROR RPT
49 030534  012311           MOV    (R3)+,(R1)  ;STORE CYLINDER WORD
50 030536  012705 000007           MOV    #7,R5       ;ALIGN IT TO LOOK LIKE HEADER
51 030542  006311           ASL   (R1)
52 030544  005305           DEC   R5
53 030546  001375           BNE   12#
54 030550  032713 000400           BIT    #BIT8,(R3)  ;TEST IF HEAD 1
55 030554  001402           BEQ   15#          ;NO - SKIP
56 030556  052711 000100           BIS    #BIT6,(R1)  ;INSERT HEAD BIT

```



```

55 030562 042715 177400          158: BIC      #177400,(R3)      ;CLEAR ALL BUT SECTOR
56 030566 052321                    BJS      (R3)..(R1)    ;INSERT SECTOR NUMBER
57 030570 020327 004472          CMP      R3,#IBUFF-256 ;CHECK IF IBUFF EMPTY
58 030574 001345                    BNE      #1          ;NO GET NEXT CYLINDER
59 030576 005737 003130          TST     TEMP3        ;ELSE TEST IF 1'S DETECTED
60 030602 001461                    BEQ      48#         ;TO MANY ERRORS REPORT
61 030604 022737 000044 003136    CMP      #36..TEMP6  ;CHECK IF SOFTWARE BAD READ
62 030612 001464                    BEQ      65#         ;YES SKIP
63 030614 012737 003502 003134 378: MOV      #SBSFIL,TEMP5 ;ELSE CHANGE POINTERS
64 030622 012737 000044 003136    MOV      #36..TEMP6  ; MAX SECTOR NUMBER
65 030630 012737 000024 003120    MOV      #20..DESSEC ; SECTOR NUMBER START
66 030636 000676                    BR       4#         ;DO READ
67 030640 005237 003450          398: INC      LOCERR      ;BUMP LOCAL ERROR COUNTER
68 030644 012777 177777 152262 408: MOV      #-1,#TEMP5  ;MOV 1'S INTO FILE STORAGE
69 030652 104420                    INLOOP  TRAP      C#INLP ;CHECK IF IN ERROR LOOP
70 030654 103667                    BCOMPLETE: 4#      ;YES GO DO READ
71 030656 023737 003120 003136 418: BCS      4#         ;CHECK IF ALL SECTORS READ
72 030664 001015                    CMP      DESSEC,TEMP6 ;NO - SKIP
73 030666 012703 006033          BNE      43#         ;SET RESULT MESSAGE POINTER
74 030672 005237 003450          MOV      #MBADSF,R3  ;BUMP LOCAL ERROR COUNTER
75 030676 104456                    INC      LOCERR
76 030700 002425                    ERRHRD  TRAP      C#ERRHD
77 030702 000000                    .WORD   1301
78 030704 012266                    .WORD   0
79 030706 022737 003502 003134    CMP      #SBSFIL,TEMP5 ;TEST IF SOFTWARE FILES CHECKED
80 030714 001423                    BEQ      65#         ;YES - EXIT
81 030716 000736                    BR       37#         ;ELSE GO CHECK SOFTWARE FILES
82 030720 062737 000004 003120 438: ADD      #4,DESSEC   ;BUMP TO NEXT SECTOR
83 030726 000642                    BR       4#         ;GO DO READ
84 030730 012703 006063          458: MOV      #MFINTER,R3 ;SET RESULT MESSAGE POINTER
85 030734 104456                    ERRHRD  TRAP      C#ERRHD
86 030736 002426                    .WORD   1302
87 030740 000000                    .WORD   0
88 030742 012266                    .WORD   ERR1
89 030744 000735                    BR       39#         ;GO CHECK FOR LOOP
90 030746 012703 006110          488: MOV      #MTMBS,R3  ;SET RESULT MESSAGE PTR
91 030752 104456                    ERRHRD  TRAP      C#ERRHD
92 030754 002427                    .WORD   1303
93 030756 000000                    .WORD   0
94 030760 012266                    .WORD   ERR1
95 030762 000730                    BR       40#         ;GO CHECK FOR LOOP
96 030764 012737 000002 003022 658: MOV      #2,ERRSWI  ;INIT ERROR SWITCH
97 030772 012737 000001 003500    MOV      #1,BSFVAL   ;SET BAD SECTOR FILES VALID FLAG
98 031000 105737 003450          TST     LOCERR      ;TEST IF LOCAL ERRORS
99 031004 001402                    BEQ      66#         ;NO - SKIP
100 031006 005237 003244          INC     ERRCNT      ;ELSE BUMP ERROR COUNT
101 031012 104401                    668:  TRAP      C#ETST
102 031012 104401                    ENDTST  L10024:
103 031012 104401                    TRAP    C#ETST
  
```

1				.SBTTL	*TEST 3	**WRITE/READ DATA (PART 1)	
2	031014			BGNTST		TEST 3	
3	031014	012737	006712	003016	MOV	#P2T14E,ERHEAD	T3:: ;SET ERROR HEADER
4	031022	004737	021226		JSR	PC,CKBSVD	;GO CHECK IF BAD SECTOR FILES VALID
5	031026	004737	016560		JSR	PC,TSTINT	;INITIALIZE TEST
6	031032	004737	016576		JSR	PC,GSTATR	;CLEAR DRIVE
7	031036	031226			T3065:		
8	031040	004737	021116		JSR	PC,CHOSHD	;GO CHOSE HEAD
9	031044	005037	003120		CLR	DESSEC	; SECTOR 0
10	031050	005037	003106		CLR	NEWCYL	; CYLINDER 0
11	031054	005037	031120		CLR	T310:	;CLEAR PATTERN SELECT
12	031060	004737	017524	T306:	JSR	PC,XSEEK	;POSITION HEADS
13	031064	031226			T3065:		
14	031066	012701	005670		MOV	#3000.,R1	;SET WAIT COUNT FOR 300 MS
15	031072	004737	022420		JSR	PC,RDYWAIT	;WAIT FOR READY
16	031076	031226			T3065:		
17	031100	004737	023032		JSR	PC,VERPOS	;VERIFY POSITION
18	031104	031226			T3065:		
19	031106	005037	031120		CLR	T310:	;CLEAR PATTERN SELECTOR
20	031112			T307:			
21	031112			BGNSUB			
	031112	104402			TRAP	C#BSUB	T3.1:
22	031114	004537	023522		JSR	R5,DATGEN	;GENERATE DATA
23	031120	000000		T310:	.WORD	0	;PATTERN SELECT WORD
24	031122	004737	024152		JSR	PC,XWRITE	;DO WRITE DATA
25	031126	031144			60:		
26	031130	004737	024212		JSR	PC,XREAD	;DO READ DATA
27	031134	031144			60:		
28	031136	004737	023662		JSR	PC,DATCOM	;COMPARE DATA
29	031142	031144			60:		
30	031144	012737	000002	003022	MOV	#2,ERRSWI	;INIT ERROR SWITCH
31	031152			60:			
	031152			ENDSUB			
	031152	104403		L10026:			
32	031154				TRAP	C#ESUB	
	031154	104410			ESCAPE	TST	;EXIT TEST IF ERROR
	031156	000050			TRAP	C#ESCAPE	
	031156	000050			.WORD	L10025-	
33	031160	022737	000010	031120	CMP	#8.,T310:	;WAS DATA PAT 8 USED?
34	031166	001403			BEQ	10:	;YES - SKIP
35	031170	005237	031120		INC	T310:	;ELSE BUMP TO NEXT PATTERN
36	031174	000746			BR	T307:	;DO TEST WITH NEW PATTERN
37	031176	004737	021142	10:	JSR	PC,SWAPHD	;GO SWAP TO HEAD 1 OR END TEST
38	031202	031226			T3065:		;ABORT RETURN
39	031204	005037	031120		CLR	T310:	;SET PATTERN SELECT TO 0
40	031210	004737	024710	11:	JSR	PC,BSCHK	;CHECK IF SECTOR BAD
41	031214	031220			13:		;YES RETURN - SKIP TO 13:
42	031216	000720			BR	T306:	;NO RETURN - DO TEST THIS SECTOR
43	031220	005237	003106	13:	INC	NEWCYL	;BUMP TO NEXT CYLINDER
44	031224	000771			BR	11:	;CHECK IF THIS ONE BAD
45	031226			T3065:			
46	031226			ENDTST			
	031226			L10025:			
47	031226	104401			TRAP	C#ETST	

1	.SBTTL	*TEST 4	**ROTATIONAL TIMING
2	031230	BGNTST	;TEST 4
3	031230	012737 006733 003016	MOV #P215E,ERHEAD ;SET ERROR HEADER T4::
4	031236	005737 003474	;CHECK FOR PRESENCE OF A P CLOCK...;BYPASS TEST IF NOT AVAILABLE
5	031242	001026	TST CLKFLG ;P-CLOCK?
6	031244		BNE 31 ;BRANCH TO PERFORM TEST IF P-CLOCK IS PRESENT
7	031244	012746 010131	PRINTF #FMT9,#NOTST4 ;ELSE, PRINT MSG. "TEST 4 CANNOT BE PERFORMED..."
	031250	012746 011554	MOV #NOTST4,-(SP)
	031254	012746 000002	MOV #FMT9,-(SP)
	031260	010600	MOV #2,-(SP)
	031262	104417	MOV SP,R0
	031264	062706 000006	TRAP C;PNTF
			ADD #6,SP
8			;P CLOCK IS NOT AVAILABLE"
9	031270		PRINTF #FMT9,#NTST4A
	031274	012746 010217	MOV #NTST4A,-(SP)
	031300	012746 000002	MOV #FMT9,-(SP)
	031304	010600	MOV #2,-(SP)
	031306	104417	MOV SP,R0
	031310	062706 000006	TRAP C;PNTF
10	031314		ADD #6,SP
	031314	104432	EXIT TST
	031316	000542	TRAP C;EXIT
11	031320	005003	.WORD L10027-.
12	031322	005004	34: CLR R3 ;CLEAR FOR TIMING STORAGE
13	031324	004737 016560	CLR R4
14	031330	004737 016576	JSR PC,TSTINT ;INITIALIZE TEST
15	031334	032052	JSR PC,GSTATR ;CLEAR DRIVE
16	031336	004537 023522	60# JSR R5,DATGEN ;GENERATE DATA
17	031342	000000	O ;PATTERN 0
18	031344	005037 003120	CLR DESSEC ;CLEAR TO SECTOR 0
19	031350	004737 021116	JSR PC,CHOSH0 ;GO SELECT HEAD
20	031354	013737 014122 003106	MOV LOLIMW,NEWCYL ;SET FOR CYLINDER
21	031362	004737 017524	JSR PC,XSEEK ;DO SEEK
22	031366	032052	60#
23	031370	012701 005670	MOV #3000.,R1 ;SET WAIT FOR 300 MS
24	031374	004737 022420	JSR PC,RDYWAIT ;WAIT FOR READY
25	031400	032052	60#
26	031402	004737 023032	JSR PC,VERPOS ;VERIFY POSITION
27	031406	032052	60#
28	031410	012701 000100	MOV #64.,R1 ;SET LOOP COUNTER
29	031414	012705 003046	54: MOV #L.MP,R5 ;SET A POINTER
30	031420	004737 024142	JSR PC,XWRITT ;DO FIRST WRITE
31	031424	032052	60#
32	031426	011562 000006	MOV (R5),RLMP(R2) ;LOAD RL REGISTERS
33	031432	014562 000004	MOV -(R5),RLDA(R2)
34	031436	014562 000002	MOV -(R5),RLBA(R2)
35	031442	014562 000000	MOV -(R5),RLCS(R2)
36	031446		WAITUS #3000.
37	031460	005737 003012	TST DONE ;TEST IF INTERRUPT
38	031464	001011	BNE 61 ;YES - SKIP
39	031466	004737 016422	JSR PC,WAITIN ;ELSE WAIT FOR TIMEOUT
40	031472	012603	MOV (SP)+,R3 ;GET MESSAGE POINTER
41	031474		ERRHRD 1501.,ERR1
	031474	104456	TRAP C;ERRRD

```

031476 002735          .WORD 1501
031500 000000          .WORD 0
031502 012266          .WORD ERR1
42 031504 000137 032052 JMP 60$
43 031510 005737 003050 6$: TST T.CS ;TEST IF ANY ERRORS
44 031514 100006          BPL 4$ ;NO SKIP
45 031516          ERRHRD 1502...ERR6
031516 104456          TRAP C$ERRHRD
031520 002736          .WORD 1502
031522 000000          .WORD 0
031524 012570          .WORD ERR6
46 031526 000137 032052 JMP 60$
47 031532 012705 003046 4$: MOV @L.MP,R5 ;SET POINTER TO RL LOAD REGS
48 031536 005037 003012 CLR DONE ;CLEAR INTERRUPT INDICATOR
49 031542          STCLK ;START P CLOCK TO INITIATE MEASUREMENT
50          ;/OF TIME INTERVAL
51 031560 011562 000006 MOV (R5),RLMP(R2) ;LOAD RL REGISTERS FOR 2ND WRITE
52 031564 014562 000004 MOV -(R5),RLDA(R2)
53 031570 014562 000002 MOV -(R5),RLBA(R2)
54 031574 014562 000000 MOV -(R5),RLCS(R2)
55 031600          WAITUS @3000. ;WAIT FOR INTERRUPT
56 031612          GETTIM R0 ;GET ELAPSED TIME
57 031622 005737 003012 TST DONE ;TEST IF INTERRUPT OCCURRED
58 031626 001010          BNE 7$ ;YES - SKIP
59 031630 004737 016422 JSR PC,WAITIN ;GO WAIT FOR INTERRUPT
60 031634 012603          MOV (SP),R3 ;GET MESSAGE POINTER
61 031636          ERRHRD 1503...ERR1 ;REPORT
031636 104456          TRAP C$ERRHRD
031640 002737          .WORD 1503
031642 000000          .WORD 0
031644 012266          .WORD ERR1
62 031646 000501          BR 60$
63 031650 005737 003050 7$: TST T.CS ;TEST IF ANY ERROR
64 031654 100005          BPL 8$ ;NO SKIP
65 031656          ERRHRD 1504...ERR6 ;REPORT ERRORS
031656 104456          TRAP C$ERRHRD
031660 002740          .WORD 1504
031662 000000          .WORD 0
031664 012570          .WORD ERR6
66 031666 000471          BR 60$
67 031670 060003 8$: ADD R0,R3 ;ADD IN TIME USED
68 031672 005504          ADC R4 ;DOUBLE PRECISION
69 031674 005301          DEC R1 ;DEC LOOP COUNTER
70 031676 001246          BNE 5$ ;LOOP UNTIL 0
71 031700 012701 000006 MOV @6,R1 ;SET DIVIDE COUNT
72 031704 000241 10$: CLC ;CLEAR CARRY FOR DIVIDE
73 031706 006004          ROR R4 ;DIVIDE SUM BY 100(8)
74 031710 006003          ROR R3
75 031712 005301          DEC R1 ;DEC DIVIDE COUNT
76 031714 001373          BNE 10$ ;LOOP UNTIL DONE
77 031716          PRINTF @FMT1.1,@SRTMES,@VALDES
031716 012746 007117 MOV @VALDES,-(SP)
031722 012746 007075 MOV @SRTMES,-(SP)
031726 012746 011342 MOV @FMT1.1,-(SP)
031732 012746 000003 MOV @3,-(SP)
031736 010600          MOV SP,R0
031740 104417          TRAP C$PNTF
  
```

```

031742 062706 000010      ADD      #10,SP
78 031746      PRINTF  #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
031746 005046      CLR      -(SP)
031750 153716 003037      BISB    RLDRV+1,(SP)
031754 012746 006142      MOV     #DRVNAM,-(SP)
031760 013746 003032      MOV     RLBAS,-(SP)
031764 012746 006131      MOV     #BASADD,-(SP)
031770 012746 011370      MOV     #FMT5,-(SP)
031774 012746 000005      MOV     #5,-(SP)
032000 010600      MOV     SP,RO
032002 104417      TRAP   C#PNTF
032004 062706 000014      ADD     #14,SP
79 032010      PRINTF  #FMT26,#RESE3,R3,#RESE4,#MAPROX,EXROT
032010 013746 003232      MOV     EXROT,-(SP)
032014 012746 007143      MOV     #MAPROX,-(SP)
032020 012746 011127      MOV     #RESE4,-(SP)
032024 010346      MOV     R3,-(SP)
032026 012746 011123      MOV     #RESE3,-(SP)
032032 012746 012211      MOV     #FMT26,-(SP)
032036 012746 000006      MOV     #6,-(SP)
032042 010600      MOV     SP,RO
032044 104417      TRAP   C#PNTF
032046 062706 000016      ADD     #16,SP
80 032052 012737 000002 003022 60#:  MOV     #2,ERRSWI      ;INITIALIZE ERROR SWITCH
81 032060      ENDTST
032060      L10027:
82 032060 104401      TRAP   C#ETST

```

```

1          .SBTTL *TEST 5          **WRITE/READ DATA (PART 2)
2 032062   BGNTST                   ;TEST 5
3 032062   012737 006756 003016     MOV    #P2T16E,ERHEAD ;SET ERROR HEADER
4 032070   004737 021226             JSR    PC,CKBSVD     ;GO CHECK IF BAD SECTOR FILES VALID
5 032074   004737 016560             JSR    PC,TSTINT    ;INITIALIZE TEST
6 032100   004737 016576             JSR    PC,GSTATR    ;CLEAR DRIVE
7 032104   033170                    T3165#
8 032106   005037 003236             CLR    PASCNT       ;CLEAR PASS TO 0
9 032112   012705 177776             MOV    # -2,R5      ;SET
10 032116  005737 003444             TST   PASNUM        ;TEST IF FIRST PASS (QUICK VERIFY)
11 032122   001006                    BNE   1#            ;NO - SKIP
12 032124   032737 000001 014120     BIT    #ALLCYL,MISWIW ;TEST IF USE ALL CYLINDERS
13 032132   001002                    BNE   1#            ;YES - SKIP
14 032134   012705 177760             MOV    # -16.,R5    ;ELSE SET PEOPLE TO NEG 8
15 032140                    1# :
16 032140   012701 002510             MOV    #T33TBL,R1   ;GET ADDRESS OF WORK TABLE
17 032144   012737 000010 002304     MOV    #10,JJJ      ;SET CLEAR COUNT
18 032152   013721 014122             MOV    LOLIMW,(R1)  ;CLEAR LOCATIONS TO LO LIMIT
19 032156   005337 002304             DEC    JJJ           ;DEC COUNT
20 032162   001373                    BNE   2#            ;LOOP UNTIL 0
21 032164   013737 014124 002514     MOV    HILIMW,T33TBL+4 ;INSERT HILIMIT
22 032172   013737 014124 002516     MOV    HILIMW,T33TBL+6 ;INTO APPROPRIATE LOCATIONS
23 032200   013737 014124 002520     MOV    HILIMW,T33TBL+10
24 032206   062705 000002             T3100# : ADD    #2,R5 ;BUMP R5 BY 2
25 032212   032737 000001 014120     BIT    #ALLCYL,MISWIW ;TEST IF USE ALL CYLINDERS
26 032220   001031                    BNE   5#            ;YES - SKIP
27 032222   005737 003444             TST   PASNUM        ;TEST IF FIRST PASS (QUICK VERIFY)
28 032226   001002                    BNE   3#            ;NO - SKIP
29 032230   062705 000016             ADD    #16,R5       ;ELSE BUMP CYLINDER POINTER BY 7
30 032234   022737 000001 002302     3# : CMP    #1,T.DRIVE ;RL01 OR RL02? THAT IS THE Q
31 032242   001404                    BEQ   44#           ;ANS IS RL01
32 032244   020527 000244             CMP    R5,#164.
33 032250   103013                    BHIS  4#
34 032252   000403                    BR    69#           ;TEST PAST TABLE-YES EXIT
35 032254   020527 000122             44# : CMP    R5,#82.
36 032260   103007                    BHIS  4#            ;TES PAST THE TABLE
37
38 032262   016537 002610 002304     69# : MOV    CYLTBL(R5),JJJ ;GET NEXT TABLE ENTRY
39 032270   043737 002310 002304     BIC    CLRBYT,JJJ   ;CLEAR UPPER BYTE
40 032276   001007                    BNE   8#
41 032300   000137 033170             4# : JMP    T3165#      ;EXIT TEST
42 032304   023705 014124             5# : CMP    HILIMW,R5   ;TEST IF ALL CYLINDERS USED
43 032310   001773                    BEQ   4#            ;YES - EXIT TEST
44 032312   010537 002304             MOV    R5,JJJ       ;USE R5 AS NEXT CYLINDER
45 032316   023737 002304 014122     8# : CMP    JJJ,LOLIMW  ;CHECK IF LOWER THAN LOLIMIT
46 032324   103730                    BLO   T3100#       ;YES - SKIP
47 032326   023737 002304 014124     CMP    JJJ,HILIMW   ;CHECK IF HIGHER THAN HILIMIT
48 032334   101324                    BHI   .3100#       ;YES - SKIP
49 032336   012703 002550             MOV    #TBT,R3
50 032342   013713 002304             MOV    JJJ,(R3)
51 032346   013763 002304 000002     MOV    JJJ,2(R3)
52 032354   013763 002304 000004     MOV    JJJ,4(R3)
53 032362   013763 002304 000006     MOV    JJJ,6(R3)
54 032370   013763 002304 000010     MOV    JJJ,10(R3)
55 032376   013763 002304 000012     MOV    JJJ,12(R3)
56 032404   010337 003030             MOV    R3,TBLSTR    ;STORE TABLE ADDRESS

```

```

57 032410 004737 021116          JSR    PC,CMOSMD      ;GO CHOSE HEAD
58
59 032414          T3101:
60 032414          BGNSUB
   032414          T5.1:
   032414 104402          TRAP    C4BSUB
61 032416 042737 003760 003010    BIC    #MQUALS,OPFLAG ;CLEAR ALL MESSAGE QUALIFIERS
62 032424 005737 003236          TST    PASCNT        ;TEST IF PASS 0
63 032430 001414          BEQ    11#           ;YES - SKIP
64 032432 023727 003236 000003    CMP    PASCNT,#3     ;TEST IF PASS 3
65 032440 001404          BEQ    10#           ;YES - SKIP
66 032442 002407          BLT    11#           ;CHECK IF LESS THAN 3, IF YES CLEAR TO 0
67 032444 012737 000003 003236    MOV    #3,PASCNT     ;ELSE SET TO 3
68 032452 052737 000020 003010 10#:  BIS    #INOUTS,OPFLAG ;SET MESSAGE QUAL
69 032460 000405          BR     12#           ;SKIP
70 032462 005037 003236          CLR    PASCNT        ;SET PASS COUNT TO 0
71 032466 052737 000040 003010    BIS    #OUTINS,OPFLAG ;SET MESSAGE QUAL
72 032474 012737 000003 003026 12#:  MOV    #3,WRTSWI     ;SET READ AND WRITE SWITCH
73 032502 013703 003030          MOV    TBLSTR,R3     ;GET STORED TABLE ADDRESS
74 032506 012701 002510          MOV    #T33TBL,R1
75 032512 012703 002550          MOV    #TBT,R3
76 032516 005037 003120          CLR    DESSEC        ;CLEAR TO SECTOR 0
77 032522 012137 003106          MOV    (R1)+,NEWCYL  ;GET NEXT TABLE ENTRY
78 032526 004737 017524          JSR    PC,XSEEK      ;DO SEEK
79 032532 033076          60#
80 032534 012701 005670          MOV    #3000.,R1     ;SET WAIT COUNT FOR 300 MS
81 032540 004737 022420          JSR    PC,RDYWAIT    ;WAIT FOR READY
82 032544 033076          60#
83 032546 012337 003106          MOV    (R3)+,NEWCYL ;GET NEXT TABLE ENTRY
84 032552 004737 017524          JSR    PC,XSEEK      ;DO SEEK
85 032556 033076          60#
86 032560 012701 005670          MOV    #3000.,R1     ;SET WAIT COUNT FOR 300 MS
87 032564 004737 022420          JSR    PC,RDYWAIT    ;WAIT FOR READY
88 032570 033076          60#
89 032572 004737 023032          JSR    PC,VERPOS     ;VERIFY POSITION
90 032576 033076          60#
91 032600 004737 024710          JSR    PC,BSCHK      ;CHECK FOR BAD SECTOR
92 032604 032736          32#
93 032606 013737 003120 032626    MOV    DESSEC,25#    ;SET DATA PATTERN - TO SECTOR NUMBER
94 032614 042737 177770 032626    BIC    #177770,25#   ;CLEAR ALL BUT LSD
95 032622 004537 023522          JSR    R5,DATGEN     ;GO GENERATE DATA
96 032626 000000          .WORD 0
97 032630 032737 000001 003026 25#:  BIT    #BIT0,WRTSWI  ;TEST IF WRITE THIS PASS
98 032636 001425          BEQ    29#           ;NO - SKIP
99 032640 004737 024152          JSR    PC,XWRITE     ;DO WRITE
100 032644 033076          60#
101 032646 005237 003120          INC    DESSEC        ;INC SECTOR
102 032652 022737 000050 003120    CMP    #40.,DESSEC   ;TEST IF ALL SECTORS USED
103 032660 001347          BNE    16#           ;NO - SKIP
104 032662 042737 000060 003010    BIC    #INOUTS!OUTINS,OPFLAG ;CLEAR QUALIFIERS
105 032670 042737 000001 003026    BIC    #BIT0,WRTSWI  ;CLEAR WRITE REQUIRED SWITCH
106 032676 052737 000100 003010    BIS    #FOLWRT,OPFLAG ;SET FOLLOWING WRITE QUALIFIER
107 032704 005037 003120          CLR    DESSEC        ;CLEAR TO SECTOR 0
108 032710 000733          BR     16#           ;SKIP
109 032712 032737 000002 003026 29#:  BIT    #BIT1,WRTSWI  ;TEST IF READ THIS PASS
110 032720 001414          BEQ    33#           ;NO - SKIP
111 032722 004737 024212          JSR    PC,XREAD      ;ELSE DO READ

```

```

112 032726 033076          60#
113 032730 004737 023662 JSR      PC,DATCOM      ;COMPARE DATA
114 032734 033076          60#
115 032736 005237 003120 32# : INC      DESSEC      ;BUMP SECTOR
116 032742 022737 000050 003120 CMP      #40.,DESSEC    ;TEST IF ALL SECTORS USED
117 032750 001313          BNE      16#           ;NO - LOOP
118 032752 005037 003120 33# : CLR      DESSEC      ;CLEAR DESIPEL SECTOR
119 032756 005037 003026 CLR      WRTSWI      ;CLEAR WRITE/READ SWITCH
120 032762 005237 003236 INC      PASCNT      ;BUMP PASS COUNT
121 032766 042737 003760 003010 BIC      #MQUALS,OPFLAG ;CLEAR ALL QUALIFIERS
122 032774 023727 003236 000003 CMP      PASCNT,#3     ;TEST IS PASS 3
123 033002 001435          BEQ      60#           ;YES - SKIP
124 033004 023727 003236 000006 CMP      PASCNT,#6     ;TEST IF PASS 6
125 033012 001431          BEQ      60#           ;YES - SKIP
126 033014 012737 000002 003026 MOV      #BIT1,WRTSWI  ;SET READ REQUIRED BIT
127 033022 023727 003236 000001 CMP      PASCNT,#1     ;TEST IF PASS 1
128 033030 001415          BEQ      40#           ;YES - SKIP
129 033032 023032 003236 000005 CMP      PASCNT,#5     ;TEST IF PASS 4
130 033040 001411          BEQ      40#           ;YES - SKIP
131 033042 000404          BR       39#           ;SKIP
132 033044 052737 002000 003010 37# : BIS      #FWDSCO,OPFLAG ;SET FWD QUALIFIER
133 033052 000407          BR       36#           ;GO DO NEXT PASS
134 033054 052737 000020 003010 39# : BIS      #INOUTS,OPFLAG ;SET QUALIFIER
135 033062 000403          BR       36#           ;SKIP
136 033064 052737 000040 003010 40# : BIS      #OUTINS,OPFLAG ;SET MESSAGE QUALIFIER
137 033072 000137 032516 36# : JMP      15#           ;GO DO NEXT PASS
138 033076 012737 000002 003022 60# : MOV      #2,ERRSWI    ;INIT ERROR SWITCH
139 033104          ENDSUB
      033104          L10031:
      033104 104403          TRAP     C#ESUB
140 033106          ESCAPE   TST           ;EXIT TEST IF ERROR
      033106 104410          TRAP     C#ESCAPE
      033110 000060          .WORD   L10030-.
141 033112 012737 000003 003026 MOV      #3,WRTSWI    ;SET FOR READ AND WRITE REQ.
142 033120 023727 003236 000003 CMP      PASCNT,#3     ;TEST IF PASS 3
143 033126 001004          BNE      45#           ;NO - SKIP
144 033130 012737 002516 003030 MOV      #T33TBL+6,TBLSTR ;STORE MID POINT IN TABLE
145 033136 000410          BR       48#           ;GO START PASS 4
146 033140 005037 003236 45# : CLR      PASCNT      ;CLEAR TO PASS 0
147 033144 004737 021142 JSR      PC,SWAPHD    ;GO SWAP TO HEAD 1 OR END TEST
148 033150 032206          T3100# ;ABORT RETURN
149 033152 012737 002510 003030 MOV      #T33TBL,TBLSTR ;STORE START OF TABLE
150 033160 062703 000006 48# : ADD      #6,R3
151 033164 000137 032414 JMP      T3101#
152 033170          T3165# :
153 033170          ENDTST
      033170          L10030:
      033170 104401          TRAP     C#ETST

```



```

*TFST 6
1
2 033172          .SBTTL *TEST 6          **WRITE LOCK ERROR AND DATA PROTECTION
   033172          BGNST          ;TEST 6
3 033172 005737 003444          TST PASNUM          ;TEST IF FIRST PASS          T6::
4 033176 001003          BNE 2$          ;NO - SKIP
5 033200 005737 014120          TST MISWIW          ;TEST IF RUN MANUAL INTERVENTION
6 033204 100402          BMI 3$          ;YES - SKIP
7 033206 000137 034206          JMP T3265$          ;EXIT TST
8 033212          2$:
9 033212          3$:          BGNSUB
   033212          TRAP C#BSUB          T6.1:
   033212 104402          MOV #P2T17E,ERHEAD ;SET ERROR HEADER
10 033214 012737 006777 003016 JSR PC,TSTINT      ;INITIALIZE TEST
11 033222 004737 016560          JSR PC,GSTATR      ;CLEAR DRIVE
12 033226 004737 016576          60$
13 033232 034054          CLR DESHD          ;SET TO HEAD 0
14 033234 005037 003116          CLR DESSEC         ;SET TO SECTOR 0
15 033240 005037 003120          CLR NEWCYL        ;CLEAR TO CYLINDER 0
16 033244 005037 003106          JSR PC,XSEEK      ;DO SEEK
17 033250 004737 017524          60$
18 033254 034054          MOV #6000.,R1     ;INITIALIZE WAIT COUNT
19 033256 012701 013560          JSR PC,RDYWAIT    ;WAIT FOR READY
20 033262 004737 022420          60$
21 033266 034054          JSR PC,VERPOS     ;VERIFY POSITION
22 033270 004737 023032          60$
23 033274 034054          BIT #WLSTAT,T.MP ;TEST IF WRITE LOCK SET
24 033276 032737 020000 003056 BNE 7$          ;YES - SKIP
25 033304 001116          JSR RS,DATGEN     ;GENERATE DATA
26 033306 004537 023522          7          ;PATTERN 7
27 033312 000007          JSR PC,XWRITE     ;WRITE DATA
28 033314 004737 024152          60$
29 033320 034054          JSR PC,XREAD     ;READ DATA
30 033322 004737 024212          60$
31 033326 034054          JSR PC,DATCOM    ;CHECK DATA
32 033330 004737 023662          60$
33 033334 034054          PRINTF #FMTOP1,#OPR004,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1> ;REQUEST SET WRT LC
34 033336          CLR -(SP)
   033336 005046          BISB RLDRV+1,(SP)
   033340 153716 003037          MOV #DRVNAM,-(SP)
   033344 012746 006142          MOV RLBAS,-(SP)
   033350 013746 003032          MOV #BASADD,-(SP)
   033354 012746 006131          MOV #OPR1A,-(SP)
   033360 012746 007366          MOV #OPR004,-(SP)
   033364 012746 007415          MOV #FMTOP1,-(SP)
   033370 012746 011243          MOV #7,-(SP)
   033374 012746 000007          MOV SP,R0
   033400 010600          TRAP C#PNTF
   033402 104417          ADD #20,SP
   033404 062706 000020          MOV #20.,R1     ;INITIALIZE WAIT COUNT
35 033410 012701 000024          5$:          WAITMS          ;CALL WAIT
36 033414          JSR PC,GSTATR      ;GET STATUS
37 033426 004737 016576          60$
38 033432 034054          BIT #WLSTAT,T.MP ;CHECK IF WRITE LOCK SET
39 033434 032737 020000 003056 BNE 7$          ;YES - SKIP
40 033442 001037          PRINTF #FMT2,#BELL ;RING BELL
41 033444          MOV #BELL,-(SP)
   033444 012746 011117

```

K

```

033450 012746 011351      MOV    #FMT2,-(SP)
033454 012746 000002      MOV    #2,-(SP)
033460 010600      MOV    SP,R0
033462 104417      TRAP   C#PNTF
033464 062706 000006      ADD    #6,SP
42 033470 005301      DEC    R1          ;DEC COUNT
43 033472 001350      BNE    S#         ;SKIP IF NOT 0
44 033474      PRINTF #FMT23,#P2T17E,#BYPSNM,#OPR1A,<B,RLDPV+1> ;RPT BYPASSED
033474 005046      CLR    -(SP)
033476 153716 003037      BLSB  RLDRV+1,(SP)
033502 012746 007366      MOV    #OPR1A,-(SP)
033506 012746 007471      MOV    #BYPSNM,-(SP)
033512 012746 006777      MOV    #P2T17E,-(SP)
033516 012746 012160      MOV    #FMT23,-(SP)
033522 012746 000005      MOV    #5,-(SP)
033526 010600      MOV    SP,R0
033530 104417      TRAP   C#PNTF
033532 062706 000014      ADD    #14,SP
45 033536      EXIT   TST
033536 104432      TRAP   C#EXIT
033540 000446      .WORD L10032-
46 033542 004537 023522      JSR    R5,DATGEN ;GENERATE DATA
47 033546 000001      1 ;PATTERN 1
48 033550 012705 003040      MOV    #L.CS,R5 ;GET ADDRESS OF L REGS
49 033554 012715 000112      MOV    #WTDATA,(R5) ;LOAD WRITE COMMAND
50 033560 053715 003036      BIS   RLDRV,(R5) ;INSERT DRIVE NUMBER
51 033564 042725 002000      BIC   #BIT10,(R5)+ ;CLEAR FOR DRIVE 4 - 7 SPEC'D
52 033570 012725 004472      MOV    #OBUFF,(R5)+ ;LOAD BUS ADDRESS
53 033574 005025      CLR    (R5)+ ;CYL 0, HD 0, SECTOR 0
54 033576 012725 177600      MOV    #177600,(R5)+ ;128 WORDS
55 033602 012701 000454      MOV    #300.,R1 ;SET WAIT COUNT FOR 30 MS
56 033606 005037 003012      CLR    DONE ;CLEAR INTERRUPT FLAG
57 033612 014562 000006      MOV    -(R5),RLMP(R2) ;LOAD RL REGS
58 033616 014562 000004      MOV    -(R5),RLDA(R2)
59 033622 014562 000002      MOV    -(R5),RLBA(R2)
60 033626 014562 000000      MOV    -(R5),RLCS(R2)
61 033632      10#: WAITUS #1
62 033644 005737 003012      TST   DONE ;CHECK IF INTERRUPT
63 033650 001013      BNE   14# ;YES - SKIP
64 033652 005301      DEC   R1 ;DEC WAIT COUNT
65 033654 001366      BNE   10# ;LOOP IF NOT 0
66 033656 004737 016422      JSR   PC,WAITIN ;WAIT FOR INTERRUPT
67 033662 012603      MOV   (SP)+,R3 ;GET RESULT MESSAGE
68 033664      ERRHRD 1701...ERR1
033664 104456      TRAP   C#ERRHD
033666 003245      .WORD 1701
033670 000000      .WORD 0
033672 012266      .WORD ERR1
69 033674      EXIT   SUB
033674 104432      TRAP   C#EXIT
033676 000164      .WORD L10033-
70 033700 004737 016626      JSR   PC,GSTAT ;GET STATUS
71 033704 034054      60#
72 033706 032737 040000 003050      BIT   #DRVERR,T.CS ;TEST IF ANY ERROR SET
73 033714 001006      BNE   15# ;YES - SKIP
74 033716 012703 010444      MOV   #MDRERR,R3 ;SET RESULT MESSAGE POINTER
75 033722      ERRHRD 1702...ERR3 ;REPORT ERROR NOT SET

```

```

033722 104456 TRAP C$ERHRD
033724 003246 .WORD 1702
033726 000000 .WORD 0
033730 012402 .WORD ERR3
76 033732 032737 002000 003056 15$: BIT #WGESTAT,T.MP ;TEST IF WGE SET
77 033740 001006 BNE 18$ ;YES - SKIP
78 033742 012703 010523 MOV #MWGERR,R3 ;GET MESSAGE FOR WGE NOT SET
79 033746 ERRHRD 1704,,ERR3
033746 104456 TRAP C$ERHRD
033750 003250 .WORD 1704
033752 000000 .WORD 0
033754 012402 .WORD ERR3
80 033756 042737 040000 003050 18$: BIC #DRVERR,T.CS ;CLEAR DRIVE ERROR BIT
81 033764 042737 002000 003056 BIC #WGESTAT,T.MP ;CLEAR WGE BIT
82 033772 032737 157400 003056 BIT #157400,T.MP ;TEST FOR ANY OTHER ERRORS
83 034000 001004 BNE 16$ ;YES - GO REPORT
84 034002 032737 036000 003050 BIT #36000,T.CS ;TEST ANY ERRORS IN CS REG
85 034010 001405 BEQ 17$ ;NO - SKIP
86 034012 16$: ERRHRD 1703,,ERR6 ;REPORT ERRORS
034012 104456 TRAP C$ERHRD
034014 003247 .WORD 1703
034016 000000 .WORD 0
034020 012570 .WORD ERR6
87 034022 000414 BR 60$ ;EXIT TEST
88 034024 004737 016576 17$: JSR PC,GSTATR ;GET STATUS AND RESET ERROR
89 034030 034054 60$
90 034032 004537 023522 JSR R5,DATGEN ;GO GENERATE DATA
91 034036 000007 7 ;PATTERN 7
92 034040 004737 024212 JSR PC,XREAD ;READ DATA
93 034044 034054 60$
94 034046 004737 023662 JSR PC,DATCOM ;COMPARE DATA
95 034052 034054 60$
96 034054 012737 000002 003022 60$: MOV #2,ERRSWI ;INIT ERROR SWITCH
97 034062 ENDSUB
034062 L10033:
034062 104403 TRAP C$ESUB
98 034064 012737 000002 003022 T3204$: MOV #2,ERRSWI ;INIT ERROR SWITCH
99 034072 PRINTF #FMTOP1,#OPR12,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1> ;REQ RESET WRT LCK
034072 005046 CLR -(SP)
034074 153716 003037 BISB RLDRV+1,(SP)
034100 012746 006142 MOV #DRVNAM,-(SP)
034104 013746 003032 MOV RLBAS,-(SP)
034110 012746 006131 MOV #BASADD,-(SP)
034114 012746 007366 MOV #OPR1A,-(SP)
034120 012746 007347 MOV #OPR12,-(SP)
034124 012746 011243 MOV #FMTOP1,-(SP)
034130 012746 000007 MOV #7,-(SP)
034134 010600 MOV SP,R0
034136 104417 TRAP C$PNTF
034140 062706 000020 ADD #20,SP
100 034144 012701 001274 MOV #700.,R1 ;INITIALIZE WAIT COUNT
101 034150 16$: WAITMS #1
102 034162 004737 016576 JSR PC,GSTATR ;GET STATUS
103 034166 034064 T3204$
104 034170 032737 020000 003056 BIT #WLSTAT,T.MP ;CHECK IF WRITE LOCK RESET
105 034176 001403 BEQ T3265$
106 034200 005301 DEC R1 ;DEC WAIT COUNT

```

```

107 034202 001362          BNE 16$      ;LOOP IF NOT 0
108 034204 000727          BR T3204$  ;ELSE REPEAT MESSAGE
109 034206                T3265$:
110 034206                ENDTST
      034206                L10032:
      034206 104401          TRAP C$ETST
111

```

119

```

1          .SBTTL *TEST 7          **ADJACENT CYLINDFR INTERFERENCE
2 034210   BGNTST          ;TEST 7
3 034210   012737 007031 003016   MOV    #P2T18E,ERHEAD ;SET ERROR HEADER
4 034216   004737 021226           JSR    PC,CKBSVD      ;GO CHECK IF BAD SECTOR FILES VALID
5 034222   004737 016560           JSR    PC,TSTINT     ;INITIALIZE TEST
6 034226   004737 016576           JSR    PC,GSTATR    ;CLEAR DRIVE
7 034232   034232 035422           T3365$
8 034234   005037 003236           CLR    PASCNT       ;CLEAR PASS TO 0
9 034240   012705 177776           MOV    #-2,R5       ;SET R5
10 034244  005737 003444           TST   PASNUM        ;TEST IF FIRST PASS (QUICK VERIFY)
11 034250   001007 000000           BNE   1$            ;NO - SKIP
12 034252  032737 000001 014120   BIT   #ALLCYL,MISWIW ;TEST IF USE ALL CYLINDERS
13 034260   001003 000000           BNE   1$            ;YES - SKIP
14 034262  012705 177730           MCV   #-40.,R5      ;ELSE SET R5 TO NEG 20
15 034266   000402 000000           BR    9$            ;SKIP
16 034270  012705 177770           1$:  MOV   #-10,R5    ;ELSE SET FOR NEG 4
17 034274  012701 002510           9$:  MOV   #T33TBL,R1 ;GET ADDRESS OF WORK TABLE
18 034300  012737 000010 002304   MOV   #10,JJJ       ;SET CLEAR COUNT
19 034306  013721 014122           2$:  MOV   LOLIMW,(R1) ;CLEAR LOCATIONS TO LOLIMIT
20 034312  005337 002304           DEC   JJJ           ;DEC COUNT
21 034316  001373 000000           BNE   2$            ;LOOP UNTIL 0
22 034320  004537 023522           JSR   R5,DATGEN     ;GO GENERATE DATA
23 034324  000011 000000           9.   ;PATTERN 9
24 034326  013737 014124 002512   MOV   HILIMW,T33TBL+2 ;INSERT HILIMIT
25 034334  013737 014124 002514   MOV   HILIMW,T33TBL+4 ;INTO APPROPRIATE LOCATIONS
26 034342  013737 014124 002520   MOV   HILIMW,T33TBL+10
27 034350  013737 014124 002526   MOV   HILIMW,T33TBL+16
28 034356  062705 000002 T3300$: ADD   #2,R5
29
30 034362  032737 000001 014120   BIT   #ALLCYL,MISWIW ;TEST IF USE ALL CYLINDERS
31 034370  001034 000000           BNE   5$            ;YES - SKIP
32 034372  005737 003444           TST   PASNUM        ;TEST IF FIRST PASS (QUICK VERIFY)
33 034376  001403 000000           BEQ   3$            ;NO - SKIP
34 034400  062705 000006           ADD   #6,R5         ;ELSE BUMP CYLINDER POINTER BY 3
35 034404  000402 000000           BR    6$            ;SKIP
36 034406  062705 000044           3$:  ADD   #36.,R5    ;BUMP TO NEXT ENTRY
37 034412  022737 000001 002302  6$:  CMP   #1,T.DRIVE
38 034420  001404 000000           BEQ   44$           ;
39 034422  020537 000244           CMP   R5,164.
40 034426  103013 000000           BHIS  4$
41 034430  000403 000000           BR    69$           ;
42
43 034432  020527 000122           44$: CMP   R5,#82.
44 034436  103007 000000           BHIS  4$
45
46 034440  016537 002610 002304  69$: MOV   CYLTBL(R5),JJJ
47 034446  043737 002310 002304   BIC   CLRBYT,JJJ
48 034454  001013 000000           BNE   8$
49 034456  000137 033170           4$:  JMP   T3165$
50 034462  005705 000000           5$:  TST   R5          ;TEST IF R5 0
51 034464  001002 000000           BNE   7$            ;NO - SKIP
52 034466  062705 000002           ADD   #2,R5
53 034472  023705 002306           7$:  CMP   HLMTW,R5    ;TEST IF ALL CYLINDERS USED
54 034476  001767 000000           BEQ   4$            ;YES - EXIT TEST
55 034500  010537 002304           MOV   R5,JJJ        ;USE R5 AS NEXT CYLINDER
56 034504  023737 002304 014122  8$:  CMP   JJJ,LOLIMW    ;CHECK IF LOWER THAN LOLIMIT
  
```

```

57 034512 103721          BLO      T33004      ;YES - SKIP
58 034514 023737 002304 014124      CMP      JJJ,HILIMW ;CHECK IF HIGHER THAN HILIMW
59 034522 101315          BMI      T33004      ;YES - SKIP
60 034524 012703 002550          MOV      @TBT,R3
61 034530 013713 002304          MOV      JJJ,(R3)
62 034534 013763 002304 000006      MOV      JJJ,6(R3)
63 034542 013763 002304 000010      MOV      JJJ,10(R3)
64 034550 013763 002304 000012      MOV      JJJ,12(R3)
65 034556 013763 002304 000016      MOV      JJJ,16(R3)
66 034564 162737 000001 002304      SUB      @1,JJJ
67 034572 013763 002304 000002      MOV      JJJ,2(R3)
68 034600 013763 002304 000012      MOV      JJJ,12(R3)
69 034606 062737 000002 002304      ADD      @2,JJJ
70 034614 013763 002304 000004      MOV      JJJ,4(R3)
71 034622 013763 002304 000014      MOV      JJJ,14(R3)
72 034630 010337 003030          MOV      R3,TBLSTR
73 034634 004737 021116          JSR      PC,CMOSMD      ;GO CHOSE HEAD
74 034640
75 034640          T33014:
034640          BGNSSUB
034640          TRAP      C@RSUB      ;7.1:
76 034642 042737 003760 003010      BIC      @MQUALS,OPFLAG ;CLEAR ALL MESSAGE QUALIFIERS
77 034650 005737 003236          TST      PASCNT      ;TEST IF PASS 0
78 034654 001414          BEQ      11#         ;YES - SKIP
79 034656 023727 003236 000004      CMP      PASCNT,@4    ;TEST IF PASS 4
80 034664 001404          BEQ      10#         ;YES - SKIP
81 034666 002407          BLT      11#         ;CHECK IF LESS THAN 4. IF YES CLEAR TO 0
82 034670 012737 000004 003236      MOV      @4,PASCNT    ;ELSE SET TO 4
83 034676 052737 000020 003010 10#:    BIS      @INOUTS,OPFLAG ;SET MESSAGE QUAL
84 034704 000405          BR       12#         ;SKIP
85 034706 005037 003236          CLR      PASCNT      ;SET PASS COUNT TO 0
86 034712 052737 000040 003010 11#:    BIS      @OUTINS,OPFLAG ;SET MESSAGE QUAL
87 034720 012737 000003 003026 12#:    MOV      @3,WRTSWI    ;SET READ AND WRITE SWITCH
88 034726 012701 002510          MOV      @T3TBL,R1
89 034732 012703 002550          MOV      @TBT,R3
90 034736 005037 003120 15#:    CLR      DESSEC      ;CLEAR TO SECTOR 0
91 034742 012137 003106          MOV      (R1)+,NEWCYL ;GET NEXT TABLE ENTRY
92 034746 004737 017524          JSR      PC,XSEEK    ;DO SEEK
93 034752 035330 60#
94 034754 012701 005670          MOV      @3000.,R1    ;SET WAIT COUNT FOR 300 MS
95 034760 004737 022420          JSR      PC,RDYWAIT   ;WAIT FOR READY
96 034764 035330 60#
97 034766 012337 003106          MOV      (R3)+,NEWCYL ;GET NEXT TABLE ENTRY
98 034772 004737 017524          JSR      PC,XSEEK    ;DO SEEK
99 034776 035330 60#
100 035000 012701 005670          MOV      @3000.,R1    ;SET WAIT COUNT FOR 300 MS
101 035004 004737 022420          JSR      PC,RDYWAIT   ;WAIT FOR READY
102 035010 035330 60#
103 035012 004737 023032          JSR      PC,VERPOS    ;VERIFY POSITION
104 035016 035330 60#
105 035020 004737 024710 16#:    JSR      PC,BSCMP     ;CHECK FOR BAD SECTOR
106 035024 035134          32#
107 035026 032737 000001 003026      BIT      @BITO,WRTSWI ;TEST IF WRITE THIS PASS
108 035034 001425          BEQ      29#         ;NO - SKIP
109 035036 004737 024152          JSR      PC,XWRITE    ;DO WRITE
110 035042 035330 60#
111 035044 005237 003120          INC      DESSEC      ;INC SECTOR

```

```

112 035050 022737 000050 003120      CMP      #40.,DESSEC      ;TEST IF ALL SECTORS USED
113 035056 001360                      BNE      16#          ;NO SKIP
114 035060 042737 000060 003010      BIC      @INOUTS,OPFLAG ;CLEAR QUALIFIERS
115 035066 042737 000001 003026      BIC      @BIT0,WRTSWI   ;CLEAR WRITE REQUIRED SWITCH
116 035074 052737 000100 003010      BIS      @FOLWRT,OPFLAG ;SET FOLLOWING WRITE QUALIFIER
117 035102 005037 003120      CLR      DESSEC        ;CLEAR TO SECTOR 0
118 035106 00074.                      BR       16#          ;SKIP
119 035110 032737 000002 003026 29# : BIT      @BIT1,WRTSWI   ;TEST IF READ THIS PASS
120 035116 001414                      BEQ      33#          ;NO - SKIP
121 035120 044737 024212                      JSR      PC,XREAD     ;ELSE DO READ
122 035124 035330                      60#
123 035126 004737 023602                      JSR      PC,DATCOM   ;COMPARE DATA
124 035132 035330                      60#
125 035134 005237 003120 32# : INC      DESSEC        ;BUMP SECTOR
126 035140 022737 000050 003120      CMP      #40.,DESSEC   ;TEST IF ALL SECTORS USED
127 035146 001324                      BNE      16#          ;NO - LOOP
128 035150 005037 003120 33# : CLR      DESSEC        ;CLEAR DESIRED SECTOR
129 035154 005037 003026      CLR      WRTSWI        ;CLEAR WRITE/READ SWITCH
130 035160 005237 003236      INC      PASCNT        ;BUMP PASS COUNT
131 035164 042737 003160 003010      BIC      @EQUALS,OPFLAG ;CLEAR ALL QUALIFIERS
132 035172 023727 003236 000004      CMP      PASCNT,#4     ;TEST IS PASS 4
133 035200 001453                      BEQ      60#          ;YES - SKIP
134 035202 023727 003236 000010      CMP      PASCNT,#8     ;TEST IF PASS 8.
135 035210 001447                      BEQ      60#          ;YES - SKIP
136 035212 023727 003236 000003      CMP      PASCNT,#3     ;TEST IF PASS 3
137 035220 001430                      BEQ      39#          ;YES - SKIP
138 035222 023727 003236 000007      CMP      PASCNT,#7     ;TEST IF PASS 7
139 035230 001430                      BEQ      40#          ;YES - SKIP
140 035232 012737 000001 003026      MOV      @BIT0,WRTSWI   ;SET WRITE REQUIRED
141 035240 023727 003236 000001      CMP      PASCNT,#1     ;TEST IF PASS 1
142 035246 001411                      BEQ      37#          ;YES - SKIP
143 035250 023727 003236 000002      CMP      PASCNT,#2     ;TEST IF PASS 2
144 035256 001405                      BEQ      37#          ;YES - SKIP
145 035260 052737 000040 003010      BIS      @OUTINS,OPFLAG ;SET MESSAGE QUALIFIER
146 035266 000137 034736 36# : JMP      15#          ;GO DO NEXT PASS
147 035272 052737 000020 003010 37# : BIS      @INOUTS,OPFLAG ;SET MESSAGE QUALIFIER
148 035300 000772                      BR       36#
149 035302 052737 000200 003010 39# : BIS      @REVSKS,OPFLAG ;SET MESSAGE QUALIFIER
150 035310 000403                      BR       41#
151 035312 052737 000400 003010 40# : BIS      @FMSKSKS,OPFLAG ;SET MESSAGE QUALIFIER
152 035320 012737 000002 003026 41# : MOV      @BIT1,WRTSWI   ;SET READ REQUIRED
153 035326 000757                      BR       36#
154 035330 012737 000002 003022 60# : MOV      @2,ERRSWI     ;INIT ERROR SWITCH
155 035336                      ENDSUB
                                L10035:
035336 104403      TRAP     CIESUB
156 035340                      ESCAPE   TST
                                TRAP     CIESCAPE
                                .WORD   L10034-.
035342 000060      MOV      #3,WRTSWI     ;SET FOR READ AND WRITE REQ.
157 035344 012737 000003 003026      CMP      PASCNT,#4     ;TEST IF PASS 4
158 035352 023727 003236 000004      BNE      45#          ;NO - SKIP
159 035360 001004                      MOV      @T33TBL+10,TBLSTR ;STORE MID POINT IN TABLE
160 035362 012737 002520 003030      MOV      48#
161 035370 000410                      BR       48#          ;GO START PASS 4
162 035372 005037 003236 45# : CLR      PASCNT        ;CLEAR TO PASS 0
163 035376 004737 021142      JSR      PC,SWAPHD     ;GO SWAP TO HEAD 1 OR END TEST
164 035402 034356                      T3300# ;ABORT RETURN

```

```
165 035404 012737 002510 003030      MOV      @T33TBL,TBLSTR  ;STORE START OF TABLE
166
167 035412 062703 000010      484:    ADD      @10,R3
168 035416 000137 034640      JMP      T33014
169 035422      T33654:
170 035422      ENDTST
    035422      L:0034
    035422 104401      TRAP    C0ETST
```



Line	Address	Op1	Op2	Op3	Op4	Op5	Op6	Op7	Op8	Op9	Op10	Op11
1												
2	035424											
3	035424	012737	007053	003016		MOV	0P2T19E,ERHEAD					T8::
4	035432	004737	021226			JSR	PC,CKBSVD					:GO CHECK IF BAD SECTOR FILES VALID
5	035436	004737	016560			JSR	PC,TSTINT					:INITIALIZE TEST
6	035442	004737	016576			JSR	PC,GSTATR					:CLEAR DRIVE
7	035446	036614					T3465:					
8	035450	005037	003236			CLR	PASCNT					:CLEAR PASS TO 0
9	035454	012705	177776			MOV	0-2,R5					:SET R5
10	035460	005737	003444			TST	PASNUM					:TEST IF FIRST PASS (QUICK VERIFY)
11	035464	001007				BNE	1:					:NO - SKIP
12	035466	032737	000001	014120		BIT	0ALLCYL,MISWIW					:TEST IF USE ALL CYLINDERS
13	035474	001003				BNE	1:					:YES - SKIP
14	035476	012705	177730			MOV	0-40.,R5					:ELSE SET R5 TO NEG 20
15	035502	000402				BR	9:					:SKIP
16	035504	012705	177770			18:	MOV	0-10,R5				:SET FOR NEXT ENTRY
17	035510	012701	002510			98:	MOV	0T33TBL,R1				:GET ADDRESS OF WORK TABLE
18	035514	012737	000010	002304			MOV	010,JJJ				:SET CLEAR COUNT
19	035522	013721	014122			28:	MOV	LOLIMW,(R1)				:CLEAR LOCATIONS TO LOLIMIT
20	035526	005337	002304				DEC	JJJ				:DEC COUNT
21	035532	001373					BNE	2:				:LOOP UNTIL 0
22	035534	013737	014124	002512			MOV	HILIMW,T33TBL*2				:INSERT HILIMIT
23	035542	013737	014124	002516			MOV	HILIMW,T33TBL*6				:INTO APPROPRIATE LOCATIONS
24	035550	013737	014124	002522			MOV	HILIMW,T33TBL*12				
25	035556	062705	000002			T3400:	ADD	02,R5				
26	035562	032737	000001	014120			BIT	0ALLCYL,MISWIW				:TEST IF USE ALL CYLINDERS
27	035570	001034					BNE	5:				:YES - SKIP
28	035572	005737	003444				TST	PASNUM				:TEST IF FIRST PASS (QUICK VERIFY)
29	035576	001003					BNE	3:				:NO - SKIP
30	035600	062705	000046				ADD	038.,R5				:ELSE BUMP CYLINDER POINTER BY 19
31	035604	000402					BR	6:				:SKIP
32	035606	062705	000006			38:	ADD	06,R5				:BUMP CYLINDER POINTER BY 3
33	035612	022737	000001	002302		68:	CMP	01.T.DRIVE				
34	035620	001404					BEQ	444:				
35	035622	020527	000244				CMP	R5,0164.				
36	035626	103013					BHIS	4:				
37	035630	000403					BR	669:				
38	035632	020527	000122			444:	CMP	R5,082.				
39	035636	103007					BHIS	4:				
40	035640	016537	002610	002304		669:	MOV	CYL TBL(R5),JJJ				
41	035646	043737	002310	002304			BIC	CLRBYT,JJJ				
42	035654	001013					BNE	8:				
43	035656	000137	036614			48:	JMP	T3465:				:EXIT TEST
44	035662	005705				58:	TST	R5				:TEST IF R5 0
45	035664	001002					BNE	7:				:NO - SKIP
46	035666	062705	000002				ADD	02,R5				
47	035672	022705	002306			78:	CMP	0MLTW,R5				:TEST IF ALL CYLINDERS USED
48	035676	001767					BEQ	4:				:YES - EXIT TEST
49	035700	010537	002304				MOV	R5,JJJ				:USE R5 AS NEXT CYLINDER
50	035704	023737	002304	014122		88:	CMP	JJJ,LOLIMW				:TEST IF PAST LO LIMIT
51	035712	103721					BLO	T3400:				:YES - SKIP
52	035714	023737	002304	014124			CMP	JJJ,HILIMW				:TEST IF PAST HILIMIT
53	035722	101315					BHI	T3400:				:YES - SKIP
54	035724	012703	002550				MOV	0TBT,R3				
55	035730	013713	002304				MOV	JJJ,(R3)				
56	035734	013763	002304	000002			MOV	JJJ,2(R3)				

```

57 035742 013763 002304 000004      MOV      JJJ,4(R3)
58 035750 013763 002304 000006      MOV      JJJ,6(R3)
59 035756 013763 002304 000010      MOV      JJJ,10(R3)
60 035764 013763 002304 000012      MOV      JJJ,12(R3)
61 035772 010337 003030      MOV      R3,TBLSTR
62 035776 004737 021116      JSR      PC,CHOSHD      ;GO CHOSE HEAD
63 036002
64 036002      T3401$:
   036002      BGNSUB
   036002      TRAP      C#BSUB      TB.1:
65 036004 042737 003760 003010      BIC      @MQUALS,OPFLAG      ;CLEAR ALL MESSAGE QUALIFIERS
66 036012 005737 003236      TST      PASCNT      ;TEST IF PASS 0
67 036015 001414      BEQ      11$      ;YES - SKIP
68 036020 023727 003236 000003      CMP      PASCNT,#3      ;TEST IF PASS 3
69 036026 001404      BEQ      10$      ;YES - SKIP
70 036030 002407      BLT      11$      ;CHECK IF LESS THAN 3, IF YES CLEAR TO 0
71 036032 012737 000003 003236      MOV      @3,PASCNT      ;ELSE SET TO 3
72 036040 052737 000020 003010 10$:      BIS      @INOUTS,OPFLAG      ;SET MESSAGE QUAL
73 036046 000405      BR       12$      ;SKIP
74 036050 005037 003236      CLR      PASCNT      ;SET PASS COUNT TO 0
75 036054 052737 000040 003010 11$:      BIS      @OUTINS,OPFLAG      ;SET MESSAGE QUAL
76 036062 012737 000003 003026 12$:      MOV      @3,WRTSWI      ;SET READ AND WRITE SWITCH
77 036070 012701 002510      MOV      @T3TBL,R1
78 036074 012703 002550      MOV      @TBT,R3
79 036100 005037 003120      15$:      CLR      DESSEC
80 036104 012137 003106      MOV      (R1)+,NEWCYL      ;GET NEXT TABLE ENTRY
81 036110 004737 017524      JSR      PC,XSEEK      ;DO SEEK
82 036114 036522      60$
83 036116 012701 005670      MOV      @3000.,R1      ;SET WAIT COUNT FOR 300 MS
84 036122 004737 022420      JSR      PC,RDYWAIT      ;WAIT FOR READY
85 036126 036522      60$
86 036130 012337 003106      MOV      (R3)+,NEWCYL      ;GET NEXT TABLE ENTRY
87 036134 004737 017524      JSR      PC,XSEEK      ;DO SEEK
88 036140 036522      60$
89 036142 012701 005670      MOV      @3000.,R1      ;SET WAIT COUNT FOR 300 MS
90 036146 004737 022420      JSR      PC,RDYWAIT      ;WAIT FOR READY
91 036152 036522      60$
92 036154 004737 023032      JSR      PC,VERPOS      ;VERIFY POSITION
93 036160 036522      60$
94 036162 004737 024710      16$:      JSR      PC,BSCHK      ;CHECK FOR BAD SECTOR
95 036166 036336      32$
96 036170 005737 003236      TST      PASCNT      ;"YES" RETURN
97 036174 001407      BEQ      17$      ;TEST IF PASS 0
98 036176 022737 000003 003236      CMP      @3,PASCNT      ;YES - SKIP
99 036204 001403      BEQ      17$      ;TEST IF PASS 3
100 036206 005037 036226      BEQ      17$      ;YES - SKIP
101 036212 000403      CLR      25$      ;ELSE CLEAR DATA PATTERN SELECTOR
102 036214 012737 000010 036226 17$:      BR       18$
103 036222 004537 023522      18$:      MOV      @8.,25$      ;SET DATA PATTERN SELECTOR TO 8
104 036226 000000      25$:      JSR      R5,DATGEN      ;GO GENERATE DATA
105 036230 032737 000001 003026      .WORD   0
106 036236 001425      BIT      @BIT0,WRTSWI      ;TEST IF WRITE THIS PASS
107 036240 004737 024152      BEQ      29$      ;NO - SKIP
108 036244 036522      JSR      PC,XWRITE      ;DO WRITE
109 036246 005237 003120      60$
110 036252 022737 000050 003120      INC      DESSEC      ;INC SECTOR
111 036260 001340      CMP      @40.,DESSEC      ;TEST IF ALL SECTORS USED
      BNE      16$      ;NO - SKIP

```

```

112 036262 042737 000060 003010 BIC #INOUTS,OPFLAG ;CLEAR QUALIFIERS
113 036270 042737 000001 003026 BIC #BIT0,WRTSWI ;CLEAR WRITE REQUIRED SWITCH
114 036276 052737 000100 003010 BIS #FOLWRT,OPFLAG ;SET FOLLOWING WRITE QUALIFIER
115 036304 005037 003120 CLR DESSEC ;CLEAR TO SECTOR 0
116 036310 000724 BR 16# ;SKIP
117 036312 032737 000002 003026 29# BIT #BIT1,WRTSWI ;TEST IF READ THIS PASS
118 036320 001414 BEQ 33# ;NO - SKIP
119 036322 004737 024212 31# JSR PC,XREAD ;ELSE DO READ
120 036326 036522 60#
121 036330 004737 023662 JSR PC,DATCOM ;COMPARE DATA
122 036334 036522 60#
123 036336 005237 003120 32# INC DESSEC ;BUMP SECTOR
124 036342 022737 000050 003120 CMP #40,,DESSEC ;TEST IF ALL SECTORS USED
125 036350 001304 BNE 16# ;NO - LOOP
126 036352 005037 003120 33# CLR DESSEC ;CLEAR DESIRED SECTOR
127 036356 005037 003026 CLR WRTSWI ;CLEAR WRITE/READ SWITCH
128 036362 005237 003236 INC PASCNT ;BUMP PASS COUNT
129 036366 042737 003760 003010 BIC #EQUALS,OPFLAG ;CLEAR ALL QUALIFIERS
130 036374 023727 003236 000003 CMP PASCNT,#3 ;TEST IS PASS 3
131 036402 001447 BEQ 60# ;YES - SKIP
132 036404 023727 003236 000006 CMP PASCNT,#6 ;TEST IF PASS 6
133 036412 001443 BEQ 60# ;YES - SKIP
134 036414 023727 003236 000001 CMP PASCNT,#1 ;TEST IF PASS 1
135 036422 001424 BEQ 39# ;YES - SKIP
136 036424 023727 003236 000004 CMP PASCNT,#4 ;TEST IF PASS 4
137 036432 001424 BEQ 40# ;YES - SKIP
138 036434 012737 000002 003026 MOV #BIT1,WRTSWI ;SET WRITE REQUIRED BIT
139 036442 023727 003236 000002 CMP PASCNT,#2 ;TEST IF PASS 2
140 036450 001405 BEQ 37# ;YES - SKIP
141 036452 052737 001000 003010 BIS #REVSKO,OPFLAG ;SET REVERSE QUALIFIER
142 036460 000137 036100 36# JMP 15# ;GO DO NEXT PASS
143 036464 052737 002000 003010 37# BIS #FWD SKO,OPFLAG ;SET FWD QUALIFIER
144 036472 000772 BR 36# ;GO DO NEXT PASS
145 036474 052737 000020 003010 39# BIS #INOUTS,OPFLAG ;SET QUALIFIER
146 036502 000403 BR 41# ;SKIP
147 036504 052737 000040 003010 40# BIS #OUTINS,OPFLAG ;SET MESSAGE QUALIFIER
148 036512 012737 000001 003026 41# MOV #BIT0,WRTSWI ;SET WRITE REQUIRED BIT
149 036520 000757 BR 36# ;GO DO NEXT PASS
150 036522 012737 000002 003022 60# MOV #2,ERRSWI ;INIT ERROR SWITCH
151 036530 ENDSUB
036530 L10037:
036530 104403 TRAP C#ESUB
152 036532 ESCAPE TST ;EXIT TEST IF ERROR
036532 104410 TRAP C#ESCAPE
036534 000060 .WORD L10036-
153 036536 012737 000003 003026 MOV #3,WRTSWI ;SET FOR READ AND WRITE REQ.
154 036544 023727 003236 000003 CMP PASCNT,#3 ;TEST IF PASS 3
155 036552 001004 BNE 45# ;NO - SKIP
156 036554 012737 002516 003030 MOV #T33TBL+6,TBLSTR ;STORE MID POINT IN TABLE
157 036562 000410 BR 48# ;GO START PASS 4
158 036564 005037 003236 45# CLR PASCNT ;CLEAR TO PASS 0
159 036570 004737 021142 JSR PC,SWAPHD ;GO SWAP TO HEAD ONE OR ABORT TEST
160 036574 035556 T3400# ;ABORT RETURN
161 036576 012737 002510 003030 MOV #T33TBL,TBLSTR ;STORE START OF TABLE
162 036604 062703 000006 48# ADD #6,R3
163 036610 000137 036002 JMP T3401#
164 036614 T3465#

```

H10

CZRLN80 RLO1/02 DRIVE TEST 3 MACRO V04.00 20-JAN 83 14:40:57 PAGE 19-3  
\*TEST 8 \*\*OVERWRITE

SEQ 0124

165 036614  
036614  
036614 104401  
166 036616

ENDTST  
L10036:  
TRAP C#ETST  
ENDMOD

1				.SBTTL	PARAMETER CODING
2	036616			BGNMOD	HRDPRM
3	036616			BGNHRD	
	036616	000030			.WORD L10040-L#HARD/2
4	036620			GPRML	CNTYPE,CNT,1,YES
	036620	005130			.WORD T#CODE
	036622	036764			.WORD CNTYPE
	036624	000001			.WORD 1
5	036626			GPRMA	CSRMSG,CSR,0,160000,177776,YES
	036626	000031			.WORD T#CODE
	036630	036700			.WORD CSRMSG
	036632	160000			.WORD T#LOLIM
	036634	177776			.WORD T#HILIM
6	036636			GPRMA	VECMG,VECT,0,0,776,YES
	036636	001031			.WORD T#CODE
	036640	036714			.WORD VECMSG
	036642	000000			.WORD T#LOLIM
	036644	000776			.WORD T#HILIM
7	036646			GPRMD	DRMSG,DRSB,0,3400,0,7,YES
	036646	004032			.WORD T#CODE
	036650	036756			.WORD DRMSG
	036652	003400			.WORD 3400
	036654	000000			.WORD T#LOLIM
	036656	000007			.WORD T#HILIM
8	036660			GPRML	DRTYPE,TYPDR,1,YES
	036660	003130			.WORD T#CODE
	036662	036734			.WORD DRTYPE
	036664	000001			.WORD 1
9	036666			GPRMD	BRMSG,PRIOR,0,340,0,7,YES
	036666	002032			.WORD T#CODE
	036670	036723			.WORD BRMSG
	036672	000340			.WORD 340
	036674	000000			.WORD T#LOLIM
	036676	000007			.WORD T#HILIM
10					
11	036700			ENDHRD	
	036700				.EVEN
12				L10040:	
13	036700	102	125	123	CSRMSG: .ASCIZ /BUS ADDRESS/
	036703	040	101	104	
	036706	104	122	105	
	036711	123	123	000	
14	036714	126	105	103	VECMG: .ASCIZ /VECTOR/
	036717	124	117	122	
	036722	000			
15	036723	102	122	040	BRMSG: .ASCIZ /BR LEVEL/
	036726	114	105	126	
	036731	105	114	000	
16	036734	104	122	111	DRTYPE: .ASCIZ /DRIVE TYPE = RL01/
	036737	126	105	040	
	036742	124	131	120	
	036745	105	040	075	
	036750	040	122	114	
	036753	060	061	000	
17	036756	104	122	111	DRMSG: .ASCIZ /DRIVE/
	036761	126	105	000	

```

18 036764      122      114      061  CNTYPE: .ASCIZ  /RL11/
    036767      061      000
19 036771
20
21
22 036772      BGNMOD  SF TPRM
23 036772      3GNSFT
    036772      000056      .WORD  L10041-L$SOFT/2
24
26 036774      GPRML   CYLQ,MISWI,1,YES
    036774      000130      .WORD  T$CODE
    036776      037130      .WORD  CYLQ
    037000      000001      .WORD  1
27 037002      GPRML   SECQ,MISWI,2,YES
    037002      000130      .WORD  T$CODE
    037004      037144      .WORD  SECQ
    037006      000002      .WORD  2
33 037010      GPRML   MANQ,MISWI,10000,YES
    037010      000130      .WORD  T$CODE
    037012      037161      .WORD  MANQ
    037014      100000      .WORD  100000
34
36 037016      GPRML   LOLIMQ,MISWI,40000,YES
    037016      000130      .WORD  T$CODE
    037020      037215      .WORD  LOLIMQ
    037022      040000      .WORD  40000
37 037024      XFERF   1$
    037024      006044      .WORD  T$CODE
38 037026      GPRMD   LIMVAL,LOLIM,D,255.,0,253.,YES
    037026      001052      .WORD  T$CODE
    037030      037234      .WORD  LIMVAL
    037032      000377      .WORD  255.
    037034      000000      .WORD  T$LOLIM
    037036      000375      .WORD  T$HILIM
39 037040      1$:    GPRML   HILIMQ,MISWI,20000,YES
    037040      000130      .WORD  T$CODE
    037042      037242      .WORD  HILIMQ
    037044      020000      .WORD  20000
40 037046      XFERF   2$
    037046      006044      .WORD  T$CODE
41 037050      GPRMD   LIMVAL,HILIM,D,255.,0,255.,YES
    037050      002052      .WORD  T$CODE
    037052      037234      .WORD  LIMVAL
    037054      000377      .WORD  255.
    037056      000000      .WORD  T$LOLIM
    037060      000377      .WORD  T$HILIM
42 037062      2$:    GPRML   HEADQ,MISWI,10000,YES
    037062      000130      .WORD  T$CODE
    037064      037263      .WORD  HEADQ
    037066      010000      .WORD  10000
43 037070      XFERF   3$
    037070      006044      .WORD  T$CODE
44 037072      GPRMD   HEADV,HEAD,D,17,0,1,YES
    037072      003052      .WORD  T$CODE
    037074      037305      .WORD  HEADV
    037076      000017      .WORD  17
    037100      000000      .WORD  T$LOLIM

```

037102	000001				.WORD	T#HILIM
46 037104					GPRMD	ERLIMQ,ERLIM,D,377,0,377,1FS
037104	004052				.WORD	T#CODE
037106	037330				.WORD	ERLIMQ
037110	000377				.WORD	377
037112	000000				.WORD	T#LOLIM
037114	000377				.WORD	T#HILIM
48 037116					GPRMD	DCLIMQ,DCLIM,D,377,1,377,YES
037116	005052				.WORD	T#CODE
037120	037352				.WORD	DCLIMQ
037122	000377				.WORD	377
037124	000001				.WORD	T#LOLIM
037126	000377				.WORD	T#HILIM
50 037130					ENDSFT	
					.EVEN	
037130					L10041:	
51						
53 037130	125	123	105	CYLQ:	.ASCIZ	/USE ALL CYL/
037133	040	101	114			
037136	114	040	103			
037141	131	114	000			
54 037144	125	123	105	SECQ:	.ASCIZ	/USE ALL SECT/
037147	040	101	114			
037152	114	040	123			
037155	105	103	124			
037160	000					
60 037161	104	117	040	MANQ:	.ASCIZ	/DO MANUAL INTERVENTION TEST/
037164	115	101	116			
037167	125	101	114			
037172	040	111	116			
037175	124	105	122			
037200	126	105	116			
037203	124	111	117			
037206	116	040	124			
037211	105	123	124			
037214	000					
62 037215	114	117	127	LOLIMQ:	.ASCIZ	/LOW SEEK LIMIT/
037220	040	123	105			
037223	105	113	040			
037226	114	111	115			
037231	111	124	000			
63 037234	126	101	114	LIMVAL:	.ASCIZ	/VALUE/
037237	125	105	000			
64 037242	125	120	120	HILIMQ:	.ASCIZ	/UPPER SEEK LIMIT/
037245	105	122	040			
037250	123	105	105			
037253	113	040	114			
037256	111	115	111			
037261	124	000				
65 037263	125	123	105	HEADQ:	.ASCIZ	/USE ONLY ONE SURF/
037266	040	117	116			
037271	114	131	040			
037274	117	116	105			
037277	040	123	125			
037302	122	106	000			
66 037305	127	110	101	HEADV:	.ASCIZ	/WHAT SURF (0 OR 1)/
037310	124	040	123			

	037313	125	122	106	
	037316	040	050	060	
	037321	040	117	122	
	037324	040	061	051	
	037327	000			
68	037330	111	116	120	ERLIMQ: .ASCIZ /INPUT ERROR LIMIT/
	037333	125	124	040	
	037336	105	122	122	
	037341	117	122	040	
	037344	114	111	115	
	037347	111	124	000	
70	037352	104	101	124	DCLIMQ: .ASCIZ /DATA CMP ERR LMT/
	037355	101	040	103	
	037360	115	120	040	
	037363	105	122	122	
	037366	040	114	115	
	037371	124	000		
72					.EVEN
73	037374				ENDMOD
74					
75	037374				LASTAD
					.EVEN
	037374	000000			.WORD 0
	037376	000000			.WORD 0
	037400				L\$LAST::
76					
77		000001			.END



ADR	=	000020	G	CLKCSR	=	172540	C#MEM	=	000031	EF.STA	=	000040	G	FMT9	=	011554	
AFMID	=	003214		CLKCTR	=	172544	C#MSG	=	000023	ERHEAD	=	003016		FOLWRT	=	000100	
AFMIDU	=	003216		CLKFLG	=	003474	C#OPEN	=	000034	ERLIM	=	000010		FRMWD	=	007463	
ALLCYL	=	000001		CLNCOD	=	015470	C#PNTB	=	000014	ERLIMQ	=	037330		FWDSCO	=	002000	
ALLSEC	=	000002		CLRBYT	=	002310	C#PNTF	=	000017	ERLIMW	=	014130		FWDSSK	=	000400	
ANYERR	=	100000		CLRPAR	=	026310	C#PNTS	=	000016	ERRCNT	=	003244		F#AU	=	000015	
ARMID	=	003220		CNT	=	000012	C#PNTX	=	000015	ERRPOI	=	003242		F#AUTO	=	000020	
ARMIDU	=	003222		CNTYPE	=	036764	C#QIO	=	000377	ERRSWI	=	003022		F#BGN	=	000040	
ASSEMB	=	000010		COMPOP	=	007777	C#RDBU	=	000007	ERRVEC	=	003234		F#CLEA	=	000007	
BADADD	=	004000		CONMSG	=	000004	C#REFG	=	000047	ERR1	=	012266	G	F#DU	=	000016	
BAMSK	=	000060		CONTIN	=	014362	C#RESE	=	000033	ERR10	=	013662	G	F#END	=	000041	
BANAM	=	006233		COSTAT	=	000040	C#REVI	=	000003	ERR2	=	012334	G	F#HARD	=	000004	
BASADD	=	006131		COUNT	=	003240	C#RFLA	=	000021	ERR3	=	012402	G	F#HW	=	000013	
BELL	=	011117		CRDYMS	=	000200	C#RPT	=	000025	ERR4	=	012450	G	F#INIT	=	000006	
BHSTAT	=	000010		CSNAM	=	006226	C#SEFG	=	000046	ERR5	=	012520	G	F#JMP	=	000050	
BIT0	=	000001	G	CSR	=	000000	C#SPRI	=	000041	ERR6	=	012570	G	F#MOD	=	000000	
BIT00	=	000001	G	CSRMSG	=	036700	C#SVEC	=	000037	ERR7	=	013452	G	F#MSG	=	000011	
BIT01	=	000002	G	CURCYL	=	003110	C#TPRI	=	000013	ERR8	=	013522	G	F#PROT	=	000021	
BIT02	=	000004	G	CYLQ	=	037130	C1OMS	=	011176	ERR9	=	013616	G	F#PWR	=	000017	
BIT03	=	000010	G	CYLTLB	=	002610	C5SEC	=	011235	EV	=	000004	G	F#RPT	=	000012	
BIT04	=	000020	G	CYLUP	=	000004	C5OMS	=	011207	EXACYL	=	003230		F#SEG	=	000003	
BIT05	=	000040	G	CYLWD	=	007456	DANAM	=	006240	EXHCYL	=	003226		F#SOFT	=	000005	
BIT06	=	000100	G	C#AU	=	000052	DATA	=	000001	EXOCYL	=	003224		F#SRV	=	000010	
BIT07	=	000200	G	C#AUTC	=	000061	DATCOM	=	023662	EXROT	=	003232		F#SUB	=	000002	
BIT08	=	000400	G	C#BRK	=	000022	DATGEN	=	023522	E#END	=	002100		F#SW	=	000014	
BIT09	=	001000	G	C#BSEG	=	000004	DKERR	=	004000	E#LOAD	=	000035		F#TEST	=	000001	
BIT1	=	000002	G	C#BSUB	=	000002	DCLIM	=	000012	FBSFIL	=	003676		GBND	=	002314	
BIT10	=	002000	G	C#CEFG	=	000045	DCLIMQ	=	037352	FMTOP1	=	011243		GETPOS	=	022704	
BIT11	=	004000	G	C#CLCK	=	000062	DCLIMW	=	014132	FMTOP2	=	011272		GETSTA	=	000003	
BIT12	=	010000	G	C#CLEA	=	000012	DESDIF	=	003112	FMTOP3	=	011314		GLBDAT	=	002230	G
BIT13	=	020000	G	C#CLOS	=	000035	DESHD	=	003116	FMT1	=	011335		GLBEQA	=	002230	G
BIT14	=	040000	G	C#CLP1	=	000006	DESSEC	=	003120	FMT1.1	=	011342		GLBERR	=	012266	G
BIT15	=	100000	G	C#CVEC	=	000036	DESSGN	=	003114	FMT11	=	011561		GLBSUB	=	015622	G
BIT2	=	000004	G	C#DCLN	=	000044	DIAGMC	=	000000	FMT12	=	011567		GLBTXT	=	005350	G
BIT3	=	000010	G	C#DDDU	=	000051	DIFAUG	=	003102	FMT13	=	011575		GSTAT	=	016626	
BIT4	=	000020	G	C#DRPT	=	000024	DIFWD	=	007432	FMT14	=	011641		GSTATC	=	016612	
BIT5	=	000040	G	C#DU	=	000053	DIRBIT	=	000004	FMT15	=	011673		GSTATG	=	016636	
BIT6	=	000100	G	C#EDIT	=	000003	DIRMSK	=	002320	FMT16	=	011727		GSTATR	=	016576	
BIT7	=	000200	G	C#ERDF	=	000055	DLTERR	=	010000	FMT17	=	011740		GTSTAT	=	000104	
BIT8	=	000400	G	C#ERHR	=	000056	DONE	=	003012	FMT18	=	011762		G#CNT0	=	000200	
BIT9	=	001000	G	C#ERRO	=	000060	DRDYMS	=	000001	FMT19	=	012014		G#DELM	=	000372	
BOE	=	000400	G	C#ERSF	=	000054	DRMSG	=	036756	FMT2	=	011351		G#DISP	=	000003	
BRMSG	=	036723		C#ERSO	=	000057	DRSB	=	000010	FMT20	=	012051		G#EXCP	=	000400	
BSCHK	=	024710		C#ESCA	=	000010	DRSEL	=	000004	FMT21	=	012101		G#HILI	=	000002	
BSFLAG	=	003024		C#ESEG	=	000005	DRSET	=	000010	FMT22	=	012124		G#LOLI	=	000001	
BSFVAL	=	003500		C#ESUB	=	000003	DRTYPE	=	036734	FMT23	=	012160		G#NO	=	000000	
BSNSTR	=	007540		C#ETST	=	000001	DRVCNT	=	003100	FMT24	=	012174		G#OFFS	=	000400	
BYPSTN	=	007471		C#EXIT	=	000032	DRVERR	=	040000	FMT25	=	012201		G#OFSI	=	000376	
CAFDT	=	011224		C#GETB	=	000026	DRVNAM	=	006142	FMT26	=	012211		G#PRMA	=	000001	
CAMSK	=	002316		C#GETW	=	000027	DRVNAV	=	006147	FMT27	=	012235		G#PRMD	=	000002	
CCYLUP	=	011215		C#GMAN	=	000043	DSESTA	=	000400	FMT28	=	012254		G#PRML	=	000000	
CHOSMD	=	021116		C#GPHR	=	000042	DSMSK	=	001400	FMT3	=	011354		G#RADA	=	000140	
CKBSVD	=	021226		C#GPLO	=	000030	DSPCOD	=	014134	FMT4	=	011357		G#RADB	=	000000	
CKDATA	=	000102		C#GPRI	=	000040	EF.CON	=	000036	FMT5	=	011370		G#RADD	=	000040	
CKERLM	=	016230		C#INIT	=	000011	EF.NEW	=	000035	FMT6	=	011410		G#RADL	=	000120	
CLKADR	=	003476		C#INLP	=	000020	EF.PWR	=	000034	FMT7	=	011452		G#RADO	=	000020	
CLKCSB	=	172542		C#MANI	=	000050	EF.RES	=	000037	FMT8	=	011522		G#XFER	=	000004	

SYMBOL TABL F

G\$YES = 000010	I\$INIT= 000041	L\$EXP1 002046 G	L10030 033170	MQUALS= 003760
HADONE 003014	I\$MOD = 000041	L\$EXP4 002064 G	L10031 033104	MREAD 005354
HCESTA= 040000	I\$MSG = 000041	L\$EXP5 002066 G	L10032 034206	MREADM 005365
HCR CER= 004000	I\$PROT= 000040	L\$HARD 036620 G	L10033 034062	MRESKO 005756
HDALIG= 000010	I\$PTAB= 000041	L\$HIME 002120 G	L10034 035422	MREVSK 005640
HDCYL 002322	I\$PWR = 000041	L\$HPCP 002016 G	L10035 035336	MRLFAL 011004
HDHSEL = 000100	I\$RPT = 000041	L\$HPTP 002022 G	L10036 036614	MRSLT 005526
HDMOVF 007313	I\$SEG = 000041	L\$HW 014102 G	L10037 036530	MSEK 005350
HORCMP= 000002	I\$SETU= 000041	L\$ICP 002104 G	L10040 036700	MSPERR 010505
HDR40 = 100000	I\$SFT = 000041	L\$INIT 014156 G	L10041 037130	MSTERR 010540
HDSEC = 000077	I\$SRV = 000041	L\$LADP 002026 G	MAJINC 003472	MTMBS 006110
HDSEL = 000020	I\$SUB = 000041	L\$LAST 037400 G	MANQ 037161	MTOSLO 006306
HDWD 007445	I\$TST = 000041	L\$LOAD 002100 G	MAPROX 007143	MULOAD 005537
HDWRD1 003056	JJJ 002304	L\$LUN 002074 G	MBADAD 006012	MUNDEF 010737
HDWRD2 003060	J\$JMP = 000167	L\$MREV 002050 G	MBADSF 006033	MWDERR 010572
HDWRD3 003062	LAB 014334	L\$NAME 002000 G	MBSET= 000001	MWGERR 010523
HEAD = 000006	LABACF 007263	L\$PRIO 002042 G	MCERR 010333	MWORD 006300
HEADLM= 010000	LABACR 007277	L\$PROT 014072 G	MCONHN 006377	MWRCHK 005375
HEADQ 037263	LABEXP 007176	L\$PRT 002112 G	MCYLOC 010707	MWRITE 005406
HEADV 037305	LABHCF 007233	L\$REPP 002062 G	MCYLUP 005550	MWRSET 005503
HEADW 014126	LABHCR 007247	L\$REV 002010 G	MDATCP 005432	MWPTAB 011043
HFIN 003174	LABIN 007153	L\$SOFT 036774 G	MDCRC 010355	M4OHDR 005467
HFINU 003176	LABMID 007161	L\$SPC 002056 G	MDHEDR 002000 G	NEWCYL 003106
HFOUT 003200	LABOCF 007207	L\$SPCP 002020 G	MDLT 010402	NOCLR = 000010
HFOUTU 003202	LABOCR 007221	L\$SPTP 002024 G	MDRDY 010322	NOCTLR 007635
HICYL = 020000	LABOUT 007170	L\$STA 002030 G	MDRERR 010444	NOERCT 003451
HILIM = 000004	LAB1 006252	L\$SW 014120 G	MDRRES 006326	NOIRPT = 000002
HILIMQ 037242	LAB2 006265	L\$TEST 002114 G	MDRVST 010472	NOOP = 000100
HILIMW 014124	LIMVAL 037234	L\$TIML 002014 G	MDSERR 010455	NOPIR 006166
HLMTW 002306	LOCERR 003450	L\$UNIT 002012 G	MERRS 011112	NOTRDY 007673
HNFERR= 010000	LOCYL = 040000	L.BA 003042	MEXERS 011055	NOTST1 007750
HOE = 100000 G	LOE = 040000 G	L.CS 003040	MFLERR 010634	NOTST4 010131
HOSTAT= 000020	LOLIM = 000002	L.DA 003044	MFMTER 006063	NTST1A 010036
HPTCOD 014100 G	LOLIMQ 037215	L.MP 003046	MFOLWR 005620	NTST4A 010217
HRDPRM 036616 G	LOLIMW 014122	L10000 012332	MFWDSK 005671	NXMERR= 020000
HRDWTS 026340 G	LOT = 000010 G	L10001 012400	MFWSKO 005722	NXTHL 002312
HRIN 003204	L\$ACP 002110 G	L10002 012446	MGTSTA 005420	NXTPAS 014402
HRINU 003206	L\$APT 002036 G	L10003 012516	MHCERR 010554	OBUFF 004472
HROUT 003210	L\$AUT 002070 G	L10004 012566	MHCRC 010345	OFIN 003144
HROUTU 003212	L\$AUTO 015132 G	L10005 013450	MHDERR 010617	OFINU 003146
HSMSK = 000100	L\$CCP 002106 G	L10006 013520	MHDRCP 005451	OFMID 003150
HSSTAT= 000100	L\$CLEA 015470 G	L10007 013614	MHFCRC 010414	OFMIDU 003152
IBE = 010000 G	L\$CO 002032 G	L10010 013660	MHNF 010366	OFOUT 003154
IBUFF 004072	L\$DEPO 002011 G	L10011 014070	MININC 003462	OFOUTU 003156
IDU = 000040 G	L\$DESC 002122 G	L10013 014116	MINOUT 005577	OLDCYL 003104
IER = 020000 G	L\$DESP 002076 G	L10014 014134	MISWI = 000000	ONSWAP 021202
INITCO 014156 G	L\$DEVP 002060 G	L10015 015130	MISWIW 014120	OPFLAG 003010
INOUTS= 000020	L\$DISP 014136 G	L10016 015466	MISTEW= 100000	OPTERR= 002000
INTEBL= 000100	L\$DLY 002116 G	L10017 015614	MNRST 010714	OPMSGS 002230
INTHLR 016150	L\$DTP 002040 G	L10020 015620	MNEERR 010662	OPROO4 007415
ISR = 000100 G	L\$DTYP 002034 G	L10021 016146	MNOCLR 006413	OPR1A 007366
IXE = 004000 G	L\$DU 015616 G	L10022 016226	MNOINT 006344	OPR1B 007372
I\$AU = 000041	L\$DUT 002072 G	L10023 030274	MOPER 005517	OPR12 007347
I\$AUTO= 000041	L\$DVTY 002216 G	L10024 031012	MOPERR 010607	ORIN 003160
I\$CLN = 000041	L\$E <sup>c</sup> 002052 G	L10025 031226	MORECE 003020	ORINU 003162
I\$DU = 000041	L\$ENVI 002044 G	L10026 031152	MOUTIN 005560	ORMID 003164
I\$HRD = 000041	L\$ETP 002102 G	L10027 032060	MPNAM 006245	ORMIDU 003164

OROUT	003170	P2T09E	006633	SPDSTA	004000	TINEST	177777	T33018	034640
OROUTU	003172	P2T10E	006636	SPTCOD	014116 G	TINSO	000000	T33658	035422
OLYINS	000040	P2T11E	006651	SRTMES	007075	TINS1	000005	T34008	035556
O1APTS	000000	P2T12E	006664	SSINDX	003006	TINS2	000002	T34018	036002
O1AU	000000	P2T13E	006676	STAMES	007527	T1PTNU	000000	T34658	036614
O1BGNR	000000	P2T14E	006712	STAMSK	000007	T1SAVL	177777	T4	031230 G
O1BGNS	000001	P2T15E	006733	STATE2	011146	T1SEGL	177777	T5	032062 G
O1DU	000001	P2T16E	006756	STATE3	011156	T1SEKO	010000	T5.1	032414
O1ERRT	000000	P2T17E	006777	STATES	011166	T1SUBN	000001	T6	033172 G
O1GNSW	000001	P2T18E	007031	STOSTA	010000	T1TAGL	177777	T6.1	033212
O1POIN	000001	P2T19E	007053	SUBSTK	002410	T1TAGN	010042	T7	034210 G
O1SETU	000000	RDALND	023154	SVCBGL	000001	T1TEMP	000000	T7.1	034640
PART2	000001 G	RDDATA	000114	SVCGBL	000000	T1TEST	000010	T8	035424 G
PASCNT	003236	RDHEAD	000110	SVCINS	000000	T1TSTM	177777	T8.1	036002
PASNEW	014410	RDNOMR	000116	SVCSUB	000001	T1TSTS	000001	UAM	000200 G
PASNUM	003444	RDYCHK	020642	SVCTAG	000000	T1TAUT	010016	ULOAD	000010
PAT1	005072	RDYMAI	022420	SVCTST	000001	T1TCLE	010017	UNDTST	007402
PAT10	005346	READRL	016370	SWAPHD	021142	T1TUDU	010020	UNXERR	006454
PAT2	005074	RELDWT	040000	SLSYM	010000	T1THAR	010040	VALDES	007117
PAT3	005134	RESE3	011123	TAG	003470	T1THW	010013	VCMRST	006433
PAT4	005174	RESE4	011127	TBLSTR	003030	T1THINI	010015	VCSTAT	001000
PAT5	005234	RESE5	011134	TBT	002550	T1THMSG	010011	VECMG	036714
PAT6	005242	RESE6	011141	TCERR	007614	T1THPRO	010012	VECT	000002
PAT7	005302	RESPAR	003066	TEMP	003464	T1THSEG	010000	VERHDR	022024
PAT8	005304	RESTAR	014352	TEMPO	003122	T1THSOF	010041	VERPOS	023032
PAT9	005344	RESTBL	002324	TEMP1	003124	T1THSRV	010022	WAITIN	016422
PM658	020604	REVSKO	001000	TEMP2	003126	T1THSUB	010037	WCMASK	017777
PNT	001000 G	REVSKS	000200	TEMP3	003130	T1THSW	010014	WCRNG	160000
POSHDS	020276	RLBA	000002	TEMP4	003132	T1THTES	010036	WDESTA	100000
POSHDO	022374	RLBAS	003032	TEMP5	003134	T.BA	003052	WGESTA	002000
POSHSB	022370	RLCS	000000	TEMP6	003136	T.CS	003050	WLSTAT	020000
POSHW1	022362	RLCSR	000000	TEMP7	003140	T.DA	003054	WRTSWI	003026
PRI	002000 G	RLDA	000004	TEMP8	003142	T.DRIV	002302	WTDATA	000112
PRIOP	000004	RLDRV	003036	TIME	015622	T.MP	003056	XDELAY	003456
PRI00	000000 G	RLMP	000006	TIM.US	003466	T.STAT	003064	XRDND	021370
PRI01	000040 G	RLVEC	003034	TOSLOW	000001	T1	026340 G	XRDNDJ	021360
PRI02	000100 G	RORWOP	020000	TRPFLC	003452	T2	030276 G	XRDNDG	021374
PRI03	000140 G	RPTOP	025060	TRPHAN	016142	T25TBL	002434	XREAD	024212
PRI04	000200 G	RPTREM	026054	TSTINT	016560	T25TB2	002462	XREADG	024220
PRI05	000240 G	RPTRES	025646	TSTLAB	006471	T3	031014 G	XSEEK	017524
PRI06	000300 G	RSTR	014270	TYPDR	000006	T3.1	031112	XSEEKT	017514
PRI07	000340 G	SAMSK	000077	T1ARGC	000007	T3068	031060	XSEEK1	017530
PSETNM	003446	SBSFIL	003502	T1CODE	005052	T30658	031226	XTIME	015766
PWCON	014660	SECQ	037144	T1ERRN	003247	T3078	031112	XWRITE	024152
PWRFLG	003454	SECWD	007451	T1EXCP	000000	T3108	031120	XWRITT	024142
P2T03E	006477	SEEK	000106	T1FLAG	000040	T31008	032206	XWRIT1	024156
P2T04E	006515	SEEKOP	010000	T1GMAN	000000	T31018	032414	X1ALWA	000000
P2T05E	006535	SEQMES	007504	T1HILI	000377	T31658	033170	X1FALS	000040
P2T06E	006555	SETDON	014436	T1LAST	000001	T32048	034064	X1OFFS	000400
P2T07E	006575	SFTRM	036772 G	T1LOLI	000001	T32658	034206	X1TRUE	000020
P2T08E	006613	SGNMD	007440	T1LSYM	010000	T33TBL	002510	YDELAY	003460
		SKTMS	007063	T1LTNO	000010	T33008	034356		

. ABS. 037400 000  
 000000 001  
 ERRORS DETECTED: 0

VIRTJAI MEMORY USED: 29696 WORDS ( 116 PAGES)

CZRLNB0 RL01-02 DRIVE TEST 3 MACRO V04.00 20 JAN 83 14:40:57 PAGE 20 7  
SYMBOL TABLE

DYNAMIC MEMORY AVAILABLE FOR 70 PAGES  
CZRLNB.BIN,CZRLNB.LST/C-[20.0]SVC34R.MLB.[20.29]CZRLNB.MAC















HDWD	4 709#	11-189	11 192	11-227															
HDWRD1	4 387#	5 245	10 659	10-734															
HDWRD2	4-388#	10-556																	
HDWRD3	4-389#																		
HEAD	4-49#	20-44	20-44	20 44															
HEADLM	4-58#	10-460	10-467	13-8															
HEADQ	20-42	20-65#																	
HEADV	20-44	20-66#																	
HEADW	5-321#	10-462	13-10																
HFIN	4-431#	12-163	12-190																
HFINU	4-432#	12-164																	
HFOUT	4-433#	12-141	12-190																
HFOUTU	4-434#	12-142	12-144																
MICYL	4-59#	6-93																	
MILIM	4-48#	20-41	20-41	20-41															
MILIMQ	20-39	20-64#																	
MILIMW	5-320#	6-95*	16-21	16-22	16-23	16-42	16-47	18-24	18-25	18-26	18-27	18-58	19-22	19-23					
	19-24	19-52																	
MLMTW	4 195#	6-77*	6-85*	6-95	10-194	10-196	10-198	10-220	11-8	12-129	12-167	12-183	13-13	18-53					
	19-47																		
MNFERR	4-111#	5-155																	
HOE	4-36#																		
HOSAT	4-145#	10-83																	
HPTCOO	5-298#																		
HRDPRM	20-2#																		
HRDWTS	12-3#																		
HRIN	4-435#	12-165	12-191																
HRINU	4-436#	12-166																	
HROUT	4-437#	12-143	12-191																
HROUTU	4-438#																		
HSMSK	4-123#	11-42																	
HSSTAT	4-147#	10-663																	
I#AU	2-8#																		
I#AUTO	2-8#	7-11#	7-38#																
I#CLN	2-8#	8-5#	8-23#																
I#DU	2-8#	8 25#	8-27#																
I#HRD	20-3#	20-11#																	
I#INIT	2-8#	6-4#	6-131#																
I#MOD	2-8#	4-7	4-7#	4-9	4-9#	4-34	4-34#	4-162	4-162#	4-168	4-168#	4-578	4-578#	4-589					
	4-589#	4-814	4-814#	5-2	5-2#	5-287	5-287#	5-298	5-298#	5-307	5-307#	5-309	5-309#	5-325					
	5-325#	5-327	5-327#	5-334	5-334#	6-3	6-3#	6-132	6-132#	8-4	8-4#	8-29	8-29#	9-3					
	9-3#	11-243	11-243#	12-3	12-3#	19-166	19-166#	20-2	20-2#	20-19	20-19#	20-22	20-22#	20-73					
	20-73#																		
I#MSG	2-8#	5-47#	5-59#	5-61#	5-73#	5-75#	5-87#	5-89#	5-102#	5-104#	5-117#	5-119#	5-222#	5 224#					
	5-236#	5-238#	5-258#	5-260#	5-272#	5-273#	5-286#												
I#PROT	2-8#	5-290#																	
I#PTAB	2-8#																		
I#PWR	2-8#																		
I#RPT	2-8#																		
I#SEG	2 8#	10-349#	10-396#	12-7	13-2	14-2	14-21	15-2	16-2	16-60	17-2	17-9	18-2	18 75					
	19-2	19-64																	
I#SETU	2-8#																		
I#SFT	20-23#	20-50#																	
I#SRV	2-8#	9-45#	9-51#	9-53#	9-67#														
I#SUB	2-8#	12-7	13-2	14-2	14-21	14-21#	14-31	14-31#	14-31#	15-2	16 2	16-60	16-60#	16-139					
	16-139#	16-139#	17-2	17-9	17-9#	17-69	17-97	17-97#	17-97#	18-2	18 75	18-75#	18-155	18 155#					















011

RLMP	4-1030	5-144	10 21	10-557	10 558	10 796	11 580	15 320	15 510	17 570				
RLVEC	4 3760	6-96	8-16											
RGPWOP	4-880	11-153	11-161	11-164	11-190									
RPTOP	5 52	5-64	5 78	5 92	5 107	5 127	5 227	5 242	5 263	5 278	11 1430			
RPTREM	5-56	5-70	5-84	5-99	5 114	5-190	5 233	5 254	5 269	11 2250				
RPTRES	5-55	5-69	5 83	5-98	5-113	5-188	5 232	5 253	5 268	11 1990				
RSTR	6-290	6-44												
S&LSM	2-80	5-590	5-730	5-870	5-1020	5 1170	5 2220	5 2360	5 2580	5 2720	5 2860	5-3060	5 3240	6 1310
	7-380	8-230	8-270	9-510	9-670	10-340	10-349	10 3490	12 1950	13 930	14 310	14 460	15 810	16-1390
	16 1530	17-970	17-1100	18-1550	18-1700	19 1510	19 1650	20-110	20 500					
SAMSK	4 1220													
SBSFIL	4 4760	10-4940	11 109	13 63	13 76									
SECO	20-27	20-540												
SECND	4-7100	11 192												
SEEK	4-660	10-229												
SEEKOP	4-870	11-153	11 158	11-187										
SEQMES	4-7140	11-147												
SETDON	6-41	6-53	6-580											
SFTPRM	20 220													
SGNMD	4 7080	11-189												
SKMES	4 6690	12-185												
SPOSTA	4 1510	10-85												
SPTCOD	5-3090													
SRTMES	4-6700	15-77												
SSINDX	4-3620	6-170	10-65	10-690	10-1620	10-180	10-1840	10-2650	10-338	10-3420	10-3970	10-412	10 4160	10-4450
	10-504	10-5080	10-5710	10-587	10-5910	10-6440	10-674	10 6780	10 7090	10-725	10-7290	10-7410	10-756	10-7600
	10-7700	10-783	10-7870	10-8330	10-891	10-8950	10-9150	11-16	11-200	11-880	11 144	11 150		
STAMES	4-7150	5-193												
STAMSK	4-1430	10 120												
STATE2	4-7700													
STATE3	4-7710													
STATES	4-7720													
STOSTA	4-1520													
SUBSTK	4-2350	10-670	10-680	10-1820	10-1830	10-3400	10-3410	10-4140	10-4150	10 5060	10-5070	10 5390	10 5900	10 6760
	10-6770	10-7270	10-7280	10 7580	10-7590	10-7850	10-7860	10 8930	10-8940	11-180	11 190	11 148		
SVCBGL	2-110													
SVCGBL	2-80	4-7	4-7	4-7	4-8	4-8	4-8	4-8	4-8	4 8	4 8	4 8	4 8	4 8
	4 8	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4 8	4 8	4 8	4 8	4 8
	4 8	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4 8	4 8	4 8	4 8	4 8
	4 8	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4 8	4 8	4 8	4 8	4 8
	4-8	4 8	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4 8	4 8	4 8
	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4 8	4 8	4 8
	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4 8	4 8	4 8
	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4 8	4 8	4 8
	4-34	4-168	4-168	4-168	4-589	4-589	4-589	5-2	5-2	5-2	5 47	5 47	5 47	5-61
	5-61	5-61	5-75	5-75	5-75	5-89	5-89	5-89	5-104	5-104	5-104	5-119	5 119	5 119
	5-224	5-224	5-224	5-238	5-238	5-238	5-260	5-260	5-260	5-273	5-273	5 273	5-290	5 290
	5-290	5-298	5-298	5-298	5-299	5-299	5-299	5-309	5-309	5-309	5-310	5-310	5 310	5 327
	5-327	5-327	5-332	5-332	5-332	6-3	6-3	6-3	6-4	6-4	6-4	7-11	7 11	7-11
	8-4	8-4	8-4	8-5	8-5	8-5	8-25	8-25	8-25	9-3	9-3	9-3	12 3	12 3
	12-3	20-2	20-2	20-2	20-3	20-3	20-3	20-22	20-22	20-22	20-23	20-23	20 23	20 75
	20-75	20-75	20-750											
SVCINS	2-80	2-120	4-8	4 8	4-8	4-8	4-8	4-8	4-8	4 8	4-8	4 8	4-8	4 8
	4-8	4 8	4-8	4 8	4-8	4-8	4-8	4-8	4 8	4-8	4 8	4 8	4 8	4-8
	4 8	4 8	4-8	4 8	4-8	4-8	4-8	4-8	4-8	4 8	4 8	4 8	4 8	4 8





17 9	17-9	17-34	17-34	17-34	17-34	17-34	17-34	17 34	17-34	17 34	17 34	17 34	17 34	17-34
17 34	17-34	17-34	17-34	17-34	17-34	17-34	17-34	17 34	17-34	17 34	17 34	17 34	17 34	17-34
17 41	17-41	17-41	17-41	17-41	17-41	17-41	17-41	17-41	17-41	17-44	17 44	17 44	17 44	17-41
17 44	17-44	17-44	17-44	17-44	17-44	17-44	17-44	17 44	17-44	17 44	17 44	17 44	17 44	17-44
17 44	17-44	17-45	17-45	17-45	17-45	17-68	17-68	17-68	17-68	17-68	17-68	17-68	17-68	17-68
17-69	17-69	17-69	17-69	17-69	17-75	17-75	17-75	17-75	17-75	17-75	17-75	17-75	17-75	17-68
17 79	17-79	17-79	17-79	17-79	17-79	17-86	17-86	17 86	17-86	17-86	17-86	17-86	17-86	17-79
17-97	17-97	17-99	17-99	17-99	17-99	17-99	17-99	17-99	17-99	17-99	17-99	17-99	17-99	17-86
17-99	17-99	17-99	17-99	17-99	17-99	17-99	17-99	17-99	17-99	17-99	17-99	17-99	17-99	17-99
18 75	18-75	18-155	18-155	18-156	18-156	18-156	18-156	18-170	18-170	19-64	19-64	19-64	19-64	17-110
19 152	19-152	19-152	19-152	19-165	19-165	20-3	20-3	20-4	20-4	20-4	20-4	20-4	20-4	17-110
20-5	20-5	20-5	20-5	20-5	20-5	20-5	20-5	20-6	20-6	20-6	20-6	20-6	20-6	19-151
20-6	20-6	20-7	20-7	20-7	20-7	20-7	20-7	20-7	20-7	20-7	20-7	20-7	20-7	20-4
20-8	20-8	20-8	20-8	20-9	20-9	20-9	20-9	20-9	20-9	20-9	20-9	20-9	20-9	20-6
20-11	20-11	20-23	20-23	20-26	20-26	20-26	20-26	20-26	20-26	20-27	20-27	20-27	20-27	20-8
20-27	20-27	20-33	20-33	20-33	20-33	20-33	20-33	20-36	20-36	20-36	20-36	20-36	20-36	20-8
20-37	20-37	20-38	20-38	20-38	20-38	20-38	20-38	20-38	20-38	20-38	20-38	20-38	20-38	20-9
20-39	20-39	20-39	20-39	20-40	20-40	20-40	20-40	20-41	20-41	20-41	20-41	20-41	20-41	20-9
20 41	20-41	20-42	20-42	20-42	20-42	20-42	20-42	20-41	20-41	20-41	20-41	20-41	20-41	20-27
20-44	20-44	20-44	20-44	20-44	20-44	20-46	20-46	20-43	20-43	20-43	20-43	20-43	20-43	20-27
20-46	20-46	20-48	20-48	20-48	20-48	20-48	20-48	20-43	20-43	20-44	20-44	20-44	20-44	20-36
20-75	20 75	20-75	20-75	20-75	20-75	20-48	20-48	20-46	20-46	20-46	20-46	20-46	20-46	20-39
SVCSUB	2-8#	2-10#	14-21	16-60	17-9	18-75	19-64	20-48	20-48	20-48	20-48	20-48	20-48	20-41
SVCTAG	2-8#	2-13#	5-59	5-59	5-59	5-73	5-73	5-87	5-87	5-87	5-87	5-102	5 102	20 50
	5-117	5-117	5-117	5-222	5-222	5-222	5-236	5-236	5-236	5-258	5-258	5-258	5-272	20 50
	5-272	5-286	5-286	5-286	5-306	5-306	5-324	5-324	5-324	6-131	6-131	6-131	6 131	20 50
	7-38	7-38	8-23	8-23	8-23	8-27	8-27	8-27	9-51	9-51	9-51	9-67	9 67	20 50
	10-396	10-396	10-396	12-195	12-195	12-195	13-93	13-93	13-93	14-31	14 31	14 31	14-46	20 50
	14-46	15-81	15-81	15-81	16-139	16-139	16-153	16-153	16-153	17-97	17-97	17-97	17 97	20 50
	17-110	17-110	18-155	18-155	18-155	18-170	18-170	19-151	19-151	19 151	19-151	19-165	19-165	20 50
	20-11	20-11	20-11	20-50	20-50	20-50	20-50	20-48	20-48	20-48	20-48	20-48	20-48	20 50
SVCTST	2-8#	2-9#	12-7	13-2	14-2	15-2	16-2	17 2	18-2	19-2				
SWAPHD	10 467#	14-37	16-147	18-163	19-159									
T##AUT	7-11#	7-38												
T##CLE	8-5#	8-23												
T##DU	8-25#	8-27												
T##HAR	20-3	20-3#	20-11											
T##HM	5-299	5-299#	5-306											
T##INI	6-4#	6-131												
T##MSG	5-47#	5-59	5-61#	5-73	5-75#	5-87	5-89#	5-102	5 104#	5 117	5 119#	5 222	5 224#	5 236
	5-238#	5-258	5-260#	5-272	5-273#	5-286								
T##PRO	5-290#													
T##SEG	10-349	10-349#	10-396	10-396#										
T##SOF	20-23	20-23#	20-50											
T##SRV	9-45#	9-51	9-53#	9-67										
T##SUB	14-21#	14-31	16-60#	16-139	17-9#	17-69	17-97	18-75#	18-155	19-64#	19 151			
T##SM	5-310	5-310#	5-324											
T##TES	12-7#	12-15	13-2#	13-12	13-93	14-2#	14-32	14-46	15-2#	15-10	15-81	16-2#	16-140	16 153
	17-2#	17-45	17-110	18-2#	18-156	18-170	19-2#	19-152	19-165					
T##ARGC	4 8	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4-8#	4 8#	4 8#
	4-8#	4-8#	4-8#	5-160	5-160	5-160	5-160	5-160	5-160#	5-160#	5-160#	5-160#	5 193	5 193
	5-193	5-193	5-193#	5-193#	5-193#	5-207	5-207	5-207	5-207	5-207	5-207	5-207	5-207	5-207#
	5-207#	5-207#	5-207#	5-207#	5-207#	5-216	5-216	5-216	5-216	5-216	5-216	5-216	5-216#	5-216#
	5-216#	5-216#	5-216#	5-279	5-279	5-279	5-279	5-279	5-279	5-279#	5-279#	5-279#	5-279#	5-279#
	5-280	5-280	5-280	5-280	5-280	5-280	5-280	5-280	5-280	5-280#	5-280#	5-280#	5-280#	5-280#
	5-280#	5-280#	5-280#	5-282	5-282	5-282	5-282	5-282	5-282	5-282	5-282	5-282#	5-282#	5-282#

	5 282#	5-282#	5-282#	5-282#	6-124	6-124	6 124	6-124#	6-124#	6-125	6 125	6 125	6 125	6-125
	6 125	6-125#	6-125#	6-125#	6-125#	6-125#	6-126	6-126	6 126#	7-19	7 19	7 19	7 19#	6-125
	7-20	7-20	7-20	7-20	7-20	7-20	7-20#	7-20#	7-20#	7-20#	7-20#	7 22	7 22	7-22#
	7 31	7 31	7-31	7-31#	7-31#	7-33	7-33	7-33	7 33	7-33	7-33	7 33#	7 33#	7-33#
	7-33#	7-33#	7-35	7-35	7-35#	10-10	10-10	10-10	10-10	10-10#	10-10#	10-10#	10-11	10-11
	10-11	10-11	10-11	10-11	10-11#	10-11#	10-11#	10-11#	10-11#	10-12	10-12	10-12#	10-491	10-491
	10-491	10-491#	10-491#	10-492	10-492	10-492	10-492	10-492	10-492	10-492#	10-492#	10-492#	10-492#	10-492#
	10-493	10-493	10-493#	10-914	10-914	10-914	10-914	10-914	10-914	10-914#	10-914#	10-914#	10-914#	10-914#
	11-147	11-147	11-147	11-147#	11-147#	11-148	11-148	11-148	11-148#	11-148#	11-152	11-152	11-152	11-152
	11-152#	11-152#	11-152#	11-165	11-165	11-165	11-165	11-165#	11-165#	11-165#	11-181	11-181	11-181	11-181#
	11-181#	11-185	11-185	11-185	11-185#	11-185#	11-189	11-189	11-189	11-189	11-189	11-189	11-189	11-189
	11-189	11-189	11-189#	11-189#	11-189#	11-189#	11-189#	11-189#	11-189#	11-189#	11-189#	11-192	11-192	11-192
	11-192	11-192	11-192	11-192	11-192	11-192#	11-192#	11-192#	11-192#	11-192#	11-192#	11-192#	11-204	11-204
	11-204	11-204	11-204#	11-204#	11-204#	11-213	11-213	11-213	11-213	11-213#	11-213#	11-213#	11-214	11-214
	11-214	11-214	11-214#	11-214#	11-214#	11-217	11-217	11-217	11-217	11-217#	11-217#	11-217#	11-225	11-225
	11-225	11-225	11-225	11-225	11-225#	11-225#	11-225#	11-225#	11-225#	11-227	11-227	11-227	11-227	11-227
	11-227	11-227	11-227	11-227#	11-227#	11-227#	11-227#	11-227#	11-227#	11-227#	11-228	11-228	11-228	11-228
	11-228	11-228	11-228	11-228#	11-228#	11-228#	11-228#	11-228#	11-228#	11-229	11-229	11-229	11-229	11-229
	11 229	11-229	11-229	11-229	11-229#	11-229#	11-229#	11-229#	11-229#	11-229#	11-229#	11-229#	12 12	12 12
	12-12	12-12#	12-12#	12-14	12-14	12-14	12-14#	12-14#	12-185	12-185	12-185	12-185	12 185#	12 185#
	12-185#	12-186	12-186	12-186	12-186	12-186	12-186#	12-186#	12-186#	12-186#	12-186#	12-186#	12-187	12-187
	12-187	12-187	12-187	12-187	12-187#	12-187#	12-187#	12-187#	12-187#	12-188	12-188	12-188	12-188	12 188
	12-188	12-188	12-188#	12-188#	12-188#	12-188#	12-188#	12-188#	12-189	12-189	12-189	12-189	12-189	12-189
	12-189	12-189#	12-189#	12-189#	12-189#	12-189#	12-189#	12-189#	12-190	12-190	12-190	12-190	12-190	12-190#
	12-190#	12-190#	12-190#	12-190#	12-191	12-191	12-191	12-191	12-191	12-191#	12-191#	12-191#	12-191#	12 191#
	12-191#	12-192	12-192	12-192	12-192	12-192	12-192#	12-192#	12-192#	12-192#	12-193	12-193	12-193	12-193
	12-193	12-193#	12-193#	12-193#	12-193#	15-7	15-7	15-7	15-7#	15-7#	15-9	15-9	15-9	15 9#
	15-9#	15-77	15-77	15-77	15-77	15-77#	15-77#	15-77#	15-78	15-78	15-78	15-78	15 78	15 78
	15-78#	15-78#	15-78#	15-78#	15-78#	15-79	15-79	15-79	15-79	15-79	15-79	15-79	15 79#	15 79#
	15-79#	15-79#	15-79#	15-79#	17-34	17-34	17-34	17-34	17-34	17-34	17-34	17-34	17-34#	17 34#
	17-34#	17-34#	17-34#	17-34#	17-34#	17-41	17-41	17-41	17-41#	17-41#	17-44	17-44	17 44	17 44
	17-44	17-44	17-44#	17-44#	17-44#	17-44#	17-44#	17-44#	17-99	17-99	17-99	17-99	17-99	17 99
	17-99	17-99#	17-99#	17-99#	17-99#	17-99#	17-99#	17-99#	17-99#	17-99#	17-99#	17-99#	17-99#	17 99
T\$CODE	20-4	20-4	20-4	20-4#	20-4#	20-4#	20-5	20-5	20-5	20-5#	20-5#	20 5#	20-6	20-6
	20-6	20-6#	20-6#	20-6#	20-7	20-7	20-7	20-7#	20-7#	20-7#	20-8	20-8	20 8	20-8#
	20-8#	20-8#	20-9	20-9	20-9	20-9#	20-9#	20-9#	20-26	20-26	20-26	20-26#	20-26#	20 26#
	20-27	20-27	20-27	20-27#	20-27#	20-27#	20-33	20-33	20-33	20-33#	20-33#	20-33#	20 36	20-36
	20-36	20-36#	20-36#	20-36#	20-37	20-37	20-37	20-37	20-37	20-37	20-37#	20-37#	20-37#	20-37#
	20-38	20-38	20-38	20-38#	20-38#	20-38#	20-39	20-39	20-39	20-39#	20-39#	20-39#	20-40	20 40
	20-40	20-40	20-40	20-40	20-40#	20-40#	20-40#	20-40#	20-41	20-41	20-41	20-41#	20-41#	20-41#
	20-42	20-42	20-42	20-42#	20-42#	20-42#	20-43	20-43	20-43	20-43	20-43	20-43	20 43#	20 43#
	20-43#	20 43#	20-44	20-44	20-44	20-44#	20-44#	20-44#	20-46	20-46	20 46	20-46#	20-46#	20 46#
	20-48	20-48	20-48	20-48#	20-48#	20-48#								
T\$ERRN	2-8#	10-97	10-97#	10-134	10-134#	10-147	10-147#	10-153	10-153#	10-258	10-258#	10-263	10-263#	10 372
	10-372#	10-388	10-388#	10-431	10-431#	10-442	10-442#	10-539	10-539#	10-550	10-550#	10-554	10-554#	10-562
	10-562#	10-611	10-611#	10-623	10-623#	10-630	10-630#	10-692	10-692#	10-702	10-702#	10-706	10 706#	10-767
	10-767#	10-820	10-820#	10-825	10-825#	10-929	10-929#	11-52	11-52#	11-67	11-67#	11-73	11 73#	11 82
	11-82#	11-86	11-86#	12-53	12-53#	12-57	12-57#	13-75	13-75#	13-82	13-82#	13 85	13-85#	15 41
	15-41#	15-45	15-45#	15-61	15-61#	15-65	15-65#	17-68	17-68#	17-75	17-75#	17-79	17-79#	17 86
	17-86#													
T\$EXCP	20-5	20-5#	20-6	20-6#	20-7	20 7#	20-9	20-9#	20-38	20-38#	20-41	20-41#	20-44	20 44#
	20-46	20-46#	20-48	20-48#										
T\$FLAG	13-12	13-12	13-12#	13-12#	14-32	14-32#	14-32#	15-10	15-10	15-10#	15-10#	16-140	16-140#	16 140#
	17-45	17-45	17-45#	17-45#	17-69	17-69	17-69#	17-69#	18-156	18-156#	18-156#	19-152	19-152#	19 152#
T\$GMAN	2-8#													
T\$MILI	20-5	20 5#	20 6	20 6#	20-7	20-7#	20-9	20 9#	20-38	20-38#	20-41	20-41#	20 44	20 44#

T\$LAST	20-46	20-460	20-48	20-480										
T\$LOLI	2-80	20-750												
T\$SYM	20-5	20-50	20-6	20-60	20-7	20-70	20-9	20-90	20-38	20-380	20-41	20-410	20-44	20-440
T\$TNO	20-46	20-460	20-48	20-480										
T\$NECT	2-8	2-80	5-59	5-73	5-87	5-102	5-117	5-222	5-236	5-258	5-272	5-286	5-306	5-324
	6-131	7-38	8-23	8-27	9-51	9-67	12-195	13-93	14-31	14-46	15-81	16-139	16-153	17-97
	17-110	18-155	18-170	19-15	19-165	20-11	20-50							
T\$PTNU	2-80	4-7	4-7	4-7	4-9	4-9	4-9	4-90	4-34	4-34	4-340	4-162	4-162	4-162
T\$SAVL	4-1620	4-168	4-168	4-1680	4-578	4-578	4-578	4-5780	4-589	4-589	4-5890	4-814	4-814	4-814
T\$SEGL	4-8140	5-2	5-2	5-2	5-47	5-47	5-470	5-59	5-59	5-59	5-590	5-61	5-61	5-610
T\$SEKO	5-73	5-73	5-73	5-730	5-75	5-75	5-750	5-87	5-87	5-87	5-870	5-89	5-89	5-890
T\$SUBN	5-102	5-102	5-102	5-1020	5-104	5-104	5-1040	5-117	5-117	5-117	5-1170	5-119	5-119	5-1190
T\$TAGL	5-222	5-222	5-222	5-2220	5-224	5-224	5-2240	5-236	5-236	5-236	5-2360	5-238	5-238	5-2380
T\$TAGN	5-258	5-258	5-258	5-2580	5-260	5-260	5-2600	5-272	5-272	5-272	5-2720	5-273	5-273	5-2730
	5-286	5-286	5-286	5-2860	5-287	5-287	5-2870	5-290	5-290	5-290	5-2900	5-294	5-294	5-2940
	5-2940	5-298	5-298	5-2980	5-299	5-299	5-2990	5-306	5-306	5-306	5-3060	5-307	5-307	5-3070
	5-3070	5-309	5-309	5-3090	5-310	5-310	5-3100	5-324	5-324	5-324	5-3240	5-325	5-325	5-3250
	5-3250	5-327	5-327	5-3270	5-334	5-334	5-3340	6-3	6-3	6-3	6-30	6-4	6-4	6-40
	6-131	6-131	6-131	6-1310	6-132	6-132	6-1320	7-11	7-11	7-11	7-110	7-38	7-38	7-380
	7-380	8-4	8-4	8-40	8-5	8-5	8-50	8-23	8-23	8-23	8-230	8-25	8-25	8-250
	8-27	8-27	8-27	8-270	8-29	8-29	8-290	9-3	9-3	9-3	9-30	9-45	9-45	9-450
	9-51	9-51	9-51	9-510	9-53	9-53	9-530	9-67	9-67	9-67	9-670	10-349	10-349	10-3490
	10-396	10-396	10-396	10-3960	11-243	11-243	11-2430	12-3	12-3	12-3	12-30	12-7	12-7	12-70
	12-195	12-195	12-195	12-1950	13-2	13-2	13-20	13-93	13-93	13-93	13-930	14-2	14-2	14-20
	14-21	14-21	14-210	14-31	14-31	14-31	14-310	14-46	14-46	14-46	14-460	15-2	15-2	15-20
	15-81	15-81	15-810	16-2	16-2	16-2	16-20	16-60	16-60	16-600	16-139	16-139	16-139	16-1390
	16-153	16-153	16-1530	17-2	17-2	17-2	17-20	17-9	17-9	17-90	17-97	17-97	17-97	17-970
	17-110	17-110	17-1100	18-2	18-2	18-2	18-20	18-75	18-75	18-750	18-155	18-155	18-155	18-1550
	18-170	18-170	18-1700	19-2	19-2	19-2	19-20	19-64	19-64	19-640	19-151	19-151	19-151	19-1510
	19-165	19-165	19-1650	19-166	19-166	19-166	19-1660	20-2	20-2	20-2	20-20	20-3	20-3	20-30
	20-11	20-11	20-110	20-19	20-19	20-19	20-190	20-22	20-22	20-22	20-220	20-23	20-23	20-230
	20-37	20-40	20-43	20-50	20-50	20-50	20-500	20-73	20-73	20-73	20-730			
T\$NS0	4-70	4-9	4-340	4-162	4-1680	4-578	4-5890	4-814	5-20	5-287	5-2900	5-294	5-2980	5-307
	5-3090	5-325	5-3270	5-334	6-30	6-132	7-110	7-38	8-40	8-29	9-30	11-243	12-30	19-166
	20-20	20-19	20-220	20-73										
T\$NS1	5-470	5-59	5-610	5-73	5-750	5-87	5-890	5-102	5-1040	5-117	5-1190	5-222	5-2240	5-236
	5-2380	5-258	5-2600	5-272	5-2730	5-286	5-2990	5-306	5-3100	5-324	6-40	6-131	8-50	8-23
	8-250	8-27	9-450	9-51	9-530	9-67	10-3490	10-396	12-70	12-195	13-20	13-93	14-20	14-46
	15-20	15-81	16-20	16-153	17-20	17-110	18-20	18-170	19-20	19-165	20-30	20-11	20-230	20-37
	20-40	20-43	20-50											
T\$NS2	14-210	14-31	16-600	16-139	17-90	17-97	18-750	18-155	19-640	19-151				
T\$PTNU	2-80													
T\$SAVL	2-80													
T\$SEGL	2-80	10-349	10-349	10-3490	10-396	10-396	10-396	10-396	10-3960					
T\$SEKO	10-3490	10-396												
T\$SUBN	2-80	12-70	13-20	14-20	14-21	14-21	14-210	15-20	16-20	16-60	16-60	16-600	17-20	17-9
	17-9	17-90	18-20	18-75	18-75	18-750	19-20	19-64	19-64	19-640				
T\$TAGL	2-80													
T\$TAGN	2-80	5-47	5-47	5-470	5-61	5-61	5-610	5-75	5-75	5-750	5-89	5-89	5-890	5-104
	5-104	5-1040	5-119	5-119	5-1190	5-224	5-2240	5-238	5-238	5-2380	5-260	5-260	5-2600	5-2600
	5-273	5-273	5-2730	5-290	5-290	5-299	5-2990	5-310	5-310	5-3100	6-4	6-4	6-40	6-40
	6-40	7-11	7-110	8-5	8-5	8-50	8-25	8-25	8-250	9-45	9-45	9-450	9-450	9-53
	9-53	9-530	12-7	12-7	12-70	13-2	13-20	13-20	14-2	14-2	14-20	14-21	14-21	14-210
	15-2	15-2	15-20	16-2	16-2	16-20	16-60	16-60	16-600	17-2	17-2	17-20	17-9	17-90
	17-90	18-2	18-2	18-20	18-75	18-750	19-2	19-2	19-20	19-64	19-64	19-640	20-3	20-30











ENDSFT	1 568#	2-8#	20 50											
ENDSRV	1 580#	2-8#	9-51	9-67										
ENDSUB	1 596#	2-8#	14 31	16-139	17-97	18-155	19-151							
ENDSW	1 614#	2-8#	5 324											
ENDTST	1-624#	2-8#	12-195	13-93	14-46	15-81	16-153	17-110	18-170	19-165				
EQUALS	1-642#	2-8#	4-36											
ERRDF	1-714#	2 8#												
ERRHRD	1 718#	2-8#	10-97	10-134	10-147	10-153	10-258	10-263	10 372	10-388	10-431	10-442	10 539	10-550
	10-554	10-502	10-611	10-623	10-630	10-692	10-702	10 706	10-767	10-820	10-825	10-929	11-52	11-67
	11-73	11-82	11-86	12-53	12-57	13-75	13-82	13-85	15-41	15-45	15-61	15 65	17 68	17-75
	17-79	17-86												
ERROR	1-722#	2-8#												
ERRSF	1-726#	2-8#												
ERRSOF	1-730#	2-8#												
ERRTBL	1-734#	2-8#												
ESCAPE	1-744#	2-8#	14-32	16-140	18-156	19-152								
EXIT	1-771#	2-8#	13-12	15-10	17-45	17-69								
FEQUAL	1-810#	2-8#												
GETBYT	1-824#	2-8#												
GETPRI	1-834#	2-8#												
GETTIM	3-21#	12-65	15-56											
GETWOR	1-829#	2-8#												
GMANIA	1-879#	2-8#												
GMANID	1-848#	2-8#												
GMANIL	1-859#	2-8#												
GPHARD	1-868#	2-8#	6-63											
GPRMA	1-874#	2-8#	20-5	20-6										
GPRMD	1-903#	2-8#	20-7	20-9	20-38	20-41	20-44	20-46	20-48					
GPRML	1-934#	2-8#	20-4	20-8	20-26	20-27	20-33	20-36	20-39	20-42				
HEADER	1-954#	2-8#	4-8											
INLOOP	1-962#	2-8#	10-8	10-350	13-69									
IOSETU	1-966#	2-8#												
IOSTAR	1-974#	2-8#												
KT11	1-982#	2-8#												
LASTAD	1-147#	2-8#	20-75											
M#BYTE	1-D00#	2-8#	4-8	4-8	4-8	4-8#								
M#CHEC	1-E18#	2-8#	13-12	13-12#	15-10	15-10#	17-45	17-45#	17-69	17-69#				
M#CNT0	1-E82#	2-8#	20-4	20-4#	20 5	20-5#	20-6	20-6#	20-7	20-7#	20-8	20-8#	20-9	20-9#
	20-26	20-26#	20-27	20-27#	20-33	20-33#	20-36	20-36#	20-38	20-38#	20-39	20-39#	20-41	20-41#
	20-42	20-42#	20-44	20-44#	20-46	20-46#	20-48	20-48#						
M#COUN	1-D66#	2-8#	5-160	5-160	5-160	5-160#	5-193	5-193	5-193#	5-207	5-207	5-207	5-207	5-207
	5-207	5-207#	5-216	5-216	5-216	5-216#	5-279	5-279	5-279#	5-279	5-279	5-279#	5-280	5-280
	5-280	5-280	5-280	5-280	5-280	5-280#	5-282	5-282	5-282#	5-282	5-282	5-282#	5-282#	6-124
	6-124#	6-125	6-125	6-125	6-125	6-125#	6-126	6-126#	7-19	7-19#	7-20	7-20	7-20	7-20
	7-20#	7-22	7-22#	7-31	7-31#	7-33	7-33	7-33#	7-33	7-33#	7-35	7-35#	10-10	10-10
	10 10#	10-11	10-11	10-11	10-11	10-11#	10-12	10-12#	10-491	10-491#	10-492	10-492	10-492	10-492
	10-492#	10-493	10 493#	10-914	10-914	10-914	10-914	10-914#	11-147	11-147#	11-148	11-148#	11-152	11 152
	11-152#	11-165	11-165	11-165#	11-181	11-181#	11-185	11-185#	11-189	11-189#	11-189	11-189	11-189	11-189
	11-189	11-189	11-189#	11-192	11-192	11-192	11-192	11-192#	11-192	11-192#	11-204	11-204	11-204#	11-213
	11-213	11-213#	11-214	11-214	11-214#	11-217	11-217	11-217#	11-225	11-225	11-225	11-225	11-225#	11-227
	11-227	11-227	11-227	11-227	11-227	11 227#	11-228	11-228	11-228	11-228	11-228	11-228#	11-229	11-229
	11-229	11-229	11-229	11-229	11-229	11-229#	12-12	12-12#	12-14	12-14#	12-185	12-185	12-185#	12 186
	12-186	12-186	12-186	12-186#	12-187	12-187	12-187	12-187#	12-187	12-187#	12-188	12-188	12-188	12-188
	12-188#	12-189	12-189	12-189	12-189	12-189	12-189	12-189#	12-190	12-190	12-190	12-190	12-190#	12-191
	12-191	12-191	12-191#	12-192	12-192	12-192	12-192	12-192#	12 193	12 193	12-193	12-193#	15-7	15 7#
	15-9#	15-77	15-77	15-77#	15-78	15-78	15 78	15-78	15-78#	15-79	15-79	15-79	15 79	15 79

	15 790	17-34	17-34	17-34	17-34	17 34	17 34	17-340	17 41	17 410	17 44	17 44	17 44	17 44
	17-440	17-99	17-99	17-99	17 99	17 99	17 99	17 490						
M\$DATA	1 8670	2 80	4 8	4-8	4-8	4 8	4 8	4 8	4 8	4 8	4 8	4 8	4 8	4 8
	4 8	4 8	4-8	4 8	4 8	4 8	4-8	4 8	4 8	4 8	4 8	4 8	4 8	4-8
	4 8	4-8	4-8	4 8	4-8	4 8	4 8	4-8	4 8	4-8	4-8	4 80	4 80	4 10
M\$DECR	4 100	4 11	4-110											
	1-D290	2-80	4-9	4-90	4-162	4-1620	4 578	4-5780	4 814	4-8140	5-59	5-590	5 73	5-730
	5 87	5-870	5-102	5-1020	5-117	5-1170	5 222	5-2220	5 236	5-2360	5-258	5-2580	5 272	5-2720
	5-286	5-2860	5-287	5-2870	5-294	5-2940	5-306	5-3060	5-307	5-3070	5 324	5-3240	5-325	5-3250
	5-334	5-3340	6-131	6-1310	6-132	6-1320	7-38	7-380	8-23	8-230	8 27	8-270	8 29	8-290
	9-51	9-510	9-67	9-670	10-396	10-396	10-3960	10-3960	11-243	11-2430	12 195	12-1950	13 93	13-930
	14-31	14-310	14-46	14-460	15-81	15-810	16 139	16-1390	16 153	16-1530	17 97	17-970	17 110	17-1100
	18-155	18-1550	18-170	18-1700	19-151	19-1510	19-165	19-1650	19-166	19-1660	20 11	20-110	20 12	20-120
M\$DEFA	20-50	20-500	20-73	20-730										
	1-E700	2-80	20-4	20-40	20-5	20-50	20-6	20-60	20-7	20-70	20-8	20-80	20 9	20-90
	20-26	20-260	20-27	20-270	20-33	20-330	20-36	20-360	20-38	20-380	20-39	20-390	20 41	20-410
	20-42	20-420	20-44	20-440	20-46	20-460	20-48	20-480						
M\$ENDE	1-D740	2-80	4-90	4-1620	4-5780	4-8140	5-590	5-730	5-870	5-1020	5 1170	5 2220	5 2360	5 2580
	5 2720	5-2860	5-2870	5-3060	5-3070	5-3240	5-3250	5-3340	6-1310	6-1320	7-380	8 230	8 270	8-290
	9-510	9-670	10-3960	11-2430	12-1950	13-930	14-310	14-460	15-810	16-1390	16-1530	17-970	17-1100	18 1550
M\$ERRI	18-1700	19-1510	19-1650	19-1660	20-110	20-190	20-500	20-730						
	1-8490	2-80	10-97	10-970	10-134	10-1340	10-147	10-1470	10-153	10-1530	10-258	10-2580	10-263	10-2630
	10-372	10-3720	10-388	10-3880	10-431	10-4310	10-442	10-4420	10-539	10-5390	10 550	10-5500	10-554	10-5540
	10-562	10-5620	10-611	10-6110	10-623	10-6230	10-630	10-6300	10-692	10-6920	10 702	10-7020	10-706	10-7060
	10-767	10-7670	10-820	10-8200	10-825	10-8250	10-929	10-9290	11-52	11-520	11-67	11-670	11 73	11-730
	11-82	11-820	11-86	11-860	12-53	12-530	12-57	12-570	13 75	13-750	13-82	13-820	13 85	13-850
	15-41	15-410	15-45	15-450	15-61	15-610	15 65	15-650	17-68	17-680	17 75	17-750	17 79	17-790
	17-86	17-860												
M\$ESCA	1-D060	2-80	14-32	14-320	16-140	16-1400	18-156	18-1560	19-152	19-1520				
M\$ESCS	1-D100	2-80	14-320	16-1400	18-1560	19-1520								
M\$EXCP	1-E010	2-80	20-5	20-50	20-50	20-6	20-60	20-7	20-70	20-70	20 70	20 9	20 9	20-90
	20-38	20 38	20-380	20-41	20-41	20-410	20-44	20-440	20-440	20-46	20 46	20 460	20 48	20-480
	20-480													
M\$EXIT	1-D140	2-80	13-12	13-120	15-10	15-100	17 45	17-450	17 69	17-690				
M\$EXSE	1-D220	2-80	13-120	15 100	17-450	17-690								
M\$EXTJ	1-D180	2-80	13-120	15-100	17-450	17-690								
M\$GEN	1-D380	2-80	4-7	4-70	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4 8	4 8	4 8
	4-8	4-8	4-8	4-8	4-8	4 8	4 8	4-8	4-8	4-8	4 8	4 8	4-8	4-8
	4 8	4-8	4-8	4-8	4-8	4-8	4 8	4-8	4-8	4-8	4 8	4 8	4 8	4 8
	4 80	4-80	4-80	4-80	4-80	4-80	4-80	4-80	4-80	4-80	4-80	4-80	4-80	4-80
	4-80	4-80	4-80	4-80	4-80	4-80	4-80	4-80	4-80	4-80	4-80	4-80	4-80	4-80
	4-80	4-80	4-80	4-80	4-80	4-80	4-80	4-80	4-80	4-80	4-10	4-100	4-11	4-110
	4-34	4-340	4-168	4-1680	4-589	4-5890	5-2	5-20	5-47	5-470	5-59	5-590	5-61	5-610
	5-73	5-730	5-75	5-750	5-87	5-870	5-89	5-890	5-102	5-1020	5-104	5-1040	5-117	5-1170
	5-119	5-1190	5-222	5-2220	5-224	5-2240	5-236	5-2360	5-238	5-2380	5-258	5-2580	5-260	5-2600
	5-272	5-2720	5-273	5-2730	5-286	5-2860	5-290	5-2900	5-298	5-2980	5-299	5-2990	5-2990	5-306
	5-3060	5-309	5-3090	5-310	5-3100	5-3100	5-324	5-3240	5-327	5-3270	5-332	5-3320	6 3	6-30
	6-4	6-40	6-131	6-1310	7-11	7-110	7-38	7-380	8-4	8-40	8-5	8-50	8-23	8-230
	8-25	8-250	8-27	8-270	9-3	9-30	9-450	9-51	9-510	9-530	9-67	9-670	10 396	10-3960
	12-3	12-30	12-7	12-70	12-195	12-1950	13-2	13-20	13-93	13-930	14-2	14-20	14 21	14-210
	14-31	14-310	14-46	14-460	15-2	15-20	15-81	15-810	16-2	16-20	16-60	16-600	16 139	16-1390
	16-153	16-1530	17-2	17-20	17-9	17-90	17-97	17-970	17-110	17-1100	17-2	18-20	18 75	18-750
	18-155	18-1550	18-170	18-1700	19-2	19-20	19-64	19-640	19-151	19-1510	19 165	19-1650	20 2	20-20
	20-3	20-30	20-11	20-110	20-22	20-220	20-23	20-230	20-50	20-500	20-75	20-750		
M\$GENB	1-C380	2-80												
M\$GETS	1-D350	2-80	4-9	4-90	4-162	4-1620	4-578	4-5780	4-814	4-8140	5 59	5-590	5 73	5-730

	5-87	5-87	5-102	5-102	5-117	5-117	5-222	5-222	5-236	5-236	5-258	5-258	5-272	5-272
	5-286	5-286	5-287	5-287	5-294	5-294	5-306	5-306	5-307	5-307	5-324	5-324	5-325	5-325
	5-334	5-334	6-131	6-131	6-132	6-132	7-38	7-38	8-23	8-23	8-27	8-27	8-29	8-29
	9-51	9-51	9-67	9-67	10-396	10-396	10-396	10-396	11-243	11-243	12-195	12-195	13-93	13-93
	14-31	14-31	14-46	14-46	15-81	15-81	16-139	16-139	16-153	16-153	17-97	17-97	17-110	17-110
	18-155	18-155	18-170	18-170	19-151	19-151	19-165	19-165	19-166	19-166	20-11	20-11	20-19	20-19
	20-37	20-37	20-40	20-40	20-43	20-43	20-50	20-50	20-73	20-73	20-73	20-73	20-110	20-190
MAGE TT	1-877	2-80	13-12	14-32	15-10	16-140	17-45	17-69	18-156	19-152	20-37	20-37	20-40	20-40
	20-43	20-43												
MIGNGB	1-C02	2-80	4-7	4-7	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4-8
	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4-8
	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4-8
	4-80	4-80	4-80	4-80	4-80	4-80	4-80	4-80	4-80	4-80	4-80	4-80	4-80	4-80
	4-80	4-80	4-80	4-80	4-80	4-80	4-80	4-80	4-80	4-80	4-80	4-80	4-80	4-80
	4-80	4-80	4-80	4-80	4-80	4-80	4-80	4-80	4-80	4-80	4-80	4-80	4-80	4-80
	4-34	4-34	4-168	4-168	4-589	4-589	5-2	5-2	5-47	5-47	5-61	5-61	4-10	4-11
	5-89	5-89	5-104	5-104	5-119	5-119	5-224	5-224	5-238	5-238	5-260	5-260	5-273	5-273
	5-290	5-290	5-298	5-298	5-299	5-299	5-299	5-309	5-309	5-310	5-310	5-310	5-327	5-327
	5-332	5-332	6-3	6-3	6-4	6-4	7-11	7-11	8-4	8-4	8-5	8-5	8-25	8-25
	9-3	9-3	9-45	9-45	9-53	9-53	12-3	12-3	20-2	20-2	20-3	20-3	20-22	20-22
	20-23	20-23	20-75	20-75										
MIGNIN	1-049	2-80	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4-8
	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4-8
	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4-8
	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4-8	4-8
	4-80	4-80	4-80	4-80	4-80	4-80	4-80	4-80	4-80	4-80	4-80	4-80	4-80	4-80
	4-80	4-80	4-80	4-80	4-80	4-80	4-80	4-80	4-80	4-80	4-80	4-80	4-80	4-80
	4-80	4-80	4-80	4-80	4-80	4-80	4-80	4-80	4-80	4-80	4-80	4-80	4-80	4-80
	4-110	4-110	5-59	5-59	5-73	5-73	5-87	5-87	5-102	5-102	5-117	5-117	5-160	5-160
	5-160	5-160	5-160	5-160	5-160	5-160	5-160	5-160	5-160	5-160	5-160	5-160	5-160	5-160
	5-193	5-193	5-193	5-193	5-193	5-193	5-193	5-193	5-193	5-193	5-193	5-193	5-207	5-207
	5-207	5-207	5-207	5-207	5-207	5-207	5-207	5-207	5-207	5-207	5-207	5-207	5-207	5-207
	5-207	5-207	5-207	5-207	5-207	5-207	5-207	5-207	5-207	5-207	5-207	5-207	5-207	5-207
	5-216	5-216	5-216	5-216	5-216	5-216	5-216	5-216	5-216	5-216	5-216	5-216	5-216	5-216
	5-272	5-272	5-279	5-279	5-279	5-279	5-279	5-279	5-279	5-279	5-279	5-279	5-279	5-279
	5-279	5-279	5-279	5-279	5-279	5-279	5-279	5-279	5-279	5-279	5-279	5-279	5-279	5-279
	5-280	5-280	5-280	5-280	5-280	5-280	5-280	5-280	5-280	5-280	5-280	5-280	5-280	5-280
	5-280	5-280	5-280	5-280	5-280	5-280	5-280	5-280	5-280	5-280	5-280	5-280	5-280	5-280
	5-280	5-280	5-280	5-280	5-280	5-280	5-280	5-280	5-280	5-280	5-280	5-280	5-280	5-280
	5-282	5-282	5-282	5-282	5-282	5-282	5-282	5-282	5-282	5-282	5-282	5-282	5-282	5-282
	5-282	5-282	5-282	5-282	5-282	5-282	5-282	5-282	5-282	5-282	5-282	5-282	5-282	5-282
	5-332	5-332	5-332	5-332	5-332	5-332	5-332	5-332	5-332	5-332	5-332	5-332	5-332	5-332
	5-332	5-332	5-332	5-332	5-332	5-332	5-332	5-332	5-332	5-332	5-332	5-332	5-332	5-332
	6-110	6-110	6-12	6-12	6-13	6-13	6-14	6-14	6-18	6-18	6-18	6-18	6-19	6-19
	6-22	6-22	6-22	6-22	6-23	6-23	6-43	6-43	6-43	6-43	6-44	6-44	6-46	6-46
	6-46	6-46	6-47	6-47	6-49	6-49	6-49	6-49	6-50	6-50	6-63	6-63	6-63	6-63
	6-63	6-64	6-64	6-64	6-96	6-96	6-96	6-96	6-96	6-96	6-96	6-96	6-96	6-96
	6-96	6-97	6-97	6-97	6-97	6-124	6-124	6-124	6-124	6-124	6-124	6-124	6-124	6-124
	6-124	6-124	6-125	6-125	6-125	6-125	6-125	6-125	6-125	6-125	6-125	6-125	6-125	6-125
	6-125	6-125	6-125	6-125	6-125	6-125	6-125	6-125	6-125	6-125	6-125	6-125	6-125	6-125
	6-126	6-127	6-127	6-127	6-127	6-128	6-128	6-128	6-131	6-131	7-13	7-13	7-13	7-13
	7-13	7-13	7-13	7-13	7-13	7-13	7-13	7-13	7-19	7-19	7-19	7-19	7-19	7-19
	7-19	7-19	7-19	7-19	7-20	7-20	7-20	7-20	7-20	7-20	7-20	7-20	7-20	7-20
	7-20	7-20	7-20	7-20	7-20	7-20	7-20	7-20	7-20	7-20	7-20	7-20	7-20	7-20
	7-22	7-22	7-22	7-22	7-24	7-24	7-24	7-24	7-31	7-31	7-31	7-31	7-31	7-31
	7-22	7-22	7-22	7-22	7-24	7-24	7-24	7-24	7-31	7-31	7-31	7-31	7-31	7-31
	7-31	7-31	7-31	7-31	7-33	7-33	7-33	7-33	7-33	7-33	7-33	7-33	7-33	7-33
	7-33	7-33	7-33	7-33	7-33	7-33	7-33	7-33	7-33	7-33	7-33	7-33	7-33	7-33
	7-35	7-35	7-35	7-35	7-36	7-36	7-36	7-36	7-37	7-37	7-37	7-37	7-37	7-37



12 187#	12 187#	12 187#	12 187#	12 188	12 188	12 188	12 188	12 188	12 188	12 188	12 188	12 188	12 188
12 188#	12 188#	12 188#	12 188#	12 188#	12 188#	12 188#	12 188#	12 188#	12 188#	12 189	12 189	12 189	12 189
12 189	12 189	12 189	12 189	12 189	12 189	12 189	12 189	12 189#	12 189#	12 189#	12 189#	12 189#	12 189#
12 190	12 190	12 190	12 190	12 190	12 190	12 190	12 190	12 190	12 190#	12 190#	12 190#	12 190#	12 190#
12 190#	12 190#	12 190#	12 191	12 191	12 191	12 191	12 191	12 191	12 191	12 191	12 191	12 191#	12 191#
12 191#	12 191#	12 191#	12 191#	12 191#	12 191#	12 192	12 192	12 192	12 192	12 192	12 192	12 192	12 192
12 192#	12 192#	12 192#	12 192#	12 192#	12 192#	12 192#	12 192#	12 193	12 193	12 193	12 193	12 193	12 193
12 193	12 193#	12 193#	12 193#	12 193#	12 193#	12 193#	12 193#	12 195	12 195#	13 12	13 12	13 12#	13 12#
13 69	13 69#	13 70	13 70#	13 75	13 75	13 75	13 75	13 75#	13 75#	13 75#	13 75#	13 75#	13 82
13 82	13 82	13 82	13 82#	13 82#	13 82#	13 82#	13 82#	13 85	13 85	13 85	13 85	13 85#	13 85#
13 85#	13 85#	13 85#	13 93	13 93#	14 21	14 21#	14 31	14 31#	14 32	14 32	14 32#	14 32#	14 46
14 46#	15 7	15 7	15 7	15 7	15 7	15 7	15 7#	15 7#	15 7#	15 7#	15 7#	15 9	15 9
15 9	15 9	15 9	15 9	15 9#	15 9#	15 9#	15 9#	15 9#	15 10	15 10	15 10#	15 10#	15 41
15 41	15 41	15 41	15 41#	15 41#	15 41#	15 41#	15 41#	15 45	15 45	15 45	15 45	15 45#	15 45#
15 45#	15 45#	15 45#	15 61	15 61#	15 61#	15 61#	15 61#	15 61#	15 61#	15 61#	15 61#	15 65	15 65
15 65	15 65	15 65#	15 65#	15 65#	15 65#	15 65#	15 65#	15 77	15 77	15 77	15 77	15 77	15 77
15 77#	15 77#	15 77#	15 77#	15 77#	15 77#	15 78	15 78	15 78	15 78	15 78	15 78	15 78	15 78
15 78	15 78	15 78#	15 78#	15 78#	15 78#	15 78#	15 78#	15 78#	15 78#	15 79	15 79	15 79	15 79
15 79	15 79	15 79	15 79	15 79	15 79	15 79#	15 79#	15 79#	15 79#	15 79#	15 79#	15 79#	15 79#
15 79#	15 81	15 81#	16 60	16 60#	16 139	16 139#	16 140	16 140	16 140#	16 140#	16 153	16 153#	17 9
17 9#	17 34	17 34	17 34	17 34	17 34	17 34	17 34	17 34	17 34	17 34	17 34	17 34	17 34#
17 34#	17 34#	17 34#	17 34#	17 34#	17 34#	17 34#	17 34#	17 34#	17 41	17 41	17 41	17 41	17 41
17 41	17 41#	17 41#	17 41#	17 41#	17 41#	17 44	17 44	17 44	17 44	17 44	17 44	17 44	17 44
17 44	17 44	17 44#	17 44#	17 44#	17 44#	17 44#	17 44#	17 44#	17 44#	17 45	17 45	17 45#	17 45#
17 68	17 68	17 68	17 68	17 68#	17 68#	17 68#	17 68#	17 68#	17 69	17 69	17 69#	17 69#	17 75
17 75	17 75	17 75	17 75#	17 75#	17 75#	17 75#	17 75#	17 75#	17 79	17 79	17 79	17 79#	17 79#
17 79#	17 79#	17 79#	17 86	17 86	17 86	17 86	17 86#	17 86#	17 86#	17 86#	17 86#	17 97	17 97#
17 99	17 99	17 99	17 99	17 99	17 99	17 99	17 99	17 99	17 99	17 99	17 99	17 99#	17 99#
17 99#	17 99#	17 99#	17 99#	17 99#	17 99#	17 99#	17 99#	17 99#	17 110	17 110#	18 75	18 75#	18 155
18 156	18 156	18 156#	18 156#	18 170	18 170#	19 64	19 64#	19 151	19 151#	19 152	19 152	19 152#	19 152#
19 165	19 165#	20 3	20 3#	20 4	20 4	20 4	20 4#	20 5	20 5	20 5	20 5	20 5#	20 6
20 6	20 6	20 6	20 6#	20 7	20 7	20 7	20 7#	20 7	20 7#	20 8	20 8	20 8	20 8#
20 9	20 9	20 9	20 9	20 9	20 9#	20 11	20 11#	20 23	20 23#	20 26	20 26	20 26	20 26#
20 27	20 27	20 27	20 27#	20 33	20 33	20 33	20 33#	20 36	20 36	20 36	20 36#	20 37	20 37#
20 38	20 38	20 38	20 38	20 38	20 38#	20 39	20 39	20 39	20 39#	20 40	20 40#	20 41	20 41
20 41	20 41	20 41	20 41#	20 42	20 42	20 42	20 42#	20 43	20 43#	20 44	20 44	20 44	20 44
20 44	20 44#	20 46	20 46	20 46	20 46	20 46	20 46#	20 48	20 48	20 48	20 48	20 48	20 48#
20 50	20 50#	20 75	20 75	20 75	20 75#	20 75#	20 75#						
M%GNLS	1 C13#	2 8#	10 396	10 396#									
M%GNSU	1 898#	2 8#	14 21	14 21#	16 60	16 60#	17 9	17 9#	18 75	18 75#	19 64	19 64#	
M%GNTA	1 890#	2 8#	5 59	5 59#	5 73	5 73#	5 87	5 87#	5 102	5 102#	5 117	5 117#	5 222
	5 236	5 236#	5 258	5 258#	5 272	5 272#	5 286	5 286#	5 306	5 306#	5 324	5 324#	6 131
	7 38	7 38#	8 23	8 23#	8 27	8 27#	9 51	9 51#	9 67	9 67#	12 195	12 195#	13 93
	14 31	14 31#	14 46	14 46#	15 81	15 81#	16 139	16 139#	16 153	16 153#	17 97	17 97#	17 110
	18 155	18 155#	18 170	18 170#	19 151	19 151#	19 165	19 165#	20 11	20 11#	20 50	20 50#	
M%GNTE	1 894#	2 8#	12 7	12 7#	13 2	13 2#	14 2	14 2#	15 2	15 2#	16 2	16 2#	17 2
	18 2	18 2#	19 2	19 2#									
M%SHAPT	1 A39#	2 8#	4 8	4 8#									
M%SHNAP	1 824#	2 8#	4 8	4 8#									
M%INCR	1 D26#	2 8#	4 7	4 7#	4 34	4 34#	4 168	4 168#	4 589	4 589#	5 2	5 2#	5 47
	5 47#	5 47#	5 59#	5 61	5 61	5 61#	5 61#	5 73#	5 75	5 75	5 75#	5 75#	5 87#
	5 89	5 89#	5 89#	5 102#	5 104	5 104	5 104#	5 104#	5 117#	5 119	5 119	5 119#	5 119#
	5 193#	5 207#	5 216#	5 222#	5 224	5 224	5 224#	5 224#	5 236#	5 238	5 238	5 238#	5 258#
	5 260	5 260	5 260#	5 260#	5 272#	5 273	5 273#	5 273#	5 273#	5 279#	5 280#	5 280#	5 280#
	5 290	5 290#	5 290#	5 298	5 298#	5 299	5 299#	5 299#	5 299#	5 309	5 309#	5 310	5 310#
	5 310#	5 327	5 327#	6 3	6 3#	6 4	6 4#	6 4#	6 4#	6 8#	6 11#	6 12#	6 18#







	12-187#	12-187#	12-187#	12-187#	12-187#	12-187#	12-188	12-188	12-188	12-188	12-188	12-188	12-188#	12-188#
	12-188#	12-188#	12-188#	12-188#	12-188#	12-188#	12-189	12-189	12-189	12-189	12-189	12-189	12-189#	12-189#
	12-189#	12-189#	12-189#	12-189#	12-189#	12-189#	12-190	12-190	12-190	12-190	12-190	12-190	12-190#	12-190#
	12-190#	12-190#	12-190#	12-190#	12-191	12-191	12-191	12-191	12-191	12-191	12-191	12-191#	12-191#	12-191#
	12-191#	12-191#	12-192	12-192	12-192	12-192	12-192	12-192#	12-192#	12-192#	12-192#	12-192#	12-193	12-193
	12-193	12-193	12-193	12-193#	12-193#	12-193#	12-193#	12-193#	15-7	15-7	15-7	15-7#	15-7#	15-7#
	15-9	15-9	15-9	15-9#	15-9#	15-9#	15-9#	15-77	15-77	15-77	15-77#	15-77#	15-77#	15-77#
	15-78	15-78	15-78	15-78	15-78	15-78	15-78#	15-78#	15-78#	15-78#	15-78#	15-78#	15-79	15-79
	15-79	15-79	15-79	15-79	15-79	15-79	15-79#	15-79#	15-79#	15-79#	15-79#	15-79#	17-34	17-34
	17-34	17-34	17-34	17-34	17-34	17-34	17-34#	17-34#	17-34#	17-34#	17-34#	17-34#	17-34#	17-34#
	17-41	17-41	17-41	17-41#	17-41#	17-41#	17-44	17-44	17-44	17-44	17-44	17-44#	17-44#	17-44#
	17-44#	17-44#	17-44#	17-44#	17-99	17-99	17-99	17-99	17-99	17-99	17-99	17-99#	17-99#	17-99#
	17-99#	17-99#	17-99#	17-99#	17-99#	17-99#	17-99#	17-99#	17-99#	17-99#	17-99#	17-99#	17-99#	17-99#
M1RADI	1-077#	2-8#	20-4	20-4#	20-5	20-5#	20-6	20-6#	20-7	20-7#	20-8	20-8#	20-9	20-9#
	20-26	20-26#	20-27	20-27#	20-33	20-33#	20-36	20-36#	20-38	20-38#	20-39	20-39#	20-41	20-41#
	20-42	20-42#	20-44	20-44#	20-46	20-46#	20-48	20-48#						
M1RBRO	1-C52#	2-8#												
M1RNRO	1-C62#	2-8#	6-8	6-8#	6-63	6-63#								
M1SETS	1-D32#	2-8#	4-7	4-7#	4-34	4-34#	4-168	4-168#	4-589	4-589#	5-2	5-2#	5-47	5-47#
	5-61	5-61#	5-75	5-75#	5-89	5-89#	5-104	5-104#	5-119	5-119#	5-224	5-224#	5-238	5-238#
	5-260	5-260#	5-273	5-273#	5-290	5-290#	5-298	5-298#	5-299	5-299#	5-309	5-309#	5-310	5-310#
	5-327	5-327#	6-3	6-3#	6-4	6-4#	7-11	7-11#	8-4	8-4#	8-5	8-5#	8-25	8-25#
	9-3	9-3#	9-45	9-45#	9-53	9-53#	10-349	10-349#	10-349#	10-349#	12-3	12-3#	12-7	12-7#
	13-2	13-2#	14-2	14-2#	14-21	14-21#	15-2	15-2#	16-2	16-2#	16-60	16-60#	17-2	17-2#
	17-9	17-9#	18-2	18-2#	18-75	18-75#	19-2	19-2#	19-64	19-64#	20-2	20-2#	20-3	20-3#
M1STAR	20-22	20-22#	20-23	20-23#										
M1SVC	1-A33#	2-8#												
	1-C33#	2-8#	5-59	5-59#	5-73	5-73#	5-87	5-87#	5-102	5-102#	5-117	5-117#	5-160	5-160#
	5-193	5-193#	5-207	5-207#	5-216	5-216#	5-222	5-222#	5-236	5-236#	5-258	5-258#	5-272	5-272#
	5-279	5-279#	5-280	5-280#	5-282	5-282#	5-286	5-286#	6-8	6-8#	6-11	6-11#	6-12	6-12#
	6-13	6-13#	6-18	6-18#	6-22	6-22#	6-43	6-43#	6-46	6-46#	6-49	6-49#	6-63	6-63#
	6-96	6-96#	6-97	6-97#	6-124	6-124#	6-125	6-125#	6-126	6-126#	6-127	6-127#	6-128	6-128#
	6-131	6-131#	7-13	7-13#	7-19	7-19#	7-20	7-20#	7-22	7-22#	7-24	7-24#	7-31	7-31#
	7-33	7-33#	7-35	7-35#	7-36	7-36#	7-37	7-37#	7-38	7-38#	8-7	8-7#	8-9	8-9#
	8-16	8-16#	8-20	8-20#	8-21	8-21#	8-23	8-23#	8-27	8-27#	9-10	9-10#	9-30	9-30#
	10-8	10-8#	10-10	10-10#	10-11	10-11#	10-12	10-12#	10-13	10-13#	10-14	10-14#	10-97	10-134
	10-147	10-153	10-258	10-263	10-349	10-349#	10-350	10-350#	10-372	10-388	10-396	10-396#	10-431	10-442
	10-491	10-491#	10-492	10-492#	10-493	10-493#	10-539	10-550	10-554	10-562	10-611	10-623	10-630	10-692
	10-702	10-706	10-767	10-820	10-825	10-914	10-914#	10-929	11-52	11-67	11-73	11-82	11-86	11-147
	11-147#	11-148	11-148#	11-152	11-152#	11-165	11-165#	11-181	11-181#	11-185	11-185#	11-189	11-189#	11-192
	11-192#	11-204	11-204#	11-213	11-213#	11-214	11-214#	11-217	11-217#	11-225	11-225#	11-227	11-227#	11-228
	11-228#	11-229	11-229#	12-12	12-12#	12-14	12-14#	12-53	12-57	12-185	12-185#	12-186	12-186#	12-187
	12-187#	12-188	12-188#	12-189	12-189#	12-190	12-190#	12-191	12-191#	12-192	12-192#	12-193	12-193#	12-195
	12-195#	13-12	13-12#	13-69	13-69#	13-75	13-82	13-85	13-93	13-93#	14-21	14-21#	14-31	14-31#
	14-32	14-32#	14-46	14-46#	15-7	15-7#	15-9	15-9#	15-10	15-10#	15-41	15-45	15-61	15-65
	15-77	15-77#	15-78	15-78#	15-79	15-79#	15-81	15-81#	16-60	16-60#	16-139	16-139#	16-140	16-140#
	16-153	16-153#	17-9	17-9#	17-34	17-34#	17-41	17-41#	17-44	17-44#	17-45	17-45#	17-68	17-69
	17-69#	17-75	17-79	17-86	17-97	17-97#	17-99	17-99#	17-110	17-110#	18-75	18-75#	18-155	18-155#
	18-156	18-156#	18-170	18-170#	19-64	19-64#	19-151	19-151#	19-152	19-152#	19-165	19-165#		
M1TLAB	1-C29#	2-8#	5-59#	5-73#	5-87#	5-102#	5-117#	5-160#	5-193#	5-207#	5-216#	5-222#	5-236#	5-258#
	5-272#	5-279#	5-280#	5-282#	5-286#	6-8#	6-11#	6-12#	6-13#	5-18#	6-22#	6-43#	6-46#	6-49#
	6-63#	6-96#	6-97#	6-124#	6-125#	6-126#	6-127#	6-128#	6-131#	7-13#	7-19#	7-20#	7-22#	7-24#
	7-31#	7-33#	7-35#	7-36#	7-37#	7-38#	8-7#	8-9#	8-16#	8-20#	8-21#	8-23#	8-27#	9-10#
	9-30#	10-8#	10-10#	10-11#	10-12#	10-13#	10-14#	10-97#	10-134#	10-147#	10-153#	10-258#	10-263#	10-349#
	10-350#	10-372#	10-388#	10-396#	10-431#	10-442#	10-491#	10-492#	10-493#	10-539#	10-550#	10-554#	10-562#	10-611#
	10-623#	10-630#	10-692#	10-702#	10-706#	10-767#	10-820#	10-825#	10-914#	10-929#	11-52#	11-67#	11-73#	11-82#



