

IDENTIFICATION

PRODUCT CODE: AC-F123B-MC
PRODUCT NAME: CZRLJB0 RL01/02 DRIVE TEST 2
DATE CREATED: 5-JAN-79
REVISED: 7-DEC-79
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) 1977, 1979 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
1.2	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

1.0 GENERAL INFORMATION1.1 PROGRAM ABSTRACT1.1.1 STRUCTURE OF PROGRAM

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

1.1.2 DIAGNOSTIC INFORMATION

THIS PROGRAM TESTS AND EXERCISES RL01/02 DISK DRIVES RL11/RLV11 CONTROLLERS (4 DRIVES PER CONTROLLER). THE ENTIRE PROGRAM IS RUN ON THE FIRST DRIVE BEFORE STARTING ON THE SECOND. THE PROGRAM STARTS BY TESTING THE SIMPLEST FUNCTIONS FIRST USING THE LOGIC TESTED IN EARLIER TESTS TO TEST MORE COMPLEX FUNCTIONS.

THIS PROGRAM TESTS THE RL01/02 OUTER AND INNER GUARD BAND DETECTION. SEEK OPERATIONS UNDERGO A BROAD RANGE OF TESTING USING SINGLE DIFFERENCES, PROCEEDING TO SEEKS OF GREATER DIFFERENCES.

1.2 SYSTEM REQUIREMENTS

1.2.1 HARDWARE REQUIREMENTS

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

1.2.2 SOFTWARE REQUIREMENTS

CZRLJ80 RL01/02 DRIVE TEST PART 2 (FORMERLY CZRLDB0)

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

CVRLAB0	RLV11 RL01 DISKLESS TEST (RLV11 ONLY)
CZRLG80	RL11/RLV11 RL01/02 CONTROLLER TEST (PART 1)
CZRLH80	RL11/RLV11 RL01/02 CONTROLLER TEST (PART 2)
CZRLIB0	RL01/02 DRIVE TEST (PART 1)

1.5 ASSUMPTIONS

THE HARDWARE OTHER THAN THE RL01/02 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 DIAGNOSTIC2.1.1 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 NAK
BOOTED VIA UNIT 0
ENTER DATE (DD-MM-YY):
```

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

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

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

WHEN THIS DIAGNOSTIC IS STARTED THE FOLLOWING STEPS WILL OCCUR:

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

THE DIAGNOSTIC WILL ISSUE THE PROMPT 'DR>'. FROM THIS POINT UNTIL THE TIME WHEN YOU RESTART XXDP+, YOU WILL BE TALKING TO THE DIAGNOSTIC, NOT XXDP+. WE WILL REFER TO THE PRESENCE OF THIS PROMPT AS BEING IN DIAGNOSTIC COMMAND MODE, AS OPPOSED TO XXDP+ COMMAND MODE.

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

```
STA/PASS:1/FLAGS:HOE
```

THINGS TO NOTE HERE:

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

PNT	PRINT NUMBER OF TEST BEING EXECUTED
LOE	LOOP ON ERROR
HOE	HALT ON ERROR
IER	INHIBIT ERROR PRINTOUT

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

* STEP 2 *

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

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

* STEP 3 *

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

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

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

* STEP 4 *

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

* STEP 5 *

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

1. IF NO ERROR IS ENCOUNTERED, THEN THE DIAGNOSTIC WILL SIMPLY EXECUTE THE DESIRED NUMBER OF PASSES AND RETURN TO COMMAND MODE (PROMPT DR>).
2. IF AN ERROR IS ENCOUNTERED, THEN ONE OF THREE THINGS HAPPENS, DEPENDING ON THE SETTINGS OF THE HOE AND LOE FLAGS.

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

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

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

2.1.2 SAMPLE RUN-THROUGH

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

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

1. ISSUE ANOTHER 'START' COMMAND (THUS GOING THRU ALL OF STEPS 1, 2, 3, 4, AND 5 AGAIN)
2. ISSUE A 'RESTART' COMMAND (SAME AS START COMMAND EXCEPT THAT THE HARDWARE QUESTIONS ARE NOT ASKED)
3. ISSUE A 'CONTINUE' COMMAND (EXECUTION WILL RESUME AT THE BEGINNING OF THE PARTICULAR HARDWARE TEST (MOST DIAGNOSTICS CONSIST OF A NUMBER OF THESE) THAT IT WAS IN WHEN THE ERROR HALT 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 ON ERROR).

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

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

THIS WILL DO THE FOLLOWING:

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

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

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

1. START
2. RESTART
3. CONTINUE

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

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

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

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

1. START
2. RESTART

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

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

	BY WHOM ENTERED: -----
.R CZRLJB	O
DRS LOADED	D
DIAG. RUN-TIME SERVICES REV. D APR-79	D
CZRLJ-B-0	D
CZRLJ TESTS OUTER & INNER GUARD	D
BAND DETECTION AND SEEK OPERATIONS	
UNIT IS RLO1, RLO2	D
DR>STA/PASS:1/FLAGS:HOE	D,O
 # UNITS (D) ? 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 = RLO1 (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 = RLO1 (L) ? N	D,O (N=RLO2)
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
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
 CZRLJ 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:HOE:IER:LOE=0       D,0
CHANGE SW (L) ? N                 D,0
CZRLJ EOP 1                        D
^C
DR>RESTART/PASS:1                 D,0
CHANGE SW (L) ? N                 D,0
-----
-----
-----
-----
```

2.2 CHAIN MODE OPERATION

CHAIN MODE OPERATION CONSISTS OF THE SEQUENTIAL EXECUTION OF PROGRAMS WITHOUT OPERATOR INTERVENTION. ONLY PROGRAMS THAT HAVE BEEN MODIFIED TO RUN IN CHAIN MODE CAN BE CHAINED. CHAINABLE PROGRAMS ARE IDENTIFIED IN THE DIRECTORY BY A BIC EXTENSION.

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 00000. 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:

<u>HOW ENTERED</u>	<u>LEGAL COMMANDS</u>
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 HOE 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 HOE FLAG SET D) OPERATOR ENTERED CONTROL/C. AFTER THE OPERATOR RESPONDS TO '# UNITS?', THE HARDWARE DIALOGUE IS INITIATED. WHEN IT IS COMPLETED, THE QUESTIONS 'CHANGE SW?' IS ISSUED, AND THE ANSWERS, IF GIVEN, BECOME THE NEW DEFAULTS. THEREFORE IT IS NECESSARY TO RELOAD THE PROGRAM IN ORDER TO RETURN TO THE LOAD DEFAULTS.

THE SWITCH ARGUMENTS ARE AS FOLLOWS:

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

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

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

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

IER INHIBIT ERROR REPORTING

IBE INHIBIT BASIC ERROR REPORTS
 IXE INHIBIT EXTENDED ERROR REPORTS
 PRI DIRECT ALL MESSAGES TO A LINE PRINTER
 PNT PRINT NUMBER OF TEST BEING EXECUTED
 BOE BELL ON ERROR
 UAM RUN IN UNATTENDED MODE, BYPASSING MANUAL INTERVENTION TESTS
 ISR INHIBIT STATISTICAL REPORTS
 IDU INHIBIT DROPPING OF UNITS BY DIAGNOSTIC
 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.

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

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

THE SWITCH ARGUMENTS ARE AS IN THE START COMMAND EXCEPT:

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

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

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

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

1. UNSPECIFIED FLAG SETTINGS ARE UNCHANGED

EXIT

RETURN TO XXDP+ PROMPT MODE.

DRO(P)/UNITS:UNIT-LIST

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

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

ADD/UNITS:UNIT-LIST

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

PRI(NT)

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

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

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

FLA(GS)

THE CURRENT SETTINGS OF ALL FLAGS ARE PRINTED.

ZFL(AGS)

ALL FLAGS ARE CLEARED.

2.4 EXTENDED P-TABLE DIALOGUE

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

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

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

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

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

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

UNITS (D) ? 8

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

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

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

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

2.5 HARDWARE PARAMETERS

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

RL11 (L) Y?

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

BUS ADDRESS (O) 174400?

ANSWER WITH THE BUS ADDRESS OF THE CONTROLLER.

VECTOR (O) 160?

ANSWER WITH THE INTERRUPT VECTOR OF THE CONTROLLER.

DRIVE (O) 0?

ANSWER WITH THE DRIVE(S) CONNECTED TO THE CONTROLLER

DRIVE TYPE = RL01 (L) ?

ANSWER NO (N) IF DRIVE IS AN RL02

BR LEVEL (O) 5?

ANSWER WITH THE INTERRUPT PRIORITY OF THE CONTROLLER.

2.6 SOFTWARE PARAMETERS

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

CHANGE S.W. ?

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

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.

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

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

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

3.1.2 SPECIFIC RESULT MESSAGES

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

THE GENERAL FORMAT FOR THE RESULT LINE IS:

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

CONT ERR	(CONTROLLER ERROR)
DRV ERR	(DRIVE ERROR)
NON-EXSTNT MEM	(NON-EXISTANT MEMORY)
HDR CRC	(HEADER CRC ERROR)
DATA CRC	
HDR NOT FND	(HEADER NOT FOUND)
DATA LATE	
HDR NOT FND/HDR CRC/OPI	(ALL 3 BITS SET)
DRV RDY	(DRIVE READY)
SELECTED HEAD	
VOL CHK	(VOLUME CHECK)
COVER OPEN	
BRUSH HME	(BRUCH HOME)
WRT LCK	(WRITE LOCK)
HDS OUT	(HEADER OUT)
DRV SEL ERR	(DRIVE SELECT ERROR)
DRV STATE	(DRIVE STATE)
SPIN TIMEOUT	(SPINDLE TIMEOUT SPD ERROR)
WRT GAT ERR	(WRITE GATE ERROR)
SEEK TIMEOUT	(SKTO ERROR)
CUR HEAD ERR	(CURRENT IN HEAD ERROR)
WRT DAT ERR	(WRITE DATA ERROR)
OP INCOMPLETE	(OPI ERROR)
HDR/DAT ERR	(HDR CRC OR DATA CRC ERROR BIT 11 OF CS REGISTER)
HDR NOT FND/DAT LATE	(HDR NOT FOUND OR DATA LATE ERROR BIT 12 OF CS REGISTER)
CYL	(CYLINDER WHEN REPORTING A SEEK ERROR)

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

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

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

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

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

'INTERRUPT TOO LATE''

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

'FAIL TO RELOAD HEADS AFTER ERR CLEAR''

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

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

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

'WRITE ABORTED''

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

'COULD NOT RETRIEVE DRIVE STATUS''

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

'DPI SET-NO DRIVE RESPONSE''

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

'NO INTERRUPT ON CMND COMPLETE''

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

'ERR DID NOT CLEAR''

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

'DRV ERR IS NOT CLEARED''

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

'UNEXPECTED ERR''

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

'BAD SEC FILE FMT ERR''

THIS IS REPORTED IF THE CONTENTS OF THE FILES DO NOT CORRESPOND TO THE EXPECTED FORMAT. (REFER TO DEC STANDARD 144 FOR FORMAT 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) CP 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 EXISTANT MEMORY ERROR
- BIT 12 - HEADER NOT FOUND (WITH BIT 10 SET)
- DATA LATE (WITH BIT 10 CLEAR)
- BIT 11 - HEADER CRC (WITH BIT 10 SET)
- DATA CRC (WITH BIT 10 CLEAR)
- BIT 10 - OPERATION INCOMPLETE
- BIT 9/8 - DRIVE SELECT (0-3)
- BIT 7 - CONTROLLER READY
- BIT 6 - INTERRUPT ENABLE
- BIT 5 - EXTENDED BUS ADDRESS (BIT 17)
- BIT 4 - EXTENDED BUS ADDRESS (BIT 16)
- BIT 3-1 - FUNCTION CODE
 - 0 - NOP (PDP-11) MAINT (LSI-11)
 - 1 - WRITE CHECK
 - 2 - GET DRIVE STATUS
 - 3 - SEEK
 - 4 - READ HEADER
 - 5 - WRITE DATA
 - 6 - READ DATA
 - 7 - READ WITHOUT HEADER COMPARE

BIT 0 - DRIVE READY

RLBA - BUS ADDRESS REGISTER (XXXXX2)

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

RLDA - DISK ADDRESS REGISTER (XXXXX4)

FOR READ/WRITE FUNCTIONS

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

FOR SEEK FUNCTION

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

FOR GET STATUS FUNCTION

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

RLMP - MULTIPURPOSE REGISTER

FOR READ/WRITE FUNCTION

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

FOR READ HEADER FUNCTION

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

FOR GET STATUS FUNCTION

HAS DRIVE STATUS

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

BIT 8 - DRIVE SELECT ERROR (DSE)
BIT 7 - DRIVE TYPE IS RLO2 IF SET
BIT 6 - SURFACE (0=UPPPER, 1=LOWER)
BIT 5 - COVER OPEN
BIT 4 - HEADS HOME
BIT 3 - BRUSHES HOME
BIT 2-0 -STATE BITS
0 - LOAD STATE
1 - SPIN UP
2 - BRUSH CYCLE
3 - LOAD HEADS
4 - SEEK - TRALK COUNTING
5 - SEEK - LINEAR MODE
6 - UNLOAD HEADS
7 - SPIN DOWN

6.0 TEST SUMMARIES

TEST 1 OUTER GUARD BAND DETECTION TEST

DO READ HEADER, WAIT FOR INTERRUPT. CHECK IF AT CYLINDER 0.
IF NOT, SEEK REVERSE 1 CYLINDER AT A TIME UNTIL CYLINDER 0 IS
REACHED. IF ANY REVERSE SEEK FAILS TO MOVE THE HEADS IN 10
TRIES:

DETECTION OF GUARD BAND PREMATURE.

WHEN AT CYLINDER 0, DO SEEK DIFFERENCE OF 1, SIGN 0, HEAD 0.
WAIT FOR INTERRUPT, WAIT FOR READY. READY SHOULD SET IN
20MS>T>15MS. IF NOT:

FAILED TO DETECT GUARD BAND

DO READ HEADER. WAIT FOR INTERRUPT. CHECK FOR CYLINDER 0.
IF NOT:

FAILED TO SEEK BACK TO ZERO

DO SEEK WITH DIFFERENCE OF 1, SIGN 0, HEAD 1. DO SAME TESTS
AS ABOVE WITH REGARD TO READY VS TIME AND CYLINDER FOUND IN
HEADER.

NOTE: CHOOSING A SINGLE SURFACE WILL LIMIT THE TESTING TO
THAT SURFACE.

TEST 2 INCREMENTAL FORWARD SEEK HEAD 0 TEST

POSITION HEADS AT CYLINDER 'LOLIMIT' USING SEEKS WITH DIFFERENCE OF ONE, HEAD 0.

DO SEEK WITH DIFFERENCE OF 1, SIGN 1, HEAD 0. WAIT FOR INTERRUPT, WAIT FOR DRIVE READY. CHECK READY IS SET IN 15 MS. IF NOT:

POSITIONING PROBLEM AT A SPECIFIC CYLINDER
MECHANICAL OBSTRUCTION

CHECK THAT THIS CYLINDER IS OLD CYLINDER + 1. IF NOT:

DIFFERENCE REGISTER OR COUNT LOGIC FAILURE
TRACK CROSSING DETECTION FAILURE

REPEAT SEEKS AND READS UNTIL CYLINDER READ IS 'HILIMIT'.

NOTE 1: IF THE 'USE ALL SECTORS' PARAMETER IS SPECIFIED AS 'Y', THE TEST WILL READ AND TEST ALL 40 HEADERS (CARTRIDGE VERIFY).

NOTE 2: TESTING WILL BE DONE BETWEEN UPPER AND LOWER CYLINDER LIMITS. CHOOSING A SINGLE SURFACE WILL LIMIT TESTING TO THAT SURFACE. THIS TEST WILL BE BYPASSED IF SURFACE 1 IS CHOSEN.

TEST 3 INCREMENTAL REVERSE SEEK HEAD 0 TEST

POSITION HEADS AT CYLINDER 'HILIMIT' USING SEEKS WITH DIFFERENCE OF 1, HEAD 0.

DO SEEK WITH DIFFERENCE OF 1, SIGN 0, HEAD 0. WAIT FOR INTERRUPT, WAIT FOR DRIVE READY. CHECK READY SET IN 15 MS:

POSITIONING PROBLEM AT A SPECIFIC CYLINDER
DO READ HEADER, WAIT FOR INTERRUPT. CHECK THAT THIS CYLINDER IS OLD CYLINDER - 1. IF NOT:

DIFFERENCE REGISTER OR COUNT LOGIC FAILURE
TRACK CROSSING DETECTION FAILURE

REPEAT SEEK AND CHECKS UNTIL CYLINDER IS 'LOLIMIT'.

NOTE: IF THE 'USE ALL SECTORS' PARAMETER IS SPECIFIED AS 'Y', THE TEST WILL READ AND TEST ALL 40 HEADERS (CARTRIDGE VERIFY).

NOTE: TESTING WILL BE DONE BETWEEN UPPER AND LOWER CYLINDER LIMITS. CHOOSING A SINGLE SURFACE WILL LIMIT TESTING TO THAT SURFACE. THIS TEST WILL BE BYPASSED IF SURFACE 1 IS CHOSEN.

TEST 4 INCREMENTAL FORWARD SEEK HEAD 1 TEST

POSITION HEADS AT CYLINDER 'HILIMIT' USING SEEKS WITH DIFFERENCE OF ONE, HEAD 0.

DO SEEK WITH DIFFERENCE OF 1, SIGN 1, HEAD 1. WAIT FOR INTERRUPT, WAIT FOR DRIVE READY. CHECK READY IS SET IN 15 MS. IF NOT:

POSITIONING PROBLEM AT A SPECIFIC CYLINDER

DO READ HEADER, WAIT FOR INTERRUPT. CHECK THAT THIS CYLINDER IS OLD CYLINDER + 1. IF NOT:

DIFFERENCE REGISTER OR COUNT LOGIC FAILURE
TRACK CROSSING DETECTION FAILURE

REPEAT SEEKS AND READS UNTIL CYLINDER READ IS 'HILIMIT'.

NOTE 1: IF THE 'USE ALL SECTORS' PARAMETER IS SPECIFIED AS 'Y', THE TEST WILL READ AND TEST ALL 40 HEADERS (CARTRIDGE VERIFY).

NOTE 2: TESTING WILL BE DONE BETWEEN UPPER AND LOWER CYLINDER LIMITS. CHOOSING A SINGLE SURFACE WILL LIMIT TESTING TO THAT SURFACE. THIS TEST WILL BE BYPASSED IF SURFACE 0 IS CHOSEN.

TEST 5 INNER GUARD BAND DETECTION TEST

POSITION HEADS AT CYLINDER 'HILIMIT' USING SEEK WITH DIFFERENCE OF 1, HEAD 0.

WHEN AT MAX CYLINDER, DO SEEK WITH DIFFERENCE OF 1, SIGN 1, HEAD 0. WAIT FOR INTERRUPT, WAIT FOR DRIVE READY. READY SHOULD SET IN 20MS>T>15MS. IF NOT:

FAILED TO DETECT GUARD BAND

DO READ HEADER. WAIT FOR INTERRUPT. CHECK FOR MAX. CYLINDER IF NOT:

FAILED TO SEEK BACK TO MAX CYLINDER

DO SEEK WITH DIFFERENCE OF 1, SIGN 1, HEAD 1. DO SAME TESTS AS ABOVE.

NOTE: CHOOSING A SINGLE SURFACE WILL LIMIT THE TESTING TO

THAT SURFACE.

TEST 6 INCREMENTAL REVERSE SEEK HEAD 1 TEST

POSITION HEADS AT CYLINDER 'HILIMIT' USING SEEKS WITH DIFFERENCE OF 1, HEAD 0.

DO SEEK WITH DIFFERENCE OF 1, SIGN 0, HEAD 1. WAIT FOR INTERRUPT, WAIT FOR DRIVE READY. CHECK READY SET IN 15 MS:

POSITIONING PROBLEM AT A SPECIFIC CYLINDER

DO READ HEADER, WAIT FOR INTERRUPT. CHECK THAT THIS CYLINDER IS OLD CYLINDER - 1. IF NOT:

DIFFERENCE REGISTER OR COUNT LOGIC FAILURE
TRACK CROSSING DETECTION FAILURE

REPEAT SEEK AND CHECKS UNTIL CYLINDER IS 'LOLIMIT'.

NOTE 1: IF PROGRAM MODE 2 IS USED AND THE 'USE ALL SECTORS' PARAMETER IS SPECIFIED AS 'Y', THE TEST WILL READ AND TEST ALL 40 HEADERS (CARTRIDGE VERIFY).

NOTE 2: TESTING WILL BE DONE BETWEEN UPPER AND LOWER CYLINDER LIMITS. CHOOSING A SINGLE SURFACE WILL LIMIT TESTING TO THAT SURFACE. THE TEST WILL BE BYPASSED IF SURFACE 0 IS CHOSEN.

TEST 7 SEEK TESTS

POSITION HEADS AT CYLINDER 'LOLIMIT' USING SEEKS WITH DIFFERENCE OF 1, HEAD 0.

DO READ HEADER, RECORD POSITION. DO SEEK WITH DIFFERENCE OF 2 (MAX DISTANCE AT 3 IPS), SIGN 1, HEAD 0. DO READ HEADER, CHECK NEW CYLINDER IS OLD CYLINDER + DISTANCE. IF NOT:

TRACK CROSSING DETECTION FAILURE
DIFFERENCE COUNTER FAILURE
COUNT PULSE GENERATION FAILURE
VELOCITY ROM FAILURE

REPEAT ABOVE UNTIL OLD CYLINDER + DISTANCE > MAX. POSITION AT MAX.

DO READ HEADER, RECORD POSITION. DO SEEK WITH DIFFERENCE OF 2 (MAX DISTANCE AT 3 IPS), SIGN 0, HEAD 0. DO READ HEADER, CHECK NEW CYLINDER IS OLD CYLINDER - DISTANCE. IF NOT:

TRACK CROSSING DETECTION FAILURE

REPEAT UNTIL OLD CYLINDER - DISTANCE < 0. REPEAT ALL OF THE ABOVE USING HEAD 1.

REPEAT ALL OF THE ABOVE TESTS USING THE FOLLOWING DISTANCES: 2, 6, 9, 12, 17, 22, 27, 34, 41, 128, 256 FOR RL01 OR 4, 12, 18, 24, 34, 44, 54, 68, 82, 256, 512 FOR RL02. THESE DISTANCES ARE SPECIFIED BECAUSE THEY REPRESENT THE MAXIMUM DISTANCE FOR EACH VELOCITY LEVEL USED IN THE DRIVE.

NOTE: TESTING WILL BE DONE BETWEEN UPPER AND LOWER CYLINDER LIMITS. CHOOSING A SINGLE SURFACE WILL LIMIT TESTING TO THAT SURFACE.

TEST 8 FORWARD OSCILLATING SEEK TEST

POSITION HEADS AT CYLINDER 0.

DO OSCILLATING SEEK USING HEAD 0 (SEEK FROM 0 TO 1 TO 0, 0 TO 2 TO 0, 0 TO 3 TO 0, ... 0 TO MAX CYL TO 0). AFTER EACH SEEK READ HEADER AND VERIFY POSITION.

REPEAT TEST USING HEAD 1.

NOTE: IF EITHER CYLINDER LIMIT IS SPECIFIED, THE TEST WILL SEEK BETWEEN UPPER AND LOWER LIMITS FOR EACH SURFACE. CHOOSING A SINGLE SURFACE WILL LIMIT TESTING TO THAT SURFACE. NOTE THAT LOOPING ON TEST THEN PROVIDES A FIXED DISTANCE SEEK LOOP.

TEST 9 REVERSE OSCILLATING SEEK TEST

POSITION HEADS AT MAX CYLINDER. DO OSCILLATING SEEK USING HEAD 0. (IF RL01 SEEK FROM 255 TO 254 TO 255, 255 TO 253 TO 255, ... 255 TO 0 TO 255.) AFTER EACH SEEK READ HEADER AND VERIFY POSITION.

REPEAT TEST USING HEAD 1.

NOTE: IF EITHER CYLINDER LIMIT IS SPECIFIED, THE TEST WILL SEEK BETWEEN UPPER AND LOWER LIMITS FOR EACH SURFACE. CHOOSING A SINGLE SURFACE WILL LIMIT TESTING TO THAT SURFACE. NOTE THAT LOOPING ON TEST THEN PROVIDES A FIXED DISTANCE SEEK LOOP.

MAIN. MACY11 30A(1052) 08-FEB-80 14:49
CZRLJB.MAC 07-DEC-79 09:06 TABLE OF CONTENTS

SEQ 0034

18	MACRO DEFINITIONS
88	GLOBAL DATA SECTION
222	GLOBAL DATA SECTION
643	GLOBAL MESSAGES
874	ERROR MESSAGES
1210	INITIALIZATION SECTION
1343	AUTO DROP SECTION
1383	CLEANUP CODE SECTION
1413	GLOBAL SUBROUTINES
2664	*TEST 1 **OUTER GUARD BAND DETECTION
2717	*TEST 2 **INCREMENTAL FORWARD SEEK HEAD 0
2765	*TEST 3 **INCREMENTAL REVERSE SEEK HEAD 0
2812	*TEST 4 **INCREMENTAL FORWARD SEEK HEAD 1
2864	*TEST 5 **INNER GUARD BAND DETECTION
2912	*TEST 6 **INCREMENTAL REVERSE SEEK HEAD 1
2961	*TEST 7 **SEEK TESTS
3025	*TEST 8 **FORWARD OSCILLATING SEEK
3084	*TEST 9 **REVERSE OSCILLATING SEEK
3147	PARAMETER CODING

1			
2	000001	PART2==1	
3			.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			

```
17
18 .SBTTL MACRO DEFINITIONS
19
20 .MACRO WAITUS ARG ;MACRO MICRO-SEC WAIT
21 MOV ARG,XDELAY ;SAVE ARGUMENT
22 JSR PC,TIME ;CALL TIMING ROUTINE
23 .ENDM
24
25 .MACRO WAITMS ARG ;MACRO MILLI-SEC WAIT
26 MOV ARG,YDELAY ;SAVE ARGUMENT
27 JSR PC,XTIME ;CALL TIMING ROUTINE
28 .ENDM
29
30 .MACRO ABORTWAIT ;MACRO CLEAR UNELAPSED TIME
31 MOV XDELAY,TEMPO ;SAVE MICRO-SEC RUN TIME
32 MOV YDELAY,TEMP ;SAVE MILLI-SEC RUN TIME
33 CLR XDELAY ;ABORT MICRO-SEC WAIT
34 CLR YDELAY ;ABORT MILLI-SEC WAIT
35 .ENDM
36
37 .MACRO GETTIM ARG ;MACRO GET ELAPSED TIME
38 MOV @#CLKCTR,ARG ;STORE CLOCK COUNTER CONTENTS
39 CLR @#CLKCSR ;EVENT FINISHED, STOP CLOCK
40 .ENDM
41
42 .MACRO STCLK ;MACRO START P-CLOCK
43 CLR @#CLKCSB ;CLEAR CLOCK COUNT SET BUFFER
44 CLR @#CLKCTR ;CLEAR CLOCK COUNTER
45 MOV #23,@#CLKCSR ;INITIALIZE CLOCK FOR COUNT-UP MODE,
46 ;/10 KHZ RATE, AND START CLOCK
47 .ENDM
48
49
50
```

```
52
53          .NLIST  CND,MD,ME
54
55
56          002000          POINTER  BGNSW,BGNSFT,BGNDU
57
58          002000          BGNMOD  MDHEDR
59          002000          HEADER  CZRLJ,B,0,30000,0
60          (4) 002000          103          .ASCII  /C/
61          (4) 002001          132          .ASCII  /Z/
62          (4) 002002          122          .ASCII  /R/
63          (4) 002003          114          .ASCII  /L/
64          (4) 002004          112          .ASCII  /J/
65          (6) 002005          000          .BYTE   0
66          (6) 002006          000          .BYTE   0
67          (5) 002007          000          .BYTE   0
68          (4) 002010          102          .ASCII  /B/
69          (4) 002011          060          .ASCII  /O/
70          (4) 002012          000000       .WORD   0
71          (4) 002014          030000       .WORD   30000
72          (4) 002016          030624       .WORD   LSHARD
73          (4) 002020          031000       .WORD   L$SOFT
74          (4) 002022          013540       .WORD   L$HW
75          (4) 002024          013556       .WORD   L$SW
76          (4) 002026          031342       .WORD   L$LAST
77          (4) 002030          000000       .WORD   0
78          (4) 002032          000000       .WORD   0
79          (4) 002034          000000       .WORD   0
80          (4) 002036          000000       .WORD   0
81          (4) 002040          013574       .WORD   L$DISPATCH
82          (4) 002042          000000       .WORD   0
83          (4) 002044          000000       .WORD   0
84          (4) 002046          000000       .WORD   0
85          (4) 002050          003          .BYTE   C$REVISION
86          (3) 002051          003          .BYTE   C$EDIT
87          (4) 002052          000000       .WORD   0
88          (5) 002054          000000       .WORD   0
89          (4) 002056          000000       .WORD   0
90          (4) 002060          002226       .WORD   L$DVTYP
91          (4) 002062          000000       .WORD   0
92          (4) 002064          000000       .WORD   0
93          (4) 002066          000000       .WORD   0
94          (4) 002070          000000       .WORD   0
95          (4) 002072          015252       .WORD   L$DU
96          (4) 002074          000000       .WORD   0
97          (4) 002076          002122       .WORD   L$DESC
98          (4) 002100          104035       EMT     E$LOAD
99          (4) 002102          000000       .WORD   0
100         (4) 002104          013616       .WORD   L$INIT
101         (4) 002106          015124       .WORD   L$CLEAN
102         (4) 002110          014566       .WORD   L$AUTO
103         (4) 002112          013530       .WORD   L$PROT
104         (4) 002114          000000       .WORD   C
105         (4) 002116          000000       .WORD   0
106         (4) 002120          000000       .WORD   0
107         65 002122          ENDMOD
```

66 002122
(3) 002122 055103 046122 020112
(3) 002130 042524 052123 020123
(3) 002136 052517 042524 020122
(3) 002144 020046 047111 042516
(3) 002152 020122 052507 051101
(3) 002160 020104 040502 042110
(3) 002166 042040 052105 041505
(3) 002174 044524 047117 040440
(3) 002202 042116 051440 042505
(3) 002210 020113 050117 051105
(3) 002216 052101 047511 051516
(3) 002224 000

DESCRIPT <CZRLJ TESTS OUTER & INNER GUARD BAND DETECTION AND SEEK OPERATIONS>
.ASCIZ /CZRLJ TESTS OUTER & INNER GUARD BAND DETECTION AND SEEK OPERATIONS/

(2) 002226 002226
67 002226
(3) 002226 046122 030460 051054
(3) 002234 030114 000062

.EVEN
DEV TYP <RL01,RL02>
.ASCIZ /RL01,RL02/

(2)

.EVEN

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

(1)

(1)

(1)

(1)

(1)

(1)

(1)

(1)

(1)

(1)

(1)

(1)

(1)

(1)

002240

002240

100000
040000
020000
010000
004000
002000
001000
000400
000200
000100

.SBTTL GLOBAL DATA SECTION

BGNMOD GLBEQAT

EQUALS

; BIT DEFINITIONS

.
BIT15== 100000
BIT14== 40000
BIT13== 20000
BIT12== 10000
BIT11== 4000
BIT10== 2000
BIT09== 1000
BIT08== 400
BIT07== 200
BIT06== 100

```
(1) 000040 BIT05== 40
(1) 000020 BIT04== 20
(1) 000010 BIT03== 10
(1) 000004 BIT02== 4
(1) 000002 BIT01== 2
(1) 000001 BIT00== 1
(1)
(1) 001000 BIT9== BIT09
(1) 000400 BIT8== BIT08
(1) 000200 BIT7== BIT07
(1) 000100 BIT6== BIT06
(1) 000040 BIT5== BIT05
(1) 000020 BIT4== BIT04
(1) 000010 BIT3== BIT03
(1) 000004 BIT2== BIT02
(1) 000002 BIT1== BIT01
(1) 000001 BIT0== BIT00
(1)
(1) :
(1) : EVENT FLAG DEFINITIONS
(1) : EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION
(1) :
(1) :
(1) 000040 EF.START== 32. ; START COMMAND WAS ISSUED
(1) 000037 EF.RESTART== 31. ; RESTART COMMAND WAS ISSUED
(1) 000036 EF.CONTINUE== 30. ; CONTINUE COMMAND WAS ISSUED
(1) 000035 EF.NEW== 29. ; A NEW PASS HAS BEEN STARTED
(1) 000034 EF.PWR== 28. ; A POWER-FAIL/POWER-UP OCCURRED
(1) :
(1) :
(1) : PRIORITY LEVEL DEFINITIONS
(1) :
(1) 000340 PRI07== 340
(1) 000300 PRI06== 300
(1) 000240 PRI05== 240
(1) 000200 PRI04== 200
(1) 000140 PRI03== 140
(1) 000100 PRI02== 100
(1) 000040 PRI01== 40
(1) 000000 PRI00== 0
(1) :
(1) : OPERATOR FLAG BITS
(1) :
(1) 000004 EVL== 4
(1) 000010 LOT== 10
(1) 000020 ADR== 20
(1) 000040 IDU== 40
(1) 000100 ISR== 100
(1) 000200 UAM== 200
(1) 000400 BOF== 400
(1) 001000 PNT== 1000
(1) 002000 PRI== 2000
(1) 004000 IXE== 4000
(1) 010000 IBE== 10000
(1) 020000 IER== 20000
(1) 040000 LOE== 40000
(1) 100000 HOE== 100000
93 : OFFSETS FOR HARDWARE P-TABLE
```

94	000000	CSR	=0	:BUS ADDRESS
95	000002	VECT	=2	:VECTOR ADDRESS
96	000004	PRIOR	=4	:PRIORITY
97	000006	TYPDR=6		
98	000010	DRSB	=10	:DRIVE SELECT BIT
99	000012	CNT	=12	:CONTROLLER TYPE
100				
101		:	OFFSET FOR SOFTWARE P-TABLE	
102	000000	MISWI	=0	:SOFTWARE PARAMETERS SWITCHES
103	000002	LOLIM	=2	:CYLINDER LOWER LIMIT
104	000004	HILIM	=4	:CYLINDER HIGH LIMIT
105	000006	HEAD	=6	:SELECTED HEAD FOR RUNNING TESTS
106	000010	ERLIM	=10	:ERROR LIMIT
107	000012	DCLIM	=12	:DATA COMPARE ERROR LIMIT
108				
109		:	BIT ASSIGNMENT FOR SOFTWARE P-TABLE SWITCHES	
110	000001	ALLCYL	=BIT00	:USE ALL CYLINDERS
111	000002	ALLSEC	=BIT01	:USE ALL SECTORS
112	000004	DRSELT	=BIT02	:EXECUTE DRIVE SELECT TEST
113	000010	HDALIGN	=BIT03	:EXECUTE HEAD ALIGNMENT TEST
114	010000	HEADLM	=BIT12	:HEAD LIMIT SPECIFIED FLAG
115	020000	HICYL	=BIT13	:HI LIMIT SPECIFIED FLAG
116	040000	LOCYL	=BIT14	:LO LIMIT SPECIFIED
117	100000	MITEST	=BIT15	:EXECUTE MANUAL INTERVENTION TESTS
118				
119		:	SUBSYSTEM FUNCTIONS	
120	000102	CKDATA	=102	:WRITE CHECK
121	000104	GTSTAT	=104	:GET STATUS
122	000106	SEEK	=106	:SEEK
123	000110	RDHEAD	=110	:READ HEADER
124	000112	WTDATA	=112	:WRITE DATA
125	000114	RDDATA	=114	:READ DATA
126	000116	RDNCHR	=116	:READ DATA, IGNORE HEADERS
127	000100	NOOP	=100	:NO OPERATION
128				
129		:	OPERATION FLAGS	
130	007777	COMPOP	=7777	:COMPOSITE OPERATION FLAGS
131	000002	HDRCMP	=BIT0	:HEADER COMPARE OPERATION
132	000001	DATAcmp	=BIT00	:DATA COMPARE OPERATION
133	000004	CYLUPL	=BIT02	:CYCLE UP OPERATION
134	000010	UNLOAD	=BIT03	:UNLOAD OPERATION
135	000020	INOUTS	=BIT04	:IN-OUT SEEK OPERATION
136	000040	OUTINS	=BIT05	:OUT-IN SEEK OPERATION
137	000100	FOLWRT	=BIT06	:FOLLOWING WRITE OPERATION
138	000200	REVSKS	=BIT07	:REV SEEK SEQ (ADJ INTERFERENCE)
139	000400	FWDSKS	=BIT08	:FWD SEEK SEQ (ADJ INTERFERENCE)
140	001000	REVSKO	=BIT09	:REV SEEK SEQ (OVERWRITE)
141	002000	FWDSKO	=BIT10	:FWD SEEK SEQ (OVERWRITE)
142	004000	BADADD	=BIT11	:BAD DISK ADDRESS
143	010000	SEEKOP	=BIT12	:SEEK OPERATION
144	020000	RORWOP	=BIT13	:READ OR WRITE OPERATION
145	040000	RFLDWT	=BIT14	:RELOAD WAIT
146	100000	HDR40	=BIT15	:40 HEADER OPERATION
147	003760	MQUALS	=OUTINS!INOUTS!FOLWRT!REVSKS!FWDSKS!REVSKO!FWDSKO	:MESSAGE QUALIFIER BITS
148				
149				


```
150      ;      ERROR FLAGS FROM SUBROUTINES
151      000001  ;TOSLOW =BIT00      ;OPERATION TOOK TOO LONG
152      000002  ;NOIRPT =BIT01      ;NO INTERRUPT FROM OPERATION
153      000004  ;CONHNG =BIT02      ;CONTROLLER HUNG
154      000010  ;NOCLR  =BIT03      ;BAD CONTROLLER CLEAR
155
156      000000  ;RICS   =0      ;CONTROL AND STATUS REGISTER
157      000002  ;RLBA   =2      ;BUS ADDRESS REGISTER
158      000004  ;RLDA   =4      ;DISK ADDRESS REGISTER
159      000006  ;RLMP   =6      ;MULTI-PURPOSE REGISTER
160
161      ;      REGISTER BIT DEFINITIONS - CONTROL STATUS REGISTER
162      000000  ;RLCSR  =0      ;CONTROL AND STATUS REGISTER
163      100000  ;ANYERR =100000   ;ANY ERROR BIT
164      040000  ;DRVERR =40000    ;DRIVE ERROR BIT
165      020000  ;NXMERR =20000    ;NON-EXISTANT MEMORY ERROR
166      010000  ;DLTERR =10000    ;DATA LATE ERROR
167      010000  ;HNFERR =10000    ;HEADER NOT FOUND ERROR
168      004000  ;DCKERR =4000     ;DATA CHECK ERROR
169      004000  ;HRCERR =4000     ;HEADER CHECK ERROR
170      002000  ;OPIERR =2000     ;OPERATION INCOMPLETE ERROR
171      001400  ;DSMSK  =1400    ;DRIVE SELECT MASK
172      000200  ;CRDYMSK =200      ;CONTROLLER READY MASK
173      000100  ;INTEBL =100      ;INTERRUPT ENABLE MASK
174      000060  ;BAMSK  =60      ;BUS ADDRESS UPPER MASK
175      000001  ;DRDYMSK =1      ;DRIVE READY MASK
176
177      ;      REGISTER BIT DEFINITIONS - DISK ADDRESS FOR DATA XFER
178      000077  ;SAMSK  =77      ;SECTOR ADDRESS MASK
179      000100  ;HSMSK  =100     ;HEAD SELECT MASK
180
181      ;      REGISTER BIT DEFINITIONS - DISK ADDRESS FOR SEEK
182      000001  ;MBSETO =1      ;MUST BE SET, BIT 0
183      000004  ;DIRBIT  =4      ;DIRECTION BIT
184      000020  ;HDSEL   =20     ;HEAD SELECT BIT
185
186      ;      REGISTER BIT DEFINITIONS - DISK ADDRESS FOR GET STATUS
187      000003  ;GETSTAT =3      ;GET STATUS SETUP
188      000010  ;DRSET   =10     ;DRIVE RESET MASK
189
190      ;      REGISTER BIT DEFINITIONS - MP FOR DATA XFER
191      017777  ;WCMSK  =17777   ;WORD COUNT MASK
192      160000  ;WCRNG  =160000  ;WORD COUNT RANGE MASK
193
194      ;      REGISTER BIT DEFINITIONS - MP FOR READ HEADER
195      000077  ;HDSEC  =77      ;SECTOR MASK
196      000100  ;HDHSEL  =100     ;HEAD SELECT MASK
197
198      ;      REGISTER BIT DEFINITIONS - MP FOR GET STATUS
199      000007  ;STAMSK  =7      ;STATE MASK
200      000010  ;BHSTAT  =10     ;BRUSH HOME STATUS
201      000020  ;HOSTAT  =20     ;HEADS OUT STATUS
202      000040  ;COSTAT  =40     ;COVER OPEN STATUS
203      000100  ;HSSTAT  =100    ;HEAD SELECT STATUS
204      000400  ;DSESTAT =400    ;DRIVE SELECT ERROR STATUS
205      001000  ;VCSTAT  =1000   ;VOLUME CHECK STATUS
```

```
206          002000          WGESTAT =2000          :WRITE GATE ERROR STATUS
207          004000          SPDSTAT =4000          :SPIN ERROR STATUS
208          010000          STOSTAT =10000         :SEEK TIMEOUT ERROR STATUS
209          020000          WLSTAT =20000         :WRITE LOCK STATUS
210          040000          HCESTAT =40000        :HEAD CURRENT ERROR STATUS
211          100000          WDESTAT =100000       :WRITE DATA ERROR STATUS
212
213          :              P-CLOCK REGISTERS
214          172540          CLKCSR =172540         :CLOCK CONTROL AND STATUS REG. STER
215          172542          CLKCSB =172542         :CLOCK COUNT SET BUFFER
216          172544          CLKCTR =172544         :CLOCK COUNTER
217
218 002240          ENDMOD
219
220
221
222          .SBTTL GLOBAL DATA SECTION
223
224 002240          BGNMOD GLBDAT
225
226          :              TABLE OF OPERATION MESSAGES
227
228 002240 000000          OPMSGs: .WORD 0          :FILLER
229 002242 005405          .WORD MWRCHK          :MESSAGE FOR WRITE CHECK
230 002244 005430          .WORD MGTSTA          :GET STATUS
231 002246 005360          .WORD MSEEK          :SEEK
232 002250 005375          .WORD MREADH          :READ HEADER
233 002252 005416          .WORD MWRITE          :WRITE DATA
234 002254 005364          .WORD MREAD          :READ DATA
235 002256 005513          .WORD MWRSET          :WITH RESET
236 002260 005442          .WORD MDATCP          :WITH DATA COMPARE
237 002262 005461          .WORD MHDRCP          :WITH HEADER COMPARE
238 002264 005560          .WORD MCYLUP          :LOAD HEADS
239 002266 005547          .WORD MLOAD          :UNLOAD HEADS
240 002270 005607          .WORD MINOUT          :IN-OUT SEQ
241 002272 005570          .WORD MOUTIN          :OUT-IN SEQ
242 002274 005630          .WORD MFOLWRT          :FOLLOWING WRITE
243 002276 005650          .WORD MREVSK          :REV SEEK
244 002300 005701          .WORD MFWDSK          :FWD SEEK
245 002302 005766          .WORD MRESKO          :REV SEEK
246 002304 005732          .WORD MFWSKO          :FWD SEEK
247 002306 006022          .WORD MBADAD          :BAD DISK ADD FOR WRITE
248 002310 005477          .WORD M40HDR          :40 HEADER OPERATION
249 002312 000000          T.DRIVE: .WORD 0
250 002314 000000          JJJ: .WORD 0
251 002316 000000          HLMTW: .WORD 0
252 002320 000000          CLRBYT: .WORD 0
253 002322 000000          NXTHL: .WORD 0
254 002324 000000          GBND: .WORD 0
255 002326 000000          CAMSK: .WORD 0
256 002330 000000          DIRMSK: .WORD 0
257 002332 000000          HDCYL: .WORD 0
258
259          :              TABLE OF RESULT NAME MESSAGE ADDRESSES
260 002334 007771          RESTBL: .WORD MCERR          :CONTROLLER ERROR
261 002336 010102          .WORD MDRERR          :DRIVE ERROR
```

262	002340	010320	.WORD	MNEERR	.NON-EXISTANT MEMORY ERROR
263	002342	010272	.WORD	MFLERR	;HEADER NOT FOUND-DATA LATE
264	002344	010255	.WORD	MHDERR	;HEADER OR DATA ERROR
265	002346	010245	.WORD	MOPERR	;OPERATION INCOMPLETE
266	002350	010352	.WORD	MNDRST	;NO DRIVE STATUS AVAILABLE
267	002352	000000	.WORD	0	
268	002354	010230	.WORD	MWDERR	;WRITE DATA ERROR
269	002356	010212	.WORD	MHCERR	;HEAD CURRENT ERROR
270	002360	000000	.WORD	0	
271	002362	010176	.WORD	MSTERR	;SEEK TIMEOUT ERROR
272	002364	010143	.WORD	MSPERR	;SPINDLE ERROR
273	002366	010161	.WORD	MWGERR	;WRITE GATE ERROR
274	002370	000000	.WORD	0	
275	002372	010113	.WORD	MDSERR	;DRIVE SELECT ERROR

276
277
278 002374 005102
279 002376 005104
280 002400 005144
281 002402 005204
282 002404 005244
283 002406 005252
284 002410 005312
285 002412 005314
286 002414 005354
287 002416 005356
288
289

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

290
291 002420 000000
292 002422 000000
293 002424 000000
294 002426 000000
295 002430 000000
296 002432 000000
297 002434 000000
298 002436 000000
299 002440 000000
300 002442 000000
301

: 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
.WORD 0

302
303 002444 000002
304 002446 000006
305 002450 000011
306 002452 000014
307 002454 000021
308 002456 000026
309 002460 000033
310 002462 000042
311 002464 000051
312 002466 000200
313 002470 000377
314

: RL01 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

315
316 002472 000004
317 002474 000014

: RL02 TABLE OF CYLINDERS
T25TB2: .WORD 4
.WORD 12

318	002476	000022	.WORD	18.
319	002500	000030	.WORD	24.
320	002502	000042	.WORD	34.
321	002504	000054	.WORD	44.
322	002506	000066	.WORD	54.
323	002510	000104	.WORD	68.
324	002512	000122	.WORD	82.
325	002514	000400	.WORD	256.
326	002516	000777	.WORD	511.

; TABLE TO BE USED TO BUILD AND STORE THE CYLINDERS

330	002520	000020	T33TBL: .BLKW	16.
331	002560	000020	TBT: .BLKW	16.

CYLtbl: .BYTE 2 ;TABLE OF DEFAULT CYLINDERS

334	002620	002	.BYTE	2.
335	002621	007	.BYTE	7.
336	002622	016	.BYTE	14.
337	002623	024	.BYTE	20.
338	002624	033	.BYTE	27.
339	002625	041	.BYTE	33.
340	002626	046	.BYTE	38.
341	002627	055	.BYTE	45.
342	002630	064	.BYTE	52.
343	002631	072	.BYTE	58.
344	002632	101	.BYTE	65.
345	002633	110	.BYTE	72.
346	002634	115	.BYTE	77.
347	002635	124	.BYTE	84.
348	002636	133	.BYTE	91.
349	002637	141	.BYTE	97.
350	002640	146	.BYTE	102.
351	002641	154	.BYTE	108.
352	002642	161	.BYTE	113.
353	002643	170	.BYTE	120.
354	002644	177	.BYTE	127.
355	002645	206	.BYTE	134.
356	002646	213	.BYTE	139.
357	002647	222	.BYTE	146.
358	002650	230	.BYTE	152.
359	002651	235	.BYTE	157.
360	002652	244	.BYTE	164.
361	002653	252	.BYTE	170.
362	002654	261	.BYTE	177.
363	002655	270	.BYTE	184.
364	002656	275	.BYTE	189.
365	002657	303	.BYTE	195.
366	002660	312	.BYTE	202.
367	002661	317	.BYTE	207.
368	002662	326	.BYTE	214.
369	002663	334	.BYTE	220.
370	002664	343	.BYTE	227.
371	002665	352	.BYTE	234.
372	002666	361	.BYTE	241.
373	002667	367	.BYTE	247.

374	002670	375	.BYTE	253.
375	002671	000	.BYTE	0
376	002672	000401	.WORD	257.
377	002674	000406	.WORD	262.
378	002676	000415	.WORD	269.
379	002700	000423	.WORD	275.
380	002702	000432	.WORD	282.
381	002704	000445	.WORD	293.
382	002706	000454	.WORD	300.
383	002710	000463	.WORD	307.
384	002712	000471	.WORD	313.
385	002714	000500	.WORD	320.
386	002716	000507	.WORD	327.
387	002720	000514	.WORD	332.
388	002722	000523	.WORD	339.
389	002724	000532	.WORD	346.
390	002726	000540	.WORD	352.
391	002730	000545	.WORD	357.
392	002732	000553	.WORD	363.
393	002734	000560	.WORD	368.
394	002736	000567	.WORD	375.
395	002740	000576	.WORD	382.
396	002742	000605	.WORD	389.
397	002744	000612	.WORD	394.
398	002746	000621	.WORD	401.
399	002750	000627	.WORD	407.
400	002752	000634	.WORD	412.
401	002754	000643	.WORD	419.
402	002756	000651	.WORD	425.
403	002760	000660	.WORD	432.
404	002762	000667	.WORD	439.
405	002764	000674	.WORD	444.
406	002766	000702	.WORD	450.
407	002770	000711	.WORD	457.
408	002772	000716	.WORD	462.
409	002774	000725	.WORD	469.
410	002776	000733	.WORD	475.
411	003000	000742	.WORD	482.
412	003002	000751	.WORD	489.
413	003004	000760	.WORD	496.
414	003006	000766	.WORD	502.
415	003010	000774	.WORD	508.
416	003012	000774	.WORD	508.
417	003014	000000	.WORD	0
418	003016	000000	.WORD	0
419				
420				
421	003020	000000	OPFLAG: .WORD	0
422	003022	000000	DONE: .WORD	0
423	003024	000000	HADONE: .WORD	0
424	003026	000000	ERHEAD: .WORD	0
425	003030	000000	MORECE: .WORD	0
426	003032	000000	ERRSWI: .WORD	0
427	003034	000000	BSFLAG: .WORD	0
428	003036	000000	WRTSWI: .WORD	0
429	003040	000000	TBLSTR: .WORD	0

:SUBROUTINE STACK INDEX POINTER

: OPERATIONAL FLAGS

:OPERATION FLAGS
:OPERATION COMPLETE FLAG
:HEAD ALIGNMENT DONE FLAG
:ADDRESS OF ERROR HEADER
:MORE THAN 1 COMPARE ERROR
:ERROR RETURN SWITCH
:BAD SECTOR FLAGS
:WRITE SWITCH
:TABLE STORAGE

430					
431	003042	000000	RLBAS: .WORD	0	;RL11 BASE ADDRESS
432	003044	000000	RLVEC: .WORD	0	;RL11 VECTOR ADDRESS
433	003046	000000	RLDRV: .WORD	0	;DRIVE NUMBER UNDER TEST
434					
435	003050	000000	L.CS: .WORD	0	;CONTROLLER REGISTER STORAGE
436	003052	000000	L.BA: .WORD	0	;BEFORE OPERATION
437	003054	000000	L.DA: .WORD	0	
438	003056	000000	L.MP: .WORD	0	
439	003060	000000	T.CS: .WORD	0	;CONTROLLER REGISTER STORAGE
440	003062	000000	T.BA: .WORD	0	; AFTER OPFRATION
441	003064	000000	T.DA: .WORD	0	
442	003066		T.MP: .WORD	0	
443	003066	000000	HDWRD1: .WORD	0	;HEADER WORD STORAGE
444	003070	000000	HDWRD2: .WORD	0	
445	003072	000000	HDWRD3: .WORD	0	
446					
447	003074	000000	T.STAT: .WORD	0	;DRIVE STATE STORAGE
448					
449	003076	000000	RESPARM: .WORD	0	;PARAM BLOCK FOR REASON REPORT
450	003100	000000	.WORD	0	
451	003102	000000	.WORD	0	
452	003104	000000	.WORD	0	
453	003106	000000	.WORD	0	
454					
455	003110	000000	DRVCNT: .WORD	0	;DRIVE COUNT FOR DRIVES UNDER TEST
456	003112	000000	DIFAUG: .WORD	0	;DIFFERENCE AUGMENT FOR SEEK
457	003114	000000	OLDCYL: .WORD	0	;OLD CYLINDER
458	003116	000000	NEWCYL: .WORD	0	;NEW CYLINDER
459	003120	000000	CURCYL: .WORD	0	;CURRENT CYLINDER
460	003122	000000	DESDIF: .WORD	0	;DESIRED DIFFERENCE
461	003124	000000	DESSGN: .WORD	0	;DESIRED SIGN
462	003126	000000	DESHD: .WORD	0	;DESIRED HEAD
463	003130	000000	DESSEC: .WORD	0	;DESIRED SECTOR
464	003132	000000	TEMP0: .WORD	0	;TEMPORARY STORAGE
465	003134	000000	TEMP1: .WORD	0	;TEMPORARY STORAGE
466	003136	000000	TEMP2: .WORD	0	;TEMPORARY STORAGE
467	003140	000000	TEMP3: .WORD	0	;TEMPORARY STORAGE
468	003142	000000	TEMP4: .WORD	0	;TEMPORARY STORAGE
469	003144	000000	TEMP5: .WORD	0	;TEMPORARY STORAGE
470	003146	000000	TEMP6: .WORD	0	;TEMPORARY STORAGE
471	003150	000000	TEMP7: .WORD	0	;TEMPORARY STORAGE
472	003152	000000	TEMP8: .WORD	0	;TEMPORARY STORAGE
473					
474					
475	003154	000000	OFIN: .WORD	0	;ONE CYLINDER FORWARD INNER
476	003156	000000	OFINU: .WORD	0	; UPPER
477	003160	000000	OFMID: .WORD	0	;ONE CYLINDER FORWARD MIDDLE
478	003162	000000	OFMIDU: .WORD	0	; UPPER
479	003164	000000	OFOUT: .WORD	0	;ONE CYLINDER FORWARD OUTER
480	003166	000000	OFOUTU: .WORD	0	; UPPER
481	003170	000000	ORIN: .WORD	0	;ONE CYLINDER REVERSE INNER
482	003172	000000	ORINU: .WORD	0	; UPPER
483	003174	000000	ORMID: .WORD	0	;ONE CYLINDER REVERSE MIDDLE
484	003176	000000	ORMIDU: .WORD	0	; UPPER
485	003200	000000	OROUT: .WORD	0	;ONE CYLINDER REVERSE OUTER
486	003202	000000	OROUTU: .WORD	0	; UPPER

487	003204	000000	HF IN: .WORD	0	:128 CYLINDER FORWARD INNER
488	003206	000000	HF INU: .WORD	0	: UPPER
489	003210	000000	HF OUT: .WORD	0	:128 CYLINDER FORWARD OUTER
490	003212	000000	HF OUTU: .WORD	0	: UPPER
491	003214	000000	HR IN: .WORD	0	:128 CYLINDER REVERSE INNER
492	003216	000000	HR INU: .WORD	0	: UPPER
493	003220	000000	HR OUT: .WORD	0	:128 CYLINDER REVERSE OUTER
494	003222	000000	HR OUTU: .WORD	0	: UPPER
495	003224	000000	AF MID: .WORD	0	:256 CYLINDER FORWARD
496	003226	000000	AF MIDU: .WORD	0	: UPPER
497	003230	000000	ARM ID: .WORD	0	:256 CYLINDER REVERSE
498	003232	000000	ARM IDU: .WORD	0	: UPPER
499					
500	003234	000226	EXOCYL: .WORD	150.	:EXPECTED TIME ONE CYLINDER
501	003236	001046	EXHCYL: .WORD	550.	:EXPECTED TIME 128 CYLINDER
502	003240	001750	EXACYL: .WORD	1000.	:EXPECTED TIME 256 CYLINDER
503	003242	000372	EXROT: .WORD	250.	:EXPECTED ROTATION TIME
505	003244	000004	ERRVEC: .WORD	4	:ERROR VECTOR
506					
507			:		MISCELLANEOUS COUNTERS
508	003246	000000	PASCNT: .WORD	0	:PASS COUNTER (LOCAL TO A TEST)
509	003250	000000	COUNT: .WORD	0	:A COUNTER (LOCAL TO A TEST)
510	003252	000000	ERRPOINT: .WORD	0	:ERROR POINTER
511	003254	000100	ERRCNT: .BLKW	64.	:ERROR COUNTER FOR PROGRAM
512	003454	000000	PASNUM: .WORD	0	:PASS NUMBER FOR PROGRAM
513	003456	000000	PSETNM: .WORD	0	:COUNTER FOR PARAMETER SET NUMBER IN USE
514	003460	000	LOCERR: .BYTE	0	:LOCAL ERROR COUNTER
515	003461	000	NOERCT: .BYTE	0	:INHIBIT ERROR COUNTING FLAG
516	003462	000000	TRPFLG: .WORD	0	:HARDWARE TRAP OCCURANCE
517	003464	000000	PWRFLG: .WORD	0	:POWER FAILURE OCCURANCE
518	003466	000000	XDELAY: .WORD	0	
519	003470	000000	YDELAY: .WORD	0	
520	003472	000000	MININC: .WORD	0	
521	003474	000000	TEMP: .WORD	0	
522	003476	000000	TIM.US: .WORD	0	
523	003500	000000	TAG: .WORD	0	
524	003502	000000	MAJINC: .WORD	0	
525	003504	000000	CLKFLG: .WORD	0	:FLAG INDICATING PRESENCE OF A P-CLOCK
526	003506	000000	CLKADR: .WORD	0	:POINTER TO DIAGNOSTIC MONITOR CLOCK TABLE
527					
528					
529			:		BAD SECTOR TABLES AND POINTERS
530	003510	000000	BSFVAL: .WORD	0	:BAD SECTORS FILES VALID FLAG
531					
532	003512	000076	SBSFIL: .BLKW	76	:SOFTWARE BAD SECTOR FILE
533	003706	000076	FBSFIL: .BLKW	76	:FACTORY BAD SECTOR FILE
534					
535	004102	000200	IBUFF: .BLKW	200	:INPUT BUFFER
536	004502	000200	OBUFF: .BLKW	200	:OUTPUT BUFFER
537					
538	005102	000000	PAT1: .WORD	0	:PATTERN 1 (ALL ZEROS)
539	005104	177772	PAT2: .WORD	177772	
540	005106	177777	.WORD	177777	
541	005110	177777	.WORD	177777	
542	005112	052525	.WORD	052525	
543	005114	052525	.WORD	052525	

544	005116	052525		.WORD	052525
545	005120	177777		.WORD	177777
546	005122	177777		.WORD	177777
547	005124	052525		.WORD	052525
548	005126	052525		.WORD	052525
549	005130	177777		.WORD	177777
550	005132	052525		.WORD	052525
551	005134	177252		.WORD	177252
552	005136	177252		.WORD	177252
553	005140	172765		.WORD	172765
554	005142	172765		.WORD	172765
555					
556	005144	000003	PAT3:	.WORD	000003
557	005146	000000		.WORD	000000
558	005150	000000		.WORD	000000
559	005152	177777		.WORD	177777
560	005154	177777		.WORD	177777
561	005156	177777		.WORD	177777
562	005160	000000		.WORD	000000
563	005162	000000		.WORD	000000
564	005164	177777		.WORD	177777
565	005166	177777		.WORD	177777
566	005170	000000		.WORD	000000
567	005172	177777		.WORD	177777
568	005174	000000		.WORD	000000
569	005176	177777		.WORD	177777
570	005200	000000		.WORD	000000
571	005202	177777		.WORD	177777
572					
573	005204	025252	PAT4:	.WORD	025252
574	005206	052525		.WORD	052525
575	005210	052525		.WORD	052525
576	005212	125252		.WORD	125252
577	005214	125252		.WORD	125252
578	005216	125252		.WORD	125252
579	005220	052525		.WORD	052525
580	005222	052525		.WORD	052525
581	005224	125252		.WORD	125252
582	005226	125252		.WORD	125252
583	005230	052525		.WORD	052525
584	005232	125252		.WORD	125252
585	005234	052525		.WORD	052525
586	005236	125252		.WORD	125252
587	005240	052525		.WORD	052525
588	005242	125252		.WORD	125252
589					
590	005244	155555	PAT5:	.WORD	155555
591	005246	133333		.WORD	133333
592	005250	066666		.WORD	066666
593					
594	005252	121105	PAT6:	.WORD	121105
595	005254	150442		.WORD	150442
596	005256	064221		.WORD	064221
597	005260	132110		.WORD	132110
598	005262	055044		.WORD	055044
599	005264	026442		.WORD	026442

600	005266	013211			.WORD	013211
601	005270	105504			.WORD	105504
602	005272	042642			.WORD	042642
603	005274	021321			.WORD	021321
604	005276	110550			.WORD	110550
605	005300	044264			.WORD	044264
606	005302	022132			.WORD	022132
607	005304	011055			.WORD	011055
608	005306	104426			.WORD	104426
609	005310	042213			.WORD	042213
610						
611	005312	177777		PAT7:	.WORD	177777
612						
613	005314	045513		PAT8:	.WORD	045513
614	005316	122645			.WORD	122645
615	005320	151322			.WORD	151322
616	005322	064551			.WORD	064551
617	005324	132264			.WORD	132264
618	005326	055132			.WORD	055132
619	005330	026455			.WORD	026455
620	005332	113226			.WORD	113226
621	005334	045513			.WORD	045513
622	005336	122645			.WORD	122645
623	005340	151322			.WORD	151322
624	005342	064551			.WORD	064551
625	005344	132264			.WORD	132264
626	005346	055132			.WORD	055132
627	005350	026455			.WORD	026455
628	005352	113226			.WORD	113226
629						
630	005354	125252		PAT9:	.WORD	125252
631						
632	005356	155555		PAT10:	.WORD	155555
633						
634	005360			ENDMOD		
635						
636						
637						
641						
642						
643				.SBTTL	GLOBAL MESSAGES	
644						
645	005360			BGNMOD	GLBTXT	
646						
647	005360	045523	000040	MSEEK:	.ASCIZ /SK /	
648	005364	042122	042040	MREAD:	.ASCIZ /RD DATA /	
649	005375	122	020104	MREADH:	.ASCIZ /RD HDR /	
650	005405	127	052122	MWRCHK:	.ASCIZ /WRT CHCK /	
651	005416	051127	020124	MWRITE:	.ASCIZ /WRT DATA /	
652	005430	042507	020124	MGTSTA:	.ASCIZ /GET STAT /	
653	005442	044527	044124	MDATCP:	.ASCIZ /WITH DATA CMP /	
654	005461	127	052111	MHDRCP:	.ASCIZ /WITH HDR CMP /	
655	005477	106	051117	M40HDR:	.ASCIZ /FOR 40 HDRS /	
656	005513	127	052111	MWRSET:	.ASCIZ /WITH RESET /	
657	005527	117	042520	MOPER:	.ASCIZ /OPER: /	
658	005536	042522	052523	MRSLT:	.ASCIZ /RESULT: /	

659	005547	125	046116	020104	MJLOAD:	.ASCIZ	/UNLD DRV/
660	005560	042114	042040	053122	MCYLUP:	.ASCIZ	/LD DRV /
661	005570	047506	020114	020060	MOUTIN:	.ASCIZ	/FOL 0 TO CC SK/
662	005607	106	046117	031040	MINOUT:	.ASCIZ	/FOL 255 TO CC SK/
663	005630	047506	020114	051127	MFOLWRT:	.ASCIZ	/FOL WRT (NO SK)/
664	005650	042101	020112	054503	MREVSK:	.ASCIZ	/ADJ CYL WRTTN AFT REV SK/
665	005701	101	045104	041440	MFWDISK:	.ASCIZ	/ADJ CYL WRTTN AFT FWD SK/
666	005732	045523	043040	042127	MFWSKO:	.ASCIZ	/SK FWD,WRT - SK REV,OVERWRT/
667	005766	045523	051040	053105	MRESKO:	.ASCIZ	/SK REV,WRT - SK FWD,OVERWRT/
668	006022	047117	041040	042101	MBADAD:	.ASCIZ	/ON BAD SEC FILES/
669	006043	103	047101	052047	MBADSF:	.ASCIZ	/CAN'T GET BAD SEC FILES/
670	006073	102	042101	051440	MFMTERR:	.ASCIZ	/BAD SEC FILE FMT ERR/
671	006120	047524	046440	047101	MTMBS:	.ASCIZ	/TO MANY BAD SEC /
672	006141	102	051525	040440	BASADD:	.ASCIZ	/BUS ADD=/
673	006152	051104	036526	000	DRVNAM:	.ASCIZ	/DRV=/
674	006157	116	020117	051104	DRVNAV:	.ASCIZ	/NO DRV FOR TST/
675	006176	051104	020126	044504	NOFWR:	.ASCIZ	/DRV DID NOT REC'R FROM PWR FAIL/
676	006236	046122	051503	000	CSNAM:	.ASCIZ	/RLCS/
677	006243	122	041114	000101	BANAM:	.ASCIZ	/RLBA/
678	006250	046122	040504	000	DANAM:	.ASCIZ	/RLDA/
679	006255	122	046514	000120	MPNAM:	.ASCIZ	/RLMP/
680	006262	050117	044440	044516	LAB1:	.ASCIZ	/OP INIT = /
681	006275	117	020120	047504	LAB2:	.ASCIZ	/OP DONE = /
682	006310	047527	042122	000040	WORD:	.ASCIZ	/WORD /
683	006316	047111	051124	052120	MTOSLOW:	.ASCIZ	/INTRPT TOO LATE/
684	006336	047516	042040	053122	MORRES:	.ASCIZ	/NO DRV RSPNSE/
685	006354	047516	044440	052116	MNOINT:	.ASCIZ	/NO INTRPT ON CMND COMPLETE/
686	006407	103	052116	051114	MCONHNG:	.ASCIZ	/CNTLR HUNG /
687	006423	105	051122	042040	MNOCLR:	.ASCIZ	/ERR DID NOT CLR/
688	006443	126	046117	041440	VCNRST:	.ASCIZ	/VOL CHK NOT RSET/
689	006464	047125	050130	052103	UNXERR:	.ASCIZ	/UNXPCTED ERR/
690	006501	040	042524	052123	TSTLAB:	.ASCIZ	/TEST/
708	006507	117	052125	043440	P2T03E:	.ASCIZ	/OUT GRD BAND /
709	006525	111	041516	051440	P2T04E:	.ASCIZ	/INC SK FWD HD 0/
710	006545	111	041516	051440	P2T05E:	.ASCIZ	/INC SK REV HD 0/
711	006565	111	041516	051440	P2T06E:	.ASCIZ	/INC SK FWD HD 1/
712	006605	111	047116	043440	P2T07E:	.ASCIZ	/INN GRD BAND /
713	006623	111	041516	051440	P2T08E:	.ASCIZ	/INC SK REV HD 1/
714	006643	123	000113		P2T09E:	.ASCIZ	/SK/
715	006646	053506	020104	051517	P2T10E:	.ASCIZ	/FWD OSC SK/
716	006661	122	053105	047440	P2T11E:	.ASCIZ	/REV OSC SK/
717	006674	045523	052040	046511	P2T12E:	.ASCIZ	/SK TIMING/
718	006706	051502	020103	042122	P2T13E:	.ASCIZ	/BSC RD DATA/
719	006722	051127	027524	042122	P2T14E:	.ASCIZ	/WRT/RD DATA (P1)&
720	006743	123	044520	042116	P2T15E:	.ASCIZ	/SPINDLE ROT TIMING/
721	006766	051127	027524	042122	P2T16E:	.ASCIZ	/WRT/RD DATA (P2)&
722	007007	127	052122	046040	P2T17E:	.ASCIZ	/WRT LCK ERR AND DATA PROT/
723	007041	191	045104	041440	P2T18E:	.ASCIZ	/ADJ CYL INTERFNCE/
724	007063	117	042526	053522	P2T19E:	.ASCIZ	/OVERWRT/
725	007073	123	020113	044524	SKTMES:	.ASCIZ	/SK TIMES /
726	007105	123	044520	042116	SRTMES:	.ASCIZ	/SPINDLE ROT TIME /
727	007127	050	047111	030440	VALDES:	.ASCIZ	/((IN 100'S OF U-SEC))
728	007153	101	050120	047522	MAPROX:	.ASCIZ	/APPROX /
729	007163	111	047116	051105	LABIN:	.ASCIZ	/INNER/
730	007171	115	042111	046104	LABMID:	.ASCIZ	/MIDDLE/
731	007200	052517	042524	000122	LABOUT:	.ASCIZ	/OUTER/

732	007206	040515	020130	044524	LABEXP: .ASCIZ	/MAX TIME/
733	007217	061	041440	046131	LABOCF: .ASCIZ	/1 CYL FWD/
734	007231	061	041440	046131	LABOCR: .ASCIZ	/1 CYL REV/
735	007243	115	042111	041440	LABHCF: .ASCIZ	/MID CYL FWD/
736	007257	115	042111	041440	LABHCR: .ASCIZ	/MID CYL REV/
737	007273	115	054101	041440	LABACF: .ASCIZ	/MAX CYL FWD/
738	007307	115	054101	041440	LABACR: .ASCIZ	/MAX CYL REV/
740	007323	110	051504	043040	HDMOVF: .ASCIZ	/HDS FAILED TO MV IN 10 TRYS/
758	007357	122	051505	052105	OPR12: .ASCIZ	/RESET WRT LCK /
759	007376	047117	000040		OPR1A: .ASCIZ	/ON /
760	007402	047117	042040	053122	OPR1B: .ASCIZ	/ON DRV /
761	007412	047125	042504	020122	UNDTST: .ASCIZ	/UNDER TEST/
762	007425	123	052105	053440	OPR004: .ASCIZ	/SET WRT LCK /
763	007442	044504	043106	000040	DIFWD: .ASCIZ	/DIFF /
764	007450	043523	020116	000	SGNWD: .ASCIZ	/SGN /
765	007455	110	020104	000	HDWD: .ASCIZ	/HD /
766	007461	123	041505	000040	SECWD: .ASCIZ	/SEC /
767	007466	054503	020114	000	CYLWD: .ASCIZ	/CYL /
768	007473	106	047522	020115	FRMWD: .ASCIZ	/FROM /
769	007501	040	054502	040520	BYPSSM: .ASCIZ	/ BYPASSED /
770	007514	047522	052125	047111	SEQMES: .ASCIZ	/ROUTINE TRACE SEQ:/
771	007537	104	053122	051440	STAMES: .ASCIZ	/DRV STAT/
772	007550	040502	020104	042523	BSNSTR: .ASCIZ	/BAD SEC FILES NOT STRD. ALL SEC ASSUMED OK./
773	007624	047524	040524	020114	TCERR: .ASCIZ	/TOTAL CMP ERRS: /
774	007645	104	044522	042526	NOCTLR: .ASCIZ	/DRIVE DROPPED - NO CONTROLLER/
775	007703	104	044522	042526	NOTRDY: .ASCIZ	/DRIVE DROPPED - DID NOT RESPOND WITH 'READY'/'
776						
777						
778						
779	007760	051104	020126	042122	MDRDY: .ASCIZ	/DRV RDY /
780	007771	103	047117	020124	MCERR: .ASCIZ	/CONT ERR /
781	010003	110	051104	041440	MHCRC: .ASCIZ	/HDR CRC/
782	010013	104	052101	020101	MDCRC: .ASCIZ	/DATA CRC/
783	010024	042110	020122	047516	MHNF: .ASCIZ	/HDR NOT FND/
784	010040	040504	040524	046040	MDLT: .ASCIZ	/DATA LATE/
785	010052	042110	020122	047516	MHFCRC: .ASCIZ	&HDR NOT FND/HDR CRC/OP1&
786	010102	051104	020126	051105	MDRERR: .ASCIZ	/DRV ERR /
795	010113	104	053122	051440	MDSERR: .ASCIZ	/DRV SEL ERR /
796	010130	051104	020126	052123	MDRVST: .ASCIZ	/DRV STATE /
797	010143	123	044520	020116	MSPERR: .ASCIZ	/SPIN TIMEOUT /
798	010161	127	052122	043440	MWGERR: .ASCIZ	/WRT GAT ERR /
799	010176	045523	052040	046511	MSTERR: .ASCIZ	/SK TIMEOUT /
800	010212	042510	042101	041440	MHCERR: .ASCIZ	/HEAD CUR ERR /
801	010230	051127	020124	040504	MWDERR: .ASCIZ	/WRT DAT ERR /
802	010245	117	051120	044455	MOPERR: .ASCIZ	/OPR-INC/
803	010255	110	051104	042057	MHDERR: .ASCIZ	&HDR/DAT ERR &
804	010272	042110	020122	047516	MFLERR: .ASCIZ	&HDR NOT FND/DAT LATE &
805	010320	047516	026516	054105	MNEERR: .ASCIZ	/NON-EXISTENT MEMORY /
806	010345	103	046131	000040	MCYLOC: .ASCIZ	/CYL /
807	010352	040503	023516	020124	MNDRST: .ASCIZ	/CAN'T GET DRV STAT/
808	010375	125	045516	020116	MUNDEF: .ASCIZ	/UNKN DRV STATE-NO RDY,NO ERR,HDS OUT/
809	010442	040506	046111	052040	MRLFAL: .ASCIZ	/FAIL TO RELD HDS AFTER ERR CLR/
810	010501	127	052122	040440	MWRTAB: .ASCIZ	/WRT ABRTD/
811	010513	040	053117	020122	MEXERS: .ASCIZ	/ OVR ERR LIMIT - UNIT DRPPD /
812	010550	042440	051122	000	MERRS: .ASCIZ	/ ERR/
813	010555	207	177777	000	BELL: .ASCIZ	<207><377><377>

```
814  
815  
816 010561 111 020123 000  
817 010565 040 041123 000040  
818  
819  
820 010572 044440 020116 000  
821 010577 040 043117 000040  
822 010604 052123 052101 020105  
823 010614 052123 052101 020105  
824 010624 052123 052101 020105  
828 010634 051461 020124 020063  
829 010645 065 030060 051515  
830 010653 103 041531 052440  
831 010662 040504 040524 054040  
832 010673 065 051440 041505  
833  
834 010701 045 022516 022524  
835 010730 047045 052045 047445  
836 010752 047045 052045 047445  
837 010773 045 022524 000124  
838 011000 047045 052045 052045  
839 011007 045 000124  
840 011012 047045 000  
841 011015 045 022516 022524  
842 011026 047045 052045 047445  
843 011046 047045 051445 030461  
844 011110 047045 052045 047445  
845 011160 047045 052045 047445  
846 011212 047045 052045 000  
847 011217 045 022524 030517  
848 011225 045 022524 031517  
849 011233 045 022516 030523  
850 011277 045 022516 022524  
851 011331 045 022516 030523  
852 011365 045 022516 032523  
853 011376 051445 030061 052045  
854 011420 047045 051445 032461  
855 011452 052045 051445 022464  
856 011507 045 022524 031123  
857 011537 045 022524 030523  
858 011562 047045 051445 030461  
859 011616 052045 052045 052045  
860 011632 047045 052045 000  
861 011637 045 022516 031104  
862 011647 045 022516 030523  
863 011673 045 022516 022524  
864 011712 047045 052045 052045  
865  
866 011723  
867  
872
```

RESULT SETTINGS
RESE3: .ASCIZ /IS /
RESE4: .ASCIZ /SB /

RESULT CONDITIONS
RESE5: .ASCIZ /IN /
RESE6: .ASCIZ /OF /
STATE2: .ASCIZ /STATE 2/
STATE3: .ASCIZ /STATE 3/
STATE5: .ASCIZ /STATE 5/
C10MS: .ASCIZ /1ST 3 MS/
C500MS: .ASCIZ /500MS/
CCYLUP: .ASCIZ /CYC UP/
CAFDT: .ASCIZ /DATA XFR/
CSSEC: .ASCIZ /5 SEC/

FMTOP1: .ASCIZ /%N%T%N%T%T%06%S%T%01%N/
FMTOP2: .ASCIZ /%N%T%01%S1%T%01%N/
FMTOP3: .ASCIZ /%N%T%01%S1%T%T%N/
FMT1: .ASCIZ /%T%T/
FMT1.1: .ASCIZ /%N%T%T/
FMT2: .ASCIZ /%T/
FMT3: .ASCIZ /%N/
FMT4: .ASCIZ /%N%T%T%N/
FMT5: .ASCIZ /%N%T%06%S1%T%01/
FMT6: .ASCIZ /%N%S11%T%S4%T%S4%T%S4%T%S4%T%S2%T/
FMT7: .ASCIZ /%N%T%06%S2%06%S2%06%S2%06%S3%03%S2%01%N/
FMT8: .ASCIZ /%N%T%06%S2%06%S2%06%S2%06/
FMT9: .ASCIZ /%N%T/
FMT11: .ASCIZ /%T%01/
FMT12: .ASCIZ /%T%03/
FMT13: .ASCIZ /%N%S11%T%03%S1%T%03%S1%T%01%S1%T%01/
FMT14: .ASCIZ /%N%T%T%D3%S1%T%06%S1%T%06/
FMT15: .ASCIZ /%N%S11%T%D3%S1%T%06%S1%T%06/
FMT16: .ASCIZ /%N%S5%06/
FMT17: .ASCIZ /%S10%T%N%S11%06%N/
FMT18: .ASCIZ /%N%S15%T%S5%T%S4%T%S5%T%N/
FMT19: .ASCIZ /%T%S4%D6%S4%D6%S4%D6%S4%D6%N/
FMT20: .ASCIZ /%T%S2%D6%S14%D6%S4%D6%N/
FMT21: .ASCIZ /%T%S12%D6%S14%D6%N/
FMT22: .ASCIZ /%N%S11%T%03%S1%T%01%S1%T%02/
FMT23: .ASCIZ /%T%T%T%01%N/
FMT24: .ASCIZ /%N%T/
FMT25: .ASCIZ /%N%D2%T/
FMT26: .ASCIZ /%N%S1%T%D4%T%T%D3%N/
FMT27: .ASCIZ /%N%T%D3%T%D3%N/
FMT28: .ASCIZ /%N%T%T%T/

ENDMOD

```
874 .SBTTL ERROR MESSAGES
875 011724 BGNMCD GLBERR
876 : ERR1 R3 POINTS TO RESULT MESSAGE
877 : RESULT: (R3)
878 :
879 : ERR2 R3 POINTS TO RESULT NAME
880 : RESULT: (R3) IS 1 SB 0
881 :
882 : ERR3 R3 POINTS TO RESULT NAME
883 : RESULT: (R3) IS 0 SB 1
884 :
885 : ERR4 R3 POINTS TO RESULT NAME
886 : R4 POINTS TO RESULT CONDITIONS
887 : RESULT: (R3) IS 1 SB 0 (R4)
888 :
889 : ERR5 R3 POINTS TO RESULT NAME
890 : R4 POINTS TO RESULT CONDITIONS
891 : RESULT: (R3) IS 0 SB 1 (R4)
892 :
893 : ERR6 RESULT ROUTINE DETERMINES WHICH ERROR(S) ARE SET AND
894 : REPORTS ALL
895 : RESULT: 'ERROR' IS 1 SB 0
896 :
897 : ERR7 DRIVE STATE ERROR REPORT
898 : R3 CONTAINS EXPECTED STATE
899 : T.STAT CONTAINS BAD STATE
900 : RESULT: DRIVE STATE IS (T.STAT) SB (R3)
901 :
902 : ERR8 HEAD POSITIONING ERROR REPORT
903 : NEWCYL CONTAINS EXPECTED CYLINDER
904 : HDWRD1 CONTAINS BAD CYLINDER
905 : RESULT: CYLINDER IS (HDWRD1) SB (NEWCYL)
906 :
907 : ERR9 UTILITY RESULT REPORT
908 : R3 POINTS TO RESULT NAME
909 : R4 POINTS TO VALUE 1
910 : R5 POINTS TO VALUE 2
911 : RESULT: (R3-NAME) IS (R4-VALUE 1) SB (R5-VALUE 2)
912 :
913 : ERR10 COMPARE ERROR REPORT
914 : R3 CONTAINS THE BAD WORD NUMBER
915 : R4 POINTS TO BAD WORD
916 : R5 POINTS TO GOOD WORD
917 : RESULT: WORD (R3) IS (R4) SB (R5)
918 :
919 :
920 011724 BGNMSG ERR1
921 011724 105737 003461 TSTB NOERCT ;TEST IF ERROR COUNTING INHIBITED
922 011730 001002 BNE 1$ ;YES - SKIP
923 011732 005277 171314 INC @ERRPOINT ;ELSE BUMP ERROR COUNT
924 011736 010146 1$: MOV R1,-(SP) ;STORE R1
925 011740 004737 024514 JSR PC,RPTOP ;REPORT OPERATION
926 011744 012721 000001 MOV #1,(R1)+ ;SET PARAM NUMBER
927 011750 010321 MOV R3,(R1)+ ;INSERT MESSAGE ADDRESS POINTER
928 011752 004737 025302 JSR PC,RPTRES ;REPORT RESULTS
929 011756 004737 025510 JSR PC,RPTREM ;REPORT REMAINDER
```

```

930 011762 012601          MOV      (SP)+,R1      ;RESTORE R1
931 011764 004737 015664   JSR      PC,CKERLM    ;GO CHECK IF ERROR COUNT EXCEEDED
932 011770                ENDMMSG
(3) 011770                L10000:
(3) 011770 104423          TRAP     C$MSG
933
934 011772                BGNMSG  ERR2
935 011772 005277 171254   INC      @ERRPOINT    ;BUMP ERROR COUNT
936 011776 010146          MOV      R1,-(SP)     ;STORE R1
937 012000 004737 024514   JSR      PC,RPTOP     ;REPORT OPERATION
938 012004 012721 000003   MOV      #3,(R1)+     ;SET PARAM NUMBER
939 012010 010321          MOV      R3,(R1)+     ;INSERT NAME ADD POINTER
940 012012 012721 000001   MOV      #1,(R1)+     ;SET IS VALUE
941 012016 005021          CLR      (R1)+        ;SET SB VALUE
942 012020 004737 025510   JSR      PC,RPTRES    ;REPORT RESULTS
943 012024 004737 025510   JSR      PC,RPTREM    ;REPORT REMAINDER
944 012030 012601          MOV      (SP)+,R1     ;RESTORE R1
945 012032 004737 015664   JSR      PC,CKERLM    ;GO CHECK IF ERROR COUNT EXCEEDED
946 012036                ENDMMSG
(3) 012036                L10001:
(3) 012036 104423          TRAP     C$MSG
947
948 012040                BGNMSG  ERR3
949 012040 005277 171206   INC      @ERRPOINT    ;BUMP ERROR COUNT
950 012044 010146          MOV      R1,-(SP)     ;STORE R1
951 012046 004737 024514   JSR      PC,RPTOP     ;REPORT OPERATION
952 012052 012721 000003   MOV      #3,(R1)+     ;SET PARAM NUMBER
953 012056 010321          MOV      R3,(R1)+     ;INSERT NAME ADD POINTER
954 012060 005021          CLR      (R1)+        ;SET IS VALUE
955 012062 012721 000001   MOV      #1,(R1)+     ;SET SB VALUE
956 012066 004737 025302   JSR      PC,RPTRES    ;REPORT RESULTS
957 012072 004737 025510   JSR      PC,RPTREM    ;REPORT REMAINDER
958 012076 012601          MOV      (SP)+,R1     ;RESTORE R1
959 012100 004737 015664   JSR      PC,CKERLM    ;GO CHECK IF ERROR COUNT EXCEEDED
960 012104                ENDMMSG
(3) 012104                L10002:
(3) 012104 104423          TRAP     C$MSG
961
962 012106                BGNMSG  ERR4
963 012106 005277 171140   INC      @ERRPOINT    ;BUMP ERROR COUNT
964 012112 010146          MOV      R1,-(SP)     ;STORE R1
965 012114 004737 024514   JSR      PC,RPTOP     ;REPORT OPERATION
966 012120 012721 000004   MOV      #4,(R1)+     ;SET PARAM NUMBER
967 012124 010321          MOV      R3,(R1)+     ;INSERT NAME ADD POINTER
968 012126 012721 000001   MOV      #1,(R1)+     ;SET IS VALUE
969 012132 005021          CLR      (R1)+        ;SET SB VALUE
970 012134 010411          MOV      R4,(R1)      ;INSERT ADD OF CONDITION POINTER
971 012136 004737 025302   JSR      PC,RPTRES    ;REPORT RESULTS
972 012142 004737 025510   JSR      PC,RPTREM    ;REPORT REMAINDER
973 012146 012601          MOV      (SP)+,R1     ;RESTORE R1
974 012150 004737 015664   JSR      PC,CKERLM    ;GO CHECK IF ERROR COUNT EXCEEDED
975 012154                ENDMMSG
(3) 012154                L10003:
(3) 012154 104423          TRAP     C$MSG
976
977 012156                BGNMSG  ERR5
  
```

```
978 012156 005277 171070      INC      @ERRPOINT      :BUMP ERROR COUNT
979 012162 010146      MOV      R1,-(SP)      :STORE R1
980 012164 004737 024514      JSR      PC,RPTOP      :REPORT OPERATION
981 012170 012721 000004      MOV      #4,(R1)+      :SET PARAM NUMBER
982 012174 010321      MOV      R3,(R1)+      :INSERT NAME ADD POINTER
983 012176 005021      CLR      (R1)+         :SET IS VALUE
984 012200 012721 000001      MOV      #1,(R1)+      :SET SB VALUE
985 012204 010411      MOV      R4,(R1)+      :INSERT ADD OF CONDITION POINTER
986 012206 004737 025302      JSR      PC,RPTRES     :REPORT RESULTS
987 012212 004737 025510      JSR      PC,RPTREM     :REPORT REMAINDER
988 012216 012601      MOV      (SP)+,R1      :RESTORE R1
989 012220 004737 015664      JSR      PC,CKERLM     :GO CHECK IF ERROR COUNT EXCEEDED
990 012224      ENDMSG
(3) 012224      L10004:
(3) 012224 104423      TRAP     CSMSG
991
992 012226      BGNMSG ERR6
993 012226 105737 003461      TSTB    NOERCT         :TEST IF ERROR COUNTING INHIBITED
994 012232 001002      BNE     17$           :YES - SKIP
995 012234 005277 171012      INC      @ERRPOINT     :ELSE BUMP ERROR COUNT
996 012240 010146      MOV      R1,-(SP)      :STORE R1
997 012242 010346      MOV      R3,-(SP)      :STORE R3
998 012244 010446      MOV      R4,-(SP)      :STORE R4
999 012246 010546      MOV      R5,-(SP)      :STORE R5
1000 012250 004737 024514      JSR      PC,RPTOP      :REPORT OPERATION
1001 012254 012721 000003      MOV      #3,(R1)+      :SET PARAM NUMBER
1002 012260 012761 000001 000002      MOV      #1,2(R1)      :INSERT IS VALUE
1003 012266 005037 003140      CLR      TEMP3         :CLEAR FOR STATUS STORAGE
1004 012272 013703 003060      MOV      T.CS,R3       :GET T.CS
1005 012276 042703 177761      BIC      #177761,R3    :AND CLEAR ALL BUT FUNCTION
1006 012302 022703 000004      CMP      #4,R3         :CHECK IF IT WAS GET STATUS
1007 012306 001434      BEQ     1$            :YES - STATUS IS IN T.MP, SKIP
1008 012310 012762 000003 000004      MOV      #GETSTAT,RLDA(R2) :ELSE DO GET STATUS
1009 012316 012703 000004      MOV      #4,R3
1010 012322 053703 003046      BIS      RLDRV,R3
1011 012326 010362 000000      MOV      R3,RLCS(R2)
1012 012332      WAITUS #10.         :WAIT FOR CONTROLLER READY
1013 012344 032762 000200 000000      BIT      #CRDYMSK,RLCS(R2) :TEST IF READY
1014 012352 001003      BNE     10$          :YES - SKIP
1015 012354 012703 001000 9$:      MOV      #BIT9,R3     :ELSE SET NO DRIVE STATUS BIT
1016 012360 000413      BR      2$            :IN MESSAGE WORD AND SKIP
1017 012362 016203 000006 10$:      MOV      RLMP(R2),R3  :STORE STATUS FOR REPORT
1018 012366 010337 003140      MOV      R3,TEMP3
1019 012372 113703 003141      MOVVB   TEMP3+1,R3    :GET ERROR BITS IN PROPER POSITION
1020 012376 000402      BR      13$
1021 012406 113703 003067 1$:      MOVVB   T.MP+1,R3    :GET ERROR BITS FROM MP REG
1022 012404 042703 177442 13$:      BIC      #177442,R3  :CLEAR UNUSED BITS
1023 012410 013704 003060 2$:      MOV      T.CS,R4     :GET ERROR BITS FROM CS REG
1024 012414 042704 001777      BIC      #1777,R4    :CLEAR UNUSED BITS
1025 012420 050403      BIS      R4,R3       :MAKE ONE WORD OF POSSIBLE ERRORS
1026 012422 032703 002000      BIT      #OPIERR,R3  :TEST IF OPI SET
1027 012426 001442      BEQ     11$          :NO - SKIP
1028 012430 032703 010000      BIT      #HNFERR,R3  :TEST IF HDR NOT FOUND ERROR
1029 012434 001026      BNE     107$        :YES - SKIP
1030 012436 032703 004000      BIT      #HRCERR,R3  :TEST IF HDR CRC ERR
1031 012442 001020      BNE     105$        :YES - SKIP
```

1032	012444	012704	010245		MOV	#MOPERR,R4	;SET OPI ALONE MESSAGE
1033	012450			100\$:	PRINTB	#FMT28,#MRSLT,R4,#MERRS	;REPORT ERROR
(10)	012450	012746	010550		MOV	#MERRS,-(SP)	
(9)	012454	010446			MOV	R4,-(SP)	
(8)	012456	012746	005536		MOV	#MRSLT,-(SP)	
(7)	012462	012746	011712		MOV	#FMT28,-(SP)	
(6)	012466	012746	000004		MOV	#4,-(SP)	
(3)	012472	010600			MOV	SP,R0	
(4)	012474	104414			TRAP	C\$PNTB	
(4)	012476	062706	000012		ADD	#12,SP	
1034	012502	000430			BR	120\$;SKIP
1035	012504	012704	010003	105\$:	MOV	#MHCRC,R4	;HDR CRC MESSAGE
1036	012510	000757			BR	100\$	
1037	012512	032703	004000	107\$:	BIT	#HCRCERR,R3	;TEST IF HCRC WITH HDR NOT FND
1038	012516	001003			BNE	109\$;YES - SKIP
1039	012520	012704	010024		MOV	#MNF,R4	;MESSAGE HEADER NOT FOUND
1040	012524	000751			BR	100\$	
1041	012526	012704	010052	109\$:	MOV	#MHFCRC,R4	;MNF AND HCRC MESSAGE
1042	012532	000746			BR	100\$;SKIP
1043	012534	032703	004000	115\$:	BIT	#DCKERR,R3	;TEST IF DATA CHECK SET, NOT OPI
1044	012540	001403			BEQ	118\$;NO - SKIP
1045	012542	012704	010013		MOV	#MDCRC,R4	;SET MESSAGE DATA CHECK
1046	012546	000740			BR	100\$;SKIP
1047	012550	032703	010000	118\$:	BIT	#DLTERR,R3	;TEST IF DATA LATE ERROR
1048	012554	001403			BEQ	120\$;NO - SKIP
1049	012556	012704	010040		MOV	#MDLT,R4	;SET MESSAGE DATA LATE
1050	012562	000732			BR	100\$;SKIP
1051	012564	012705	100000	120\$:	MOV	#BIT15,R5	;SET BIT POINTER FOR TEST
1052	012570	005004			CLR	R4	;CLEAR R4 FOR TABLE COUNT
1053	012572	030503		3\$:	BIT	R5,R3	;TEST IF BIT IS SET
1054	012574	001005			BNE	6\$;YES - SKIP TO REPORT
1055	012576	005724		4\$:	TST	(R4)+	;ELSE BUMP TABLE POINTER
1056	012600	000241			CLC		;CLEAR CARRY
1057	012602	006005			ROR	R5	;SHIFT BIT POINTER TO NEXT BIT
1058	012604	001372			BNE	3\$;LOOP IF NOT 0
1059	012606	000405			BR	7\$;ELSE REPORT REMAINDER
1060	012610	016411	002334	6\$:	MOV	RESTBL(R4),(R1)	;INSERT NAME ADDRESS
1061	012614	004737	025302		JSR	PC,RPTRES	;REPORT RESULTS
1062	012620	000766			BR	4\$;GET NEXT BIT
1063	012622	004737	025510	7\$:	JSR	PC,RPTREM	;REPORT REMAINDER
1064	012626	005737	003140		TST	TEMP3	;TEST IF ANY NEW STATUS
1065	012632	001414			BEQ	15\$;NO - SKIP
1066	012634				PRINTB	#FMT17,#STAMES,TEMP3	
(9)	012634	013746	003140		MOV	TEMP3,-(SP)	
(8)	012640	012746	007537		MOV	#STAMES,-(SP)	
(7)	012644	012746	011376		MOV	#FMT17,-(SP)	
(6)	012650	012746	000003		MOV	#3,-(SP)	
(3)	012654	010600			MOV	SP,R0	
(4)	012656	104414			TRAP	C\$PNTB	
(4)	012660	062706	000010		ADD	#10,SP	
1067	012664	032737	004000	003060	15\$:	BIT	#DCKERR,T.CS ;TEST IF DATA CHECK ERROR
1068	012672	001453			BEQ	25\$;NO - SKIP
1069	012674	032737	002000	003060	BIT	#OPIERR,T.CS	;TEST IF OPI SET
1070	012702	001047			BNE	25\$;YES - SKIP
1071	012704	005037	003030		CLR	MORECE	;CLEAR COMPARE ERROR COUNT
1072	012710	012701	000200		MOV	#128,R1	;SET COMPARE LENGTH

1073	012714	012703	000001		MOV	#1,R3	:SET WORD COUNT
1074	012720	012705	004502		MOV	#0BUFF,R5	:SET GOOD WORD POINTER
1075	012724	012704	004102		MOV	#1BUFF,R4	:SET TEST WORD POINTER
1076	012730	021514		18\$:	CMP	(R5),(R4)	:CHECK WORD
1077	012732	001427			BEQ	19\$:GOOD - SKIP
1078	012734	023727	003030	000012	CMP	MORECE,#10.	:TEST IF COMPARE LIMIT REACHED
1079	012742	003021			BGT	20\$:YES - SKIP
1080	012744				PRINTB	#FMT15,#MWORD,R3,#RESE3,(R4),#RESE4,(R5)	
(13)	012744	011546			MOV	(R5),-(SP)	
(12)	012746	012746	010565		MOV	#RESE4,-(SP)	
(11)	012752	011446			MOV	(R4),-(SP)	
(10)	012754	012746	010561		MOV	#RESE3,-(SP)	
(9)	012760	010346			MOV	R3,-(SP)	
(8)	012762	012746	006310		MOV	#MWORD,-(SP)	
(7)	012766	012746	011331		MOV	#FMT15,-(SP)	
(6)	012772	012746	000007		MOV	#7,-(SP)	
(3)	012776	010600			MOV	SP,R0	
(4)	013000	104414			TRAP	C\$PNTB	
(4)	013002	062706	000020		ADD	#20,SP	
1081	013006	005237	003030	20\$:	INC	MORECE	:BUMP ERROR COUNTER
1082	013012	022524		19\$:	CMP	(R5)+,(R4)+	:BUMP POINTERS
1083	013014	005203			INC	R3	:BUMP COUNTER
1084	013016	005301			DEC	R1	:DEC LENGTH COUNT
1085	013020	001343			BNE	18\$:LOOP IF NOT DONE
1086	013022	005737	003030	25\$:	TST	MORECE	:TEST IF ANY COMPARE ERRORS
1087	013026	001421			BEQ	27\$:NO - SKIP
1088	013030	012701	000200		MOV	#128,R1	:SET COMPARE LENGTH
1089	013034				PRINTB	#FMT27,#TCERR,MORECE,#RESF6,R1	
(11)	013034	010146			MOV	R1,-(SP)	
(10)	013036	012746	010577		MOV	#RESE6,-(SP)	
(9)	013042	013746	003030		MOV	MORECE,-(SP)	
(8)	013046	012746	007624		MOV	#TCERR,-(SP)	
(7)	013052	012746	011673		MOV	#FMT27,-(SP)	
(6)	013056	012746	000005		MOV	#5,-(SP)	
(3)	013062	010600			MOV	SP,R0	
(4)	013064	104414			TRAP	C\$PNTB	
(4)	013066	062706	000014		ADD	#14,SP	
1090	013072	012605		27\$:	MOV	(SP)+,R5	:RESTORE R5, 4, 3, 1
1091	013074	012604			MOV	(SP)+,R4	
1092	013076	012603			MOV	(SP)+,R3	
1093	013100	012601			MOV	(SP)+,R1	
1094	013102	004737	015664		JSR	PC,CKERLM	:GO CHECK IF ERROR COUNT EXCEEDED
1095	013106			ENDMSG			
(3)	013106			L10005:			
(3)	013106	104423			TRAP	C\$MSG	
1096				BGNMSG	ERR7		
1097	013110				INC	@ERRPOINT	:BUMP ERROR COUNT
1098	013110	005277	170136		MOV	R1,-(SP)	:STORE R1
1099	013114	010146			JSR	PC,RPTOP	:REPORT OPERATION
1100	013116	004737	024514		MOV	#3,(R1)+	:SET PARAM NUMBER
1101	013122	012721	000003		MOV	#MDRVST,(R1)+	:INSERT NAME ADD POINTER
1102	013126	012721	010130		MOV	T_STAT,(R1)+	:INSERT IS VALUE
1103	013132	013721	003074		MOV	R3,(R1)	:INSERT SB VALUE
1104	013136	010311			JSR	PC,RPTRES	:REPORT RESULTS
1105	013140	004737	025302		JSR	PC,RPTREM	:REPORT REMAINDER
1106	013144	004737	025510				

1107	013150	012601		MOV	(SP)+,R1	:RESTORE R1
1108	013152	004737	015664	JSR	PC,CKERLM	:GO CHECK IF ERROR COUNT EXCEEDED
1109	013156			ENDMSG		
(3)	013156			L10006:		
(3)	013156	104423		TRAP	C\$MSG	
1110						
1111	013140			BGNMSG	ERR8	
1112	013160	005277	170066	INC	@ERRPOINT	:BUMP ERROR COUNT
1113	013164	010146		MOV	R1,-(SP)	:STORE R1
1114	013166	010346		MOV	R3,-(SP)	:STORE R3
1115	013170	004737	024514	JSR	PC,RPTOP	:REPORT OPERATION
1116	013174	012721	000003	MOV	#3,(R1)+	:SET PARAM NUMBER
1117	013200	012721	010345	MOV	#MCYLOC,(R1)+	:INSERT NAME ADD POINTER
1118	013204	013711	003066	MOV	HDWRD1,(R1)	:GET HEADER WORD
1119	013210	012703	000007	MOV	#7,R3	:SET SHIFT COUNT
1120	013214	000241		3\$:	CLC	
1121	013216	006011		ROR	(R1)	:ALIGN CHAR FOR PRINTING
1122	013220	005303		DEC	R3	: AS IS VALUE
1123	013222	001374		BNE	3\$	
1124	013224	005721		TST	(R1)+	:BUMP PARAM POINTER
1125	013226	013711	003116	MOV	NEWCYL,(R1)	:INSERT SB VALUE
1126	013232	004737	025302	JSR	PC,RPTRES	:REPORT RESULTS
1127	013236	004737	025510	JSR	PC,RPTREM	:REPORT REMAINDER
1128	013242	012603		MOV	(SP)+,R3	:RESTORE R3
1129	013244	012601		MOV	(SP)+,R1	:RESTORE R1
1130	013246	004737	015664	JSR	PC,CKERLM	:GO CHECK IF ERROR COUNT EXCEEDED
1131	013252			ENDMSG		
(3)	013252			L10007:		
(3)	013252	104423		TRAP	C\$MSG	
1132						
1133	013254			BGNMSG	ERR9	
1134	013254	005277	167772	INC	@ERRPOINT	:BUMP ERROR COUNT
1135	013260	010146		MOV	R1,-(SP)	:STORE R1
1136	013262	004737	024514	JSR	PC,RPTOP	:REPORT OPERATION
1137	013266	012721	000003	MOV	#3,(R1)+	:SET PARAM NUMBER
1138	013272	010321		MOV	R3,(R1)+	:INSERT NAME ADD POINTER
1139	013274	010421		MOV	R4,(R1)+	:SET IS VALUE
1140	013276	010521		MOV	R5,(R1)+	:SET SB VALUE
1141	013300	004737	025302	JSR	PC,RPTRES	:REPORT RESULTS
1142	013304	004737	025510	JSR	PC,RPTREM	:REPORT REMAINDER
1143	013310	012601		MOV	(SP)+,R1	:RESTORE R1
1144	013312	004737	015664	JSR	PC,CKERLM	:GO CHECK IF ERROR COUNT EXCEEDED
1145	013316			ENDMSG		
(3)	013316			L10010:		
(3)	013316	104423		TRAP	C\$MSG	
1146	013320			BGNMSG	ERR10	
1147	013320	010146		MOV	R1,-(SP)	:STORE R1
1148	013322	005737	003030	TST	MORECE	:TEST IF 2ND BAD LINE
1149	013326	001051		BNE	3\$:YES - SKIP
1150	013330	005277	167716	INC	@ERRPOINT	:BUMP ERROR COUNT
1151	013334	004737	024514	JSR	PC,RPTOP	:REPORT OPERATION
1152	013340			PRINTB	#FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>	:REPORT ID
(11)	013340	005046		CLR	-(SP)	
(11)	013342	153716	003047	BISB	RLDRV+1,(SP)	
(10)	013346	012746	006152	MOV	#DRVNAM,-(SP)	
(9)	013352	013746	003042	MOV	RLBAS,-(SP)	

```

(8) 013356 012746 006141      MOV      #BASADD,-(SP)
(7) 013362 012746 011026      MOV      #FMT5,-(SP)
(6) 013366 012746 000005      MOV      #5,-(SP)
(3) 013372 010600                MOV      SP,R0
(4) 013374 104414                TRAP     C$PNTB
(4) 013376 062706 000014      ADD      #14,SP
1153 013402                PRINTB  #FMT14,#MRSLT,#MMWORD,R3,#RESE3,(R4),#RESE4,(R5)
(14) 013402 011546                MOV      (R5),-(SP)
(13) 013404 012746 010565      MOV      #RESE4,-(SP)
(12) 013410 011446                MOV      (R4),-(SP)
(11) 013412 012746 010561      MOV      #RESE3,-(SP)
(10) 013416 010346                MOV      R3,-(SP)
(9) 013420 012746 006310      MOV      #MMWORD,-(SP)
(8) 013424 012746 005536      MOV      #MRSLT,-(SP)
(7) 013430 012746 011277      MOV      #FMT14,-(SP)
(6) 013434 012746 000010      MOV      #10,-(SP)
(3) 013440 010600                MOV      SP,R0
(4) 013442 104414                TRAP     C$PNTB
(4) 013444 062706 000022      ADD      #22,SP
1154 013450 000421                BR       4$
1155 013452                3$: PRINTB #FMT15,#MMWORD,R3,#RESE3,(R4),#RESE4,(R5) ;REPORT DATA
(13) 013452 011546                MOV      (R5),-(SP)
(12) 013454 012746 010565      MOV      #RESE4,-(SP)
(11) 013460 011446                MOV      (R4),-(SP)
(10) 013462 012746 010561      MOV      #RESE3,-(SP)
(9) 013466 010346                MOV      R3,-(SP)
(8) 013470 012746 006310      MOV      #MMWORD,-(SP)
(7) 013474 012746 011331      MOV      #FMT15,-(SP)
(6) 013500 012746 000007      MOV      #7,-(SP)
(3) 013504 010600                MOV      SP,R0
(4) 013506 104414                TRAP     C$PNTB
(4) 013510 062706 000020      ADD      #20,SP
1156 013514 005237 003030      4$: INC   MORECE ;INC COMPARE ERROR COUNT
1157 013520 012601                MOV      (SP)+,R1 ;RESTORE R1
1158 013522 004737 015664      JSR     PC,CKERLM ;GO CHECK IF ERROR COUNT EXCEEDED
1159 013526                ENDMSG
(3) 013526                L10011:
(3) 013526 104423                TRAP     C$MSG
1160 013530                ENDMOD
1161
1162                ;LOAD PROTECTION TABLE
1163 013530                BGNPROT
1164 013530 000000                .WORD   0 ;OFFSET OF CSR IN P-TABLE
1165 013532 177777                .WORD  -1 ;NOT A MASS-BUS DRIVE
1166 013534 000010                .WORD   10 ;OFFSET OF DRIVE IN P-TABLE
1167 013536                ENDPROT
1168
1169                .EVEN
1170
1171 013536                BGNMOD  HPTCODE
1172 013536                BGNHW
(3) 013536 000006                .WORD  L10013-L$HW/2
1173 013540 174400                .WORD  174400 ;CSR BASE ADDRESS DEFAULT
1174 013542 000160                .WORD   160 ;VECTOR DEFAULT
1175 013544 000240                .WORD   240 ;PRIORITY DEFAULT
1176 013546 000001                .WORD    1 ;TYPE OF DRIVE

```

1177 013550 000000
1178 013552 000001
1179 013554
(3) 013554
1180 013554
1181
1182 013554
1183 013554
(3) 013554 000006
1184 013556 000000
1185
1186
1187
1188
1189
1190
1191 013560 000000
1192 013562 000377
1193 013564 000000
1194 013566 000024
1195 013570 000012
1196 013572
(3) 013572
1197 013572
1198
1199 013572
1204 013572
(4) 013572 000011
(6) 013574 025774
(6) 013576 026216
(6) 013600 026426
(6) 013602 026636
(6) 013604 027062
(6) 013606 027270
(6) 013610 027516
(6) 013612 030026
(6) 013614 030324
1206 013616
1207
1208

.WORD 0
.WORD 1
ENDHW
L10013:
ENDMOD
BGNMOD SPFCODE
BGNSW
MISWIW: .WORD L10014-L\$SW/2
.WORD 0
LOLIMW: .WORD 0
HILIMW: .WORD 255.
HEADW: .WORD 0
ERLIMW: .WORD 20.
DCLIMW: .WORD 10.
ENDSW
L10014:
ENDMOD
BGNMOD DSPCODE
DISPATCH
.WORD 9
.WORD T1
.WORD T2
.WORD T3
.WORD T4
.WORD T5
.WORD T6
.WORD T7
.WORD T8
.WORD T9
ENDMOD

:DRIVE NUMBER DEFAULT
:RL11 CONTROLLER
:BIT 0 = USE ALL CYLINDERS
:BIT 1 = USE ALL SECTORS
:BIT 2 = EXECUTE DRIVE SELECT TEST
:BIT 3 = EXECUTE HEAD ALIGNMENT
:BIT 12 = HEAD SELECT SUPPLIED FLAG
:BIT 13 = HILIMIT SPECIFIED FLAG
:BIT 14 = LO LIMIT SPECIFIED FLAG
:ERROR LIMIT
:COMPARE ERROR LIMIT

```

1210 .SBTTL  INITIALIZATION SECTION
1211
1212 013616 BGNMOD  INITCODE
1213 013616 BGNINIT
1214
1215 ;CHECK FOR PRESENCE OF A P-CLOCK
1216 013616 005037 003504 CLR  CLKFLG ;CLEAR CLOCK FLAG
1217 013622 CLOCK  P,CLKADR ;P-CLOCK?
(3) 013622 012700 000120 MOV  #P,RO
(3) 013626 104462 TRAP C$CLK
(3) 013630 010037 003506 MOV  RO,CLKADR
1218 013634 BNCOMPLETE 1$ ;BRANCH IF NO P-CLOCK
(2) 013634 103002 BCC 1$
1219 013636 005237 003504 1$: INC  CLKFLG ;INDICATE PRESENCE OF A P-CLOCK
1220 013642 SETPRI #340 ;SET PRIORITY TO 7 TO INHIBIT ALL INTERRUPTS
(3) 013642 012700 000340 MOV  #340,RO
(3) 013646 104441 TRAP C$SPRI
1221 013650 BRESET ;FOR LSI-11 CPU'S
(3) 013650 104433 TRAP C$RESET
1222 013652 042737 100014 013556 BIC  #MITEST!DRSELT!HDALIGN,MISWIW ;CLEAR ALL MANUAL
1223 ; INTERVENTION FLAGS
1224 013660 005037 003016 CLR  SSINDX ;CLEAR SUBROUTINE STACK INDEX
1225 013664 REDEF  #EF.PWR ;POWER FAILURE
(3) 013664 012700 000034 MOV  #EF.PWR,RO
(3) 013670 104447 TRAP C$REFG
1226 013672 BNCOMPLETE 4$ ;NO, GO CHECK NEW PASS
(2) 013672 103005 BCC 4$
1227 013674 013737 002012 003464 MOV  LSUNIT,PWRFLG ;SET POWER FAIL FLAG
1228 013702 000137 014314 JMP  PWCON ;GO SERVICE POWER FAIL
1229 013706 4$: REDEF  #EF.START ;CHECK IF START
(3) 013706 012700 000040 MOV  #EF.START,RO
(3) 013712 104447 TRAP C$REFG
1230 013714 BNCOMPLETE RESTART ;NO - SKIP
(2) 013714 103034 BCC  RESTART
1231
1232 ; ON START INITIALIZE TO START AT FIRST DRIVE, CLEAR INTERNAL
1233 ; PASS COUNT, AND ERROR COUNT.
1234
1235 013716 013737 002012 003110 MOV  LSUNIT,DRVCNT ;SET UP UNIT COUNT
1236 013724 005037 003454 RSTRT: CLR  PASNUM ;CLEAR PASS NUMBER
1237 013730 012700 003254 MOV  #ERRCNT,RO
1238 013734 012701 000100 MOV  #64,R1 ;GET A COUNT
1239 013740 005020 1$: CLR  (R0)+ ;CLEAR AN ERROR COUNTER STORAGE AREA
1240 013742 005301 DEC  R1
1241 013744 001375 BNE  1$ ;LOOP TILL ALL CLEARED
1242 013746 012737 003252 003252 MOV  #ERRCNT-2,ERRPOINT ;INIT ERROR POINTER
1243 013754 012737 177777 003456 MOV  #-1,PSETNM ;SET PARAM SELECT TO INITIAL VALUE
1244 013762 012737 177777 003024 MOV  #-1,HADONE ;PRESET HEAD ALIGN DONE FLAG
1245 013770 032737 040000 013556 LAB: BIT  #LOCYL,MISWIW ;TEST IF LO LIMIT SET
1246 013776 001002 BNE  5$ ;YES - SKIP
1247 014000 005037 013560 CLR  LOLIMW ;ELSE CLEAR LO LIMIT
1248 014004 000432 5$: BR  SETDON
1249 014006 RESTART:
1250 014006 REDEF  #EF.RESTART ;CHECK IF RESTART
(3) 014006 012700 000037 MOV  #EF.RESTART,RO
(3) 014012 104447 TRAP  C$REFG

```

```
1251 014014          BCOMPLETE RSTRT ;NO - SKIP
(2) 014014 103743   BCS RSTRT
1252 014016          CONTINUE:
1253 014016          READF #EF.CONTINUE ;TEST IF CONTINUE
(3) 014016 012700 000036 MOV #EF.CONTINUE,RO
(3) 014022 104447   TRAP C$REFG
1254 014024          BCOMPLETE PWCON
(2) 014024 103533   BCS PWCON
1255          ; ON CONTINUE PICK UP UNIT LAST UNDER TEST
1256 014026          READF #EF.NEW ;CHECK IF STARTING NEW PASS
(3) 014026 012700 000035 MOV #EF.NEW,RO
(3) 014032 104447   TRAP C$REFG
1257 014034          BCOMPLETE PASNEW
(2) 014034 103403   BCS PASNEW
1258 014036          NXTPAS:
1259 014036 005737 003110 TST DRVCNT ;TEST IF ALL UNITS CHECKED
1260 014042 001013 BNE SETDON ;NO - SKIP
1261 014044 005237 003454 PASNEW: INC PASNUM ;ELSE BUMP PASS COUNT
1262 014050 012737 003252 003252 MOV #ERRCNT-2,ERRPOINT ;INIT ERROR POINTER
1263 014056 013737 002012 003110 MOV L$UNIT,DRVCNT ;GET ALL DRIVES
1264 014064 012737 177777 003456 MOV #-1,PSETNM ;SET PARAM SELECT TO INITIAL
1265 014072 005237 003456 SETDON: INC PSETNM ;NEXT SET OF PARAMETERS
1266 014076 005337 003110 DEC DRVCNT ;DOWN COUNT DRIVE TOTAL
1267 014102 062737 000002 003252 ADD #2,ERRPOINT ;UPDATE THE ERROR POINTER
1268 014110 013700 003456 MOV PSETNM,RO ;SET UP TO GET PARAMETERS
1269 014114 012702 003042 MOV #RLBAS,R2
1270 014120          GPHARD RO,R1
(3) 014120 104442 TRAP C$GPHRD
(3) 014122 010001 MOV RO,R1
1271 014124          BCOMPLETE 7$ ;SKIP IF GOOD PARAM
(2) 014124 103406 BCS 7$
1272 014126 005737 003464 TST PWRFLG ;RECENT POWER FAILURE
1273 014132 001741 BEQ NXTPAS ;NO
1274 014134 005337 003464 DEC PWRFLG ;ACCOUNT FOR DRIVE
1275 014140 000736 BR NXTPAS
1276 014142 012122 7$: MOV (R1)+,(R2)+ ;STORE PARAMETERS CSR
1277 014144 012122 MOV (R1)+,(R2)+ ; VECTOR
1278 014146 005721 TST (R1)+ ;BUMP PAST PRIORITY
1279 014150 012137 002312 MOV (R1)+,T.DRIVE
1280 014154 012122 MOV (R1)+,(R2)+
1281 014156 022737 000001 002312 CMP #1,T.DRIVE
1282 014164 001426 BEQ 65$
1283 014166 012737 000776 002322 MOV #510.,NXTHL
1284 014174 012737 000777 002316 MOV #511.,HLMTW
1285 014202 012737 001000 002324 MOV #512.,GBND
1286 014210 012737 177600 002326 MOV #177600,CAMSK
1287 014216 012737 177600 002330 MOV #177600,DIRMSK
1288 014224 012737 177600 002332 MOV #177600,HDCYL
1289 014232 012737 177000 002320 MOV #177000 CLRBYT
1290 014240 000425 BR PWCON
1291
1292 014242 012737 000377 002316 65$: MOV #255.,HLMTW
1293 014250 012737 000400 002324 MOV #256.,GBND
1294 014256 012737 077600 002326 MOV #77600,CAMSK
1295 014264 012737 077600 002330 MOV #77600,DIRMSK
1296 014272 012737 077600 002332 MOV #77600,HDCYL
```

```

1297 014300 012737 000376 002322      MOV      #254,NXTHL
1298 014306 012737 177400 002320      MOV      #177400,CLRBYT
1299
1300 014314 032737 020000 013556  PWCON:  BIT      #HICYL,MISWIW
1301 014322 001003          BNE      1$
1302 014324 013737 002316 013562      MOV      HLMTW,HILIMW
1303 014332          1$:  SETVEC  RLVEC,#INTHLR,#340      ;SET UP VECTOR
(7) 014332 012746 000340      MOV      #340,-(SP)
(6) 014336 012746 015604      MOV      #INTHLR,-(SP)
(5) 014342 013746 003044      MOV      RLVEC,-(SP)
(4) 014346 012746 000003      MOV      #3,-(SP)
(3) 014352 104437          TRAP    C$SVEC
(2) 014354 062706 000010      ADD     #10,SP
1304 014360          SETPRI  #0      ;SET PRIORITY
(3) 014360 012700 000000      MOV     #0,R0
(3) 014364 104441          TRAP    C$SPP,
1305 014366 013702 003042      MOV     RLBAS,R2      ;SET RL11 BASE ADDRESS POINTER
1316          ;CHECK IF POWER FAILURE WAIT IS NEEDED
1317
1318 014372 005737 003464          TST     PWRFLG      ;NEEDED???
1319 014376 001472          BEQ     8$      ;NO, SKIP
1320
1321 014400 013705 003046          MOV     RLDRV,R5      ;DRIVE SELECT
1322 014404 052705 000200      BIS     #CRDYMSK,R5  ;SET CRDY
1323 014410 010562 000000      MOV     R5,RLCS(R2)  ;SELECT DRIVE
1324 014414 012701 000170      MOV     #120,R1      ;INITIALIZE WAIT COUNT
1325 014420 032762 000001 000000  9$:  BIT     #DRDYMSK,RLCS(R2) ;DRIVE UP YET?
1326 014426 001056          BNE     8$      ;YES START TEST
1327
1328 014430          WAITMS #10.      ;WAIT A SECOND
1329 014442 005301          DEC     R1      ;SIXTY GONE BY
1330 014444 001365          BNE     9$      ;NO
1331 014446          PRINTF #FMT24,#NOPWR
(8) 014446 012746 006176      MOV     #NOPWR,-(SP)
(7) 014452 012746 011632      MOV     #FMT24,-(SP)
(6) 014456 012746 000002      MOV     #2,-(SP)
(3) 014462 010600          MOV     SP,R0
(4) 014464 104417          TRAP    C$PNTF
(4) 014466 062706 000006      ADD     #6,SP
1332 014472          PRINTF #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
(11) 014472 005046          CLR     -(SP)
(11) 014474 153716 003047      BISB   RLDRV+1,(SP)
(10) 014500 012746 006152      MOV     #DRVNAM,-(SP)
(9) 014504 013746 003042      MOV     RLBAS,-(SP)
(8) 014510 012746 006141      MOV     #BASADD,-(SP)
(7) 014514 012746 011026      MOV     #FMT5,-(SP)
(6) 014520 012746 000005      MOV     #5,-(SP)
(3) 014524 010600          MOV     SP,R0
(4) 014526 104417          TRAP    C$PNTF
(4) 014530 062706 000014      ADD     #14,SP
1333 014534          PRINTF #FMT3
(7) 014534 012746 011012      MOV     #FMT3,-(SP)
(6) 014540 012746 000001      MOV     #1,-(SP)
(3) 014544 010600          MOV     SP,R0
(4) 014546 104417          TRAP    C$PNTF
(4) 014550 062706 000004      ADD     #4,SP
  
```

1334	014554			DODU	PSETNM		;DROP DRIVE
(3)	014554	013700	003456	MOV	PSETNM,RO		
(3)	014560	104451		TRAP	CSDODU		
1335	014562			DOCLN			
(3)	014562	104444		TRAP	CSDCLN		
1336	014564						
1337				8\$:			
1338	014564			ENDINIT			
(3)	014564			L10015:			
(3)	014564	104411		TRAP	C\$INIT		
1339	014566			ENDMOD			
1340							

1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
(7)
(6)
(5)
(4)
(3)
(2)
1355
1356
1357
1358
1359
1360
(8)
(7)
(6)
(3)
(4)
(4)
1361
(11)
(11)
(10)
(9)
(8)
(7)
(6)
(3)
(4)
(4)
1362
1363
(7)
(6)
(3)
(4)
(4)
1364
1365
(3)
(3)
1366
1367
1368

014566
014566 005037 003462
014572
014572 012746 000340
014576 012746 015576
014602 013746 003244
014606 012746 000003
014612 104437
014614 062706 000010

014620 013702 003042
014624 005762 000000
014630 005737 003462
014634 001447
014636
014636 012746 007645
014642 012746 011632
014646 012746 000002
014652 010600
014654 104417
014656 062706 000006
014662
014662 005046
014664 153716 003047
014670 012746 006152
014674 013746 003042
014700 012746 006141
014704 012746 011026
014710 012746 000005
014714 010600
014716 104417
014720 062706 000014

014724
014724 012746 011012
014730 012746 000001
014734 010600
014736 104417
014740 062706 000004

014744
014744 013700 003456
014750 104451
014752 000460
014754 013705 003046
014760 052705 000200

.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 #NOCTLR,-(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

```
1369 014764 010562 000000      MOV      R5,RLCS(R2)      ;LOAD IN THE DRIVE NUMBER
1370 014770 032762 000001 000000  BIT      #DRDYMSK,RLCS(R2) ;IS DRIVE READY?
1371 014776 001046          BNE      2$              ;BRANCH TO PERFORM TESTS IF DRIVE IS READY
1372 015000          PRINTF  #FMT24,#NOTRDY ;PRINT MSG. 'DRIVE DROPPED - DID NOT RESPOND
(8) 015000 012746 007703      MOV      #NOTRDY,-(SP)
(7) 015004 012746 011632      MOV      #FMT24,-(SP)
(6) 015010 012746 000002      MOV      #2,-(SP)
(3) 015014 010400          MOV      SP,RO
(4) 015016 1044 7          TRAP    C$PNTF
(4) 015020 062706 000006      ADD      #6,SP
1373          ;/WITH 'READY''
1374 015024          PRINTF  #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
(11) 015024 005046          CLR      -(SP)
(11) 015026 153716 003047      BISB    RLDRV+1,(SP)
(10) 015032 012746 006152      MOV      #DRVNAM,-(SP)
(9) 015036 013746 003042      MOV      RLBAS,-(SP)
(8) 015042 012746 006141      MOV      #BASADD,-(SP)
(7) 015046 012746 011026      MOV      #FMT5,-(SP)
(6) 015052 012746 000005      MOV      #5,-(SP)
(3) 015056 1044 7          MOV      SP,RO
(4) 015060 104417          TRAP    C$PNTF
(4) 015062 062706 000014      ADD      #14,SP
1375          ;PRINT DRIVE INFORMATION
1376 015066          PRINTF  #FMT3
(7) 015066 012746 011012      MOV      #FMT3,-(SP)
(6) 015072 012746 000001      MOV      #1,-(SP)
(3) 015076 010600          MOV      SP,RO
(4) 015100 104417          TRAP    C$PNTF
(4) 015102 062706 000004      ADD      #4,SP
1377 015106          DODU    PSETNM          ;DO DROP UNIT ON DRIVE
(3) 015106 013700 003456      MOV      PSETNM,RO
(3) 015112 104451          TRAP    C$DODU
1378 015114          2$:    CLRVEC  ERRVEC          ;RELEASE ERROR VECTOR
(3) 015114 013700 003244      MOV      ERRVEC,RO
(3) 015120 104436          TRAP    C$CVEC
1379 015122          ENDAUTO
(3) 015122          L10016:
(3) 015122 104461          TRAP    C$AUTO
1380
```

```
1382
1383
1384
1385 015124
1386 015124
1387
1388 015124
      (7) 015124 012746 000340
      (6) 015130 012746 015576
      (5) 015134 013746 003244
      (4) 015140 012746 000003
      (3) 015144 104437
      (2) 015146 062706 000010
1389
1390 015152
      (3) 015152 012700 000007
      (3) 015156 104441
1391 015160 032762 000200 000000 2$:
1392 015166 001407
1393 015170 053762 003046 000000
1394 015176 032762 000001 000000
1395 015204 001005
1396 015206 3$:
1397 015220 5$:
      (3) 015220 013700 003044
      (3) 015224 104436
1398 015226 005737 003464
1399 015232 001402
1400 015234 005337 003464
1401 015240 7$:
      (3) 015240 013700 003244
      (3) 015244 104436
1402 015246
      (3) 015246 104433
1403
1404 015250
      (3) 015250
      (3) 015250 104412
1405
1406 015252
1407 015252 000240
1408 015254
      (3) 015254
      (3) 015254 104453
1409
1410 015256
1411
```

.SBTTL 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,RO
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
3\$: WAITMS #3 ;WAIT 300 MS
5\$: CLRVEC RLVEC ;RELEASE VEC
MOV RLVEC,RO
TRAP C\$CVEC
TST PWRFLG ;PWR FAIL SET
BEQ 7\$;NO
DEC PWRFLG
7\$: CLRVEC ERRVEC
MOV ERRVEC,RO
TRAP C\$CVEC
BRESET ;TAKE CARE OF LSI-11
TRAP C\$RESET

ENDCLN
L10017:
TRAP C\$CLEAN

BGNDU
NOP

ENDDU
L10020:
TRAP C\$DU

ENDMOD

```
1413 .SBTTL GLOBAL SUBROUTINES
1414
1415 015256 BGNMOD GLBSUB
1416
1417
1418 015256 012737 000160 002116 TIME: MOV #160,L$DLY ;GET OUTER DELAY LOOP
1419 015264 005237 003476 INC TIM.US ;US-WAIT ROUTINE INDICATOR
1420 015270 013737 003466 003472 MOV XDELAY,MININC ;SAVE ORIGINAL US-WAIT
1421 015276 005437 003466 NEG XDELAY ;GET NEGATIVE OF FACTOR
1422 015302 READBUS ;Q - BUS?
(3) 015302 104407 TRAP C$RDBU
1423 015304 BCOMPLETE 2$ ;BRANCH - IF YES
(2) 015304 103420 BCS 2$
1424 015306 1$: DELAY #1. ;WAIT
(2) 015306 012727 000001 MOV ##1.,(PC)+
(2) 015312 000000 .WORD 0
(2) 015314 013727 002116 MOV L$DLY,(PC)+
(2) 015320 000000 .WORD 0
(2) 015322 005367 177772 DEC -6(PC)
(2) 015326 001375 BNE -4
(2) 015330 005367 177756 DEC -22(PC)
(2) 015334 001367 BNE -20
1425 015336 005237 003466 INC XDELAY ;WAIT FACTOR EXPIRED?
1426 015342 002761 BLT 1$ ;BRANCH - IF NO
1427 015344 000422 BR 4$ ;GET TIME
1428 015346 012737 000065 002116 2$: MOV #65,L$DLY ;GET OUTER DELAY LOOP
1429 015354 3$: DELAY #1. ;WAIT WITH RESPECT TO FONZ BUS
(2) 015354 012727 000001 MOV ##1.,(PC)+
(2) 015360 000000 .WORD 0
(2) 015362 013727 002116 MOV L$DLY,(PC)+
(2) 015366 000000 .WORD 0
(2) 015370 005367 177772 DEC -6(PC)
(2) 015374 001375 BNE -4
(2) 015376 005367 177756 DEC -22(PC)
(2) 015402 001367 BNE -20
1430 015404 005237 003466 INC XDELAY ;WAIT FACTOR EXPIRED?
1431 015410 002761 BLT 3$ ;BRANCH - IF NO
1432 015412 063737 003472 003132 4$: ADD MININC,TEMPO ;GET TIME EXPIRED
1433 015420 000207 RTS PC ;RETURN
1434
1435
1436 015422 012737 000160 00:116 XTIME: MOV #160,L$DLY ;GET OUTER DELAY LOOP
1437 015430 005037 003476 CLR TIM.US ;MS. WAIT INDICATOR
1438 015434 013737 003470 003502 MOV YDELAY,MAJINC ;SAVE ORIGINAL WAIT MS
1439 015442 006337 003470 ASL YDELAY ;MULTIPLY BY FACTOR 4
1440 015446 006337 003470 ASL YDELAY
1441 015452 005437 003470 NEG YDELAY ;GET NEGATIVE OF RESULT
1442 015456 READBUS ;Q - BUS?
(3) 015456 104407 TRAP C$RDBU
1443 015460 BNCMPLETE 1$ ;BRANCH - IF NO
(2) 015460 103023 BCC 1$
1444 015462 012737 000150 002116 2$: MOV #150,L$DLY ;GET OUTER DELAY LOOP
1445 015470 DELAY #20 ;WAIT WITH RESPECT TO FONZ BUS
(2) 015470 012727 000020 MOV ##20,(PC)+
(2) 015474 000000 .WORD 0
(2) 015476 013727 002116 MOV L$DLY,(PC)+
```

```
(2) 015502 000000 .WORD 0
(2) 015504 005367 177772 DEC -6(PC)
(2) 015510 001375 BNE -4
(2) 015512 005367 177756 DEC -22(PC)
(2) 015516 001367 BNE -20
1446 015520 005237 003470 INC YDELAY ;WAIT FACTOR EXPIRED
1447 015524 002761 BLT 2$ ;BRANCH - IF NO
1448 015526 000417 BR 3$ ;GET TIME
1449 015530 1$: DELAY #10 ;WAIT
(2) 015530 012727 000010 MOV ##10,(PC)+
(2) 015534 000000 .WORD 0
(2) 015536 013727 002116 MOV L$DLY,(PC)+
(2) 015542 000000 .WORD 0
(2) 015544 005367 177772 DEC -6(PC)
(2) 015550 001375 BNE -4
(2) 015552 005367 177756 DEC -22(PC)
(2) 015556 001367 BNE -20
1450 015560 005237 003470 INC YDELAY ;WAIT FACTOR EXPIRED?
1451 015564 002761 BLT 1$ ;BRANCH - IF NO
1452 015566 063737 003502 003474 3$: ADD MAJINC,TEMP ;GET EXPIRED TIME
1453 015574 000207 RTS PC ;RETURN
1454
1455
1456
1457 015576 BGNSRV
1458 ;TRAP HANDLER. INDICATES OCCURRENCE OF A TRAP.
1459
1460
1461 015576 005237 003462 TRPHAN: INC TRPFLG
1462
1463 015602 ENDSRV
(3) 015602 L10021:
(2) 015602 000002 RTI
1464
1465 015604 BGNSRV
1466 ;INTERRUPT HANDLER. ABORTS WAIT TIMER AND STORES RL11 REGISTERS.
1467
1468
1469 015604 INTHLR:
1470
1471 015604 012237 003060 MOV (R2)+,T.CS ;STORE RL REGISTERS
1472 015610 012237 003062 MOV (R2)+,T.BA
1473 015614 012237 003064 MOV (R2)+,T.DA
1474 015620 011237 003066 MOV (R2),T.MP
1475 015624 012737 177777 003022 MOV #-1,DONE ;SET DONE FLAG
1476 015632 013702 003042 MOV RLBAS,R2 ;RESTORE R2
1477 015636 ABORTWAIT
1478
1479 015662 ENDSRV
(3) 015662 L10022:
(2) 015662 000002 RTI
1480
```

```

1482
1483
1484          :      ERROR LIMIT CHECKING ROUTINE
1485          :
1486
1487          :      CKERLM: CMP      @ERRPOINT,ERLIMW      ;TEST IF ERROR LIMIT EXCEEDED
1488          :      BLT      1$      ;NO - SKIP
1489          :      INLOOP      ;CHECK IF IN ERROR LOOP
1490          :      TRAP      C$INLP
1491          :      BCOMPLETE      1$      ;YES - SKIP
1492          :      BCS      1$
1493          :      PRINTF      #FMT25,ERLIMW,#MEXERS
1494          :      MOV      #MEXERS,-(SP)
1495          :      MOV      ERLIMW,-(SP)
1496          :      MOV      #FMT25,-(SP)
1497          :      MOV      #3,-(SP)
1498          :      MOV      SP,RO
1499          :      TRAP      C$PNTF
1500          :      ADD      #10,SP
1501          :      PRINTF      #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
1502          :      CLR      -(SP)
1503          :      BISB      RLDRV+1,(SP)
1504          :      MOV      #DRVNAM,-(SP)
1505          :      MOV      RLBAS,-(SP)
1506          :      MOV      #BASADD,-(SP)
1507          :      MOV      #FMT5,-(SP)
1508          :      MOV      #5,-(SP)
1509          :      MOV      SP,RO
1510          :      TRAP      C$PNTF
1511          :      ADD      #14,SP
1512          :      PRINTF      #FMT3
1513          :      MOV      #FMT3,-(SP)
1514          :      MOV      #1,-(SP)
1515          :      MOV      SP,RO
1516          :      TRAP      C$PNTF
1517          :      ADD      #4,SP
1518          :      DODU      PSETNM      ;DROP DRIVE
1519          :      MOV      PSETNM,RO
1520          :      TRAP      C$DODU
1521          :      DOCLN
1522          :      TRAP      C$DCLN      ;GO TO CLEAN UP
1523          :      RTS      PC
1524
1525          :
1526          :      READ AND STORE ALL RL11 REGISTERS
1527          :      READRL: MOV      RLCSR(R2),T.CS      ;GET CS REG
1528          :      MOV      RLBA(R2),T.BA      ;GET BUS ADDRESS REG
1529          :      MOV      RLDA(R2),T.DA      ;GET DISK ADDRESS
1530          :      MOV      RLMP(R2),T.MP      ;GET MULTI-PURPOSE REG
1531          :      RTS      PC      ;RETURN
1532
1533          :
1534          :      WAIT FOR CONTROLLER TIMEOUT TO FORCE INTERRUPT ROUTINE
1535          :      WAITIN: MOV      (SP) -(SP)      ;MAKE ROOM FOR ERROR POINTER
1536          :      CLR      2(SP)      ;CLEAR FOR POINTER
1537          :      BIT      #CRDYMSK,RLCSR(R2)      ;TEST IF CONTROLLER READY
1538          :      BEQ      4$      ;NO - SKIP TO WAIT
1539          :      JSR      PC,READRL      ;READ ALL RL REGS
1540
1541          :
1542          :
1543          :
1544          :
1545          :
1546          :
1547          :
1548          :
1549          :
1550          :
1551          :
1552          :
1553          :
1554          :
1555          :
1556          :
1557          :
1558          :
1559          :
1560          :
1561          :
1562          :
1563          :
1564          :
1565          :
1566          :
1567          :
1568          :
1569          :
1570          :
1571          :
1572          :
1573          :
1574          :
1575          :
1576          :
1577          :
1578          :
1579          :
1580          :
1581          :
1582          :
1583          :
1584          :
1585          :
1586          :
1587          :
1588          :
1589          :
1590          :
1591          :
1592          :
1593          :
1594          :
1595          :
1596          :
1597          :
1598          :
1599          :
1600          :
1601          :
1602          :
1603          :
1604          :
1605          :
1606          :
1607          :
1608          :
1609          :
1610          :
1611          :
1612          :
1613          :
1614          :
1615          :
1616          :
1617          :
1618          :
1619          :
1620          :
1621          :
1622          :
1623          :
1624          :
1625          :
1626          :
1627          :
1628          :
1629          :
1630          :
1631          :
1632          :
1633          :
1634          :
1635          :
1636          :
1637          :
1638          :
1639          :
1640          :
1641          :
1642          :
1643          :
1644          :
1645          :
1646          :
1647          :
1648          :
1649          :
1650          :
1651          :
1652          :
1653          :
1654          :
1655          :
1656          :
1657          :
1658          :
1659          :
1660          :
1661          :
1662          :
1663          :
1664          :
1665          :
1666          :
1667          :
1668          :
1669          :
1670          :
1671          :
1672          :
1673          :
1674          :
1675          :
1676          :
1677          :
1678          :
1679          :
1680          :
1681          :
1682          :
1683          :
1684          :
1685          :
1686          :
1687          :
1688          :
1689          :
1690          :
1691          :
1692          :
1693          :
1694          :
1695          :
1696          :
1697          :
1698          :
1699          :
1700          :
1701          :
1702          :
1703          :
1704          :
1705          :
1706          :
1707          :
1708          :
1709          :
1710          :
1711          :
1712          :
1713          :
1714          :
1715          :
1716          :
1717          :
1718          :
1719          :
1720          :
1721          :
1722          :
1723          :
1724          :
1725          :
1726          :
1727          :
1728          :
1729          :
1730          :
1731          :
1732          :
1733          :
1734          :
1735          :
1736          :
1737          :
1738          :
1739          :
1740          :
1741          :
1742          :
1743          :
1744          :
1745          :
1746          :
1747          :
1748          :
1749          :
1750          :
1751          :
1752          :
1753          :
1754          :
1755          :
1756          :
1757          :
1758          :
1759          :
1760          :
1761          :
1762          :
1763          :
1764          :
1765          :
1766          :
1767          :
1768          :
1769          :
1770          :
1771          :
1772          :
1773          :
1774          :
1775          :
1776          :
1777          :
1778          :
1779          :
1780          :
1781          :
1782          :
1783          :
1784          :
1785          :
1786          :
1787          :
1788          :
1789          :
1790          :
1791          :
1792          :
1793          :
1794          :
1795          :
1796          :
1797          :
1798          :
1799          :
1800          :
1801          :
1802          :
1803          :
1804          :
1805          :
1806          :
1807          :
1808          :
1809          :
1810          :
1811          :
1812          :
1813          :
1814          :
1815          :
1816          :
1817          :
1818          :
1819          :
1820          :
1821          :
1822          :
1823          :
1824          :
1825          :
1826          :
1827          :
1828          :
1829          :
1830          :
1831          :
1832          :
1833          :
1834          :
1835          :
1836          :
1837          :
1838          :
1839          :
1840          :
1841          :
1842          :
1843          :
1844          :
1845          :
1846          :
1847          :
1848          :
1849          :
1850          :
1851          :
1852          :
1853          :
1854          :
1855          :
1856          :
1857          :
1858          :
1859          :
1860          :
1861          :
1862          :
1863          :
1864          :
1865          :
1866          :
1867          :
1868          :
1869          :
1870          :
1871          :
1872          :
1873          :
1874          :
1875          :
1876          :
1877          :
1878          :
1879          :
1880          :
1881          :
1882          :
1883          :
1884          :
1885          :
1886          :
1887          :
1888          :
1889          :
1890          :
1891          :
1892          :
1893          :
1894          :
1895          :
1896          :
1897          :
1898          :
1899          :
1900          :
1901          :
1902          :
1903          :
1904          :
1905          :
1906          :
1907          :
1908          :
1909          :
1910          :
1911          :
1912          :
1913          :
1914          :
1915          :
1916          :
1917          :
1918          :
1919          :
1920          :
1921          :
1922          :
1923          :
1924          :
1925          :
1926          :
1927          :
1928          :
1929          :
1930          :
1931          :
1932          :
1933          :
1934          :
1935          :
1936          :
1937          :
1938          :
1939          :
1940          :
1941          :
1942          :
1943          :
1944          :
1945          :
1946          :
1947          :
1948          :
1949          :
1950          :
1951          :
1952          :
1953          :
1954          :
1955          :
1956          :
1957          :
1958          :
1959          :
1960          :
1961          :
1962          :
1963          :
1964          :
1965          :
1966          :
1967          :
1968          :
1969          :
1970          :
1971          :
1972          :
1973          :
1974          :
1975          :
1976          :
1977          :
1978          :
1979          :
1980          :
1981          :
1982          :
1983          :
1984          :
1985          :
1986          :
1987          :
1988          :
1989          :
1990          :
1991          :
1992          :
1993          :
1994          :
1995          :
1996          :
1997          :
1998          :
1999          :
2000          :

```

```

1511 016100 005737 003022          TST     DONE          ;TEST IF INTERRUPT OCCURRED
1512 016104 001435                   BEQ     5$            ;NO - GO SET NO INTERRUPT ERR FLAG
1513 016106 012766 006316 000002 1$:    MOV     #MTOSLOW,2(SP) ;ELSE SET TOO SLOW ERROR POINTER
1514 016114 032737 002000 003060          BIT     #OPIERR,T.CS ;TEST IF OPI SET
1515 016122 001403                   BEQ     2$            ;NO - SKIP
1516 016124 012766 006336 000002          MOV     #MDRRES,2(SP) ;SET MESSAGE FOR NO DRIVE RESPONSE
1517 016132 000207                   RTS     PC            ;RETURN
1518 016134                   4$:    WAITMS #3          ;WAIT 300 MS FOR TIMEOUT
1519 016146 032762 000200 000000          BIT     #CRDYMSK,RLCS(R2) ;TEST IF READY NOW SET
1520 016154 001006                   BNE     3$            ;YES - SKIP
1521 016156 004737 016024                   JSR     PC,READRL    ;READ RL REGS
1522 016162 012766 006407 000002          MOV     #MCONHNG,2(SP) ;SET MESSAGE FOR CONTROLLER HUNG
1523 016170 000760                   BR      2$            ;SKIP
1524 016172 005737 003022          3$:    TST     DONE          ;ELSE CHECK IF INTERRUPT OCCURRED
1525 016176 001343                   BNE     1$            ;YES - SKIP TO SET TOO SLOW
1526 016200 004737 016024          5$:    JSR     PC,READRL    ;READ RL REGS
1527 016204 012766 006354 000002          MOV     #MNOINT,2(SP) ;ELSE SET NO INTERRUPT FLAG
1528 016212 000747                   BR      2$            ;GO TO RETURN
1529
1530          ; OPERATION AND TEST INITIALIZE ROUTINE
1531 016214 005037 003020          TSTINT: CLR     OPFLAG    ;CLEAR OPERATION FLAGS
1532 016220 105037 003461          CLR     MOERCT      ;RESET INHIBIT ERROR COUNTING
1533 016224 005037 003030          CLR     MORECE      ;RESET MORE COMPARE ERRORS
1534 016230 000207          RTS     PC
1535
1536          ; GET STATUS AND GET STATUS WITH RESET ROUTINE
1537 016232 013746 003142          GSTAT:  MOV     TEMP4,-(SP) ;STORE TEMP4
1538 016236 012737 000013 003142          MOV     #GETSTAT!DRSET,TEMP4 ;SET FOR RESET
1539 016244 000412                   BR      GSTATG
1540 016246 013746 003142          GSTATC: MOV     TEMP4,-(SP) ;STORE TEMP4
1541 016252 012737 000003 003142          MOV     #GETSTAT,TEMP4 ;SET FOR NO RESET
1542 016260 000404                   BR      GSTATG
1543 016262 013746 003142          GSTAT:  MOV     TEMP4,-(SP) ;STORE TEMP4
1544 016266 005037 003142          CLR     TEMP4        ;SET FOR SAVE L. AND T. REGS
1545 016272 010346          GSTATG: MOV     R3,-(SP) ;STORE R3
1546 016274 013703 003016          MOV     SSINDX,R3    ;GET SUBROUTINE INDEX
1547 016300 005723          TST     (R3)+        ;BUMP IT FOR NEXT ENTRY
1548 016302 016663 000004 002420          MOV     4(SP),SUBSTK(R3) ;INSERT THIS CALL
1549 016310 162763 000004 002420          SUB     #4,SUBSTK(R3) ;ADJUST IT TO CALLING LOCATION
1550 016316 010337 003016          MOV     R3,SSINDX   ;STORE IT BACK
1551 016322 010046          MOV     R0,-(SP)    ;STORE R0
1552 016324 010146          MOV     R1,-(SP)    ;STORE R1
1553 016326 012737 000002 003032          MOV     #2,ERRSWI   ;SET FOR NO ERROR RETURN
1554 016334 032737 000010 003142          BIT     #DRSET,TEMP4 ;TEST IF DRIVE RESET
1555 016342 001460          BEQ     11$         ;NO - SKIP
1556 016344 032762 040000 000000          BIT     #DRVERR,RLCS(R2) ;TEST IF DRIVE ERROR SET
1557 016352 001405          BEQ     49$         ;NO - SKIP
1558 016354          WAITMS #3          ;WAIT FOR 300 MS FOR DRIVE TO SETTLE
1559 016366 012701 000062          49$:    MOV     #50.,R1    ;INITIALIZE WAIT COUNT
1560 016372 004737 016262          50$:    JSR     PC,GSTAT    ;GET DRIVE STATUS
1561 016376 017062          3$:    BR      3$
1562 016400 032737 000001 003060          BIT     #DRDYMSK,T.CS ;TEST IF DRIVE READY
1563 016406 001054          BNE     5$            ;YES - GO DO CLEAR
1564 016410 032737 000020 003066          BIT     #HOSTAT,T.MP ;ELSE TEST IF HEADS OUT
1565 016416 001010          BNE     51$         ;YES - BYPASS RELOAD WAIT FLAG SETTING
1566 016420 032737 144000 003066          BIT     #SPDSTAT!HCESTAT!WDESTAT,T.MP ;TEST IF DRIVE HAS ERROR

```

```

1567                                     ;THAT CAUSED HEADS TO
1568                                     ;UNLOAD
1569 016426 001444                      BEQ      5$      ;NO - SKIP
1570 016430 052737 040000 003020      BIS      #RELDWT,OPFLAG ;ELSE SET WAIT FLAG
1571 016436 000440                      BR       5$      ;SKIP TO CLEAR
1572 016440 032737 040000 003060 51$: BIT      #DRVERR,T.CS  ;TEST IF DRIVE ERROR NOW
1573 016446 001034                      BNE     5$      ;YES - SKIP TO CLEAR
1574 016450                              WAITMS  #1      ;WAIT FOR DRIVE TO GET ERROR, RDY, OR HEADS OUT
1575 016462 005301                      DEC      R1     ;DEC WAIT COUNTER
1576 016464 001342                      BNE     50$     ;IF NOT DONE, LOOP
1577 016466 012703 010375              MOV      #UNDEF,R3 ;MESSAGE FOR UNDEFINED STATE
1578 016472                              ERRHRD  10001,,,ERR1
(4) 016472 104456                      TRAP    C$ERHRD
(5) 016474 023421                      .WORD   10001
(5) 016476 000000                      .WORD   0
(5) 016500 011724                      .WORD   ERR1
1579 016502 000565                      BR       14$     ;EXIT
1580 016504 005737 003142 11$:        TST     TEMP4   ;TEST IF SAVE REGISTERS
1581 016510 001011:                    BNE     5$      ;NO SKIP
1582 016512 012701 000004              MOV      #4,R1   ;SET SAVE COUNT
1583 016516 012703 003060              MOV      #L,MP+2,R3 ;SET ADDRESS OF FIRST SAVE
1584 016522 014346 8$:                MOV      -(R3),-(SP) ;PUT REG ON STACK
1585 016524 005301                      DEC      R1     ;DEC COUNT
1586 016526 001375                      BNE     8$      ;LOOP UNTIL ALL SAVED
1587 016530 012737 000003 003054      MOV      #GETSTAT,L.DA ;SET FOR GET STATUS
1588 016536 000403                      BR       6$      ;SKIP
1589 016540 013737 003142 003054 5$: MOV      TEMP4,L.DA ;INSERT PRESET FOR STATUS
1590 016546 6$:
1591 016546 005037 003022              CLR      DONE   ;CLEAR INTERRUPT FLAG
1592 016552 013737 003046 003050      MOV      RLDRV,L.CS ;SET UP TO GET STATUS
1593 016560 042737 002000 003050      BIC      #BIT10,L.CS ;CLEAR FOR DRIVE 4 - 7 SPEC'D
1594 016566 052737 000104 003050      BIS      #GTSTAT,L.CS
1595 016574 013762 003054 000004      MOV      L.DA,RLDA(R2) ;LOAD RL REGS
1596 016602 013762 003050 000000      MOV      L.CS,RLCSR(R2) ;LOAD CS REG
1597 016610                              WAITUS  #1      ;WAIT 100 US FOR INTERRUPT
1598 016622 005737 003022              TST     DONE   ;CHECK IF INTERRUPT OCCURRED
1599 016626 001504                      BEQ     1$      ;NO - SKIP
1600 016630 013737 003066 003074 4$: MOV      T.MP,T.STAT ;STORE MP REGISTER
1601 016636 042737 177770 003074      BIC      #*C<STAMSK>,T.STAT ;CLEAR ALL BUT STATE
1602 016644 032737 000010 003054      BIT      #DRSET,L.DA ;TEST IF RESET WAS SPECIFIED
1603 016652 001503                      BEQ     3$      ;NO - SKIP TO EXIT
1604 016654 032737 040000 003020      BIT      #RELDWT,OPFLAG ;TEST IF RELOAD WAIT FLAG SET
1605 016662 001427                      BEQ     12$     ;NO - SKIP
1606 016664 012701 001130              MOV      #600,R1 ;SET WAIT COUNT FOR 60 SECONDS
1607 016670 032762 000001 000000 13$: BIT      #DRDYMSK,RLCS(R2) ;TEST IF DRIVE NOW READY
1608 016676 001021                      BNE     12$     ;YES - SKIP
1609 016700                              WAITMS  #1      ;CALL WAIT
1610 016712 005301                      DEC      R1     ;DEC COUNT
1611 016714 001365                      BNE     13$     ;LOOP IF NOT 0
1612 016716 004737 016262              JSR     PC,GSTAT ;GET DRIVE STATUS
1613 016722 017062                      3$      ;ERROR RETURN
1614 016724 012703 010442              MOV      #MRLFAL,R3 ;SET RESULT MESSAGE POINTER
1615 016730                              ERRHRD  10003,,,ERR1
(4) 016730 104456                      TRAP    C$ERHRD
(5) 016732 023423                      .WORD   10003
(5) 016734 000000                      .WORD   0

```



```
(5) 016736 011724 .WORD ERR1
1616 016740 000446 BR 14$ ;GO TO EXIT
1617 016742 12$: WAITUS #10. ;WAIT FOR 1MS
1618 016754 004737 016262 JSR PC,GSTAT ;GET DRIVE STATUS
1619 016760 017062 3$ ;
1620 016762 032737 100000 003060 BIT #ANYERR,T.CS ;TEST IF ANY ERROR
1621 016770 001434 3$ ;NO - SKIP
1622 016772 032737 001000 003066 BIT #VCSTAT,T.MP ;CHECK IF VOLUME CHECK RESET
1623 017000 001403 BEQ 7$ ;YES SKIP
1624 017002 012703 006443 MOV #VCNRST,R3 ;SET REASON POINTER
1625 017006 000417 BR 2$ ;EXIT
1626 017010 032737 040000 003060 7$: BIT #DRVERR,T.CS ;CHECK IF DRIVE ERROR
1627 017016 001405 BEQ 9$ ;NO - SKIP
1628 017020 ERRHRD 10004...ERR6
(4) 017020 104456 TRAP C$ERHRD
(5) 017022 023424 .WORD 10004
(5) 017024 000000 .WORD 0
(5) 017026 012226 .WORD ERR6
1629 017030 000412 BR 14$ ;EXIT
1630 017032 012703 006464 9$: MOV #UNXERR,R3 ;SET REASON POINTER
1631 017036 000403 BR 2$ ;EXIT
1632 017040 004737 016056 1$: JSR PC,WAITIN ;WAIT FOR INTERRUPT
1633 017044 012603 MOV (SP)+,R3 ;STORE REASON POINTER FOR RETURN
1634 017046 2$: ERRHRD 10002...ERR1
(4) 017046 104456 TRAP C$ERHRD
(5) 017050 023422 .WORD 10002
(5) 017052 000000 .WORD 0
(5) 017054 011724 .WORD ERR1
1635 017056 005037 003032 14$: CLR ERRSWI ;CLEAR FOR ERROR RETURN
1636 017062 005737 003142 3$: TST TEMP4 ;TEST IF REGISTERS WERE SAVED
1637 017066 001007 BNE 22$ ;NO - SKIP
1638 017070 012703 003050 MOV #L.CS,R3 ;SET POINTER TO RESTORE
1639 017074 012701 000004 MOV #4,R1 ;SET REGISTER COUNT
1640 017100 012623 20$: MOV (SP)+,(R3)+ ;RESTORE REG
1641 017102 005301 DEC R1 ;DEC COUNT
1642 017104 001375 BNE 20$ ;LOOP UNTIL ALL ARE RESTORED
1643 017106 162737 000002 003016 22$: SUB #2,SSINDX ;REMOVE ENTRY FROM SUBROUT STACK
1644 017114 012601 MOV (SP)+,R1 ;RESTORE R1
1645 017116 012600 MOV (SP)+,R0 ;RESTORE R0
1646 017120 012603 MOV (SP)+,R3 ;RESTORE R3
1647 017122 012637 003142 MOV (SP)+,TEMP4 ;RESTORE TEMP4
1648 017126 005737 003032 TST ERRSWI ;TEST IF ERROR RETURN
1649 017132 001403 BEQ 99$ ;YES - SKIP
1650 017134 063716 003032 ADD ERRSWI,(SP) ;ADD IN ERROR RETURN
1651 017140 000207 RTS PC
1652 017142 017616 000000 99$: MOV @ (SP),(SP) ;SET ERROR RETURN ADDRESS
1653 017146 000207 RTS PC
1654
1655
1656 ; SEEK ROUTINE
1657 017150 012737 177777 003134 XSEKT: MOV #-1,TEMP1 ;SET SPECIAL TIMING SEEK FLAG
1658 017156 000402 BR XSEK1
1659 017160 005037 003134 XSEK: CLR TEMP1 ;CLEAR SPECIAL SEEK FOR TIMING FLAG
1660 017164 010346 XSEK1: MOV R3,-(SP) ;STORE R3
1661 017166 013703 003016 MOV SSINDX,R3 ;GET SUBROUTINE INDEX
1662 017172 005723 TST (R3)+ ;BUMP IT FOR NEXT ENTRY
```

1663	017174	016663	000002	002420	MOV	2(SP),SUBSTK(R3)	:INSERT THIS CALL
1664	017202	162763	000004	002420	SUB	#4,SUBSTK(R3)	:ADJUST IT TO CALLING LOCATION
1665	017210	010337	003016		MOV	R3,SSINDX	:STORE IT BACK
1666	017214	010046			MOV	R0,-(SP)	
1667	017216	010146			MOV	R1,-(SP)	
1668	017220	010546			MOV	R5,-(SP)	:STORE REG
1669	017222	012737	000002	003032	MOV	#2,ERRSWI	:SET FOR NO ERROR RETURN
1670	017230	005037	003112		CLR	DIFAUG	:CLEAR DIFFERENCE AUGMENT (FOR SEEKING : PAST GUARD BAND)
1671							
1672	017234	004737	022340		JSR	PC,GETPOS	:GET PRESENT POSITION
1673	017240	017672			65\$		
1674	017242	013737	003120	003114	MOV	CURCYL,OLDCYL	:MOVE CURRENT TO OLD CYLINDER
1675	017250	023737	003116	002316	CMP	NEWCYL,HLMTW	:TEST IF NEW IS GREATER THAN 255
1676	017256	003427			BLE	3\$:NO - SKIP
1677	017260	163737	002316	003116	SUB	HLMTW,NEWCYL	:ELSE SUBTRACT 255.
1678	017266	013737	003116	003112	MOV	NEWCYL,DIFAUG	:STORE DIFFERENCE AS AUGMENT
1679	017274	013737	002316	003116	MOV	HLMTW,NEWCYL	:SET NEWCYL AS 255.
1680	017302	022737	000001	002312	CMP	#1,T.DRIVE	
1681	017310	001424			BEQ	6\$	
1682	017312	162737	000001	003116	SUB	#1,NEWCYL	
1683	017320	012737	000001	003124	MOV	#1,DESSGN	
1684	017326	012737	000001	003122	MOV	#1,DESDIF	
1685	017334	000451			BR	18\$	
1686	017336	005737	003116		3\$: TST	NEWCYL	:TEST IF NEWCYL HAS NEGATIVE VALUE
1687	017342	100007			BPL	6\$:NO - SKIP
1688	017344	005437	003116		NEG	NEWCYL	:ELSE MAKE IT POSITIVE
1689	017350	013737	003116	003112	MOV	NEWCYL,DIFAUG	:AND STORE IT AS AUGMENT
1690	017356	005037	003116		CLR	NEWCYL	:AND SET NEWCYL TO 0
1691	017362	013705	003120		6\$: MOV	CURCYL,R5	:COMPUTE DIFFERENCE AND NEW CYLINDER
1692	017366	163705	003116		SUB	NEWCYL,R5	:SUB NEWCYL FROM CURCYL
1693	017372	100005			BPL	13\$:IF DIFF IS POSITIVE - SKIP(REV SEEK)
1694	017374	012737	000001	003124	MOV	#1,DESSGN	:ELSE SET SIGN FOR FORWARD
1695	017402	005405			NEG	R5	:MAKE DIFFERENCE POSITIVE
1696	017404	000402			BR	14\$:SKIP
1697	017406	005037	003124		13\$: CLR	DESSGN	:SET SIGN FOR REVERSE
1698	017412	010537	003122		14\$: MOV	R5,DESDIF	:STORE DIFFERENCE
1699	017416	005737	003112		TST	DIFAUG	:IS THERE A DIFFERENCE AUGMENT
1700	017422	001416			BEQ	18\$:NO - SKIP
1701	017424	023737	003116	002316	CMP	NEWCYL,HLMTW	:CHECK IF NEW CYL IS 255.
1702	017432	001007			BNE	17\$:NO - SKIP
1703	017434	012737	000001	003124	MOV	#1,DESSGN	:ELSE FORCE SIGN FOR FORWARD
1704							: (INNER GUARD BAND)
1705	017442	022737	000001	002312	CMP	#1,T.DRIVE	
1706	017450	001003			BNE	18\$	
1707	017452	063737	003112	003122	17\$: ADD	DIFAUG,DESDIF	
1708	017460				18\$:		
1709	017460	012705	003050		MOV	#L,CS,R5	:GET L REG ADDRESS
1710	017464	012715	000106		MOV	#SEEK,(R5)	:SET FOR SEEK
1711	017470	053715	003046		BIS	RLDRV,(R5)	:INSERT DRIVE NUMBER
1712	017474	042725	002000		BIC	#BIT10,(R5)+	:CLEAR IF DRIVE 4 - 7 SPEC'D
1713	017500	005025			CLR	(R5)+	:CLEAR BUS ADDRESS
1714	017502	013715	003122		MOV	DESDIF,(R5)	:LOAD DIFFERENCE
1715	017506	012700	000007		MOV	#7,R0	:SET TO SHIFT DIFFERENCE
1716	017512	006315			21\$: ASL	(R5)	
1717	017514	005300			RO		
1718	017516	001375			BNE	21\$:LOOP UNTIL ALIGNED

1719	017520	005737	003124		TST	DESSGN		:TEST SIGN
1720	017524	001402			BEQ	23\$:SKIP IF 0
1721	017526	052715	000004		BIS	#DIRBIT,(R5)		:ELSE INSERT SIGN
1722	017532	005737	003126	23\$:	TST	DESHD		:TEST IF HEAD 0
1723	017536	001402			BEQ	25\$:YES - SKIP
1724	017540	052715	000020		BIS	#HDSSEL,(R5)		:ELSE SET HEAD BIT
1725	017544	052725	000001	25\$:	BIS	#MBSETO,(R5)+		:INSERT MARKER BIT
1726	017550	004737	020276		JSR	PC,RDYCHK		:CHECK IF DRIVE READY
1727	017554	017672			65\$			
1728	017556	005037	003022		CLR	DONE		:CLEAR INTERRUPT FLAG
1729	017562	005737	003134		TST	TEMP1		:CHECK IF SPECIAL SEEK FLAG SFT
1730	017566	001041			BNE	65\$:YES - SKIP, DO NOT START SEEK
1731	017570	014562	000004		MOV	-(R5),RLDA(R2)		:LOAD RL REGISTERS
1732	017574	014562	000002		MOV	-(R5),RLBA(R2)		
1733	017600	014562	000000		MOV	-(R5),RLCS(R2)		
1734	017604			30\$:	WAITUS	#10.		
1735	017616	005737	003022		TST	DONE		:TEST IF INTERRUPT DONE
1736	017622	001012			BNE	32\$:YES - SKIP
1737	017624	004737	016056		JSR	PC,WAITIN		:GO WAIT FOR INTERRUPT
1738	017630	012603			MOV	(SP)+,R3		:GET RESULT MESSAGE POINTER
1739	017632				ERRHRD	10005,,,ERR1		
(4)	017632	104456			TRAP	C\$ERRHD		
(5)	017634	023425			.WORD	10005		
(5)	017636	000000			.WORD	0		
(5)	017640	011724			.WORD	ERR1		
1740	017642	005037	003032		CLR	ERRSWI		:CLEAR FOR ERROR RETURN
1741	017646	000411			BR	65\$		
1742	017650	005737	003060	32\$:	TST	T.CS		:TEST IF ANY ERROR
1743	017654	100006			BPL	65\$:NO - SKIP
1744	017656				ERRHRD	10006,,,ERR6		
(4)	017656	104456			TRAP	C\$ERRHD		
(5)	017660	023426			.WORD	10006		
(5)	017662	000000			.WORD	0		
(5)	017664	012226			.WORD	ERR6		
1745	017666	005037	003032		CLR	ERRSWI		:CLEAR FOR ERROR RETURN
1746	017672	162737	000002	003016	65\$:	SUB	#2,SSINDX	:REMOVE ENTRY FROM SUBROUT STACK
1747	017700	012605			MOV	(SP)+,R5		:RESTORE REGISTERS
1748	017702	012601			MOV	(SP)+,R1		
1749	017704	012600			MOV	(SP)+,R0		
1750	017706	012603			MOV	(SP)+,R3		
1751	017710	005737	003032		TST	ERRSWI		:TEST IF ERROR RETURN
1752	017714	001403			BEQ	99\$:YES - SKIP
1753	017716	063716	003032		ADD	ERRSWI,(SP)		:ADD IN ERROR RETURN
1754	017722	000207			RTS	PC		
1755	017724	017616	000000	99\$:	MOV	@(SP),(SP)		:SET ERROR RETURN ADDRESS
1756	017730	000207			RTS	PC		
1757								
1814								
1816								
1817					:	POSITION HEADS ROUTINE. POSITIONS HEADS USING 1 CYLINDER SEEKS		
1818					:	TO CYLINDER SPECIFIED IN R5 BY THE CALLING ROUTINE		
1818	017732	010346			POSHDS:	MOV	R3,-(SP)	:SAVE REGS
1819	017734	013703	003016		MOV	SSINDX,R3		:GET SUBROUTINE INDEX
1820	017740	005723			TST	(R3)+		:BUMP IT FOR NEXT ENTRY
1821	017742	016663	000002	002420	MOV	2(SP),SUBSTK(R3)		:INSERT THIS CALL
1822	017750	162763	000004	002420	SUB	#4,SUBSTK(R3)		:ADJUST IT TO CALLING LOCATION
1823	017756	010337	003016		MOV	R3,SSINDX		:STORE IT BACK

1824	017762	010346			MOV	R3,-(SP)	
1825	017764	010446			MOV	R4,-(SP)	
1826	017766	012737	000002	003032	MOV	#2,ERRSWI	;SET FOR NO ERROR RETURN
1827	017774	004737	022340		JSR	PC,GETPOS	;GET CURRENT POSITION
1828	020000	020240			PH65\$		
1829	020002	012704	000012		MOV	#10.,R4	;SET RETRY COUNT
1830	020006				BGNSEG		
(3)	020006	104404			TRAP	C\$BSEG	
1831	020010				1\$: INLOOP		;CHECK IF IN ERROR LOOP
(3)	020010	104420			TRAP	C\$INLP	
1832	020012				BNCOMPLETE	5\$;NO - SKIP
(2)	020012	103012			BCC	5\$	
1833	020014	004737	022340		JSR	PC,GETPOS	;ELSE GET POSITION
1834	020020	020236			60\$		
1835	020022	023737	003120	003116	CMP	CURCYL,NEWCYL	;CHECK IF AT INTENDED POSITION
1836	020030	001017			BNE	8\$;NO - SKIP
1837	020032	004737	020636		JSR	PC,ONSWAP	;SWAP OLDCYL AND NEWCYL
1838	020036	000414			BR	8\$;SKIP
1839	020040	013737	003120	003114	5\$: MOV	CURCYL,OLDCYL	;IN NOT LOOPING, STORE CURCYL AS OLDCYL
1840	020046	023705	003120		CMP	CURCYL,R5	;CHECK IF HDS AT FINAL POSITION
1841	020052	001471			BEQ	60\$;YES - GO TO EXIT
1842	020054	003003			BGT	7\$;IF CURCYL > FINAL POSITION - SKIP
1843	020056	005237	003116		INC	NEWCYL	;ELSE BUMP NEWCYL (MOVE HDS IN)
1844	020062	000402			BR	8\$;SKIP
1845	020064	005337	003116		7\$: DEC	NEWCYL	;DEC NEWCYL (MOVE HDS OUT)
1846	020070	004737	017160		8\$: JSR	PC,XSEEK	;DO SEEK
1847	020074	020236			60\$		
1848	020076	012701	005670		MOV	#3000.,R1	;SET WAIT COUNT 300 MS
1849	020102	004737	022054		JSR	PC,RDYWAIT	;WAIT FOR DRIVE READY
1850	020106	020236			60\$		
1851	020110	005737	003060		TST	T,C5	;TEST IF ANY ERROR
1852	020114	100007			BPL	10\$;NO - SKIP
1853	020116				ERRHRD	10008.,ERR6	
(4)	020116	104456			TRAP	C\$ERHRD	
(5)	020120	023430			.WORD	10008	
(5)	020122	000000			.WORD	0	
(5)	020124	012226			.WORD	ERR6	
1854	020126	005037	003032		CLR	ERRSWI	;CLEAR FOR ERROR ERROR RETURN
1855	020132	000441			BR	60\$	
1856	020134	004737	022340		10\$: JSR	PC,GETPOS	;GET POSITION
1857	020140	020236			60\$		
1858	020142	023737	003120	003116	CMP	CURCYL,NEWCYL	;CHECK IF ARRIVED AT DESIRED PLACE
1859	020150	001003			BNE	15\$;NO - SKIP
1860	020152	012704	000012		14\$: MOV	#10.,R4	;ELSE INIT RETRY COUNT
1861	020156	000714			BR	1\$;GO DO NEXT SEEK
1862	020160	005737	003124		15\$: TST	DESSGN	;TEST IF GOING IN
1863	020164	001017			BNE	17\$;YES - SKIP
1864	020166	023737	003120	003116	CMP	CURCYL,NEWCYL	;CHECK IF HEADS DID NOT MOVE IN
1865	020174	003366			BGT	14\$;YES - SKIP
1866	020176	005304			16\$: DEC	R4	;DEC RETRY COUNT
1867	020200	001333			BNE	8\$;DO ANOTHER SEEK IF NOT 0
1868	020202	012703	007323		MOV	#HDMOVF,R3	;ELSE SET RESULT MESSAGE POINTER
1869	020206				ERRHRD	10009.,ERR1	
(4)	020206	104456			TRAP	C\$ERHRD	
(5)	020210	023431			.WORD	10009	
(5)	020212	000000			.WORD	0	

```

(5) 020214 011724 .WORD ERR1
1870 020216 005037 003032 CLR ERRSWI ;CLEAR FOR ERROR ERROR RETURN
1871 020222 000405 BR 60$
1872 020224 023737 003120 003116 17$: CMP CURCYL,NEWCYL ;HDS SHOULD MOVE OUT, CHK THEY DID
1873 020232 002747 BLT 14$ ;YES - SKIP
1874 020234 000760 BR 16$ ;ELSE GO DEC AND RETRY
1875 020236 20$:
1876 020236 60$:
1877 020236 ENDSEG
(3) 020236 10000$:
(3) 020236 104405 TRAP C$ESEG
1878 020240 162737 000002 003016 PH65$: SUB #2,SSINDX ;REMOVE ENTRY FROM SUBROUT STACK
1879 020246 012604 MOV (SP)+,R4 ;RESTORE REGISTERS
1880 020250 012600 MOV (SP)+,R0
1881 020252 012603 MOV (SP)+,R3
1882 020254 005737 003032 TST ERRSWI ;TEST IF ERROR RETURN
1883 020260 001403 BEQ 99$ ;YES - SKIP
1884 020262 063716 003032 ADD ERRSWI,(SP) ;ADD IN ERROR RETURN
1885 020266 000207 RTS PC
1886 020270 017616 000000 99$: MOV @ (SP),(SP) ;SET ERROR RETURN ADDRESS
1887 020274 000207 RTS PC
1888
1890 ; DRIVE READY TEST ROUTINE. CHECKS DRIVE IS READY. IF NOT, WAIT
1891 ; 500MS FOR READY TO SET.
1892 020276 010346 RDYCHK: MOV R3,-(SP) ;STORE REGS
1893 020300 013703 003016 MOV SSINDX,R3 ;GET SUBROUTINE INDEX
1894 020304 005723 TST (R3)+ ;BUMP IT FOR NEXT ENTRY
1895 020306 016663 000002 002420 MOV 2(SP),SUBSTK(R3) ;INSERT THIS CALL
1896 020314 162763 000004 002420 SUB #4,SUBSTK(R3) ;ADJUST IT TO CALLING LOCATION
1897 020322 010337 003016 MOV R3,SSINDX ;STORE IT BACK
1898 020326 010046 MOV R0,-(SP)
1899 020330 010146 MOV R1,-(SP)
1900 020332 010446 MOV R4,-(SP)
1901 020334 012737 000002 003032 MOV #2,ERRSWI ;SE' FOR NO ERROR RETURN
1902 020342 012701 011610 MOV #5000,R1 ;SE' WAIT COUNT
1903 020346 004737 016262 1$: JSR PC,GSTAT ;GET DRIVE STATUS
1904 020352 020506 4$
1905 020354 032737 000001 003060 BIT #DRDYMSK,T.CS ;TEST IF DRIVE READY
1906 020362 001053 BNE 5$ ;YES - EXIT
1907 020364 WAITUS #1
1908 020376 005301 DEC R1 ;DEC WAIT COUNT
1909 020400 001362 BNE 1$ ;LOOP IF NOT 0
1910 020402 012703 007760 MOV #MDRDY,R3 ;SET RESULT MESSAGE POINTER
1911 020406 012704 010645 MOV #C500MS,R4 ;SET CONDITION MESSAGE POINTER
1912 020412 ERRHRD 10010...ERR5
(4) 020412 104456 TRAP C$ERHRD
(5) 020414 023432 .WORD 10010
(5) 020416 000000 .WORD 0
(5) 020420 012156 .WORD ERR5
1913 020422 012701 000062 MOV #50,R1 ;SET WAIT COUNT FOR 5 SECONDS
1914 020426 004737 016262 2$: JSR PC,GSTAT ;GET DRIVE STATUS
1915 020432 020506 4$
1916 020434 032737 000001 003060 BIT #DRDYMSK,T.CS ;TEST IF DRIVE READY
1917 020442 001007 BNE 3$ ;YES - SKIP
1918 020444 WAITMS #1 ;WAIT FOR 100MS
1919 020456 005301 DEC R1 ;DEC WAIT COUNTER
  
```

```

1920 020460 001362          BNE      2$          ;LOOP UNTIL TIME DONE
1921 020462 032737 100000 003060 3$: BIT      #ANYERR,T.CS ;TEST IF ANYERR SET
1922 020470 001406          BEQ      4$          ;NO - SKIP
1923 020472          ERRHRD 10011,,,ERR6 ;REPORT ALL ERRORS
(4) 020472 104456          TRAP    C$ERRHRD
(5) 020474 023433          .WORD  10011
(5) 020476 000000          .WORD  0
(5) 020500 012226          .WORD  ERR6
1924 020502 005337 003254          DEC     ERRCNT      ;REDUCE ERROR COUNT FOR DUAL ERRORS
1925 020506 005037 003032          CLR     ERRSWI     ;CLEAR FOR ERROR RETURN
1926 020512 162737 000002 003016 5$: SUB     #2,SSINDX  ;REMOVE ENTRY FROM SUBROUT STACK
1927 020520 012604          MOV     (SP)+,R4   ;RESTORE REGS
1928 020522 012601          MOV     (SP)+,R1
1929 020524 012600          MOV     (SP)+,R0
1930 020526 012603          MOV     (SP)+,R3
1931 020530 005737 003032          TST     ERRSWI     ;TEST IF ERROR RETURN
1932 020534 001403          BEQ     99$        ;YES - SKIP
1933 020536 063716 003032          ADD     ERRSWI,(SP) ;ADD IN ERROR RETURN
1934 020542 000207          RTS     PC
1935 020544 017616 000000          99$: MOV     @ (SP),(SP) ;SET ERROR RETURN ADDRESS
1936 020550 000207          RTS     PC
1937
1938 ;          CHOOSE HEAD ROUTINE. PICKS HEAD 0 UNLESS SPECIFIC HEAD IS
1939 ;          SELECTED BY SOFTWARE PARAMETER.
1940 020552 005037 003126          CHOSHD: CLR     DESHD ;CLEAR TO HEAD 0
1941 020556 032737 010000 013556          BIT     #HEADLM,MISWIW ;TEST IF HEAD SPECIFIED
1942 020564 001403          BEQ     1$          ;NO - SKIP
1943 020566 013737 013564 003126          MOV     HEADW,DESHD ;INSERT SPECIFIED HEAD
1944 020574 000207          1$: RTS     PC
1945
1946 ;          SWAP HEAD ROUTINE. CHANGES SELECTED HEAD TO HEAD 1
1947 ;          UNLESS HEAD 0 SPECIFICALLY SELECTED BY SOFTWARE PARAMETER.
1948 020574 032737 010000 013556          SWAPHD: BIT     #HEADLM,MISWIW ;TEST IF HEAD SPECIFIED
1949 020604 001011          BNE     2$          ;YES - TAKE ABORT EXIT
1950 020606 005737 003126          TST     DESHD     ;TEST IF HEAD ONE USED
1951 020612 001006          BNE     2$          ;YES - TAKE ABORT EXIT
1952 020614 012737 000001 003126          MOV     #1,DESHD  ;ELSE SET FOR HEAD ONE
1953 020622 062716 000002          ADD     #2,(SP)   ;BUMP PAST ABORT RETURN
1954 020626 000207          RTS     PC        ;RETURN
1955 020630 017616 000000          2$: MOV     @ (SP),(SP) ;GET ABORT DESTINATION
1956 020634 000207          3$: RTS     PC
1957
1958 ;          SWAP OLD CYLINDER AND NEW CYLINDER ROUTINE.
1959 020636 010046          ONSWAP: MOV     R0,-(SP) ;STORE R0
1960 020640 013700 003114          MOV     OLD CYL,R0 ;MOVE OLD TO R0
1961 020644 013737 003116 003114          MOV     NEW CYL,OLD CYL ;MOVE NEW TO OLD
1962 020652 010037 003116          MOV     R0,NEW CYL ;PUT OLD IN NEW
1963 020656 012600          MOV     (SP)+,R0  ;RESTORE R0
1964 020660 000207          RTS     PC
1965
1966 ;          BAD SECTOR FILES VALID CHECK ROUTINE. CHECKS IF BAD SECTOR
1967 ;          FILES HAVE BEEN READ AND STORED. IF NOT, REPORT AND FORCE
1968 ;          FILES TO LOOK LIKE ALL SECTORS OK.
1969 ;          CKBSVD:
1970 020662 005737 003510          CKBSVD: TST     BSFVAL ;TEST IF BAD SECTORS STORED
1971 020666 001051          BNE     5$          ;YES - EXIT
1972 020670          PRINTF #FMT9,#BSNSTR ;REPORT

```

(8)	020670	012746	007550		MOV	#BSNSTR,-(SP)	
(7)	020674	012746	011212		MOV	#FMT9,-(SP)	
(6)	020700	012746	000002		MOV	#2,-(SP)	
(3)	020704	010600			MOV	SP,R0	
(4)	020706	104417			TRAP	C\$PNTF	
(4)	020710	062706	000006		ADD	#6,SP	
1973	020714				PRINTF	#FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>	
(11)	020714	005046			CLR	-(SP)	
(11)	020716	153716	003047		BISB	RLDRV+1,(SP)	
(10)	020722	012746	006152		MOV	#DRVNAM,-(SP)	
(9)	020726	013746	003042		MOV	RLBAS,-(SP)	
(8)	020732	012746	006141		MOV	#BASADD,-(SP)	
(7)	020736	012746	011026		MOV	#FMT5,-(SP)	
(6)	020742	012746	000005		MOV	#5,-(SP)	
(3)	020746	010600			MOV	SP,R0	
(4)	020750	104417			TRAP	C\$PNTF	
(4)	020752	062706	000014		ADD	#14,SP	
1974	020756				PRINTF	#FMT3	
(7)	020756	012746	011017		MOV	#FMT3,-(SP)	
(6)	020762	012746	000001		MOV	#1,-(SP)	
(3)	020766	010600			MOV	SP,R0	
(4)	020770	104417			TRAP	C\$PNTF	
(4)	020772	062706	000004		ADD	#4,SP	
1975	020776	012737	177777	003512	MOV	#-1,SBSFIL	;FORCE FILES TO NO ENTRIES
1976	021004	012737	177777	003706	MOV	#-1,FBSFIL	
1977	021012	000207			RTS	PC	
1978							
1980							
1981	021014	012737	000001	003142	: XRDHDC: MOV	#1,TEMP4	;SET FLAG TO BYPASS REG STORAGE
1982	021022	000402			BR	XRDHDC	;GO DO IT
1983	021024	005037	003142		XRDHD: CLR	TEMP4	;SET FLAG TO SAVE T. AND L. REGS
1984	021030	010346			XRDHDG: MOV	R3,-(SP)	;STORE REGISTERS
1985	021032	013703	003016		MOV	SSINDEX,R3	;GET SUBROUTINE INDEX
1986	021036	005723			TST	(R3)+	;BUMP IT FOR NEXT ENTRY
1987	021040	016663	000002	002420	MOV	2(SP),SUBSTK(R3)	;INSERT THIS CALL
1988	021046	162763	000004	002420	SUB	#4,SUBSTK(R3)	;ADJUST IT TO CALLING LOCATION
1989	021054	010337	003016		MOV	R3,SSINDEX	;STORE IT BACK
1990	021060	010046			MOV	R0,-(SP)	
1991	021062	010146			MOV	R1,-(SP)	
1992	021064	010446			MOV	R4,-(SP)	
1993	021066	012737	000002	003032	MOV	#2,ERRSWI	;SET FOR NO ERROR RETURN
1994	021074	005737	003142		TST	TEMP4	;TEST IF REGISTERS TO BE SAVED
1995	021100	001007			BNE	2\$;NO - SKIP
1996	021102	012703	003060		MOV	#L.MP+2,R3	;SET POINTER FOR REGS
1997	021106	012701	000004		MOV	#4,R1	;SET COUNT
1998	021112	014346			1\$: MOV	-(R3),-(SP)	;SAVE REGISTER
1999	021114	005301			DEC	R1	;DEC COUNT
2000	021116	001375			BNE	1\$;LOOP UNTIL ALL ARE SAVED
2001	021120	004737	020276		2\$: JSR	PC,RDYCHK	;CHECK DRIVE READY
2002	021124	021374			65\$		
2003	021126	005037	003022		CLR	DONE	;CLEAR INTERRUPT FLAG
2004	021132	012701	003050		MOV	#L.CS,R1	;GET ADDRESS OF LOAD REGS
2005	021136	013711	003046		MOV	RLDRV,(R1)	;LOAD DRIVE NUMBER
2006	021142	042711	002000		BIC	#BIT10,(R1)	;CLEAR FOR DRIVE 4 - 7 SPEC'D
2007	021146	052721	000110		BIS	#RDHEAD,(R1)+	;INSERT COMMAND
2008	021152	005021			CLR	(R1)+	;CLEAR BA

2009	021154	005021			CLR	(R1)+	:CLEAR DA	
2010	021156	014162	000004		MOV	-(R1),RLDA(R2)	:LOAD RL11 REGS	
2011	021162	014162	000002		MOV	-(R1),RLBA(R2)		
2012	021166	014162	000000		MOV	-(R1),RLCSR(R2)		
2013	021172			3\$:	WAITUS	#10.	:WAIT 1MS FOR INTERRUPT	
2014	021204	005737	003022		TST	DONE	:TEST IN INTERRUPT FLAG SET	
2015	021210	001460			BEQ	14\$:NO - SKIP	
2016	021212	032737	000001	003060	5\$:	BIT	#DRDYMSK,T.CS	:TEST IF DRIVE READY
2017	021220	001035			BNE	10\$:YES - SKIP	
2018	021222	012703	007760		MOV	#MDRDY,R3	:SET NO READY MESSAGE	
2019	021226	012704	010662		MOV	#CAFDT,R4	:CONDITION OF AFTER DATA XFER	
2020	021232				ERRHRD	10017,,,ERR5		
(4)	021232	104456			TRAP	C\$ERHRD		
(5)	021234	023441			.WORD	10017		
(5)	021236	000000			.WORD	0		
(5)	021240	012156			.WORD	ERR5		
2021	021242	012701	000062		MOV	#50.,R1	:SET WAIT COUNT FOR 5 SECONDS	
2022	021246	004737	016262		4\$:	JSR	PC,GSTAT	:GET STATUS
2023	021252	021370			60\$			
2024	021254	032737	000001	003060	BIT	#DRDYMSK,T.CS	:TEST IF DRIVE HAS COME READY	
2025	021262	001403			BEQ	11\$:NO - SKIP	
2026	021264	005037	003032		CLR	ERRSWI	:CLEAR ERROR SWITCH	
2027	021270	000411			BR	10\$:SKIP	
2028	021272	005301			11\$:	DEC	R1	:DEC WAIT COUNT
2029	021274	001364			BNE	4\$:LOOP UNTIL TIME DONE	
2030	021276	012704	010673		MOV	#CSSEC,R4	:SET CONDITION AFTER 5 SECONDS	
2031	021302				ERRHRD	10014,,,ERR5		
(4)	021302	104456			TRAP	C\$ERHRD		
(5)	021304	023436			.WORD	10014		
(5)	021306	000000			.WORD	0		
(5)	021310	012156			.WORD	ERR5		
2032	021312	000426			BR	60\$:EXIT	
2033	021314	005737	003060		10\$:	TST	T.CS	:CHECK FOR ANY ERRORS
2034	021320	100005			BPL	12\$:NO - SKIP	
2035	021322				ERRHRD	10016,,,ERR6	:REPORT ALL ERRORS	
(.)	021322	104456			TRAP	C\$ERHRD		
(5)	021324	023440			.WORD	10016		
(5)	021326	000000			.WORD	0		
(5)	021330	012226			.WORD	ERR6		
2036	021332	000416			BR	60\$		
2037	021334	012701	003070		12\$:	MOV	#HDWRD2,R1	:GET POINTER
2038	021340	016221	000006		MOV	RLMP(R2),(R1)+	:STORE LAST TWO HEADER WORDS	
2039	021344	016221	000006		MOV	RLMP(R2),(R1)+		
2040	021350	000411			BR	65\$:EXIT	
2041	021352	004737	016056		14\$:	JSR	PC,WAITIN	:WAIT FOR INTERRUPT
2042	021356	012603			MOV	(SP)+,R3	:GET RESULTS	
2043	021360				ERRHRD	10015,,,ERR1	:REPORT	
(4)	021360	104456			TRAP	C\$ERHRD		
(5)	021362	023437			.WORD	10015		
(5)	021364	000000			.WORD	0		
(5)	021366	011724			.WORD	ERR1		
2044	021370	005037	003032		60\$:	CLR	ERRSWI	:CLEAR FOR ERROR ERROR RETURN
2045	021374	005737	003142		65\$:	TST	TEMP4	:TEST IF REGISTERS WERE SAVED
2046	021400	001007			BNE	22\$:NO - SKIP	
2047	021402	012703	003050		MOV	#L.CS,R3	:SET POINTER TO RESTORE REGS	
2048	021406	012701	000004		MOV	#4,R1	:SET COUNT	


```

2049 021412 012623      20$:  MOV      (SP)+,(R3)+  ;RESTORE REGISTER
2050 021414 005301      DEC      R1           ;DEC COUNT
2051 021416 001375      BNE     20$          ;LOOP UNTIL ALL ARE RESTORED
2052 021420 162737 000002 003016 22$:  SUB      #2,SSINDX   ;REMOVE ENTRY FROM SUBROUT STACK
2053 021426 012604      MOV      (SP)+,R4    ;RESTORE REGS
2054 021430 012601      MOV      (SP)+,R1
2055 021432 012600      MOV      (SP)+,R0
2056 021434 012603      MOV      (SP)+,R3
2057 021436 005737 003032  TST     ERRSWI      ;TEST IF ERROR RETURN
2058 021442 001403      BEQ     99$         ;YES - SKIP
2059 021444 063716 003032  ADD     ERRSWI,(SP) ;ADD IN ERROR RETURN
2060 021450 000207      RTS     PC
2061 021452 017616 000000 99$:  MOV      @ (SP),(SP) ;SET ERROR RETURN ADDRESS
2062 021456 000207      RTS     PC
2063
2065 ;
2066 ;
2067 021460 010346      VERHDR: MOV     R3,-(SP) ;STORE REGS
2068 021462 013703 003016  MOV     SSINDX,R3   ;GET SUBROUTINE INDEX
2069 021466 005723      TST     (R3)+       ;BUMP IT FOR NEXT ENTRY
2070 021470 016663 000002 002420  MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
2071 021476 162763 000004 002420  SUB     #4,SUBSTK(R3) ;ADJUST IT TO CALLING LOCATION
2072 021504 010337 003016  MOV     R3,SSINDX   ;STORE IT BACK
2073 021510 010046      MOV     R0,-(SP)
2074 021512 010146      MOV     R1,-(SP)
2075 021514 010446      MOV     R4,-(SP)
2076 021516 010546      MOV     R5,-(SP)
2077 021520 012737 000002 003032  MOV     #2,ERRSWI   ;SET FOR NO ERROR RETURN
2078 021526 052737 000002 003020  BIS     #HDCMP,OPFLAG ;SET HEADER COMPARE FLAG
2079 021534 005037 003030  CLR     MORECE      ;CLEAR MORE ERRORS FLAG
2080 021540 012704 004102  MOV     #IBUFF,R4   ;SET POINTER TO HEADERS
2081 021544 012705 003132  MOV     #TEMPO,R5   ;SET POINTER TO WORK AREA
2082 021550 005003      CLR     R3          ;CLEAR FOR WORD COUNTER
2083 021552 011415      MOV     (R4),(R5)   ;MOVE HDR WORD TO WORK AREA
2084 021554 011401      MOV     (R4),R1     ;PUT WORD IN REG 1
2085 021556 012701 000177  BIC     #177,R1     ;CLEAR ALL BUT CYLINDER
2086 021562 012700 000007  MOV     #7,R0       ;SET SHIFT COUNT
2087 021566 006201 3$:   ASR     R1          ;SHIFT
2088 021570 005300      DEC     R0          ;DEC
2089 021572 001375      BNE     3$         ;LOOP
2090 021574 020137 003116  CMP     R1,NEWCYL   ;CHECK IF CYLINDER PART GOOD
2091 021600 001407      BEQ     4$         ;YES - SKIP
2092 021602  ERRHRD 10018...ERR10 ;REPORT ERROR
    (4) 021602 104456  TRAP   C$ERRHRD
    (5) 021604 023442  .WORD 10018
    (5) 021606 000000  .WORD 0
    (5) 021610 013320  .WORD ERR10
2093 021612 005037 003032  CLR     ERRSWI      ;CLEAR FOR ERROR ERROR RETURN
2094 021616 000456      BR     65$
2095 021620 012701 000050 4$:  MOV     #40,R1      ;SET HEADER COUNT
2096 021624 042715 000100  BIC     #HDHSEL,(R5) ;CLEAR HEAD SELECT AND 0 BIT
2097 021630 005737 003126  TST     DESHD       ;ARE WE USING HD 0?
2098 021634 001402      BEQ     5$         ;YES - SKIP
2099 021636 052715 000100  BIS     #HDHSEL,(R5) ;INSERT HEAD BIT
2100 021642 005065 000002 5$:  CLR     2(R5)       ;CLEAR 2ND WORD OF WORK AREA
2101 021646 021524 6$:  CMP     (R5),(R4)+  ;TEST FIRST WORD OK
    
```

```

2102 021650 001410      BEQ      6$      ;YES - SKIP
2103 021652 005744      TST      -(R4)   ;ELSE SET POINTER FOR ERROR
2104 021654      ERRHRD 10018.,,ERR10 ;REPORT
      (4) 021654 104456      TRAP     C$ERHRD
      (5) 021656 023442      .WORD   10018
      (5) 021660 000000      .WORD   0
      (5) 021662 013320      .WORD   ERR10
2105 021664 005037 003032 CLR      ERRSWI   ;CLEAR FOR ERROR RETURN
2106 021670 005724      TST      (R4)+   ;RESET POINTER
2107 021672 005203      8$: INC      R3      ;BUMP WORD COUNTER
2108 021674 005724      TST      (R4)+   ;TEST 2ND WORD IS 0
2109 021676 001410      BEQ      12$     ;YES - SKIP
2110 021700 022544      CMP      (R5)+,-(R4) ;ADJUST POINTERS FOR REPORT
2111 021702      ERRHRD 10018.,,ERR10 ;REPORT
      (4) 021702 104456      TRAP     C$ERHRD
      (5) 021704 023442      .WORD   10018
      (5) 021706 000000      .WORD   0
      (5) 021710 013320      .WORD   ERR10
2112 021712 005037 003032 CLR      ERRSWI   ;CLEAR FOR ERROR RETURN
2113 021716 024524      CMP      -(R5),(R4)+ ;RESET POINTERS
2114 021720 005724      12$: TST      (R4)+   ;BUMP PAST ECC WORD
2115 021722 005203      INC      R3      ;BUMP WORD COUNTER
2116 021724 005215      INC      (R5)    ;BUMP SECTOR OF EXPECTED HEADER
2117 021726 011500      MOV      (R5),R0 ;MOVE EXPECTED HDR TO R0
2118 021730 042700 177700 BIC      #^CHDSEC,R0 ;CLEAR ALL BUT SECTOR
2119 021734 022700 000050 CMP      #40.,R0   ;TEST IF AT SECTOR 40
2120 021740 001002      BNE      15$     ;NO - SKIP
2121 021742 042715 000077 BIC      #HDSEC,(R5) ;CLEAR SECTOR TO 0
2122 021746 005203      15$: INC      R3      ;BUMP HDR WORD COUNTER
2123 021750 005301      DEC      R1      ;DEC HEADER COUNT
2124 021752 001335      BNE      6$      ;LOOP IF NOT YET DONE
2125 021754 162737 000002 003016 65$: SUB      #2,SSINDX ;REMOVE ENTRY FROM SUBROUT STACK
2126 021762 012605      MOV      (SP)+,R5 ;RESTORE REGISTERS
2127 021764 012604      MOV      (SP)+,R4
2128 021766 012601      MOV      (SP)+,R1
2129 021770 012600      MOV      (SP)+,R0
2130 021772 012603      MOV      (SP)+,R3
2131 021774 005737 003032 TST      ERRSWI   ;TEST IF ERROR RETURN
2132 022000 001403      BEQ      99$     ;YES - SKIP
2133 022002 063716 003032 ADD      ERRSWI,(SP) ;ADD IN ERROR RETURN
2134 022006 000207      RTS      PC
2135 022010 017616 000000 99$: MOV      @ (SP),(SP) ;SET ERROR RETURN ADDRESS
2136 022014 000207      RTS      PC
2137
2139      ; POSITION HEAD BIT FROM HEADER OR MULTIPURPOSE REGISTER TO LSB.
2140 022016 013705 003066 POSHW1: MOV      HDWRD1,R5 ;START FOR POSITION HD BIT IN WD 1
2141 022022 000402      BR      POSHDO  ;SKIP
2142 022024 013705 003066 POSHSB: MOV      T.MP,R5 ;START FOR POSITION HD BIT IN MP
2143 022030 010146      POSHDO: MOV      R1,-(SP) ;STORE R1
2144 022032 042705 177677 BIC      #^CHSSTAT,R5 ;CLEAR ALL BUT HEAD SEL BIT
2145 022036 012701 000006 MOV      #6,R1    ;SET SHIFT COUNT
2146 022042 006205      1$: ASR      R5      ;SHIFT FOR RIGHT JUSTIFY
2147 022044 005301      DEC      R1
2148 022046 001375      BNE      1$
2149 022050 012601      MOV      (SP)+,R1 ;RESTORE R1
2150 022052 000207      RTS      PC      ;RETURN

```

```
2151  
2152  
2153 ; WAIT FOR READY ROUTINE. DURATION OF WAIT PASSED TO THE ROUTINE  
; FROM THE CALLING ROUTINE IN R1.  
2154 022054 010346 PDYWAIT: MOV R3,-(SP) ;STORE R3  
2155 022056 013703 003016 MOV SSINDEX,R3 ;GET SUBROUTINE INDEX  
2156 022062 005723 TST (R3)+ ;BUMP IT FOR NEXT ENTRY  
2157 022064 016663 000002 002420 MOV 2(SP),SUBSTK(R3) ;INSERT THIS CALL  
2158 022072 162763 000004 002420 SUB #4,SUBSTK(R3) ;ADJUST IT TO CALLING LOCATION  
2159 022100 010337 003016 MOV R3,SSINDEX ;STORE IT BACK  
2160 022104 010046 MOV R0,-(SP)  
2161 022106 010146 MOV R1,-(SP)  
2162 022110 010446 MOV R4,-(SP)  
2163 022112 012737 000002 003032 MOV #2,ERRSWI ;SET FOR NO ERROR RETURN  
2164 022120 004737 016262 5$: JSR PC,GSTAT ;GET DRIVE STATUS  
2165 022124 022274 10$  
2166 022126 032737 000001 003060 BIT #DRDYMSK,T.CS ;CHECK IF READY  
2167 022134 001061 BNE 9$ ;YES - SKIP  
2168 022136 005301 DEC R1 ;DEC WAIT COUNT  
2169 022140 001406 BEQ 7$ ;SKIP IF 0  
2170 022142 WAITUS #1  
2171 022154 000761 BR 5$  
2172 022156 012703 007760 7$: MOV #MDRDY,R3 ;SET NAME MESSAGE PTR  
2173 022162 ERRHRD 10020,,,ERR3 ;REPORT READY ERROR  
(4) 022162 104456 TRAP C$ERRHD  
(5) 022164 023- 4 .WORD 10020  
(5) 022166 000000 .WORD 0  
(5) 022170 012040 .WORD ERR3  
2174 022172 012701 000062 MOV #50,,R1 ;SET WAIT COUNT FOR 5 SECONDS  
2175 022176 004737 016262 6$: JSR PC,GSTAT ;GET DRIVE STATUS  
2176 022202 022274 10$  
2177 022204 032737 000001 003060 BIT #DRDYMSK,T.CS ;TEST IF DRIVE READY  
2178 022212 001016 BNE 8$ ;YES - SKIP  
2179 022214 WAITMS #1 ;WAIT 100 MS  
2180 022226 005301 DEC R1 ;DEC WAIT COUNT  
2181 022230 001362 BNE 6$ ;LOOP UNTIL TIME DONE  
2182 022232 012704 010673 MOV #C5SEC,R4 ;SET CONDITION AFTER 5 SECDS  
2183 022236 ERRHRD 10021,,,ERR5  
(4) 022236 104456 TRAP C$ERRHD  
(5) 022240 023445 .WORD 10021  
(5) 022242 000000 .WORD 0  
(5) 022244 012156 .WORD ERR5  
2184 022246 000410 BR 11$  
2185 022250 032737 100000 003060 8$: BIT #ANYERR,T.CS ;EXIT  
2186 022256 001406 BEQ 10$ ;TEST IF ANY ERROR SET  
2187 022260 ERRHRD 10022,,,ERR6 ;NO - SKIP  
(4) 022260 104456 TRAP C$ERRHD ;REPORT ALL ERRORS  
(5) 022262 023446 .WORD 10022  
(5) 022264 000000 .WORD 0  
(5) 022266 012226 .WORD ERR6  
2188 022270 005337 003254 11$: DEC ERRCNT ;DEC FOR DOUBLE ERROR REPORT  
2189 022274 005037 003032 10$: CLR ERRSWI ;CLEAR FOR ERROR ERROR RETURN  
2190 022300 162737 000002 003016 9$: SUB #2,SSINDEX ;REMOVE ENTRY FROM SUBROUT STACK  
2191 022306 012604 MOV (SP)+,R4 ;RESTORE REGISTERS  
2192 022310 012601 MOV (SP)+,R1  
2193 022312 012600 MOV (SP)+,R0  
2194 022314 012603 MOV (SP)+,R3 ;RESTORE R3
```

```

2195 022316 005737 003032      TST  ERRSWI      ;TEST IF ERROR RETURN
2196 022322 001403      BEQ  99$        ;YES - SKIP
2197 022324 063716 003032      ADD  ERRSWI,(SP);ADD IN ERROR RETURN
2198 022330 000207      RTS  PC
2199 022332 017616 000000      99$: MOV  @ (SP),(SP);SET ERROR RETURN ADDRESS
2200 022336 000207      RTS  PC

:      GET POSITION ROUTINE. READS A HEADER FROM CURRENT CYLINDER
:      (WHERE IT IS PRESENTLY POSITIONED) AND STORES CYLINDER
:      NUMBER IN CURCYL.
2205 022340 010346      GETPOS: MOV  R3,-(SP)      ;STORE REGISTERS
2206 022342 013703      MOV  SSINDX,R3      ;GET SUBROUTINE INDEX
2207 022346 005723      TST  (R3)+          ;BUMP IT FOR NEXT ENTRY
2208 022350 016663 000002 002420      MOV  2(SP),SUBSTK(R3);INSERT THIS CALL
2209 022356 162763 000004 002420      SUB  #4,SUBSTK(R3)  ;ADJUST IT TO CALLING LOCATION
2210 022364 010337 003016      MOV  R3,SSINDX      ;STORE IT BACK
2211 022370 010046      MOV  R0,-(SP)
2212 022372 010546      MOV  R5,-(SP)
2213 022374 004737 021024      JSR  PC,XRDHD      ;DO READ HEADER
2214 022400 022430      65$
2215 022402 013703 003066      MOV  HDWRD1,R3      ;GET HEADER WORD
2216 022406 012705 000007      MOV  #7,R5          ;SET SHIFT COUNT
2217 022412 006203      4$:  ASR  R3          ;SHIFT TO RIGHT JUSTIFY
2218 022414 005305      DEC  R5
2219 022416 001375      BNE  4$
2220 022420 042703 177000      BIC  #177000,R3
2221 022424 010337 003120      MOV  R3,CURCYL      ;STORE AS CURRENT CYLINDER
2222 022430 162737 000002 003016 65$: SUB  #2,SSINDX      ;REMOVE ENTRY FROM SUBROUT STACK
2223 022436 012605      MOV  (SP)+,R5      ;RESTORE REGISTERS
2224 022440 012600      MOV  (SP)+,R0
2225 022442 012603      MOV  (SP)+,R3
2226 022444 005737 003032      TST  ERRSWI      ;TEST IF ERROR RETURN
2227 022450 001403      BEQ  99$        ;YES - SKIP
2228 022452 063716 003032      ADD  ERRSWI,(SP);ADD IN ERROR RETURN
2229 022455 000207      RTS  PC
2230 022460 017616 000000      99$: MOV  @ (SP),(SP);SET ERROR RETURN ADDRESS
2231 022464 000207      RTS  PC

:      VERIFY POSITION ROUTINE. READS A HEADER (USING GETPOS) AND
:      CHECKS HEADS ARE POSITIONED AT NEW CYLINDER (CURCYL = NEWCYL).
2236 022466 010346      VERPOS: MOV  R3,-(SP)      ;STORE R3
2237 022470 013703 003016      MOV  SSINDX,R3      ;GET SUBROUTINE INDEX
2238 022474 005723      TST  (R3)+          ;BUMP IT FOR NEXT ENTRY
2239 022476 016663 000002 002420      MOV  2(SP),SUBSTK(R3);INSERT THIS CALL
2240 022504 162763 000004 002420      SUB  #4,SUBSTK(R3)  ;ADJUST IT TO CALLING LOCATION
2241 022512 010337 003016      MOV  R3,SSINDX      ;STORE IT BACK
2242
2243 022516 012737 000002 003032      MOV  #2,ERRSWI      ;SET FOR NO ERROR RETURN
2244 022524 004737 022340      JSR  PC,GETPOS      ;GET POSITION
2245 022530 022556      65$
2246 022532 023737 003116 003120      CMP  NEWCYL,CURCYL ;CHECK IF CURRENT CYL IS NEW CYL
2247 022540 001406      BEQ  1$          ;YES - SKIP
2248 022542      ERRHRD 10022,,ERR8
(4) 022542 104456      TRAP C$ERRHRD
(5) 022544 023446      .WORD 10022
(5) 022546 000000      .WORD 0
  
```

```

(5) 022550 013160          .WORD  ERR8
2249 022552 005037 003032 CLR      ERRSWI          ;CLEAR FOR ERROR ERROR RETURN
2250 022556          1$:          SUB      #2,SSINDX      ;REMOVE ENTRY FROM SUBROUT STACK
2251 022556 162737 00000? 003016 65$:      MOV      (SP)+,R3      ;RESTORE R3
2252 022564 012603          TST      ERRSWI          ;TEST IF ERROR RETURN
2253 022566 005737 003032          BEQ      99$           ;YES - SKIP
2254 022572 001403          ADD      ERRSWI,(SP)    ;ADD IN ERROR RETURN
2255 022574 063716 003032          RTS      PC
2256 022600 000207          99$:      MOV      @ (SP), (SP)    ;SET ERROR RETURN ADDRESS
2257 022602 017616 000000          RTS      PC
2258 022606 000207
2259
2261          ;          READ ALL HEADERS ROUTINE. 40 HEADERS ARE READ AND STORED
2262          ;          IN Ibuff.
2263 022610 010346          RDALHD: MOV      R3,-(SP)      ;STORE REGISTERS
2264 022612 013703 003016          MOV      SSINDX,R3      ;GET SUBROUTINE INDEX
2265 022616 005723          TST      (R3)+          ;BUMP IT FOR NEXT ENTRY
2266 022620 016663 000002 002420          MOV      2(SP),SUBSTK(R3) ;INSERT THIS CALL
2267 022626 162763 000004 002420          SUP      #4,SUBSTK(R3)  ;ADJUST IT TO CALLING LOCATION
2268 022634 010337 003016          MOV      R3,SSINDX      ;STORE IT BACK
2269 022640 010446          MOV      R0,-(SP)
2270 022642 010146          MOV      R1,-(SP)
2271 022644 010446          MOV      R4,-(SP)
2272 022646 012737 000002 003032          MOV      #2,ERRSWI      ;SET FOR NO ERROR RETURN
2273 022654 012701 000050          MOV      #40,R1         ;SET HEADER COUNT
2274 022660 052737 100000 003020          BIS      #HDR40,OPFLAG  ;SET 40 HDR OP FLAG
2275 022666 012703 004102          MOV      #IBUFF,R3      ;SET POINTER TO STORE HDRS
2276 022672 013704 003042          MOV      RLBAS,R4       ;GET BASE ADDRESS
2277 022676 062704 000006          ADD      #RLMP,R4       ;MAKE IT POINT TO MP REG
2278 022702 012737 000010 003050          MOV      #10,L.CS       ;LOAD FOR READ HEADER, NO INTERRUPT
2279 022710 053737 003046 003050          BIS      RLDRV,L.CS     ;INSERT DRIVE NUMBER
2280 022716 042737 002000 003050          BIC      #BIT10,L.CS    ;CLEAR FOR DRIVE 4 - 7 SPEC'D
2281 022724 005037 003052          CLR      L.BA           ;CLEAR BA
2282 022730 005037 003054          CLR      L.DA           ;CLEAR DA
2283 022734 005737 003126          TST      DESHD          ;TEST IF HEAD 0
2284 022740 001403          BEQ      3$            ;YES - SKIP
2285 022742 052737 000020 003054          BIS      #HDSSEL,L.DA   ;ELSE INSERT HEAD 0
2286 022750 013762 003054 000004 3$:      MOV      L.DA,RLDA(R2)   ;LOAD RLDA REG
2287 022756 013762 003052 000002          MOV      L.BA,RLBA(R2)  ;LOAD RLBA
2288 022764 032762 000200 000000          BIT      #CRDYMSK,RLCS(R2) ;TEST IF CONTROLLER READY
2289 022772 001003          BNE      6$            ;YES - SKIP
2290 022774 004737 020276          JSR      PC,RDYCHK      ;ELSE CHECK READY
2291 023000 023116          6$:      MOV      L.CS,RLCS(R2)  ;LOAD RLCS REG
2292 023002 013762 003050 000000          MOV      #77777,R0      ;SET COUNT FOR WAIT
2293 023010 012700 077777          MOV      #CRDYMSK,RLCS(R2) ;CHECK THAT OPERATION COMPLETED
2294 023014 032762 000200 000000 7$:      BIT      8$            ;YES - SKIP
2295 023022 001016          BNE      8$            ;DEC COUNT
2296 023024 005300          DEC      R0            ;DEC COUNT
2297 023026 001372          BNE      7$            ;SKIP IF NOT YET 0
2298 023030 004737 016024          JSR      PC,READRL      ;ELSE GET ALL REGISTERS
2299 023034 004737 016056          JSR      PC,WAITIN      ;ELSE WAIT FOR TIMEOUT
2300 023040 012603          MOV      (SP)+,R3       ;GET RESULT MESSAGE POINTER
2301 023042          ERRHRD 10025,,,ERR1
(4) 023042 104456          TRAP   C$ERHRD
(5) 023044 023451          .WORD 10025
(5) 023046 000000          .WORD 0
  
```

```

(5) 023050 011724 .WORD ERR1
2302 023052 005037 003032 CLR ERRSWI ;CLEAR FOR ERROR RETURN
2303 023056 000417 BR 65$
2304 023060 005737 003060 8$: TST T.CS ;TEST FOR ANY ERRORS
2305 023064 100007 BPL 12$ ;NO - SKIP
2306 023066 ERRHRD 10026.,,ERR6
(4) 023066 104456 TRAP C$ERHRD
(5) 023070 023452 .WORD 10026
(5) 023072 000000 .WORD 0
(5) 023074 012226 .WORD ERR6
2307 023076 005037 003032 CLR ERRSWI ;CLEAR FOR ERROR RETURN
2308 023102 000405 BR 65$
2309 023104 011423 12$: MOV (R4),(R3)+ ;STORE HEADER WORDS
2310 023106 011423 MOV (R4),(R3)+
2311 023110 011423 MOV (R4),(R3)+
2312 023112 005301 DEC R1 ;DEC HEADER COUNT
2313 023114 001332 BNE 6$
2314 023116 162737 000002 003016 65$: SUB #2,SSINDX ;REMOVE ENTRY FROM SUBROUT STACK
2315 023124 012604 MOV (SP)+,R4 ;RESTORE REGISTERS
2316 023126 012601 MOV (SP)+,R1
2317 023130 012600 MOV (SP)+,R0
2318 023132 012603 MOV (SP)+,R3
2319 023134 005737 003032 TST ERPSWI ;TEST IF ERROR RETURN
2320 023140 001403 BEQ 99$ ;YES - SKIP
2321 023142 063716 003032 ADD ERRSWI,(SP) ;ADD IN ERROR RETURN
2322 023146 000207 RTS PC
2323 023150 017616 000000 99$: MOV @ (SP),(SP) ;SET ERROR RETURN ADDRESS
2324 023154 000207 RTS PC
2325
2326
2328 :
2329 : GENERATE DATA ROUTINE. PATTERN TO BE GENERATED IS GIVEN
2330 : IN THE WORD FOLLOWING THE CALL. 128 WORDS ARE GENERATED
2331 : IN OBUFF.
2331 023156 010146 DATGEN: MOV R1,-(SP) ;STORE REGISTERS
2332 023160 010346 MOV R3,-(SP)
2333 023162 010446 MOV R4,-(SP)
2334 023164 012701 004502 MOV #OBUFF,R1 ;SET POINTER TO OBUFF
2335 023170 012504 MOV (R5)+,R4 ;GET DATA PATTERN SELECTOR
2336 023172 006304 ASL R4 ;ADJUST IT FOR INDEXING
2337 023174 016403 002374 MOV PATTBL(R4),R3 ;GET ADDRESS OF PATTERN
2338 023200 011321 MOV (R3),(R1)+ ;MOVE FIRST PATTERN WORD
2339 023202 001421 BEQ 5$ ;SKIP IF PATTERN IS 0
2340 023204 021327 177777 CMP (R3),#-1 ;CHECK IF PATTERN IS ALL 1'S
2341 023210 001416 BEQ 5$ ;YES - SKIP
2342 023212 020427 000010 CMP R4,#8. ;TEST IF PATTERN 5
2343 023216 001403 BEQ 3$ ;YES - SKIP
2344 023220 020427 000020 CMP R4,#16. ;CHECK IF PATTERN 9 OR 10
2345 023224 002413 BLT 6$ ;NO - SKIP
2346 023226 005723 3$: TST (R3)+ ;BUMP SOURCE POINTER
2347 023230 012321 MOV (R3)+,(R1)+ ;MOVE TWO MORE WORDS FORM SOURCE
2348 023232 012321 MOV (R3)+,(R1)+
2349 023234 012704 000015 MOV #13,R4 ;SET COUNT
2350 023240 012703 004502 MOV #OBUFF,R3 ;RESET POINTER
2351 023244 000406 BR 8$
2352 023246 012703 004502 5$: MOV #OBUFF,R3 ;ELSE SET OBUFF AS PATTERN SOURCE
2353 023252 000401 BR 7$ ;GO TO FILL
  
```

```

2354 023254 005723      6$: TST      (R3)+      ;RUMP SOURCE POINTER
2355 023256 012704 000017 7$: MOV      #15.,R4      ;SET MOVE COUNT
2356 023262 012321      8$: MOV      (R3)+,(R1)+  ;MOVE 15 WORDS INTO BUFFER
2357 023264 005304      DEC      R4
2358 023266 001375      BNE      8$
2359 023270 012703 004502      MOV      #OBUFF,R3      ;SET SOURCE TO TOP OF OBUFF
2360 023274 012704 000160      MOV      #112.,R4      ;SET COUNT FOR REST OF BUFFER
2361 023300 012321      10$: MOV     (R3)+,(R1)+  ;REPEAT PATTERN IN BUFFER
2362 023302 005304      DEC      R4
2363 023304 001375      BNE      10$
2364 023306 012604      MOV      (SP)+,R4      ;RESTORE REGISTERS
2365 023310 012603      MOV      (SP)+,R3
2366 023312 012601      MOV      (SP)+,R1
2367 023314 000205      RTS      R5      ;RETURN
2368
2369 ;
2370 ; DATA COMPARE ROUTINE. COMPARES THE CONTENTS OF Ibuff AND OBUFF.
2371 ; ERROR REPORTING IS LIMITED BY SOFTWARE PARAMETER.
2371 023316 010346      DATCOM: MOV    R3,-(SP)      ;STORE R3
2372 023320 013703 003016      MOV    SSINDX,R3      ;GET SUBROUTINE STACK INDEX
2373 023324 005723      TST    (R3)+      ;BUMP INDEX TO NEXT ENTRY
2374 023326 016663 000002 002420      MOV    2(SP),SUBSTK(R3) ;INSERT THIS CALL
2375 023334 162763 000004 002420      SUB    #4,SUBSTK(R3)  ;ADJUST IT TO CALLING LOCATION
2376 023342 010337 003016      MOV    R3,SSINDX      ;STORE IT BACK
2377 023346 010146      MOV    R1,-(SP)      ;STORE OTHER REGISTERS
2378 023350 010446      MOV    R4,-(SP)
2379 023352 010546      MOV    R5,-(SP)
2380 023354 052737 000001 003020      BIS    #DATACMP,OPFLAG ;SET DATA COMPARE FLAG
2381 023362 005037 003030      CLR    MORECE        ;CLEAR MORE ERROR FLAG
2382 023366 012705 004502      MOV    #OBUFF,R5      ;SET POINTERS TO DATA FOR COMPARE
2383 023372 012704 004102      MOV    #IBUFF,R4
2384 023376 012703 000001      MOV    #1,R3          ;SET WORD COUNTER
2385 023402 012701 000200      MOV    #128.,R1      ;SET COMPARE COUNT
2386 023406 022425      5$: CMP    (R4)+,(R5)+  ;COMPARE DATA
2387 023410 001052      BNE    10$          ;ERROR - SKIP TO REPORT
2388 023412 005203      7$: INC    R3          ;BUMP WORD COUNT
2389 023414 005301      DEC    R1          ;DEC COMPARE COUNT
2390 023416 001373      BNE    5$          ;LOOP IF NOT 0
2391 023420 042737 000001 003020      9$: BIC    #DATACMP,OPFLAG ;CLEAR DATA COMPARE FLAG
2392 023426 005737 003032      TST    ERRSWI        ;TEST IF ANY COMPARE ERRORS
2393 023432 001021      BNE    15$        ;NO - SKIP
2394 023434 012701 000200      MOV    #128.,R1      ;SET REPORT VALUE
2395 023440      PRINTB #FMT27,#TCERR,MORECE,#RESE6,R1
(11) 023440 010146      MOV    R1,-(SP)
(10) 023442 012746 010577      MOV    #RESE6,-(SP)
(9) 023446 013746 003030      MOV    MORECE,-(SP)
(8) 023452 012746 007624      MOV    #TCERR,-(SP)
(7) 023456 012746 011573      MOV    #FMT27,-(SP)
(6) 023462 012746 000005      MOV    #5,-(SP)
(3) 023466 010600      MOV    SP,R0
(4) 023470 104414      TRAP  CSPNTB
(4) 023472 062706 000014      ADD    #14,SP
2396 023476 162737 000002 003016      15$: SUB    #2,SSINDX      ;REMOVE ENTRY FROM SUBROUT STACK
2397 023504 012605      MOV    (SP)+,R5      ;RESTORE REGS
2398 023506 012604      MOV    (SP)+,R4
2399 023510 012601      MOV    (SP)+,R1
2400 023512 012603      MOV    (SP)+,R3
  
```

2401	023514	005737	003032		TST	ERRSWI		:TEST IF ERROR RETURN
2402	023520	001403			BEQ	99\$:YES - SKIP
2403	023522	063716	003032		ADD	ERRSWI,(SP)		:ADD IN ERROR RETURN
2404	023526	000207			RTS	PC		
2405	023530	017616	000000	99\$:	MOV	@(SP),(SP)		:SET ERROR RETURN ADDRESS
2406	023534	000207			RTS	PC		
2407	023536	023737	003030	013570 10\$:	CMP	MORECE,DCLIMW		:TEST IF COMPARE ERRORS LIMIT EXCEEDED
2408	023544	002011			BGE	13\$:YES - SKIP
2409	023546	024445			CMP	-(R4),-(R5)		:SET PTRS BACK TO ERROR WORDS
2410	023550				ERRHRD	10035,,ERR10		:REPORT ERROR
(4)	023550	104456			TRAP	L\$ERRHRD		
(5)	023552	023463			.WORD	10035		
(5)	023554	000000			.WORD	0		
(5)	023556	013320			.WORD	ERR10		
2411	023560	005037	003032		CLR	ERRSWI		:CLEAR ERROR SWITCH
2412	023564	022425			CMP	(R4)+,(R5)+		:BUMP PTRS PAST ERROR WORDS
2413	023566	000711			BR	7\$:DO NEXT COMPARE
2414	023570	005237	003030	13\$:	INC	MORECE		:BUMP ERROR COUNTER
2415	023574	000706			BR	7\$:DO NEXT COMPARE


```

2417
2418
2419
2420 023576 012737 177777 003134 XWRITT: MOV # -1,TEMP1 ;SET SPECIAL WRITE FOR TIMING FLAG
2421 023604 000402 BP XWRIT1
2422 023606 005037 003134 XWRITE: CLR TEMP1 ;CLEAR SPECIAL WRITE FLAG
2423 023612 012737 000112 003150 XWRIT1: MOV #WTDATA,TEMP7 ;SET FOR WRITE
2424 023620 023737 002316 003120 CMP HLMTW,CURCYL ;TEST IF CYLINDER 255 (BAD SEC)
2425 023626 001006 BNE 1$ ;NO - SKIP
2426 023630 005737 003126 TST DESHD ;TEST IF HEAD 1 (BAD SECTOR FILES)
2427 023634 001403 BEQ 1$ ;NO - SKIP
2428 023636 052737 004000 003020 1$: BIS #BADADD,OPFLAG ;SET BAD ADDRESS FLAG
2429 023644 000403 BR XREADG ;SKIP TO EXECUTE
2430 023646 012737 000114 003150 XREAD: MOV #RDDATA,TEMP7 ;SET FOR READ
2431 023654 010346 XREADG: MOV R3,-(SP) ;STORE R3
2432 023656 013703 MOV SSINDX,R3 ;SET SUBROUTINE INDEX
2433 023662 005723 TST (R3)+ ;BUMP TO NEXT STACK ENTRY
2434 023664 016663 000002 002420 MOV 2(SP),SUBSTK(R3) ;INSERT THIS CALL
2435 023672 162763 000004 002420 SUB #4,SUBSTK(R3) ;ADJUST TO POINT TO CALL
2436 023700 010337 003016 MOV R3,SSINDX ;STORE IT BACK
2437 023704 010046 MOV R0,-(SP)
2438 023706 010146 MOV R1,-(SP) ;STORE OTHER REGISTERS
2439 023710 010446 MOV R4,-(SP)
2440 023712 004737 020276 JSR PC,RDYCHK ;CHECK IF DRIVE READY
2441 023716 024304 65$
2442 023720 012703 003050 MOV #L,CS,R3 ;GET ADDRESS OF LOAD REGS
2443 023724 013713 003150 MOV TEMP7,(R3) ;SET COMMAND
2444 023730 053713 003046 BIS RLDRV,(R3) ;INSERT DRIVE NUMBER
2445 023734 042713 002000 BIC #BIT10,(R3) ;CLEAR FOR DRIVE 4 - 7 SPEC'D
2446 023740 032723 000004 BIT #BIT2,(R3)+ ;TEST IF WRITE DATA
2447 023744 001403 BEQ 3$ ;YES - SKIP
2448 023746 012723 004102 MOV #IBUFF,(R3)+ ;ELSE SET BA FOR READ
2449 023752 000402 BR 4$
2450 023754 012723 004502 3$: MOV #OBUFF,(R3)+ ;SET BA FOR WRITE
2451 023760 013713 003120 4$: MOV CURCYL,(R3) ;GET CURRENT CYLINDER
2452 023764 012704 000007 5$: MOV #7,R4 ;ALIGN IT IN DA
2453 023770 006313 ASL (R3)
2454 023772 005304 DEC R4
2455 023774 001375 BNE 5$
2456 023776 005737 003126 TST DESHD ;TEST IF HEAD 0
2457 024002 001402 BEQ 7$ ;YES - SKIP
2458 024004 052713 000100 BIS #HSMASK,(R3) ;SET FOR HEAD 1
2459 024010 053723 003130 7$: BIS DESSEC,(R3)+ ;INSERT DESIRED SECTOR
2460 024014 012713 177600 MOV #177600,(R3) ;INSERT WORD COUNT
2461 024020 005737 003134 TST TEMP1 ;CHECK IF SPECIAL WRITE FOR TIMING
2462 024024 001402 BEQ 8$ ;NO - SKIP
2463 024026 012713 177777 MOV #177777,(R3) ;ELSE SET FOR 1 WORD TRANSFER
2464 024032 032737 004000 003020 8$: BIT #BADADD,OPFLAG ;TEST IF BAD ADDRESS FLAG SET
2465 024040 001414 BEQ 2$ ;NO - SKIP
2466 024042 042737 173777 003020 BIC #CBADADD,OPFLAG ;CLEAR ALL BUT THIS FLAG
2467 024050 012703 010501 MOV #MRTAB,R3 ;SET RESULT MESSAGE POINTER
2468 024054 ERRHRD 10032,,,ERR1
(4) 024054 104456 TRAP C$ERRHD
(5) 024056 023460 .WORD 10032
(5) 024060 000000 .WORD 0
(5) 024062 011724 .WORD ERR1
  
```

2469	024064	005037	003020		CLR	OPFLAG		:CLEAR ALL FLAGS
2470	024070	000503			BR	64\$		
2471	024072	005037	003022	2\$:	CLR	DONE		:CLEAR INTERRUPT FLAG
2472	024076	005737	003134		TST	TEMP1		:CHECK IF SPECIAL WRITE FLAG SET
2473	024102	001100			BNE	65\$:YES - DO NOT START WRITE
2474	024104	011362	000006		MOV	(R3),RLMP(R2)		:LOAD RL REGS
2475	024110	014362	000004		MOV	-(R3),RLDA(R2)		
2476	024114	014362	000002		MOV	-(R3),RLBA(R2)		
2477	024120	014362	000000		MOV	-(R3),RLCS(R2)		
2478	024124			10\$:	WAITUS	#3000.		:WAIT 300MS FOR INTERRUPT
2479	024136	005737	003022		TST	DONE		:CHECK IF INTERRUPT
2480	024142	001010			BNE	14\$:YES - SKIP
2481	024144	004737	016056		JSR	PC,WAITIN		:WAIT FOR INTERRUPT
2482	024150	012603			MOV	(SP)+,R3		:GET RESULT MESSAGE
2483	024152				ERRHRD	10030,,,ERR1		
(4)	024152	104456			TRAP	C\$ERHRD		
(5)	024154	023456			.WORD	10030		
(5)	024156	000000			.WORD	0		
(5)	024160	011724			.WORD	ERR1		
2484	024162	000446			BR	64\$		
2485	024164	032737	000001	003060	14\$:	BIT	#DRDYMSK,T.CS	:TEST IF DRIVE READY
2486	024172	001033			BNE	20\$:YES - SKIP
2487	024174	012703	007760		MOV	#MDRDY,R3		:SET RESULT MESSAGE
2488	024200	012704	010662		MOV	#CAFDT,R4		:CONDITION AFTER DATA XFER
2489	024204				ERRHRD	10032,,,ERR5		
(4)	024204	104456			TRAP	C\$ERHRD		
(5)	024206	023460			.WORD	10032		
(5)	024210	000000			.WORD	0		
(5)	024212	012156			.WORD	ERR5		
2490	024214	012701	000062		MOV	#50,,R1		:SET WAIT COUNT FOR 5 SECS
2491	024220	004737	016262	17\$:	JSR	PC,GSTAT		:GET DRIVE STATUS
2492	024224	024300			64\$			
2493	024226	032737	000001	003060	17\$:	BIT	#DRDYMSK,T.CS	:TEST IF DRIVE READY NOW
2494	024234	001012			BNE	20\$:YES - SKIP
2495	024236	005301			DEC	R1		:DEC WAIT COUNT
2496	024240	001367			BNE	17\$:LOOP IF NOT TIME DONE
2497	024242	012704	010673		MOV	#C5SEC,R4		:SET CONDITION 5 SECONDS
2498	024246				ERRHRD	10033,,,ERR5		
(4)	024246	104456			TRAP	C\$ERHRD		
(5)	024250	023461			.WORD	10033		
(5)	024252	000000			.WORD	0		
(5)	024254	012156			.WORD	ERR5		
2499	024256	005037	003032		CLR	ERRSWI		:CLEAR ERROR SWITCH
2500	024262	005737	003060	20\$:	TST	T.CS		:CHECK IF ANY ERROR
2501	024266	100006			BPL	65\$:NO - SKIP
2502	024270				ERRHRD	10031,,,ERR6		
(4)	024270	104456			TRAP	C\$ERHRD		
(5)	024272	023457			.WORD	10031		
(5)	024274	000000			.WORD	0		
(5)	024276	012226			.WORD	ERR6		
2503	024300	005037	003032	64\$:	CLR	ERRSWI		:CLEAR ERROR SWITCH
2504	024304	162737	000002	003016	65\$:	SUB	#2,SSINDX	:REMOVE ENTRY FROM SUBROUT STACK
2505	024312	012604			MOV	(SP)+,R4		:RESTORE REGISTERS
2506	024314	012601			MOV	(SP)+,R1		
2507	024316	012600			MOV	(SP)+,R0		
2508	024320	012603			MOV	(SP)+,R3		

```

2509 024322 005737 003032      TST      ERRSWI      ;TEST IF ERROR RETURN
2510 024326 001403              BEQ      99$         ;YES - SKIP
2511 024330 063716 003032      ADD      ERRSWI,(SP) ;FALSE ADD IN ERROR RETURN
2512 024334 000207              RTS      PC
2513 024336 017616 000000      99$:    MOV      @ (SP),(SP) ;ADJUST FOR ERROR RETURN
2514 024342 000207              RTS      PC
2515
2516 ;      BAD SECTOR CHECK ROUTINE. CHECKS IF SECTOR SPECIFIED IN CURCYL,
2517 ;      DESHD, AND DESSEC IS LISTED AS BAD IN THE BAD SECTOR FILFS.
2518 024344 010046      BSCHK:  MOV      R0,-(SP) ;STORE REGISTERS
2519 024346 010146              MOV      R1,-(SP)
2520 024350 010346              MOV      R3,-(SP)
2521 024352 005037 003034      CLR      BSFLAG      ;CLEAR FLAG
2522 024356 012703 003706      MOV      #FBSFIL,R3 ;GET POINTER TO FACTORY FILE
2523 024362 022713 177777      CMP      #-1,(R3)    ;CHECK IF ALL ONES
2524 024366 001005              BNE      4$         ;NO SKIP TO TEST
2525 024370 012703 003512      2$:    MOV      #SBSFIL,R3 ;ELSE SET POINTER TO SOFTWARE FILE
2526 024374 022713 177777      CMP      #-1,(R3)    ;CHECK IF ALL ONES
2527 024400 001431              BEQ      20$        ;YES - EXIT
2528 024402 013700 003116      4$:    MOV      NEWCYL,R0 ;BUILD HEADER OF ADDRESS IN QUESTION
2529 024406 012701 000007      MOV      #7,R1      ;POSITION CYLINDER
2530 024412 006300      5$:    ASL      R0
2531 024414 005301              DEC      R1
2532 024416 001375              BNE      5$
2533 024420 005737 003126      TST      DESHD      ;CHECK IF HEAD 0
2534 024424 001402              BEQ      7$         ;YES - SKIP
2535 024426 052700 000100      BIS      #BIT6,R0    ;INSERT HEAD 1
2536 024432 053700 003130      7$:    BIS      DESSEC,R0  ;INSERT SECTOR
2537 024436 022300      8$:    CMP      (R3)+,R0    ;CHECK THIS WORD IN FILE
2538 024440 001402              BEQ      12$        ;YES - EXIT,ERROR
2539 024442 101005              BHI      15$        ;EXIT- NO ERROR
2540 024444 000774              BR       8$
2541 024446 012737 000001 003034 12$:  MOV      #1,BSFLAG  ;SET ERROR FLAG
2542 024454 000403              BR       20$        ;GO TO EXIT
2543 024456 020327 003706      15$:  CMP      R3,#FBSFIL ;DONE BOTH FILES?
2544 024462 003342              BGT      2$         ;NO GO DO SOFTWARE FILE
2545 024464 012603      20$:  MOV      (SP)+,R3    ;ELSE RESTORE REGISTERS
2546 024466 012601              MOV      (SP)+,R1
2547 024470 012600              MOV      (SP)+,R0
2548 024472 005737 003034      TST      BSFLAG      ;CHECK IF ERROR
2549 024476 001003              BNE      99$        ;YES - SKIP
2550 024500 062716 000002      ADD      #2,(SP)    ;ELSE BUMP ERROR RETURN
2551 024504 000207              RTS      PC
2552 024506 017616 000000      99$:  MOV      @ (SP),(SP) ;SET FOR ERROR RETURN
2553 024512 000207              RTS      PC
2554
2555 ;      REPORT OPERATION ROUTINE. PRINTS SUBROUTINE TRACE SEQUENCE AND
2556 ;      OPERATION BEING PERFORMED PORTION OF ALL
2557 ;      ERROR MESSAGES.
2558 ;
2559 024514 010446      RPTOP: MOV      R4,-(SP)
2560 024516 005737 003016      TST      SSINDX      ;TEST SUBROUTINE INDEX 0
2561 024522 001433              BEQ      1$         ;SKIP IF 0
2562 024524 012704 000002      MOV      #2,R4      ;SET INDEXER TO FIRST ENTRY.
2563 024530              PRINTB  #FMT9,#SEQMES ;PRINT 'SUBROUTINE CALL SEQ'
(8) 024530 012746 007514      MOV      #SEQMES,-(SP)
(7) 024534 012746 011212      MOV      #FMT9,-(SP)

```

(6)	024540	012746	000002		MOV	#2,-(SP)		
(3)	024544	010600			MOV	SP,R0		
(4)	024546	104414			TRAP	(SPNTB		
(4)	024550	062706	000006		ADD	#6,SP		
2564	024554			3\$:	PRINTB	#FMT16,SUBSTK(R4)	;PRINT CALLING LOCATION	
(8)	024554	016446	002420		MOV	SUBSTK(R4),-(SP)		
(7)	024560	012746	011365		MOV	#FMT16,-(SP)		
(6)	024564	012746	000002		MOV	#2,-(SP)		
(3)	024570	010600			MOV	SP,R0		
(4)	024572	104414			TRAP	(SPNTB		
(4)	024574	062706	000006		ADD	#6,SP		
2565	024600	062704	000002		ADD	#2,R4	;BUMP INDEX	
2566	024604	020437	003016		CMF	R4,SSINDX	;CHECK IF ALL PRINTED	
2567	024610	003761			BLE	3\$;LOOP IF NOT ALL PRINTED YET	
2568	024612			1\$:	PRINTB	#FMT4,ERHEAD,#TSTLAB	;PRINT ERROR HEADER	
(9)	024612	012746	006501		MOV	#TSTLAB,-(SP)		
(8)	024616	013746	003026		MOV	ERHEAD,-(SP)		
(7)	024622	012746	011015		MOV	#FMT4,-(SP)		
(6)	024626	012746	000003		MOV	#3,-(SP)		
(3)	024632	010600			MOV	SP,R0		
(4)	024634	104414			TRAP	(SPNTB		
(4)	024636	062706	000010		ADD	#10,SP		
2569	024642	042737	030000	003020	BIC	#SEEKOP!RORWOP,OPFLAG	;CLEAR SK & RD OR WRT FLAG	
2570	024650	013701	003050		MOV	L,CS,R1	;GET COMMAND EXECUTED	
2571	024654	042701	177741		BIC	#177741,R1	;STRIP ALL BUT FUNCTION CODE	
2572	024660	022701	000006		CMF	#6,R1	;TEST IF SEEK OPERATION	
2573	024664	001003			BNE	2\$;NO - SKIP	
2574	024666	052737	010000	003020	BIS	#SEEKOP,OPFLAG	;ELSE SET SEEK FLAG	
2575	024674	022701	000012	2\$:	CMF	#12,R1	;TEST IF WRITE	
2576	024700	001003			BNE	20\$;NO - SKIP	
2577	024702	052737	020000	003020	BIS	#RORWOP,OPFLAG	;SET RD OR WRT FLAG	
2578	024710	022701	000014	20\$:	CMF	#14,R1	;TEST IF READ	
2579	024714	001003			BNE	22\$;NO - SKIP	
2580	024716	052737	020000	003020	BIS	#RORWOP,OPFLAG	;SET RD OR WRT FLAG	
2581	024724			22\$:	PRINTB	#FMT1,#MOPER,OPMSGs(R1)	;PRINT OPERATION	
(9)	024724	016146	002240		MOV	OPMSGs(R1),-(SP)		
(8)	024730	012746	005527		MOV	#MOPER,-(SP)		
(7)	024734	012746	010773		MOV	#FMT1,-(SP)		
(6)	024740	012746	000003		MOV	#3,-(SP)		
(3)	024744	010600			MOV	SP,R0		
(4)	024746	104414			TRAP	(SPNTB		
(4)	024750	062706	000010		ADD	#10,SP		
2582	024754	020127	000004		CMF	R1,#4	;CHECK IF GET STATUS	
2583	024760	001007			BNE	4\$;NO - SKIP	
2584	024762	032737	000010	003054	BIT	#DRSET,L,DA	;TEST IF RESET INCLUDED	
2585	024770	001403			BEQ	4\$;NO - SKIP	
2586	024772	012701	000016		MOV	#16,R1	;SET TO PRINT WITH RESET	
2587	024776	000436			BR	9\$		
2588	025000	032737	007777	003020	4\$:	BIT	#COMPOP,OPFLAG	;TEST IF ANY OTHER OPERATION
2589	025006	001424			BEQ	8\$;NO - SKIP	
2590	025010	013704	003020		MOV	OPFLAG,R4	;SET UP TO DETERMINE WHICH ONE	
2591	025014	012701	000020		MOV	#20,R1	;PRESET THE POINTER	
2592	025020	032704	000001	5\$:	BIT	#BIT00,R4	;CHECK THE BIT	
2593	025024	001003			BNE	6\$;IF SET - SKIP	
2594	025026	005721			TST	(R1)+	;BUMP POINTER	
2595	025030	006204			ASR	R4		

```
2596 025032 000772          BR          5$
2597 025034          6$: PRINTB #FMT2,OPMSG5(R1)
      (8) 025034 016146 002240      MOV        OPMSG5(R1),-(SP)
      (7) 025040 012746 011007      MOV        #FMT2,-(SP)
      (6) 025044 012746 000002      MOV        #2,-(SP)
      (3) 025050 010600          MOV        SP,RO
      (4) 025052 104414          TRAP      C$PNTB
      (4) 025054 062706 000006      ADD        #6,SP
2598 025060 032737 100000 003020 8$: BIT        #HDR40,OPFLAG      ;TEST IF 40 HEADER OPERATION
2599 025066 001415          BEQ        10$              ;NO - SKIP
2600 025070 012701 000050          MOV        #50,R1          ;ELSE PRINT IT
2601 025074          9$: PRINTB #FMT2,OPMSG5(R1)
      (8) 025074 016146 002240      MOV        OPMSG5(R1),-(SP)
      (7) 025100 012746 011007      MOV        #FMT2,-(SP)
      (6) 025104 012746 000002      MOV        #2,-(SP)
      (3) 025110 010600          MOV        SP,RO
      (4) 025112 104414          TRAP      C$PNTB
      (4) 025114 062706 000006      ADD        #6,SP
2602 025120 000434          BR          15$
2603 025122 032737 010000 003020 10$: BIT        #SEEKOP,C$FLAG      ;TEST IF SEEK
2604 025130 001430          BEQ        15$              ;NO - SKIP
2605 025132          PRINTB #FMT13,#FRMWD,OLDCYL,#DIFWD,DESDIF,#SGNWD,DESSGN,#HDWD,DESHD
      (15) 025132 013746 003126      MOV        DESHD,-(SP)
      (14) 025136 012746 007455      MOV        #HDWD,-(SP)
      (13) 025142 013746 003124      MOV        DESSGN,-(SP)
      (12) 025146 012746 007450      MOV        #SGNWD,-(SP)
      (11) 025152 013746 003122      MOV        DESDIF,-(SP)
      (10) 025156 012746 007442      MOV        #DIFWD,-(SP)
      (9) 025162 013746 003114      MOV        OLDCYL,-(SP)
      (8) 025166 012746 007473      MOV        #FRMWD,-(SP)
      (7) 025172 012746 011233      MOV        #FMT13,-(SP)
      (6) 025176 012746 000011      MOV        #11,-(SP)
      (3) 025202 010600          MOV        SP,RO
      (4) 025204 104414          TRAP      C$PNTB
      (4) 025206 062706 000024      ADD        #24,SP
2606 025212 032737 020000 003020 15$: BIT        #RORWOP,OPFLAG      ;TEST IF READ OR WRITE SET
2607 025220 001424          BEQ        17$              ;NO - SKIP
2608 025222          PRINTB #FMT22,#CYLWD,CURCYL,#HDWD,DESHD,#SECWD,DESSEC
      (13) 025222 013746 003130      MOV        DESSEC,-(SP)
      (12) 025226 012746 007461      MOV        #SECWD,-(SP)
      (11) 025232 013746 003126      MOV        DESHD,-(SP)
      (10) 025236 012746 007455      MOV        #HDWD,-(SP)
      (9) 025242 013746 003120      MOV        CURCYL,-(SP)
      (8) 025246 012746 007466      MOV        #CYLWD,-(SP)
      (7) 025252 012746 011562      MOV        #FMT22,-(SP)
      (6) 025256 012746 000007      MOV        #7,-(SP)
      (3) 025262 010600          MOV        SP,RO
      (4) 025264 104414          TRAP      C$PNTB
      (4) 025266 062706 000020      ADD        #20,SP
2609 025272 004737 025744          17$: JSR      PC,CLPPARM      ;CLEAR PARAM TABLE
2610 025276 012604          MOV        (SP)+,R4        ;RESTORE R4
2611 025300 000207          RTS        PC
2612
2613          ; REPORT REASON ROUTINE
2614          ; PRINTS REASON PORTION FOR ALL ERROR REPORTS.
2615 025302 010146          RPTRES: MOV      R1,-(SP)      ;STORE R1
```

2616	025304	010346		MOV	R3,-(SP)	:STORE R3
2617	025306	010446		MOV	R4,-(SP)	:STORE R4
2618	025310	012701	003076	MOV	#RESPARM,R1	:GET START OF PARAM
2619	025314	012103		MOV	(R1)+,R3	:GET NUMBER OF PARAM
2620	025316			PRINTB	#FMT1.1,#MRSLT,(R1)	:PRINT NAME
(9)	025316	011146		MOV	(R1)-,(SP)	
(8)	025320	012746	005536	MOV	#MRSLT,-(SP)	
(7)	025324	012746	011000	MOV	#FMT1.1,-(SP)	
(6)	025330	012746	000003	MOV	#3,-(SP)	
(3)	025334	010600		MOV	SP,R0	
(4)	025336	104414		TRAP	C\$PNTB	
(4)	025340	062706	000010	ADD	#10,SP	
2621	025344	021127	010352	COMP	(R1),#MNRST	:TEST IF MESSAGE IS NO DRV STATUS
2622	025350	001453		BEQ	6\$:YES - SKIP REST OF REPORT
2623	025352	012704	011217	MOV	#FMT11,R4	:PRISET FOR FORMAT 11
2624	025356	022127	010345	COMP	(R1)+,#MCYLOC	:CHECK IF REPORTING CYLINDER LOC
2625	025362	001002		BNE	3\$:NO - SKIP
2626	025364	012704	011225	MOV	#FMT12,R4	:ELSE CHANGE TO FORMAT 12
2627	025370	005303		DEC	R3	:DEC PARAM COUNT
2628	025372	001442		BEQ	6\$:IF 0 - EXIT
2629	025374			PRINTB	R4,#RESE3,(R1)+	:REPORT IS VALUE
(9)	025374	012146		MOV	(R1)+,-(SP)	
(8)	025376	012746	010561	MOV	#RESE3,-(SP)	
(7)	025402	010446		MOV	R4,-(SP)	
(6)	025404	012746	000003	MOV	#3,-(SP)	
(3)	025410	010600		MOV	SP,R0	
(4)	025412	104414		TRAP	C\$PNTB	
(4)	025414	062706	000010	ADD	#10,SP	
2630	025420			PRINTB	R4,#RESE4,(R1)+	:REPORT SB VALUE
(9)	025420	012146		MOV	(R1)+,-(SP)	
(8)	025422	012746	010565	MOV	#RESE4,-(SP)	
(7)	025426	010446		MOV	R4,-(SP)	
(6)	025430	012746	000003	MOV	#3,-(SP)	
(3)	025434	010600		MOV	SP,R0	
(4)	025436	104414		TRAP	C\$PNTB	
(4)	025440	062706	000010	ADD	#10,SP	
2631	025444	162703	000002	SUB	#2,R3	:DEC PARAM COUNT
2632	025450	001413		BEQ	6\$:IF 0 - EXIT
2633	025452			PRINTB	#FMT1,#RESE5,(R1)+	:REPORT CONDITION
(9)	025452	012146		MOV	(R1)+,-(SP)	
(8)	025454	012746	010572	MOV	#RESE5,-(SP)	
(7)	025460	012746	010773	MOV	#FMT1,-(SP)	
(6)	025464	012746	000003	MOV	#3,-(SP)	
(3)	025470	010600		MOV	SP,R0	
(4)	025472	104414		TRAP	C\$PNTB	
(4)	025474	062706	000010	ADD	#10,SP	
2634	025500	012604		MOV	(SP)+,R4	:RESTORE REGS
2635	025502	012603		MOV	(SP)+,R3	
2636	025504	012601		MOV	(SP)+,R1	
2637	025506	000207		RTS	PC	:RETURN
2638						
2639				:	REPORT PHYSICAL ADDRESS OF DEVICE UNDER TEST	
2640				:	AND ALL REGISTER CONTENTS.	
2641	025510			RPTREM:	PRINTB #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>	
(11)	025510	005046		CLR	-(SP)	
(11)	025512	153716	003047	BISB	RLDRV+1,(SP)	

```
(10) 025516 012746 006152      MOV    #DRVNAM,-(SP)
(9)  025522 013746 003042      MOV    RLBAS,-(SP)
(8)  025526 012746 006141      MOV    #BASADD,-(SP)
(7)  025532 012746 011026      MOV    #FMT5,-(SP)
(6)  025536 012746 000005      MOV    #5,-(SP)
(3)  025542 010600                MOV    SP,R0
(4)  025544 104414                TRAP   C$PNTB
(4)  025546 062706 000014      ADD    #14,SP
2642 : REPORT RL11 REGISTERS
2643 PRINTB #FMT6,#CSNAM,#DANAM,#BANAM,#MPNAM,#CYLWD,#HDWD
(13) 025552 012746 007455      MOV    #HDWD,-(SP)
(12) 025556 012746 007466      MOV    #CYLWD,-(SP)
(11) 025562 012746 006255      MOV    #MPNAM,-(SP)
(10) 025566 012746 006243      MOV    #BANAM,-(SP)
(9)  025572 012746 006250      MOV    #DANAM,-(SP)
(8)  025576 012746 006236      MOV    #CSNAM,-(SP)
(7)  025602 012746 011046      MOV    #FMT6,-(SP)
(6)  025606 012746 000007      MOV    #7,-(SP)
(3)  025612 010600                MOV    SP,R0
(4)  025614 104414                TRAP   C$PNTB
(4)  025616 062706 000020      ADD    #20,SP
2644 PRINTB #FMT8,#LAB1,L.CS,L.DA,L.BA,L.MP
(12) 025622 013746 003056      MOV    L.MP,-(SP)
(11) 025626 013746 003052      MOV    L.BA,-(SP)
(10) 025632 013746 003054      MOV    L.DA,-(SP)
(9)  025636 013746 003050      MOV    L.CS,-(SP)
(8)  025642 012746 006262      MOV    #LAB1,-(SP)
(7)  025646 012746 011160      MOV    #FMT8,-(SP)
(6)  025652 012746 000006      MOV    #6,-(SP)
(3)  025656 010600                MOV    SP,R0
(4)  025660 104414                TRAP   C$PNTB
(4)  025662 062706 000016      ADD    #16,SP
2645 PRINTB #FMT7,#LAB2,T.CS,T.DA,T.BA,T.MP,CURCYL,DESHD
(14) 025666 013746 003126      MOV    DESHD,-(SP)
(13) 025672 013746 003120      MOV    CURCYL,-(SP)
(12) 025676 013746 003066      MOV    I.MP,-(SP)
(11) 025702 013746 003062      MOV    T.BA,-(SP)
(10) 025706 013746 003064      MOV    T.DA,-(SP)
(9)  025712 013746 003060      MOV    T.CS,-(SP)
(8)  025716 012746 006275      MOV    #LAB2,-(SP)
(7)  025722 012746 011110      MOV    #FMT7,-(SP)
(6)  025726 012746 000010      MOV    #10,-(SP)
(3)  025732 010600                MOV    SP,R0
(4)  025734 104414                TRAP   C$PNTB
(4)  025736 062706 000022      ADD    #22,SP
2646 RTS PC
2647
2648 : CLEAR PARAMETER BLOCK FOR REPORTING
2649 CLRPARM: MOV    R5,-(SP) ;STORE R5
2650 MOV    #RESPARM,R1 ;GET ADDRESS OF BLOCK
2651 MOV    #5,R5 ;SET COUNT
2652 2$: CLR    (R1)+ ;CLEAR WORD
2653 DEC    R5 ;DEC COUNT
2654 BNE   2$ ;LOOP UNTIL 0
2655 MOV    #RESPARM,R1 ;RESET POINTER
2656 MOV    (SP)+,R5 ;RESTORE R5
```

CZRLJBO RL01/02 DRIVE TEST 2
CZRLJB.MAC 07-DEC-79 09:06

MACY11 30A(1052) 08-FEB-80 14:49 F 8
GLOBAL SUBROUTINES PAGE 2-7

SEQ 0096

2657 025772 000207
2658
2659 025774

RTS PC

ENDMOD


```
2661  
2662 .TITLE CZRLJBO RL01/02 DRIVE TEST 2  
2663  
2664 .SBTTL *TEST 1 **OUTER GUARD BAND DETECTION  
2665  
2666 BGNMOD HRDWTST  
2667 BGNTST ;TEST 1  
(3) 025774  
2668 T1::  
2669 025774 012737 006507 003026 MOV #P2T03E,ERHEAD ;SET ERROR HEADER  
2670 026002 004737 016214 JSR PC,TSTINT ;INITIALIZE TEST  
2671 026006 004737 016232 JSR PC,GSTATR ;CLEAR DRIVE  
2672 026012 026214 T1965$  
2673 026014 004737 020552 JSR PC,CHOSHD ;GO CHOSE HEAD  
2674 026020 005005 T197$: CLR R5 ;CLEAR FOR POSITION TO 0  
2675 026022 004737 017732 JSR PC,POSHDS ;POSITION HEADS  
2676 026026 026214 T1965$  
2677 BGNSUB  
(3) 026030 T1.1:  
(3) 026030 104402 TRAP C$BSUB  
2678 026032 012737 177777 003116 MOV #-1,NEWCYL ;SET FOR GUARD BAND SEEK  
2679 026040 004737 017160 JSR PC,XSEEK ;DO SEEK  
2680 026044 026170 60$  
2681 026046 012701 000002 MOV #2,R1 ;INITIALIZE WAIT COUNT  
2682 026052 032762 000001 000000 8$: BIT #DRDYMSK,RLCS(R2) ;TEST IF DRIVE READY  
2683 026060 001414 BEQ 9$ ;NO-SKIP  
2684 026062 004737 016262 JSR PC,GSTAT ;GET DRIVE STATUS  
2685 026066 026170 60$  
2686 026070 012703 007760 MOV #MDRDY,R3 ;SET NAME MESSAGE PTR  
2687 026074 012704 010634 MOV #C10MS,R4 ;SET CONDITION MESSAGE PTR  
2688 026100 ERRHRD 301...ERR4 ;REPORT READY ERROR  
(4) 026100 104456 TRAP C$ERHRD  
(5) 026102 000455 .WORD 301  
(5) 026104 000000 .WORD 0  
(5) 026106 012106 .WORD ERR4  
2689 026110 000427 BR 60$ ;EXIT TEST  
2690 026112 005301 9$: DEC R1 ;DEC WAIT COUNT  
2691 026114 001406 BEQ 12$ ;SKIP IF 0  
2692 026116 WAITUS #10. ;WAIT 1MS  
2693 026130 000750 BR 8$ ;LOOP  
2694 026132 012701 000226 12$: MOV #150.,R1 ;SET WAIT COUNT FOR 15 MS  
2695 026136 004737 022054 JSR PC,RDYWAIT ;WAIT FOR READY & REPORT IF NOT READY  
2696 026142 026170 60$  
2697  
2698 026144 004737 022340 JSR PC,GETPOS ;GET POSITION  
2699 026150 026170 60$  
2700 026152 005737 003120 TST CURCYL ;CHECK IF HEADS STILL AT 0  
2701 026156 001404 BEQ 15$ ;YES-SKIP  
2702 026160 ERRHRD 302...ERR8 ;ELSE REPORT CYLINDER ERROR  
(4) 026160 104456 TRAP C$ERHRD  
(5) 026162 000456 .WORD 302  
(5) 026164 000000 .WORD 0  
(5) 026166 013160 .WORD ERR8  
2703 026170 15$:  
2704 026170 012737 000002 003032 60$: MOV #2,ERRSWI ;INIT ERROR SWITCH  
2705 026176 ENDSUB
```

CZRLJBO RL01/02 DRIVE TEST 2
CZRLJB.MAC 07-DEC-79 09:06

MACY11 30A(1052)
*TEST 1

08-FEB-80 14:49 PAGE 3-1
**OUTER GUARD BAND DETECTION

SEQ 0098

(3) 026176
 (3) 026176 104403
 2706 026200
 (3) 026200 104410
 (3) 026202 000012
 2707 026204 004737 020576
 2708 026210 026214
 2709 026212 000702
 2710 026214
 2711 026214
 2712
 2713 026214
 (3) 026214
 (3) 026214 104401
 2714

L10024:

TRAP C\$ESUB
 ESCAPE TST ;EXIT TEST IF ERROR
 TRAP C\$ESCAPE
 .WORD L10023-
 JSR PC,SWAPHD ;GO SWAP TO HEAD 1 OR END TEST
 17\$;ABORT RETURN
 BR T197\$;REDO TEST

17\$:
T1965\$:

ENDTST
L10023:

TRAP C\$ETST

```
2716  
2717 .SBTTL *TEST 2 **INCREMENTAL FORWARD SEEK HEAD 0  
2718 BGNTST ;TEST 2  
2719 (3) 026216 012737 006525 003026 MOV #P2T04E,ERHEAD ;SET ERROR HEADER  
2720 026216 004737 016214 JSR PC,TSTINT ;INITIALIZE TEST  
2721 026230 004737 016232 JSR PC,GSTATR ;CLEAR DRIVE  
2722 026234 026424 T2065$  
2723 026236 004737 020552 JSR PC,CHOSHD ;GO CHOSE HEAD  
2724 026242 005737 003126 TST DESHD ;TEST IF THIS IS HEAD 0  
2725 026246 001402 BEQ 2$ ;YES - SKIP  
2726 026250 EXIT TST ;ELSE EXIT TEST  
2727 (3) 026250 104432 TRAP C$EXIT  
2728 (3) 026252 000152 .WORD L10025-  
2729 026254 013705 013560 2$: MOV LOLIMW,R5 ;CLEAR TO POSITION HEADS TO LOLIMIT  
2730 026260 004737 017732 JSR PC,POSHDS ;POSITION HEADS  
2731 026264 026424 T2065$  
2732 (3) 026266 BGNSUB  
2733 (3) 026266 104402 TRAP C$BSUB ;T2.1:  
2734 026270 004737 022340 T206$: JSR PC,GETPOS ;GET POSITION  
2735 026274 026414 60$  
2736 026276 INLOOP ;CHECK IF IN ERROR LOOP  
2737 (3) 026276 104420 TRAP C$INLP  
2738 026300 BNCOMPLETE 5$ ;NO - SKIP  
2739 (2) 026300 103007 BCC 5$  
2740 026302 023737 003120 003116 CMP CURCYL,NEWCYL ;CHECK IF POSITIONED AT DESIRED LOC  
2741 026310 001003 BNE 5$ ;NO - SKIP  
2742 026312 004737 020636 JSR PC,ONSWAP ;ELSE SWAP NEW AND OLD CYLINDERS  
2743 026316 000405 BR 7$ ;SKIP  
2744 026320 013737 003120 003116 5$: MOV CURCYL,NEWCYL ;PLACE CURRENT INTO NEW  
2745 026326 005237 003116 7$: INC NEWCYL ;BUMP FOR ONE CYLINDER SEEK  
2746 026332 004737 017160 JSR PC,XSEEK ;DO SEEK  
2747 026336 026414 60$  
2748 026340 012701 000226 MOV #150.,R1 ;SET WAIT TIME 15 MS  
2749 026344 004737 022054 JSR PC,RDYWAIT ;WAIT FOR READY  
2750 026350 026414 60$  
2751 026352 004737 022466 JSR PC,VERPOS ;GO VERIFY POSITON  
2752 026356 026414 60$  
2753 026360 032737 000002 013556 BIT #ALLSEC,MISWIW ;TEST IF CHECK ALL SECTORS  
2754 026366 001406 BEQ 11$ ;NO-SKIP  
2755 026370 004737 022610 JSR PC,RDALHD ;GO READ ALL HEADERS  
2756 026374 026414 60$  
2757 026376 004737 021460 JSR PC,VERHDR ;GO VERIFY HEADER  
2758 026402 026414 60$  
2759 026404 023737 013562 003116 11$: CMP HILIMW,NEWCYL ;CHECK IF HILIMIT REACHED  
2760 026412 103726 BLO T206$ ;NO-LOOP  
2761 026414 012737 000002 003032 60$: MOV #2,ERRSWI ;INIT ERROR SWITCH  
2762 (3) 026422 ENDSUB  
2763 (3) 026422 104403 L10026:  
2764 026424 T2065$: TRAP C$ESUB
```

CZRLJBO RL01/02 DRIVE TEST 2
CZRLJB.MAC 07-DEC-79 09:06

MACY11 30A(1052) 08-FEB-80 14:49 J 8 PAGE 3-3
*TEST 2 **INCREMENTAL FORWARD SEEK HEAD 0

SEQ 0100

2763 026424
(3) 026424
(3) 026424 104401

ENDTST
L10025:
TRAP C\$ETST

```
2765      .SBTTL *TEST 3          **INCREMENTAL REVERSE SEEK HEAD 0
2766      BGNTST ;TEST 3
(3) 026426
2767 026426 012737 006545 003026 MOV #P<TOSE,ERHEAD ;SET ERROR HEADER
2768 026434 004737 016214 JSR PC,TSTINT ;INITIALIZE TEST
2769 026440 004737 016232 JSR PC,GSTATR ;CLEAR DRIVE
2770 026444 026634 T2165$
2771 026446 004737 020552 JSR PC,CHOSHD ;GO CHOSE HEAD
2772 026452 005737 003126 TST DESHD ;TEST IF HEAD 0 SELECTED
2773 026456 001402 BEQ 2$ ;YES - SKIP
2774 026460 EXIT TST ;ELSE EXIT TEST
(3) 026460 104432 TRAP C$EXIT
(3) 026462 000152 .WORD L10027-
2775 026464 013705 013562 2$: MOV HILIMW,R5 ;SET TO POSITION HDS TO HILIMIT
2776 026470 004737 017732 JSR PC,POSHDS ;POSITION HEADS
2777 026474 026634 T2165$
2778 026476 BGNSUB
(3) 026476
(3) 026476 104402 T3.1:
2779 026500 004737 022340 T216$: TRAP C$BSUB
2780 026504 026624 JSR PC,GETPOS ;GET POSITION
2781 026506 INLOOP 60$
(3) 026506 104420 TRAP C$INLP ;CHECK IF IN ERROR LOOP
2782 026510 BNCOMPLETE 5$ ;NO - SKIP
(2) 026510 103007 BCC 5$
2783 026512 023737 003120 003116 CMP CURCYL,NEWCYL ;CHECK IF POSITIONED AT DES LOC
2784 026520 001003 BNE 5$ ;NO - SKIP
2785 026522 004737 020636 JSR PC,ONSWAP ;ELSE SWAP OLD AND NEW CYLINDERS
2786 026526 000405 BR 7$ ;SKIP
2787 026530 013737 003120 003116 5$: MOV CURCYL,NEWCYL ;PUT CURRENT INTO NEW
2788 026536 005337 003116 DEC NEWCYL ;DEC FOR ONE CYLINDER REVERSE SEEK
2789 026542 004737 017160 7$: JSR PC,XSEEK ;SEEK TO IT
2790 026546 026624 60$
2791 026550 012701 000226 MOV #150.,R1 ;SET WAIT FOR 15 MS
2792 026554 004737 022054 JSR PC,RDYWAIT ;WAIT FOR READY
2793 026560 026624 60$
2794
2795 026562 004737 022466 JSR PC,VERPOS ;VERIFY POSITION
2796 026566 026624 60$
2797
2798 026570 032737 000002 003020 BIT #ALLSEC,OPFLAG ;TEST IF USE ALL SECTORS
2799 026576 001406 BEQ 11$ ;NO-SKIP
2800 026600 004737 022610 JSR PC,RDALHD ;ELSE READ ALL THE HDRS
2801 026604 026624 60$
2802 026606 004737 021460 JSR PC,VERHDR ;VERIFY THE HEADERS
2803 026612 026624 60$
2804 026614 11$:
2805 026614 023737 013560 003116 CMP LOLIMW,NEWCYL ;CHECK IF REACHED LOLIMIT
2806 026622 103726 BLO T216$ ;NO - LOOP
2807 026624 012737 000002 003032 60$: MOV #2,ERRSWI ;INIT ERROR SWITCH
2808 026632 ENDSUB
(3) 026632 L10030:
(3) 026632 104403 TRAP C$ESUB
2809 026634 T2165$:
2810 026634 ENDTST
(3) 026634 L10027:
```

CZRLJBO RLO1/02 DRIVE TEST 2
CZRLJB.MAC 07-DEC-79 09:06

MACY11 30A(1052) 08-FEB-80 14:49 L 8 PAGE 3-5
*TEST 3 **INCREMENTAL REVERSE SEEK HEAD 0

FO 0102

(3) 026634 104401

TRAP C\$ETST

```
2812 .SBTTL *TEST 4 **INCREMENTAL FORWARD SEEK HEAD 1
2813 BGNSTST ;TEST 4
(3) 026636 T4::
2814 026636 012737 006565 003026 MOV #P2T06E,ERHEAD ;SET ERROR HEADER
2815 026644 004737 016214 JSR PC,TSTINT ;INITIALIZE TEST
2816 026650 004737 016232 JSR PC,GSTATR ;CLEAR DRIVE
2817 026654 027060 T2265$
2818 026656 005037 003126 CLR DESHD ;SET HEAD TO 0
2819 026662 013705 013560 MOV LOLIMW,R5 ;CLEAR FOR POSITION HDS TO LOLIMIT
2820 026666 004737 017732 JSR PC,POSHDS ;POSITION HDS
2821 026672 027060 T2265$
2822 026674 012737 000001 003126 MOV #1,DESHD ;SET TO HEAD 1
2823 026702 032737 010000 013556 BIT #HEADLM,MISWIW ;TEST IF HEAD SPECIFIED
2824 026710 001405 BEQ 2$ ;NO - SKIP
2825 026712 005737 013564 TST HEADW ;TEST IF IT IS HEAD 0
2826 026716 001002 BNE 2$ ;NO - SKIP
2827 026720 EXIT TST ;ELSE EXIT TEST
(3) 026720 104432 TRAP C$EXIT
(3) 026722 000136 .WORD L10031-.
2828 026724 2$:
2829 026724 BGNSTST T4.1:
(3) 026724 104402 TRAP C$BSUB
2830 026726 004737 022340 T227$: JSR PC,GETPOS ;GET CURRENT POSITION
2831 026732 INLOOP ;CHECK IF IN ERROR LOOP
(3) 026732 104420 TRAP C$INLP
2832 026734 BNCOMPLETE 5$ ;NO - SKIP
(2) 026734 103007 BCC 5$
2833 026736 023737 003120 003116 CMP CURCYL,NEWCYL ;CHECK IF AT DESIRED LOCATION
2834 026744 001003 BNE 5$ ;NO - SKIP
2835 026746 004737 020636 JSR PC,ONSWAP ;SWAP OLD AND NEW CYLINDER
2836 026752 000405 BR 7$ ;SKIP
2837 026754 013737 003120 003116 5$: MOV CURCYL,NEWCYL ;MOVE CURRENT INTO NEW
2838 026762 005237 003116 7$: INC NEWCYL ;BUMP NEWCYL FOR ONE CYL FWRD SEEK
2839 026766 JSR PC,XSEEK ;DO SEEK
2840 026766 004737 017160 60$
2841 026772 027050 MOV #150.,R1 ;SET WAIT COUNT 15 MS
2842 026774 012701 000226 JSR PC,RDYWAIT ;WAIT FOR READY
2843 027000 004737 022054 60$
2844 027004 027050 JSR PC,VERPOS ;VERIFY POSITION IS CORRECT
2845 027006 004737 022466 60$
2846 027012 027050
2847
2848 027014 032737 000002 013556 BIT #ALLSEC,MISWIW ;CHECK IF USE ALL SECTORS
2849 027022 001406 BEQ 9$ ;NO-SKIP
2850 027024 004737 022610 JSR PC,RDALHD ;ELSE READ ALL HEADERS
2851 027030 027050 60$
2852 027032 004737 021460 JSR PC,VERHDR ;VERIFY HEADERS
2853 027036 027050 60$
2854 027040 9$:
2855 027040 023737 013562 003116 CMP #1,ILIMW,NEWCYL ;CHECK IF DONE
2856 027046 101327 BHI T227$ ;NO - LOOP
2857 027050 012737 000002 003032 60$: MOV #2,ERRSWI ;INIT ERROR SWITCH
2858 027056 ENDSUB
(3) 027056 L10032:
(3) 027056 104403 TRAP C$ESUB
```

CZRLJB0 RL01/02 DRIVE TEST 2
CZRLJB.MAC 07-DEC-79 09:06

MACY11 30A(1052)
*TEST 4

08-FEB-80 14:49 N 8
PAGE 3-7
**INCREMENTAL FORWARD SEEK HEAD 1

SEQ 0104

2859 027060
2860 027060
(3) 027060
(3) 027060 104401
2861

T2265\$:
ENDTST
L10031:
TRAP C\$ETST


```

2863
2864
2865
2866 027062
(3) 027062
2867
2868 027062 012737 006605 00302C
2869 027070 004737 016214
2870 027074 004737 016232
2871 027100 027266
2872 027102 004737 020552
2873 027106 013705 002316
2874 027112 004737 017732
2875 027116 027266
2876 027120
(3) 027120
(3) 027120 104402
2877 027122 013737 002324 003116
2878 027130 004737 017160
2879 027134 027242
2880 027136 012701 000001
2881 027142 032762 000001 000000 7$:
2882 027150 001414
2883 027152 004737 016262
2884 027156 027242
2885 027160 012703 007760
2886 027164 012704 010634
2887 027170
(4) 027170 104456
(5) 027172 001275
(5) 027174 000000
(5) 027176 027106
2888 027200 000420
2889 027202 005301
2890 027204 001406
2891 027206
2892 027220 000750
2893 027222 012701 000226
2894 027226 004737 022054
2895 027232 027242
2896
2897 027234 004737 022466
2898 027240 027242
2899 027242 012737 000002 003032
2900 027250
(3) 027250
(3) 027250 104403
2901 027252
(3) 027252 104410
(3) 027254 000012
2902 027256 004737 020576
2903 027262 027266
2904 027264 000710
2905 027266
2906 027266
2907

```

```

.SBTL *TEST 5 **INNER GUARD BAND DETECTION
RGNTST ;TEST 5
T5::
MOV #P2T07E.ERHEAD ;SET ERROR HEADER
JSR PC,ISTINT ;INITIALIZE TEST
JSR PC,GSTATR ;CLEAR DRIVE
T233$: JSR PC,CHOSHU ;GO CHOSE HEAD
MOV HLMTW,R5 ;SET FOR POSITION TO 255.
JSR PC,POSHDS ;POSITION HEADS
T2365$
BGNSUB
T5.1:
TRAP CSBSUB
MOV GRND,NEWCYL ;SET FOR INNER GUARD BAND SEEK
JSR PC,XSEEK ;DO IT
60$
MOV #1.,R1 ;INITIALIZE WAIT COUNT
BIT #DRDYMSK,RLCS(R2) ;CHECK IF READY
BEQ 9$ ;NO-SKIP
JSR PC,GSTAT ;GET DRIVE STATUS
60$
MOV #MDRDY,R3 ;SET NAME MESSAGE PTR
MOV #C10MS,R4 ;SET CONDITION MESSAGE PTR
ERRHRD 701.,ERR4 ;REPORT READY ERROR
TRAP CSERHRD
WORD 701
WORD 0
WORD ERR4
BR 60$ ;EXIT TEST
9$: DEC R1 ;DEC WAIT COUNT
BEQ 11$ ;SKIP IF 0
WAITUS #10. ;WAIT 1MS
BR 7$ ;LOOP
11$: MOV #150.,R1 ;SET WAIT COUNT 15 MS
JSR PC,RDYWAIT ;GO WAIT FOR READY
60$
JSR PC,VERPOS ;GO VERIFY POSITION IS 255
60$
MOV #2.,ERRSW1 ;INIT ERROR SWITCH
ENDSUB
L10034:
TRAP CS$SUB
ESCAPE TST ;EXIT TEST IF ERROR
TRAP CS$ESCAPE
WORD L10033-.
JSR PC,SWAPHD ;GO SWAP TO HEAD 1 OR END TEST
15$: JSR PC,SWAPHD ;ABORT RETURN
BR T233$ ;REPEAT THE TESTS
T2365$:

```

CZRLJBO RL01/02 DRIVE TEST 2
CZRLJB.MAC 07-DEC-79 09:06

MACY11 30A(1052) 08-FEB-80 14:49 PAGE 3-9
*TEST 5 **INNER GUARD BAND DETECTION

C 9

SEQ 0106

2908 027266
(3) 027266
(3) 027266 104401
2909

ENDTST
L10033:
TRAP LSETST

```
2911
2912
2913 027270          .SBTTL *TEST 6          **INCREMENTAL REVERSE SEEK HFAD 1
      (3) 027270          BGNST          ;TEST 6
2914 027270 012737 006623 003026      MOV #P2TOBE,ERHEAD ;SET ERROR HEADER
2915 027276 004737 016214          JSR PC,TSTINT      ;INITIALIZE TEST
2916 027302 004737 016232          JSR PC,GSTATR     ;GET STATUS & CLEAR
2917 027306 027514          T2465$
2918 027310 005037 003126          CLR DESHD         ;SET TO HEAD 0
2919 027314 013705 013562          MOV HILIMW,R5    ;SET TO POSITION HDS AT HILIMIT
2920 027320 004737 017732          JSR PC,POSHDS    ;POSITION HDS
2921 027324 027514          T2465$
2922 027326 012737 000001 003126      MOV #1,DESHD     ;SET TO SELECT HD 1
2923 027334 032737 010000 013556      BIT #HEADLM,MISWIW ;TEST IF HEAD SPECIFIED
2924 027342 001405          BEQ 2$          ;NO - SKIP
2925 027344 005737 013564          TST HEADW       ;TEST IF HEAD SPECIFIED IS 0
2926 027350 001002          BNE 2$          ;NO - SKIP
2927 027352          EXIT TST        ;ESLE EXIT TEST
      (3) 027352 104432          TRAP C$EXIT
      (3) 027354 000140          .WORD L10035-.
2928 027356          2$:
2929 027356          BGNSUB
      (3) 027356
      (3) 027356 104402          T6.1:
2930 027360 004737 022340          T247$: TRAP C$BSUB
2931 027364 027504          JSR PC,GETPOS   ;GET CURRENT POSITION
2932 027366          60$
      (3) 027366 104420          INLOOP          ;CHECK IF IN ERROR LOOP
2933 027370          TRAP C$INLP
      (2) 027370 103007          BNCOMPLETE 5$   ;NO - SKIP
2934 027372 023737 003120 003116      BCC 5$
2935 027400 001003          CMP CURCYL,NEWCYL ;CHECK IF POSITIONED AT DESIRED LOC
2936 027402 004737 020636          BNE 5$          ;NO - SKIP
2937 027406 000405          JSR PC,ONSWAP   ;ELSE SWAP OLD AND NEW CYLINDER
2938 027410 013737 003120 003116 5$: BR 7$          ;SKIP
2939 027416 005337 003116          MOV CURCYL,NEWCYL ;MOV CUR TO NEW
2940 027422 004737 017160 7$: DEC NEWCYL      ;DEC NEWCYL FOR 1 CYL REV SEEK
2941 027426 027504          JSR PC,XSEEK   ;DO SEEK
2942 027430 012701 000226          MOV #150.,R1    ;SET WAIT FOR 15 MS
2943 027434 004737 022054          JSR PC,RDYWAIT ;WAIT FOR READY
2944 027440 027504          60$
2945 027442 004737 022466          JSR PC,VERPOS   ;VERIFY POSITION
2946 027446 027504          60$
2947 027450 032737 000002 013556      BIT #ALLSEC,MISWIW ;TEST IF ALL SECTORS
2948 027456 001406          BEQ 9$          ;NO-EXIT
2949 027460 004737 022610          JSR PC,RDALHD  ;READ ALL HEADERS
2950 027464 027504          60$
2951 027466 004737 021460          JSR PC,VERHDR  ;VERIFY HEADER
2952 027472 027504          60$
2953 027474          9$:
2954 027474 023737 013560 003116      CMP LOLIMW,NEWCYL ;CHECK IF AT LOLIMIT
2955 027502 103726          BLO T247$      ;NO - LOOP
2956 027504 012737 000002 003032 60$: MOV #2,ERRSWI    ;INIT ERROR SWITCH
2957 027512          ENDSUB
      (3) 027512          L10036:
      (3) 027512 104403          TRAP C$ESUB
```

CZRLJBO RL01/02 DRIVE TEST 2
CZRLJB.MAC 07-DEC-79 09:06

MACV11 30A(1052) 08-FEB-80 14:49 PAGE 3-11
*TEST 6 **INCREMENTAL REVERSE SEEK HEAD 1

E 9

SFO 0108

2958 027514
2959 027514
(3) 027514
(3) 027514 104401

T24658:
ENDTST
L10035:
TRAP CSETST

```
2961 .SBTTL *TEST 7 **SEEK TESTS
2962 027516 BGNTST ;TEST 7
(3) 027516 T7::
2963 027516 012737 006643 003026 MOV #P2T09E,ERHEAD ;SET ERROR HEADER
2964 027524 004737 016214 JSR PC,TSTINT ;INITIALIZE TEST
2965 027530 004737 016232 JSR PC,GSTATR ;CLEAR DRIVE
2966 027534 030024 T2565$
2967 027536 004737 020552 JSR PC,CHOSHD ;GO CHOSE HEAD
2968 027542 013705 013560 MOV LOLIMW,R5 ;SET TO POSITION HEADS TO LOLIMIT
2969 027546 004737 017732 JSR PC,POSHDS ;POSITION HDS TO LOWLIMIT
2970 027552 030024 T2565$
2971 027554 004737 022340 T256$: JSR PC,GETPOS ;GET CURRENT POSITION
2972 027560 030024 T2565$
2973 027562 013737 003120 003116 MOV CURCYL,NEWCYL ;PUT CURRENT INTO NEW
2974 027570 012704 002444 MOV #T25TBL,R4 ;SET POINTER TO TABLE OF SEEK DIFF FOR RLO1
2975 027574 022737 000001 002312 T258$: CMP #1,T.DRIVE ;CHECK TYPE OF DRIVE
2976 027602 001402 BEQ T2588$ ;BRANCH IF RLO1
2977 027604 012704 002472 MOV #T25TB2,R4 ;POINT TO THE RLO2 TABLE OF CYLINDERS
2978
2979 027610 012405 T2588$: MOV (R4)+,R5 ;PUT FIRST IN R5
2980 027612 013701 013562 MOV HILIMW,R1 ;GET HILIMIT
2981 027616 163701 013560 SUB LOLIMW,R1 ;SUBTRACT LOLIMIT
2982 027622 021401 CMP (R4),R1 ;CHECK IF NEW DIFFERENCE IS IN BOUNDS
2983 027624 101073 BHI T2517$$ ;NO - SKIP TEST
2984 027626 060537 003116 T257$: ADD R5,NEWCYL ;ADD TO PRESENT POSITION
2985 027632 023737 003116 013560 CMP NEWCYL,LOLIMW ;CHECK IF AT OR PAST LOLIMIT
2986 027640 002004 BGE 9$ ;NO - SKIP
2987 027642 013737 013560 003116 MOV LOLIMW,NEWCYL ;ELSE SET TO LOLIMIT
2988 027650 000407 BR 11$
2989 027652 023737 003116 013562 9$: CMP NEWCYL,HILIMW ;CHECK IF AT HILIMIT OR GREATER
2990 027660 003403 BLE 11$ ;NO - SKIP
2991 027662 013737 013562 003116 MOV HILIMW,NEWCYL ;ELSE SET FOR HILIMIT
2992 027670 11$:
2993 027670 BGNSUB
(3) 027670 T7.1:
(3) 027670 104402 TRAP C$BSUB
2994 027672 INLOOP ;CHECK IF IN ERROR LOOP
(3) 027672 104420 TRAP C$INLP
2995 027674 BNCOMplete 13$ ;NO - SKIP
(2) 027674 103011 BCC 13$
2996 027676 004737 022340 JSR PC,GETPOS ;GET CURRENT POSITION
2997 027702 027746 60$
2998 027704 023737 003120 003116 CMP CURCYL,NEWCYL ;CHECK IF HEADS AT DESIRED POSITION
2999 027712 001002 BNE 13$ ;NO - SKIP
3000 027714 004737 020636 JSR PC,ONSWAP ;ELSE SWAP CURRENT AND NEW CYLINDERS
3001 027720 004737 017160 13$: JSR PC,XSEEK ;DO SEEK
3002 027724 027746 60$
3003 027726 012701 005670 MOV #3000.,R1 ;SET WAIT COUNT
3004 027732 004737 022054 JSR PC,RDYWAIT ;WAIT FOR READY
3005 027736 027746 60$
3006 027740 004737 022466 JSR PC,VERPOS ;VERIFY POSITION
3007 027744 027746 60$
3008 027746 012737 000002 003032 60$: MOV #2,ERRSWI ;INITIALIZE ERROR SWITCH
3009 027754 ENDSUB
(3) 027754 L10040:
(3) 027754 104403 TRAP C$ESUB
```

3010	027756			ESCAPE	TST	:EXIT TEST IF ERROR		
(3)	027756	104410		TRAP	C\$ESCAPE			
(3)	027760	000044		.WORD	L10037-			
3011	027762	023737	013562	003116	CMP	HILIMW,NEWCYL	:CHECK IF SEEK WAS TO HILIMIT	
3012	027770	001002			BNE	15\$:NO - SKIP	
3013	027772	005405			NEG	R5	:ELSE SET R5 TO REPEAT DIFF IN REVERSE	
3014	027774	000714			BR	T257\$		
3015	027776	023737	013560	003116	15\$:	CMP	LOLIMW,NEWCYL	:TEST IF LAST SEEK WAS TO LOLIMIT
3016	030004	001310			BNE	T257\$:NO - GO DO SEEK TEST	
3017	030006	021437	002316		CMP	(R4),HLMTW	:CHECK IF ALL TABLE DIFF USED	
3018	030012	001276			BNE	T2588\$:NO - SKIP	
3019	030014	004737	020576		T2517\$:	JSR	PC,SWAPHD	:GO SWAP TO HEAD 1 OR END TEST
3020	030020	030024			T2565\$	BR		:ABORT RETURN
3021	030022	000654						:REPEAT TEST HEAD 1
3022	030024				T2565\$:			
3023	030024				ENDTST			
(3)	030024				L10037:			
(3)	030024	104401			TRAP	C\$ETST		

```
3025 .SBTTL *TEST 8 **FORWARD OSCILLATING SEEK
3026 BGNTST ;TEST 8
(3) 030026
3027 030026 012737 006646 003026 MOV #P2T10E,ERHEAD ;SET ERROR HEADER
3028 030034 004737 016214 JSR PC,TSTINT ;INITIALIZE TEST
3029 030040 004737 016232 JSR PC,GSTATR ;CLEAR DRIVE
3030 030044 030322 T2665$
3031 030046 004737 020552 JSR PC,CHOSHD ;GO CHOSE HEAD
3032 030052 012705 000001 T266$: MOV #1,R5 ;LOAD R5 FOR FIRST SEEK
3033 030056 032737 020000 013556 BIT #H1CYL,MISWIW ;TEST IF HI CYLINDER SPECED
3034 030064 001402 BEQ 2$ ;NO - SKIP
3035 030066 013705 013562 MOV #H1CYL,R5 ;ELSE SET UPPER LIMIT
3036 030072 005037 003116 2$: CLR NEWCYL ;SET TO SEEK TO CYL 0
3037 030076 032737 040000 013556 BIT #LOCYL,MISWIW ;CHECK IF LO CYL SPEC'D
3038 030104 001403 BEQ 5$ ;NO - SKIP
3039 030106 013737 013560 003116 MOV #LOCYL,NEWCYL ;ELSE SET LOWER LIMIT
3040 030114 004737 017160 5$: JSR PC,XSEEK ;DO SEEK
3041 030120 030322 T2665$
3042 030122 012701 005670 MOV #3000.,R1 ;SET WAIT COUNT FOR 120 MS
3043 030126 004737 022054 JSR PC,RDYWAIT ;WAIT FOR READY
3044 030132 030322 T2665$
3045 030134 004737 022340 T267$: JSR PC,GETPOS ;GET HEAD POSITION
3046 030140 030322 T2665$
3047 030142 010537 003116 MOV R5,NEWCYL ;LOAD NEW CYLINDER INTO NEWCYL
3048 030146 BGNSUB
(3) 030146 T8.1:
(3) 030146 104402 TRAP C$SUB
3049 030150 INLOOP ;CHECK IF IN ERROR LOOP
(3) 030150 104420 TRAP C$INLP
3050 030152 BNCOMPLETE 18$ ;NO - SKIP
(2) 030152 103011 BCC 18$
3051 030154 004737 022340 JSR PC,GETPOS ;GET POSITION
3052 030160 030256 60$
3053 030162 023737 003120 003116 CMP #CURCYL,NEWCYL ;CHECK IF HEADS AT DESIRED LOC
3054 030170 001002 BNE 18$ ;NO - SKIP
3055 030172 004737 020636 JSR PC,ONSWAP ;SWAP OLD AND NEW
3056 030176 004737 017160 18$: JSR PC,XSEEK ;DO SEEK
3057 030202 030256 60$
3058 030204 012701 005670 MOV #3000.,R1 ;SET WAIT COUNT 120 MS
3059 030210 004737 022054 JSR PC,RDYWAIT ;WAIT FOR READY
3060 030214 030256 60$
3061 030216 004737 022466 JSR PC,VERPOS ;VERIFY HEAD POSITION
3062 030222 030256 60$
3063 030224 005737 003124 TST DESSGN ;TEST IF JUST SEEK REV
3064 030230 001412 BEQ 60$ ;YES - SKIP
3065 030232 005037 003116 CLR NEWCYL ;ELSE SET TO SEEK TO 0
3066 030236 032737 040000 013556 BIT #LOCYL,MISWIW ;CHECK IF LO LIMIT SPEC'D
3067 030244 001754 BEQ 18$ ;NO - SKIP
3068 030246 013737 013560 003116 MOV #LOCYL,NEWCYL ;ELSE SET LOW LIMIT FOR SEEK
3069 030254 000750 BR 18$
3070 030256 012737 000002 003032 60$: MOV #2,ERRSWI ;INIT ERROR SWITCH
3071 030264 ENDSUB
(3) 030264 L10042:
(3) 030264 104403 TRAP C$ESUB
3072 030266 104410 ESCAPE TST ;EXIT TEST IF ERROR
(3) 030266 TRAP C$ESCAPE
```

(3)	030270	000032			.WORD	L10041-	
3073	030272	032737	020000	(013556	BIT	#HICYL,MISWIW	:TEST IF UPPER LIMIT SPEC'D
3074	030300	001004			BNE	20\$:YES - SKIP
3075	030302	005205			INC	R5	:BUMP R5
3076	030304	020537	002324		CMP	R5,GBND	:ALL CYLINDERS DONE
3077	030310	001311			BNE	T267\$:NO - GO DO NEXT CYLINDER
3078	030312	004737	020576	20\$:	JSR	PC,SWAPHD	:GO SWAP TO HEAD 1 OR END TEST
3079	030316	030322			T2665\$:ABORT RETURN
3080	030320	000654			BR	T266\$:GO DO TESTS
3081	030322			T2665\$:			
3082	030322			ENDTST			
(3)	030322			L10041:			
(3)	030322	104401			TRAP	C\$ETST	


```
3084 .SBTTL *TEST 9 **REVERSE OSCILLATING SEEK
3085 (3) 030324 BGNTST ;TEST 9
3086 030324 012737 006661 003026 MOV #P2T11E,ERHEAD ;SET ERROR HEADER
3087 030332 004737 016214 JSR PC,TSTINT ;INITIALIZE TEST
3088 030336 004737 016232 JSR PC,GSTATR ;CLEAR DRIVE
3089 030342 030620 T2765$
3090 030344 004737 020552 JSR PC,CHOSHD ;GO CHOSE HEAD
3091 030350 013737 002316 003116 T275$: MOV HLMTW,NEWCYL ;SEEK OUT TO 255.
3092 030356 032737 020000 013556 BIT #HICYL,MISWIW ;TEST IF UPPER LIMIT SPEC'D
3093 030364 001403 BEQ 2$ ;NO - SKIP
3094 030366 013737 013562 003116 MOV HILIMW,NEWCYL ;ELSE SET UPPER LIMIT
3095 030374 013705 002322 2$: MOV NXTHL,R5 ;SET R5 FOR FIRST SEEKS
3096 030400 032737 040000 013556 BIT #LOCYL,MISWIW ;CHECK IF LO LIMIT SPEC'D
3097 030406 001402 BEQ 5$ ;NO - SKIP
3098 030410 013705 013560 MOV LOLIMW,R5 ;SET LOWER LIMIT
3099 030414 004737 017160 5$: JSR PC,XSEEK ;DO SEEK
3100 030420 030620 T2765$
3101 030422 012701 005670 MOV #3000.,R1 ;SET WAIT TO 120 MS
3102 030426 004737 022054 JSR PC,RDYWAIT ;WAIT FOR DRIVE READY
3103 030432 030620 T2765$
3104 030434 004737 022340 T276$: JSR PC,GETPOS ;GET POSITION
3105 030440 030620 T2765$
3106 030442 010537 003116 MOV R5,NEWCYL ;SET FOR NEXT SEEK
3107 030446 (3) 030446 BGNSUB
3108 (3) 030446 104402 TRAP C$SUB T9.1:
3109 (3) 030452 104420 INLOOP ;CHECK IF IN ERROR LOJP
3110 (2) 030452 103011 TRAP C$INLP
3111 030454 004737 022340 BNCOMPLETE 18$ ;NO - SKIP
3112 030462 023737 003120 003116 BCC 18$
3113 030470 001002 JSR PC,GETPOS ;ELSE GET POSITION
3114 030472 004737 020636 60$
3115 030476 004737 017160 18$: JSR PC,ONSWAP ;CHECK IF AT DESIRED CYL
3116 030502 030560 60$ ;NO - SKIP
3117 030504 012701 005670 JSR PC,ONSWAP ;ELSE SWAP OLD AND NEW CYL
3118 030510 004737 022054 60$ ;DO SEEK
3119 030514 030560 MOV #3000.,R1 ;SET WAIT FOR 120 MS
3120 030516 004737 022466 JSR PC,RDYWAIT ;WAIT FOR READY
3121 030522 030560 60$
3122 030524 005737 003124 JSR PC,VERPOS ;VERIFY POSITION
3123 030530 001013 TST DESSGN ;CHECK IF JUST SEEK FWD
3124 030532 013737 002316 003116 BNE 60$ ;YES - SKIP
3125 030540 032737 020000 013556 MOV HLMTW,NEWCYL ;ELSE SEEK TO TO 255
3126 030546 001753 BIT #HICYL,MISWIW ;TEST IF HILIMIT SPEC'D
3127 030550 013737 013562 003116 BEQ 18$ ;NO - SKIP
3128 030556 000747 MOV HILIMW,NEWCYL ;SET TO UPPER LIMIT
3129 030560 012737 000002 003032 60$: BR 18$
3130 (3) 030566 ENDSUB L10044:
3131 (3) 030566 104403 TRAP C$ESUB ;EXIT TEST IF ERROR
3132 (3) 030570 104410 ESCAPE TST
3133 (3) 030570 TRAP C$ESCAPE
```

CZRLJBO RL01/02 DRIVE TEST 2
CZRLJB.MAC 07-DEC-79 09:06

MACY11 30A(1052)
*TEST 9

08-FEB-80 14:49 PAGE 3-17

**REVERSE OSCILLATING SEEK

SEQ 0114

(3)	030572	000026			.WORD	L10043-	
3132	030574	032737	040000	013556	BIT	#LOCYL,MISWIW	:TEST IF LOLIMIT SPEC'D
3133	030602	001002			BNE	20\$:YES - SKIP
3134	030604	005305			DEC	R5	:DEC CYLINDER COUNT
3135	030606	100312			BPL	T276\$:IF STILL POSITIVE, DO SEEKS AGAIN
3136	030610	004737	020576		JSR	PC,SWAPHD	:GO SWAP TO HEAD 1 OR END TEST
3137	030614	030620			T2765\$:ABORT RETURN
3138	030616	000654			BR	T275\$:LOOP AGAIN
3139	030620				T2765\$:		
3140	030620				ENDTST		
(3)	030620				L10043:		
(3)	030620	104401			TRAP	C\$ETST	
3141							
3142	030622				ENDMOD		
3143							
3144							
3145							

3147					.SBTTL	PARAMETER CODING
3148	030622				BGNMOD	HRDPRM
3149	030622				BGNHRD	
(3)	030622	000030				.WORD L10045-L\$HARD/2
3150	030624				GPRML	CNTYPE,CNT,1,YES
(4)	030624	005130				.WORD T\$CODE
(4)	030626	030770				.WORD CNTYPE
(4)	030630	000001				.WORD 1
3151	030632				GPRMA	CSRMSG,CSR,0,160000,177776,YES
(4)	030632	000031				.WORD T\$CODE
(4)	030634	030704				.WORD CSRMSG
(4)	030636	160000				.WORD T\$LLOLIM
(4)	030640	177776				.WORD T\$HILIM
3152	030642				GPRMA	VECMMSG,VECT,0,0,776,YES
(4)	030642	001031				.WORD T\$CODE
(4)	030644	030720				.WORD VECMSG
(4)	030646	000000				.WORD T\$LLOLIM
(4)	030650	000776				.WORD T\$HILIM
3153	030652				GPRMD	DRMSG,DRSB,0,3400,0,7,YES
(4)	030652	004032				.WORD T\$CODE
(4)	030654	030762				.WORD DRMSG
(4)	030656	003400				.WORD 3400
(4)	030660	000000				.WORD T\$LLOLIM
(4)	030662	000007				.WORD T\$HILIM
3154	030664				GPRML	DRTYPE,TYPDR,1,YES
(4)	030664	003130				.WORD T\$CODE
(4)	030666	030740				.WORD DRTYPE
(4)	030670	000001				.WORD 1
3155	030672				GPRMD	BRMSG,PRIOR,0,340,0,7,YES
(4)	030672	002032				.WORD T\$CODE
(4)	030674	030727				.WORD BRMSG
(4)	030676	000340				.WORD 340
(4)	030700	000000				.WORD T\$LLOLIM
(4)	030702	000007				.WORD T\$HILIM
3156						
3157	030704				ENDHRD	
(2)						.EVEN
(3)	030704				L10045:	
3158						
3159	030704	052502	020123	042101	CSRMSG:	.ASCIZ /BUS ADDRESS/
	030712	051104	051505	000123		
3160	030720	042526	052103	051117	VECMMSG:	.ASCIZ /VECTOR/
	030726	000				
3161	030727	102	020122	042514	BRMSG:	.ASCIZ /BR LEVEL/
	030734	042526	000114			
3162	030740	051104	053111	020105	DRTYPE:	.ASCIZ /DRIVE TYPE = RL01/
	030746	054524	042520	036440		
	030754	051040	030114	000061		
3163	030762	051104	053111	000105	DRMSG:	.ASCIZ /DRIVE/
3164	030770	046122	030461	000	CNTYPE:	.ASCIZ /RL11/
3165	030775				ENDMOD	
3166		030776				.EVEN
3167						
3168	030776				BGNMOD	SFTPRM
3169	030776				BGNSFT	
(3)	030776	000053				.WORD L10046-L\$SOFT/2

3170				
3172	031000		GPRML	CYLQ,MISWI,1,YES
(4)	031000	000130	.WORD	T\$CODE
(4)	031002	031126	.WORD	CYLQ
(4)	031004	000001	.WORD	1
3173	031006		GPRML	SECO,MISWI,2,YES
(4)	031006	000130	.WORD	T\$CODE
(4)	031010	031142	.WORD	SECO
(4)	031012	000002	.WORD	2
3179				
3181	031014		GPRML	LOLIMQ,MISWI,40000,YES
(4)	031014	000130	.WORD	T\$CODE
(4)	031016	031157	.WORD	LOLIMQ
(4)	031020	040000	.WORD	40000
3182	031022		XFERF	1\$
(5)	031022	006044	.WORD	T\$CODE
3183	031024		GPRMD	LIMVAL,LOLIM,D,255.,0,253.,YES
(4)	031024	001052	.WORD	T\$CODE
(4)	031026	031176	.WORD	LIMVAL
(4)	031030	000377	.WORD	255.
(4)	031032	000000	.WORD	T\$LOLIM
(4)	031034	000375	.WORD	T\$HILIM
3184	031036		1\$: GPRML	HILIMQ,MISWI,20000,YES
(4)	031036	000130	.WORD	T\$CODE
(4)	031040	031204	.WORD	HILIMQ
(4)	031042	020000	.WORD	20000
3185	031044		XFERF	2\$
(5)	031044	006044	.WORD	T\$CODE
3186	031046		GPRMD	LIMVAL,HILIM,D,255.,0,255.,YES
(4)	031046	002052	.WORD	T\$CODE
(4)	031050	031176	.WORD	LIMVAL
(4)	031052	000377	.WORD	255.
(4)	031054	000000	.WORD	T\$LOLIM
(4)	031056	000377	.WORD	T\$HILIM
3187	031060		2\$: GPRML	HEADQ,MISWI,10000,YES
(4)	031060	000130	.WORD	T\$CODE
(4)	031062	031225	.WORD	HEADQ
(4)	031064	010000	.WORD	10000
3188	031066		XFERF	3\$
(5)	031066	006044	.WORD	T\$CODE
3189	031070		GPRMD	HEADV,HEAD,D,17,0,1,YES
(4)	031070	003052	.WORD	T\$CODE
(4)	031072	031247	.WORD	HEADV
(4)	031074	000017	.WORD	17
(4)	031076	000000	.WORD	T\$LOLIM
(4)	031100	000001	.WORD	T\$HILIM
3191	031102		3\$: GPRMD	ERLIMQ,ERLIM,D,377,0,377,YES
(4)	031102	004052	.WORD	T\$CODE
(4)	031104	031277	.WORD	ERLIMQ
(4)	031106	000377	.WORD	377
(4)	031110	000000	.WORD	T\$LOLIM
(4)	031112	000377	.WORD	T\$HILIM
3193	031114		GPRMD	DCLIMQ,DCLIM,D,377,1,377,YES
(4)	031114	005052	.WORD	T\$CODE
(4)	031116	031314	.WORD	DCLIMQ
(4)	031120	000377	.WORD	377

(4)	031122	000001				.WORD	T\$LOLIM	
(4)	031124	000377				.WORD	T\$HILIM	
3195	031126				ENDSFT			
(2)						.EVEN		
(3)	031126				L10046:			
3196								
3198	031126	051525	020105	046101	CYLQ:	.ASCIZ	/USE ALL CYL/	
	031134	020114	054503	000114				
3199	031142	051525	020105	046101	SECO:	.ASCIZ	/USE ALL SECT/	
	031150	020114	042523	052103				
	031156	000						
3206	031157	114	053517	051440	LOLIMQ:	.ASCIZ	/LOW SEEK LIMIT/	
	031164	042505	020113	044514				
	031172	044515	000124					
3207	031176	040526	052514	000105	LIMVAL:	.ASCIZ	/VALUE/	
3208	031204	050125	042520	020122	HILIMQ:	.ASCIZ	/UPPER SEEK LIMIT/	
	031212	042523	045505	046040				
	031220	046511	052111	000				
3209	031225	125	042523	047440	HEADQ:	.ASCIZ	/USE ONLY ONE SURF/	
	031232	046116	020131	047117				
	031240	020105	052523	043122				
	031246	000						
3210	031247	127	040510	020124	HEADV:	.ASCIZ	/WHAT SURF (0 OR 1)/	
	031254	052523	043122	024040				
	031262	020060	051117	030440				
	031270	000051						
3212	031272	047111	052520	020124	ERLIMQ:	.ASCIZ	/INPUT ERROR LIMIT/	
	031300	051105	047522	020122				
	031306	044514	044515	000124				
3214	031314	040504	040524	041440	DCLIMQ:	.ASCIZ	/DATA CMP ERR LMT/	
	031322	050115	042440	051122				
	031330	046040	052115	000				
3216		031336				.EVEN		
3217	031336				ENDMOD			
3218								
3219	031336				LASTAD			
(2)						.EVEN		
(4)	031336	000000				.WORD	0	
(4)	031340	000000				.WORD	0	
(3)	031342				L\$LAST::			
3220								
3221		000001			.END			

VECT = 000002	95#	3152												
VERHDR 021460	2067#	2755	2802	2852	2951									
VERPOS 022466	2236#	2748	2795	2845	2897	2945	3006	3061	3120					
WAITIN 016056	1506#	1632	1737	2041	2299	2481								
WCMSK = 017777	191#													
WCRNG = 160000	192#													
WDESTA= 100000	211#	1566												
WGESTA= 002000	206#													
WLSTAT= 020000	209#													
WRTSWI 003036	428#													
WTDATA= 000112	124#	2423												
XDELAY 003466	518#	1012*	1420	1421*	1425*	1430*	1477*	1597*	1617*	1734*	1907*	2013*	2170*	
	2478*	2692*	2891*											
XRDHD 021024	1983#	2213												
XRDHDC 021014	1981#													
XRDHDG 021030	1982	1984#												
XREAD 023646	2430#													
XREADG 023654	2429	2431#												
XSEEK 017160	1659#	1846	2679	2742	2789	2840	2878	2940	3001	3040	3056	3099	3115	
XSEEKT 017150	1657#													
XSEEK1 017164	1658	1660#												
XTIME 015422	1328	1396	1436#	1518	1558	1574	1609	1918	2179					
XWRITE 023606	2422#													
XWRIT1 023576	2420#													
XWRIT1 023612	2421	2423#												
X\$ALWA= 000000	8#													
X\$FALS= 000040	8#	3182	3185	3188										
X\$OFFS= 000400	8#	3182	3185	3188										
X\$TRUE= 000020	8#													
YDELAY 003470	519#	1328*	1396*	1438	1439*	1440*	1441*	1446*	1450*	1477*	1518*	1558*	1574*	
	1609*	1918*	2179*											
	5#	66#	330#	331#	511#	532#	533#	535#	536#	869#	1424	1429	1445	
	1449	2706	2726	2774	2827	2901	2927	3010	3072	3131	3166#	3182	3185	
	3188	3216#												

- 031342

MSBYTE	63#														
MSCHEC	2726#	2774#	2827#	2927#											
MSCNTO	3150#	3151#	3152#	3153#	3154#	3155#	3172#	3173#	3181#	3183#	3184#	3186#	3187#	3189#	3191#
	3193#														
MSCOUN	1033#	1066#	1080#	1089#	1152#	1153#	1155#	1331#	1332#	1333#	1360#	1361#	1363#	1372#	1374#
	1376#	1491#	1492#	1493#	1972#	1973#	1974#	2395#	2563#	2564#	2568#	2581#	2597#	2601#	2605#
	2608#	2620#	2629#	2630#	2633#	2641#	2643#	2644#	2645#						
MSDATA	63#	66#	67#												
MSDECR	65#	218#	634#	866#	932#	946#	960#	975#	990#	1095#	1109#	1131#	1145#	1159#	1160#
	1167#	1179#	1180#	1196#	1197#	1206#	1338#	1339#	1379#	1404#	1408#	1410#	1463#	1479#	1877#
	2659#	2705#	2713#	2761#	2763#	2808#	2810#	2858#	2860#	2900#	2908#	2957#	2959#	3009#	3023#
	3071#	3082#	3130#	3140#	3142#	3157#	3165#	3195#	3217#						
MSDEFA	3150#	3151#	3152#	3153#	3154#	3155#	3172#	3173#	3181#	3183#	3184#	3186#	3187#	3189#	3191#
	3193#														
MSENDE	65#	218#	634#	866#	932#	946#	960#	975#	990#	1095#	1109#	1131#	1145#	1159#	1160#
	1179#	1180#	1196#	1197#	1206#	1338#	1339#	1379#	1404#	1408#	1410#	1463#	1479#	1877#	2659#
	2705#	2713#	2761#	2763#	2808#	2810#	2858#	2860#	2900#	2908#	2957#	2959#	3009#	3023#	3071#
	3082#	3130#	3140#	3142#	3157#	3165#	3195#	3217#							
MSERRI	1578#	1615#	1628#	1634#	1739#	1744#	1853#	1869#	1912#	1923#	2020#	2031#	2035#	2043#	2092#
	2104#	2111#	2173#	2183#	2187#	2248#	2301#	2306#	2410#	2468#	2483#	2489#	2498#	2502#	2688#
	2702#	2887#													
MSERSCA	2706#	2901#	3010#	3072#	3131#										
MSERSCS	2706#	2901#	3010#	3072#	3131#										
MSEXCP	3151#	3152#	3153#	3155#	3183#	3186#	3189#	3191#	3193#						
MSEXIT	2726#	2774#	2827#	2927#											
MSXSE	2726#	2774#	2827#	2927#											
MSXTJ	2726#	2774#	2827#	2927#											
MSGEN	58#	63#	66#	67#	90#	224#	645#	875#	920#	932#	934#	946#	948#	960#	962#
	975#	977#	990#	992#	1095#	1097#	1109#	1111#	1131#	1133#	1145#	1146#	1159#	1163#	1171#
	1172#	1179#	1182#	1183#	1196#	1199#	1204#	1212#	1213#	1338#	1352#	1379#	1385#	1386#	1406#
	1406#	1408#	1415#	1457#	1463#	1465#	1479#	1877#	2666#	2667#	2677#	2705#	2713#	2718#	2730#
	2761#	2763#	2766#	2778#	2808#	2810#	2813#	2829#	2858#	2860#	2866#	2876#	2900#	2908#	2913#
	2929#	2957#	2959#	2962#	2993#	3009#	3023#	3026#	3048#	3071#	3082#	3085#	3107#	3130#	3140#
	3148#	3149#	3157#	3168#	3169#	3195#	3219#								
MSGETS	65#	218#	634#	866#	932#	946#	960#	975#	990#	1095#	1109#	1131#	1145#	1159#	1160#
	1167#	1179#	1180#	1196#	1197#	1206#	1338#	1339#	1379#	1404#	1408#	1410#	1463#	1479#	1877#
	2659#	2705#	2713#	2761#	2763#	2808#	2810#	2858#	2860#	2900#	2908#	2957#	2959#	3009#	3023#
	3071#	3082#	3130#	3140#	3142#	3157#	3165#	3182#	3185#	3188#	3195#	3217#			
MSGETT	2706#	2726#	2774#	2827#	2901#	2927#	3010#	3072#	3131#	3182#	3185#	3188#			
MSGNGB	58#	63#	66#	67#	90#	224#	645#	875#	920#	934#	948#	962#	977#	992#	1097#
	1111#	1133#	1146#	1163#	1171#	1172#	1182#	1183#	1199#	1204#	1212#	1213#	1352#	1385#	1386#
	1406#	1415#	1457#	1465#	2666#	3148#	3149#	3168#	3169#	3219#					
MSGNIN	63#	66#	67#	932#	946#	960#	975#	990#	1033#	1066#	1080#	1089#	1095#	1109#	1131#
	1145#	1152#	1153#	1155#	1159#	1172#	1183#	1204#	1217#	1218#	1220#	1221#	1225#	1226#	1229#
	1230#	1250#	1251#	1253#	1254#	1256#	1257#	1270#	1271#	1303#	1304#	1331#	1332#	1333#	1334#
	1335#	1338#	1354#	1360#	1361#	1363#	1365#	1372#	1374#	1376#	1377#	1378#	1379#	1388#	1390#
	1397#	1401#	1402#	1404#	1408#	1422#	1423#	1424#	1429#	1442#	1443#	1445#	1449#	1463#	1479#
	1489#	1490#	1491#	1492#	1493#	1494#	1495#	1578#	1615#	1628#	1634#	1739#	1744#	1830#	1831#
	1832#	1853#	1869#	1877#	1912#	1923#	1972#	1973#	1974#	2020#	2031#	2035#	2043#	2092#	2104#
	2111#	2173#	2183#	2187#	2248#	2301#	2306#	2395#	2410#	2468#	2483#	2489#	2498#	2502#	2563#
	2564#	2568#	2581#	2597#	2601#	2605#	2608#	2620#	2629#	2630#	2633#	2641#	2643#	2644#	2645#
	2677#	2688#	2702#	2705#	2706#	2713#	2726#	2730#	2733#	2734#	2761#	2763#	2774#	2778#	2781#
	2782#	2808#	2810#	2827#	2829#	2831#	2832#	2858#	2860#	2876#	2887#	2900#	2901#	2908#	2927#
	2929#	2932#	2933#	2957#	2959#	2993#	2994#	2995#	3009#	3010#	3023#	3048#	3049#	3050#	3071#
	3072#	3082#	3107#	3108#	3109#	3130#	3131#	3140#	3149#	3150#	3151#	3152#	3153#	3154#	3155#
	3169#	3172#	3173#	3181#	3182#	3183#	3184#	3185#	3186#	3187#	3188#	3189#	3191#	3193#	

MSGNLS	3195#	3219#													
MSGNSU	1877#														
MSGNTA	2677#	2730#	2778#	2829#	2876#	2929#	2993#	3048#	3107#						
	932#	946#	960#	975#	990#	1095#	1109#	1131#	1145#	1159#	1179#	1196#	1338#	1379#	1404#
	1408#	1463#	1479#	2705#	2713#	2761#	2763#	2808#	2810#	2858#	2860#	2900#	2908#	2957#	2959#
	3009#	3023#	3071#	3082#	3130#	3140#	3157#	3195#							
MSGNTE	2667#	2718#	2766#	2813#	2866#	2913#	2962#	3026#	3085#						
MSHAPT	63#														
MSHAP	63#														
MSINCR	58#	90#	224#	645#	875#	920#	932#	934#	946#	948#	960#	962#	975#	977#	990#
	992#	1033#	1066#	1080#	1089#	1095#	1097#	1109#	1111#	1131#	1133#	1145#	1146#	1152#	1153#
	1155#	1159#	1163#	1171#	1172#	1182#	1183#	1199#	1212#	1213#	1217#	1220#	1221#	1225#	1229#
	1250#	1253#	1256#	1270#	1303#	1304#	1331#	1332#	1333#	1334#	1335#	1338#	1352#	1354#	1360#
	1361#	1363#	1365#	1372#	1374#	1376#	1377#	1378#	1379#	1385#	1386#	1388#	1390#	1397#	1401#
	1402#	1404#	1406#	1408#	1415#	1422#	1442#	1457#	1465#	1489#	1491#	1492#	1493#	1494#	1495#
	1578#	1615#	1628#	1634#	1739#	1744#	1830#	1831#	1853#	1869#	1877#	1912#	1923#	1972#	1973#
	1974#	2020#	2031#	2035#	2043#	2092#	2104#	2111#	2173#	2183#	2187#	2248#	2301#	2306#	2395#
	2410#	2468#	2483#	2489#	2498#	2502#	2563#	2564#	2568#	2581#	2597#	2601#	2605#	2608#	2620#
	2529#	2630#	2633#	2641#	2643#	2644#	2645#	2666#	2667#	2677#	2688#	2702#	2705#	2706#	2713#
	2718#	2726#	2730#	2733#	2761#	2763#	2766#	2774#	2778#	2781#	2808#	2810#	2813#	2827#	2829#
	2831#	2858#	2860#	2866#	2876#	2887#	2900#	2901#	2908#	2913#	2927#	2929#	2932#	2957#	2959#
	2962#	2993#	2994#	3009#	3010#	3023#	3026#	3048#	3049#	3071#	3072#	3082#	3085#	3107#	3108#
	3130#	3131#	3140#	3148#	3149#	3168#	3169#								
MSLDRO	1217#	1220#	1225#	1229#	1250#	1253#	1256#	1270#	1304#	1334#	1365#	1377#	1378#	1390#	1397#
	1401#	1494#													
MSMCHI	8#														
MSMCLO	8#														
MSPOP	65#	218#	634#	866#	932#	946#	960#	975#	990#	1095#	1109#	1131#	1145#	1159#	1160#
	1167#	1179#	1180#	1196#	1197#	1206#	1338#	1339#	1379#	1404#	1408#	1410#	1463#	1479#	1877#
	2659#	2705#	2713#	2761#	2763#	2808#	2810#	2858#	2860#	2900#	2908#	2957#	2959#	3009#	3023#
	3071#	3082#	3130#	3140#	3142#	3157#	3165#	3195#	3217#						
MSPRIN	1033#	1066#	1080#	1089#	1152#	1153#	1155#	1331#	1332#	1333#	1360#	1361#	1363#	1372#	1374#
	1376#	1491#	1492#	1493#	1972#	1973#	1974#	2395#	2563#	2564#	2568#	2581#	2597#	2601#	2605#
	2608#	2620#	2629#	2630#	2633#	2641#	2643#	2644#	2645#						
MSPUSH	58#	90#	224#	645#	875#	920#	934#	948#	962#	977#	992#	1097#	1111#	1133#	1146#
	1163#	1171#	1172#	1182#	1183#	1199#	1212#	1213#	1352#	1385#	1386#	1406#	1415#	1457#	1465#
	1830#	2666#	2667#	2677#	2718#	2730#	2766#	2778#	2813#	2829#	2866#	2876#	2913#	2929#	2962#
	2993#	3026#	3048#	3085#	3107#	3148#	3149#	3168#	3169#						
MSPUT	1033#	1066#	1080#	1089#	1152#	1153#	1155#	1303#	1331#	1332#	1333#	1354#	1360#	1361#	1363#
	1372#	1374#	1376#	1388#	1491#	1492#	1493#	1972#	1973#	1974#	2395#	2563#	2564#	2568#	2581#
	2597#	2601#	2605#	2608#	2620#	2629#	2630#	2633#	2641#	2643#	2644#	2645#	2645#	2645#	2645#
MSPUT1	1033#	1066#	1080#	1089#	1152#	1153#	1155#	1303#	1331#	1332#	1333#	1354#	1360#	1361#	1363#
	1372#	1374#	1376#	1388#	1491#	1492#	1493#	1972#	1973#	1974#	2395#	2563#	2564#	2568#	2581#
	2597#	2601#	2605#	2608#	2620#	2629#	2630#	2633#	2641#	2643#	2644#	2645#	2645#	2645#	2645#
MSRADI	3150#	3151#	3152#	3153#	3154#	3155#	3172#	3173#	3181#	3183#	3184#	3186#	3187#	3189#	3191#
	3193#														
MSRNRO	1217#	1270#													
MSSETS	58#	90#	224#	645#	875#	920#	934#	948#	962#	977#	992#	1097#	1111#	1133#	1146#
	1163#	1171#	1172#	1182#	1183#	1199#	1212#	1213#	1352#	1385#	1386#	1406#	1415#	1457#	1465#
	1830#	2666#	2667#	2677#	2718#	2730#	2766#	2778#	2813#	2829#	2866#	2876#	2913#	2929#	2962#
	2993#	3026#	3048#	3085#	3107#	3148#	3149#	3168#	3169#						
MS SVC	932#	946#	960#	975#	990#	1033#	1066#	1080#	1089#	1095#	1109#	1131#	1145#	1152#	1153#
	1155#	1159#	1217#	1220#	1221#	1225#	1229#	1250#	1253#	1256#	1270#	1303#	1304#	1331#	1332#
	1333#	1334#	1335#	1338#	1354#	1360#	1361#	1363#	1365#	1372#	1374#	1376#	1377#	1378#	1379#
	1388#	1390#	1397#	1401#	1402#	1404#	1408#	1422#	1442#	1489#	1491#	1492#	1493#	1494#	1495#
	1578	1615	1628	1634	1739	1744	1830#	1831#	1853	1869	1877#	1912	1923	1972#	1973#

	1974#	2020	2031	2035	2043	2092	2104	2111	2173	2183	2187	2248	2301	2306	2395#
	2410	2468	2483	2489	2498	2502	2563#	2564#	2568#	2581#	2597#	2601#	2605#	2608#	2620#
	2629#	2630#	2633#	2641#	2643#	2644#	2645#	2677#	2688	2702	2705#	2706#	2713#	2726#	2730#
	2733#	2761#	2763#	2774#	2778#	2781#	2808#	2810#	2827#	2829#	2831#	2858#	2860#	2876#	2887
	2900#	2901#	2908#	2927#	2929#	2932#	2957#	2959#	2993#	2994#	3009#	3010#	3023#	3048#	3049#
	3071#	3072#	3082#	3107#	3108#	3130#	3131#	3140#							
MSSTLAB	932#	946#	960#	975#	990#	1033#	1066#	1080#	1089#	1095#	1109#	1131#	1145#	1152#	1153#
	1155#	1159#	1217#	1220#	1221#	1225#	1229#	1250#	1253#	1256#	1270#	1303#	1304#	1331#	1332#
	1333#	1334#	1335#	1338#	1354#	1360#	1361#	1363#	1365#	1372#	1374#	1376#	1377#	1378#	1379#
	1388#	1390#	1397#	1401#	1402#	1404#	1408#	1422#	1442#	1489#	1491#	1492#	1493#	1494#	1495#
	1578#	1615#	1628#	1634#	1739#	1744#	1830#	1831#	1853#	1869#	1877#	1912#	1923#	1972#	1973#
	1974#	2020#	2031#	2035#	2043#	2092#	2104#	2111#	2173#	2183#	2187#	2248#	2301#	2306#	2395#
	2410#	2468#	2483#	2489#	2498#	2502#	2563#	2564#	2568#	2581#	2597#	2601#	2605#	2608#	2620#
	2629#	2630#	2633#	2641#	2643#	2644#	2645#	2677#	2688#	2702#	2705#	2706#	2713#	2726#	2730#
	2733#	2761#	2763#	2774#	2778#	2781#	2808#	2810#	2827#	2829#	2831#	2858#	2860#	2876#	2887#
	2900#	2901#	2908#	2927#	2929#	2932#	2957#	2959#	2993#	2994#	3009#	3010#	3023#	3048#	3049#
	3071#	3072#	3082#	3107#	3108#	3130#	3131#	3140#							
MSSTSL	932#	946#	960#	975#	990#	1033#	1066#	1080#	1089#	1095#	1109#	1131#	1145#	1152#	1153#
	1155#	1159#	1217#	1220#	1221#	1225#	1229#	1250#	1253#	1256#	1270#	1303#	1304#	1331#	1332#
	1333#	1334#	1335#	1338#	1354#	1360#	1361#	1363#	1365#	1372#	1374#	1376#	1377#	1378#	1379#
	1388#	1390#	1397#	1401#	1402#	1404#	1408#	1422#	1442#	1489#	1491#	1492#	1493#	1494#	1495#
	1578#	1615#	1628#	1634#	1739#	1744#	1830#	1831#	1853#	1869#	1877#	1912#	1923#	1972#	1973#
	1974#	2020#	2031#	2035#	2043#	2092#	2104#	2111#	2173#	2183#	2187#	2248#	2301#	2306#	2395#
	2410#	2468#	2483#	2489#	2498#	2502#	2563#	2564#	2568#	2581#	2597#	2601#	2605#	2608#	2620#
	2629#	2630#	2633#	2641#	2643#	2644#	2645#	2677#	2688#	2702#	2705#	2706#	2713#	2726#	2730#
	2733#	2761#	2763#	2774#	2778#	2781#	2808#	2810#	2827#	2829#	2831#	2858#	2860#	2876#	2887#
	2900#	2901#	2908#	2927#	2929#	2932#	2957#	2959#	2993#	2994#	3009#	3010#	3023#	3048#	3049#
	3071#	3072#	3082#	3107#	3108#	3130#	3131#	3140#							
MSWORD	63#	1204#	1578#	1615#	1628#	1634#	1739#	1744#	1853#	1869#	1912#	1923#	2020#	2031#	2035#
	2043#	2092#	2104#	2111#	2173#	2183#	2187#	2248#	2301#	2306#	2410#	2468#	2483#	2489#	2498#
	2502#	2688#	2702#	2726#	2774#	2827#	2887#	2927#	3150#	3151#	3152#	3153#	3154#	3155#	3172#
	3173#	3181#	3182#	3183#	3184#	3185#	3186#	3187#	3188#	3189#	3191#	3193#	3219		
MSXFER	3182#	3185#	3188#												
POINTE	56														
PRINTB	1033	1066	1080	1089	1152	1153	1155	2395	2563	2564	2568	2581	2597	2601	2605
	2608	2620	2629	2630	2633	2641	2643	2644	2645						
PRINTF	1331	1332	1333	1360	1361	1363	1372	1374	1376	1491	1492	1493	1972	1973	1974
READBU	1422	1442													
REDEF	1225	1229	1250	1253	1256										
SETPRI	1220	1304	1390												
SETVEC	1303	1354	1388												
STCLK	42#														
SVC	6#	8													
WAITMS	25#	1328	1396	1518	1558	1574	1609	1918	2179						
WAITUS	20#	1012	1597	1617	1734	1907	2013	2170	2478	2692	2891				
XFER	2726#	2774#	2827#	2927#											
XFERF	3182	3185	3188												

. ABS. 031342 000

ERRORS DETECTED: 0

CZRLJB,CZRLJB/CRF=SVC33/ML,CZRLJB.MAC
 RUN-TIME: 84 89 10 SECONDS

CZRLJBO RLO1/02 DRIVE TEST 2
CZRLJB.MAC 07-DEC-79 09:06

MACY11 30A(1052) 08-FEB-80 14:49 K11 PAGE 5-4
CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0140

RUN-TIME RATIO: 485/184=2.6
CORE USED: 16K (31 PAGES)