

1
2
3
4
5
6
7
8
9

.REM 8

10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51

IDENTIFICATION

PRODUCT CODE: AC-F843C-MC
PRODUCT NAME: CZRLNCO RL01/02 DRIVE TEST 3
DATE CREATED: 05-JAN-1979
REVISED: 06-JAN-1986
MAINTAINER: CXO DIAGNOSTIC ENGINEERING
AUTHORS: D. DEKNIS, C. CAMPBELL
REVISED BY: M. LEAVITT

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

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

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

COPYRIGHT (C) 1979, 1983, 1986 DIGITAL EQUIPMENT CORPORATION

C1

CZRLNCO RL01/02 DRIVE TEST 3 MACRO V05.03b Monday 06-Jan-86 00:23 Page 2

SEQ 0002

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

Page 2

HISTORY

AUTHOR: DAN DEKNIS 05-JAN-1979 VERSION A0

MODIFIED BY:

CHUCK CAMPBELL 1983 VERSION B0
MIKE LEAVITT 06-JAN-1986 VERSION C0

B0 Problem:

Unknown

Solution:

Unknown

C0 Problem:

Prism Report PR00486. Diagnostic will not read Bad Sector File if TEST 2 is not included in test sequence.

Solution:

All tests in the diagnostic which require the bad sec file data, will test to see if the bad sec file had previously been read. If not, the test will read the bad sector file before executing the desired test sequence.

D1

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52

Page 3

TABLE OF CONTENTS

1.0	GENERAL INFORMATION
1.1	PROGRAM ABSTRACT
1.1.1	STRUCTURE OF PROGRAM
1.1.2	DIAGNOSTIC INFORMATION
1.1.3	DIAGNOSTIC RUN TIME
1.2	SYSTEM REQUIREMENTS
1.2.1	HARDWARE REQUIREMENTS
1.2.2	SOFTWARE REQUIREMENTS
1.3	RELATED DOCUMENTS AND STANDARDS
1.4	DIAGNOSTIC HIERARCHY PREREQUISITES
1.5	ASSUMPTIONS
2.0	OPERATING INSTRUCTIONS
2.1	HOW TO RUN THIS DIAGNOSTIC
2.1.1	THE FIVE STEPS OF EXECUTION
2.1.2	SAF FILE RUN-THROUGH
2.2	CHAIN MODE OPERATION
2.3	DETAILS OF COMMANDS AND SYNTAX
2.3.1	TABLE OF COMMAND VALIDITY
2.3.2	COMMAND SYNTAX
2.4	EXTENDED P-TABLE DIALOGUE
2.5	HARDWARE PARAMETERS
2.6	SOFTWARE PARAMETERS
3.0	ERROR INFORMATION
3.1	ERROR REPORTING
3.1.1	SPECIFIC RESULT MESSAGES
3.1.2	OTHER MESSAGES
3.1.3	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

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

Page 4

1.0 GENERAL INFORMATION

1.1 PROGRAM ABSTRACT

1.1.1 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 DIAGNOSTIC PROGRAM INTERFACES TO THE ENVIRONMENT AS IT EXECUTES.

WHEN THIS DIAGNOSTIC IS STARTED AT ADDRESS 200, CONTROL GOES FIRST TO THE SUPERVISOR PORTION, WHICH WILL ASK CERTAIN "HARD CORE" QUESTIONS ABOUT THE ENVIRONMENT. THEN IT WILL ENTER COMMAND MODE INDICATED BY A PROMPT CHARACTER (DR>). AT COMMAND MODE THE OPERATOR MAY ENTER ANY OF SEVERAL COMMANDS AS DESCRIBED IN 2.0 "OPERATING INSTRUCTIONS".

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

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

1

Page 5

2

3

4

1.1.3 DIAGNOSTIC RUN TIME

5

6

7

8

9

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

10

11

12

13

14

15

16

1.2 SYSTEM REQUIREMENTS

17

18

19

20

21

22

23

24

25

26

27

1.2.1 HARDWARE REQUIREMENTS

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

- * 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'

- * KW11-P CLOCK (REQUIRED TO PERFORM TESTS 1 AND 4)

- * LINE PRINTER (OPTIONAL)

1.2.2 SOFTWARE REQUIREMENTS

CZRLJ RL01/02 DRIVE TEST PART 2

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:

CVRLA	RLV11 RL01 DISKLESS TEST (RLV11 ONLY)
CZRLG	RL11/RLV11 RL01/02 CONTROLLER TEST (PART 1)
CZRLH	RL11/RLV11 RL01/02 CONTROLLER TEST (PART 2)
CZRRI	RL01/02 DRIVE TEST (PART 1)

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

Page 6

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 DIAGNOSTIC

2.1.1 THE FIVE STEPS OF EXECUTION

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

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

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:

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

Page 7

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

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

Page 8

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

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

Page 9

NO ERROR HAS 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.

Page 10

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

1. START
2. RESTART
3. CONTINUE

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

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

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

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

1. START
2. RESTART

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

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

BY
WHOM
ENTERED:

45 .R CZRLN??	O
46 DRS LOADED	D
47 DIAG. RUN-TIME SERVICES REV. x mmm-yy	D
48 CZRLN-?-?	D
49 CZRLN TESTS SEEK AND ROTATIONAL	D
50 TIMING & WRITE & READ DATA	D
51 UNIT IS RL01, RL02	D
52 DR>STA/PASS:1/FLAGS:HOE	D,O
53	
54 # UNITS (D) ? 2	D,O
55	
56 UNIT O	D
57 RL11 (L) Y ?	D,O

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

BUS ADDRESS (0) 174400 ?	D,0
VECTOR (0) 160 ?	0,0
DRIVE (0) 0 ?	D,0
DRIVE TYPE = RL01 (L) Y ?	D,0
BR LEVEL (0) 5 ?	D,0
UNIT 1	0
RL11 (L) Y ?	0,0
BUS ADDRESS (0) 174400 ?	0,0
VECTOR (0) 160 ?	0,0
DRIVE (0) 0 ? 1	0,0
DRIVE TYPE = RL01 (L) ? N	0,0 (N=RL02)
BR LEVEL (0) 5 ?	0,0
CHANGE SW (L) ? Y	D,0
USE ALL CYL (L) N ?	0,0
USE ALL SECT (L) N ?	0,0
DO MANUAL INTERVENTION TEST (L) N ?	0,0
LOW SEEK LIMIT (L) N ?	0,0
UPPER SEEK LIMIT (L) N ?	0,0
USE ONLY ONE SURF (L) N ?	0,0
INPUT ERROR LIMIT (D) 20 ?	0,0
DATA CMP ERR LMT (D) 10 ?	0,0
PRINT ERRORS DETECTED WHILE READING BAD SEC FILE (N) ? 0,0	
CZRLN HRD ERR 00004 TST 003 SUB 002 PC:004130	
ERR HLT	
DR>PRO/FLAGS:IER:LOE:HOE=0	D,0
***** AT THIS POINT THE DIAGNOSTIC IS LOOPING ON THE ERROR WITHOUT PRINTING ANYTHING. YOU CAN SCOPE THE ERROR UNTIL YOU HAVE LOCATED IT, THEN †C OUT *****	
†C	0
DR>CON/FLAGS:HOE:IER:LOE=0	D,0
CHANGE SW (L) ? N	0,0
CZRLN EOP 1	D
†C	
DR>RESTART/PASS:1	D,0
CHANGE SW (L) ? N	D,0

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

2.2

CHAIN MODE OPERATION

Page 12

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

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

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

TO EXECUTE A CHAIN FILE THE USER TYPES:

C FILNAM <CR> OR
C FILNAM/QV <CR>

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

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

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

2.3

DETAILS OF COMMANDS AND SYNTAX

2.3.1

TABLE OF COMMAND VALIDITY

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
Page 13

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

HOW ENTERED	LEGAL COMMANDS
1. OPERATOR ENTERED 'RUN DIAG'	START PRINT DISPLAY FLAGS ZFLAGS EXIT
2. DIAGNOSTIC HAS FINISHED ALL ITS REQUESTED PASSES	START RESTART PRINT DISPLAY FLAGS ZFLAGS EXIT
3. OPERATOR INTERRUPTED THE	START PRINT DISPLAY FLAGS ZFLAGS EXIT
4. AN ERROR WAS ENCOUNTERED WITH THE HDE 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

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
Page 14

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

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

Page 15

LOT LOOP ON TEST

EVL EVALUATE

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

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

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

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

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

THE SWITCH ARGUMENTS ARE AS IN THE START COMMAND EXCEPT:

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

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

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

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
Page 16

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)

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

Page 17

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 (\$

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

Page 18

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 RL01'S AND THE LAST 4 DRIVES ARE RL02'S (ON THE SECOND CONTROLLER):

UNITS (D) ? 8

UNIT 0

RL11 (L) Y ?

BUS ADDRESS (0) 174400 ?

VECTOR (0) 160 ?

DRIVE (0) 0 ? 0-3

DRIVE TYPE = RL01 (L) Y ?

BR LEVEL (0) 5 ?

UNIT 4

RL11 (L) Y ?

BUS ADDRESS (0) 174400 ? 175400

VECTOR (0) 160 ? 164

DRIVE (0) 0 ? 0-3

DRIVE TYPE = RL01 (L) Y ? N

BR LEVEL (0) 5 ?

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

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

2.5 HARDWARE PARAMETERS

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

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56

Page 19

RL11 (L) Y?

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

BUS ADDRESS (0) 174400?

ANSWER WITH THE BUS ADDRESS OF THE CONTROLLER.

VECTOR (0) 160?

ANSWER WITH THE INTERRUPT VECTOR OF THE CONTROLLER.

DRIVE (0) 0?

ANSWER WITH THE DRIVE(S) CONNECTED TO THE CONTROLLER

DRIVE TYPE = RL01 (L) ?

ANSWER NO (N) IF DRIVE IS AN RL02

BR LEVEL (0) 5?

ANSWER WITH THE INTERRUPT PRIORITY OF THE CONTROLLER.

2.6 SOFTWARE PARAMETERS

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

CHANGE S.W. ?

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

USE ALL CYLINDERS (N)?

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

USE ALL SECTORS (N)?

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

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

Page 20

EXECUTE MANUAL INTERVENTION TESTS (N)?

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

LOWER SEEK LIMIT (N)?

IF "YES", THE NEXT PARAMETER IS REQUESTED.

ENTER VALUE (DECIMAL) (0)?

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

UPPER SEEK LIMIT (N)?

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

ENTER VALUE (DECIMAL) (255)?

USE ONLY ONE SURFACE (N)?

IF "YES", THE NEXT PARAMETER IS REQUESTED.

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

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

SPECIFY ERROR LIMIT (DECIMAL) (20)?

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

DATA COMPARE ERROR LIMIT (DECIMAL) (20)?

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

12
1
2
3
4
5
6
7
Page 21

TOTAL IS REPORTED AT THE END OF THE COMPARE.

PRINT ERRORS DETECTED WHILE READING BAD SEC FILE (N)?

IF "YES", ALL ERRORS DETECTED WHILE READING THE BAD SECTOR FILE, WILL BE PRINTED TO THE OUTPUT DEVICE. IF "NO" ONLY HARD ERRORS WILL BE PRINTED TO THE OUTPUT DEVICE. THIS IS USEFUL IF THE USER WISHES TO SEE WHAT ERRORS ARE DETECTED IN ANY BAD COPIES OF THE BAD SECTOR FILES.

14
15 3.0 ERROR INFORMATION
16
17

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.

18
19 3.1 ERROR REPORTING
20
21

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

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

Page 22

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.

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

Page 23

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

3.1.2 SPECIFIC RESULT MESSAGES

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

THE GENERAL FORMAT FOR THE RESULT LINE IS:

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

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

VARIABLE 2 WILL BE A VALUE THAT DEFINES WHAT THE RESULT
ACTUALLY IS. THIS CAN BE A 1 OR 0 TO INDICATE A SET OF RESULT

Page 24

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 HOME 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
ORV RDY IS 0 SB 1 IN STATE 5
ORV RDY IS 1 SB 0 IN SEEK W/O MOTION
DRV ROY IS 0 SB 1 IN 10MS
DRV ROY IS 0 SB 1 IN 500MS
ORV ROY 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 LOOSING READY BEFORE STARTING A READ HEADER AND THEREFORE NOT COMPLETING THE READ HEADER IN 1MS.

"FAIL TO RELOAD HEADS AFTER ERR CLEAR"

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

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

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

"WRITE ABORTED"

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

"COULD NOT RETRIEVE DRIVE STATUS"

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

"OPI SET-NO DRIVE RESPONSE"

THIS IS REPORTED AS THE RESULT WHEN THE GET STATUS COMMAND IS TIMED OUT (OPI SETS) WHEN THAT COMMAND IS BEING USED IN THE

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

Page 25

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:

"*WARNING* ALL SECTORS ASSUMED GOOD FOR TESTS REQUIRING BAD SEC DATA"

THIS MESSAGE IS PRINTED WHEN THE BAD SECTOR FILES COULD NOT BE READ OR IF DATA READ IS CORRUPT. THIS WARNING IS TO PRINTED TO LET THE USER KNOW THAT ANY ERRORS COULD BE A RESULT OF TESTING A KNOWN BAD SECTOR.

"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)

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

Page 26

(ADDRESS)

(ADDRESS)

(3) TEST DESCRIPTION
(4) OPERATION:
(5) RESULT:
(6) ADDRESS OF UNIT UNDER TEST RLCS RLDA RLBA RLMP CYL HD
(7)
(8) OP INIT
(9) OP DONE
(10) DRIVE STATUS
(11) WORD NUM IS (XXXXXX) SB (YYYYYY)
(12) TOTAL COMPARE ER'S: (ZZZ) OF (128)

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

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

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

THE ERROR PC IS THE PHYSICAL MEMORY LOCATION WHERE THE ERROR REPORT WAS INITIATED. SINCE MANY FUNCTIONS ARE SUBROUTINED AND ERRORS ARE REPORTED FROM SUBROUTINES, THE ERROR PC 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

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

Page 27

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.

Page 28

4.0 PERFORMANCE AND PROGRESS REPORTS

4.1 PERFORMANCE REPORTS

THIS PROGRAM WILL NOT GIVE ANY PERFORMANCE REPORTS.

4.2 PROGRESS REPORTS

THIS PROGRAM WILL NOT GIVE ANY PROGRESS REPORTS.

5.0 DEVICE INFORMATION TABLES

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

RLCS - CONTROL AND STATUS REGISTER (XXXXX0)

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

BIT 0 - DRIVE READY

RLBA - BUS ADDRESS REGISTER (XXXXX2)

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

Page 29

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

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

Page 30

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 RL02 IF SET
BIT 6 - SURFACE (0=UPPER, 1=LOWER)
BIT 5 - COVER OPEN
BIT 4 - HEADS HOME
BIT 3 - BRUSHES HOME
BIT 2-0 - STATE BITS
 0 - LOAD STATE
 1 - SPIN UP
 2 - BRUSH CYCLE
 3 - LOAD HEADS
 4 - SEEK - TRACK COUNTING
 5 - SEEK - LINEAR MODE
 6 - UNLOAD HEADS
 7 - SPIN DOWN

6.0 TEST SUMMARIES

TEST 1 SEEK TIMING

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

POSITION HEADS AT CYLINDER 0.

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

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

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

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

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

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

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

Page 31

MID CYL FWD	X			X	
MID CYL REV	X			X	X
MAX CYL FWD		X			X
MAX CYL REV		X			X

THE X INDICATES WHERE TIME WILL BE REPORTED.

TEST 2 BASIC READ DATA TEST

POSITION HEADS AT MAX CYLINDER (BAD SEC FILE).

DO READ DATA ON 1ST COPY OF THE FACTORY BAD SEC FILE (SECTORS 0 & 1, HEAD 1). IF AN ERROR IS DETECTED, PROCEED BY READING THE NEXT COPY OF THE FACTORY BAD SEC FILE UNTIL A GOOD COPY IS FOUND (SECTORS 4 & 5, 8 & 9, 12 & 13, 16 & 17). REPORT ALL ERRORS IF BAD SEC FILE ERROR REPORTING IS ON (SEE SW QUESTIONS), BUT DO NOT INCREMENT ERROR COUNT. IF NO GOOD COPIES CAN BE FOUND, REPORT THAT FACTORY BAD SECTOR FILE CANNOT BE READ, INCREMENT ERROR COUNT AND PROCEED WITH READING FIELD BAD SEC FILE AT SECTOR 20.

DO READ DATA ON 1ST COPY OF THE FIELD BAD SEC FILE (SECTORS 20 & 21, HEAD 1). IF AN ERROR IS DETECTED, PROCEED BY READING THE NEXT COPY OF THE FIELD BAD SEC FILE UNTIL A GOOD COPY IS FOUND (SECTORS 24 & 25, 28 & 29, 32 & 33, 36 & 37). REPORT ALL ERRORS IF BAD SEC FILE ERROR REPORTING IS ON (SEE SW QUESTIONS), BUT DO NOT INCREMENT ERROR COUNT. IF NO GOOD COPIES CAN BE FOUND, REPORT THAT FIELD BAD SECTOR FILE CANNOT BE READ, INCREMENT ERROR COUNT AND EXIT.

UPON FINDING A GOOD COPY OF THE BAD SEC FILE, VERIFY DATA FORMAT (WORD 0 & 1 ARE NOT 0 & NOT NEGATIVE, WORD 2 & 3 ARE 0, WORD 254 & 255 ARE ALL ONE'S LOCATE 1ST WORD OF ALL ONE'S & MAKE SURE THAT ALL REMAIN WORDS TO WORD 255 ARE ALL 1'S) STORE BAD SECTOR DATA.

NOTE: IF HEAD 1 IS DESELECTED VIA THE SW QUESTIONS, THIS TEST WILL BE BYPASSED AND A MESSAGE PRINTED TO THAT AFFECT.

TEST 3 WRITE/READ DATA TEST (PART 1)

READ THE BAD SECTOR FILE IF NOT ALREADY READ.

POSITION HEADS AT CYLINDER 0.

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

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

REPEAT FOR OTHER DATA PATTERNS (2 THROUGH 8).

1

Page 32

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

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

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

TEST 4 ROTATIONAL TIMING TEST

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

POSITION HEADS TO CYLINDER 0.

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

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

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

TEST 5 WRITE/READ TEST (PART 2)

READ THE BAD SECTOR FILE IF NOT ALREADY READ.

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

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

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

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

READ/COMPARE ALL DATA.

REPEAT ABOVE TEST FOR HEAD 1.

REPEAT ABOVE TESTS FOR ALL CYLINDERS IN SELECTED CYLINDER SET.

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

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
Page 33

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

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

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

TEST 6 WRITE LOCK ERROR AND DATA PROTECTION TEST

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

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

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

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

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

TEST 7 ADJACENT CYLINDER INTERFERENCE TEST

READ THE BAD SECTOR FILE IF NOT ALREADY READ.

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

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

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

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
Page 34

OPPOSITE OF CENTER CYLINDER.)

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

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

REPEAT ABOVE TESTS ON HEAD 1.

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

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

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

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

TEST 8 OVERWRITE TEST

READ THE BAD SECTOR FILE IF NOT ALREADY READ.

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

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

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

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

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
Page 35

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

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

REPEAT ABOVE TESTS ON HEAD 1.

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

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

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

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

```
4 000000 000001          PART2==1
5 .ENABL LC,AMA,ABS
6 .NLIST MC,BEX,TOC
7           =2000
8           :MCALL SVC
9
10
11 000000          SVCTST=0
12 000000          SVCSUB=0
13 000001          SVCBGL=1
14 000000          SVCINS=0
15 000000          SVCTAG=0
```

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

1 .NLIST CND,MD,ME
2
3
4
5
6 002000 103 .ASCII /C/
002001 132 .ASCII /Z/
002002 122 .ASCII /R/
002003 114 .ASCII /L/
002004 116 .ASCII /N/
002005 000 .BYTE 0
002006 000 .BYTE 0
002007 000 .BYTE 0
002010 103 .ASCII /C/
002011 060 .ASCII /O/
002012 000000 .WORD 0
002014 030000 .WORD 30000
002016 037352 .WORD L\$HARD
002020 037526 .WORD L\$SOFT
002022 014462 .WORD L\$HW
002024 014500 .WORD L\$SW
002026 040220 .WORD L\$LAST
002030 000000 .WORD 0
002032 000000 .WORD 0
002034 000000 .WORD 0
002036 000000 .WORD 0
002040 014520 .WORD L\$DISPATCH
002042 000000 .WORD 0
002044 000000 .WORD 0
002046 000000 .WORD 0
002050 004 .BYTE C\$REVISION
002051 001 .BYTE C\$EDIT
002052 000000 .WORD 0
002054 000000 .WORD 0
002056 000000 .WORD 0
002060 002214 .WORD L\$DVTYP
002062 000000 .WORD 0
002064 000000 .WORD 0
002066 000000 .WORD 0
002070 000000 .WORD 0
002072 016204 .WORD L\$DU
002074 000000 .WORD 0
002076 002122 .WORD L\$DESC
002100 104035 EMT E\$LOAD
002102 000000 .WORD 0
002104 014540 .WORD L\$INIT
002106 016056 .WORD L\$CLEAN
002110 015520 .WORD L\$AUTO
002112 014452 .WORD L\$PROT
002114 000000 .WORD 0
002116 000000 .WORD 0
002120 000000 .WORD 0
8 002122 103 132 122 .ASCIZ /CZRLN TESTS SEEK, ROTATIONAL TIMING AND WRITE & READ DATA/
9 002214 122 114 060 .EVEN
10 .ASCTZ *RL01,RL02\$
11 .EVE.J
12 .SBTTL GLOBAL EQUATE SECTION

14
15

```
: BIT DEFINITIONS
100000      BIT15== 100000
040000      BIT14== 40000
020000      BIT13== 20000
010000      BIT12== 10000
004000      BIT11== 4000
002000      BIT10== 2000
001000      BIT09== 1000
000400      BIT08== 400
000200      BIT07== 200
000100      BIT06== 100
000040      BIT05== 40
000020      BIT04== 20
000010      BIT03== 10
000004      BIT02== 4
000002      BIT01== 2
000001      BIT00== 1
.
001000      BIT9==  BIT09
000400      BIT8==  BIT08
000200      BIT7==  BIT07
000100      BIT6==  BIT06
000040      BIT5==  BIT05
000020      BIT4==  BIT04
000010      BIT3==  BIT03
000004      BIT2==  BIT02
000002      BIT1==  BIT01
000001      BIT0==  BIT00
:
: EVENT FLAG DEFINITIONS
: EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION
:
000040      EF.START== 32.          : BIT POSITION IN SECOND STATUS WORD
000037      EF.RESTART== 31.        : (100000) START COMMAND WAS ISSUED
000036      EF.CONTINUE== 30.        : (040000) RESTART COMMAND WAS ISSUED
000035      EF.NEW== 29.           : (020000) CONTINUE COMMAND WAS ISSUED
000034      EF.PWR== 28.           : (010000) A NEW PASS HAS BEEN STARTED
000033      EF.XM== 27.            : (004000) A POWER-FAIL/POWER-UP OCCURRED
                                         : (002000) Diag is good of extended environment
:
: PRIORITY LEVEL DEFINITIONS
000340      PRI07== 340
000300      PRI06== 300
000240      PRI05== 240
000200      PRI04== 200
000140      PRI03== 140
000100      PRI02== 100
000040      PRI01== 40
000000      PRI00== 0
:
: OPERATOR FLAG BITS
:
```

```

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

16          : OFFSETS FOR HARDWARE P-TABLE
17 000000      CSR =0      ;BUS ADDRESS
18 000002      VECT =2      ;VECTOR ADDRESS
19 000004      PRIOR =4      ;PRIORITY
20 000006      TYPDR =6      ;DRIVE TYPE
21 000010      DRSB =10     ;DRIVE SELECT BIT
22 000012      CNT =12      ;CONTROLLER TYPE

23          : OFFSET FOR SOFTWARE P-TABLE
24 000000      MISWI =0      ;SOFTWARE PARAMETERS SWITCHES
25 000002      LOLIM =2      ;CYLINDER LOWER LIMIT
26 000004      HILIM =4      ;CYLINDER HIGH LIMIT
27 000006      HEAD =6      ;SELECTED HEAD FOR RUNNING TESTS
28 000010      ERLIM =10     ;ERROR LIMIT
29 000012      DCLIM =12     ;DATA COMPARE ERROR LIMIT
30 000014      BSERR =14     ;BAD SEC FILE PRINT ERROR FLAG

33          : BIT ASSIGNMENT FOR SOFTWARE P-TABLE SWITCHES
34 000001      ALLCYL =BIT00   ;USE ALL CYLINDERS
35 000002      ALLSEC =BIT01   ;USE ALL SECTORS
36 000004      DRSELT =BIT02   ;EXECUTE DRIVE SELECT TEST
37 000010      HDALIGN =BIT03   ;EXECUTE HEAD ALIGNMENT TEST
38 010000      HEADLM =BIT12   ;HEAD LIMIT SPECIFIED FLAG
39 020000      HICYL =BIT13   ;HI LIMIT SPECIFIED FLAG
40 040000      LOCYL =BIT14   ;LO LIMIT SPECIFIED
41 100000      MITEST =BIT15   ;EXECUTE MANUAL INTERVENTION TEST'S

43          : SUBSYSTEM FUNCTIONS
44 000102      CKDATA =102     ;WRITE CHECK
45 000104      GTSTAT =104     ;GET STATUS
46 000106      SEEK =106      ;SEEK
47 000110      RDHEAD =110     ;READ HEADER
48 000112      WTDATA =112     ;WRITE DATA
49 000114      RDDATA =114     ;READ DATA
50 000116      RDNOHR =116     ;READ DATA, IGNORE HEADERS
51 000100      NOOP =100      ;NO OPERATION

53          : OPERATION FLAGS
54 007777      COMPOP =7777    ;COMPOSITE OPERATION FLAGS
55 000002      HDRCMP =BIT01   ;HEADER COMPARE OPERATION
56 000001      DATACMP =BIT00   ;DATA COMPARE OPERATION
57 000004      CYLUP =BIT02    ;CYCLE UP OPERATION
58 000010      ULOAD =BIT03    ;UNLOAD OPERATION

```

59 000020 INOUTS =BIT04 ;IN-OUT SEEK OPERATION
60 000040 OUTINS =BIT05 ;OUT-IN SEEK OPERATION
61 000100 FOLWRT =BIT06 ;FOLLOWING WRITE OPERATION
62 000200 REVSKS =BIT07 ;REV SEEK SEQ (ADJ INTERFERENCE)
63 000400 FWOSKS =BIT08 ;FWD SEEK SEQ (ADJ INTERFERENCE)
64 001000 REVSKO =BIT09 ;REV SEEK SEQ (OVERWRITE)
65 002000 FWDSKO =BIT10 ;FWD SEEK SEQ (OVERWRITE)
66 004000 BADADD =BIT11 ;BAD DISK ADDRESS
67 010000 SEEKOP =BIT12 ;SEEK OPERATION
68 020000 RORWOP =BIT13 ;READ OR WRITE OPERATION
69 040000 RELDWT =BIT14 ;RELOAD WAIT
70 100000 HDR40 =BIT15 ;40 HEADER OPERATION
71 003760 MQUALS =OUTINS!INOUTS!FOLWRT!REVSKS!FWOSKS!REVSKO!FWDSKO
72 ;MESSAGE QUALIFIER BITS
73
74 ;ERROR FLAGS FROM SUBROUTINES
75 000001 TOSLOW =BIT00 ;OPERATION TOOK TOO LONG
76 000002 NOIRPT =BIT01 ;NO INTERRUPT FROM OPERATION
77 000004 CONHNG =BIT02 ;CONTROLLER HUNG
78 000010 NOCLR =BIT03 ;BAD CONTROLLER CLEAR
79
80 000000 RLCS =0 ;CONTROL AND STATUS REGISTER
81 000002 RLBA =2 ;BUS ADDRESS REGISTER
82 000004 RLDA =4 ;DISK ADDRESS REGISTER
83 000006 RLMP =6 ;MULTI-PURPOSE REGISTER
84
85 000000 ;REGISTER BIT DEFINITIONS - CONTROL STATUS REGISTER
86 100000 RLCSR =0 ;CONTROL AND STATUS REGISTER
87 040000 YERR =100000 ;ANY ERROR BIT
88 020000 DRVERR =40000 ;DRIVE ERROR BIT
89 010000 NXMERR =20000 ;NON-EXISTENT MEMORY ERROR
90 010000 DLTERR =10000 ;DATA LATE ERROR
91 010000 MNFERR =10000 ;HEADER NOT FOUND ERROR
92 004000 DCKERR =4000 ;DATA CHECK ERROR
93 004000 MCRCERR =4000 ;HEADER CHECK ERROR
94 002000 OPTERR =2000 ;OPERATION INCOMPLETE ERROR
95 001400 DSMASK =1400 ;DRIVE SELECT MASK
96 200200 CRDYMSK =200 ;CONTROLLER READY MASK
97 000100 INTEBL =100 ;INTERRUPT ENABLE MASK
98 000060 BAMS =60 ;BUS ADDRESS UPPER MASK
99 000001 DRDYMSK =1 ;DRIVE READY MASK
100
101 000077 ;REGISTER BIT DEFINITIONS - DISK ADDRESS FOR DATA XFER
102 000100 SAMSK =77 ;SECTOR ADDRESS MASK
103 HSMSK =100 ;HEAD SELECT MASK
104
105 ;REGISTER BIT DEFINITIONS - DISK ADDRESS FOR SEEK
106 000001 MBSETO =1 ;MUST BE SET BIT 0
107 000004 DIRBIT =4 ;DIRECTION BIT
108 000020 HDSSEL =20 ;HEAD SELECT BIT
109
110 ;REGISTER BIT DEFINITIONS - DISK ADDRESS FOR GET STATUS
111 000003 GETSTAT =3 ;GET STATUS SETUP
112 000010 DRSET =10 ;DRIVE RESET MASK
113
114 ;REGISTER BIT DEFINITIONS - MP FOR DATA XFER
115 017777 WCMSK =17777 ;WORD COUNT MASK

GLOBAL EQUATE SECTION

```

116      160000      WCRNG  =160000      ;WORD COUNT RANGE MASK
117
118
119      000077      HDSEC   =77        ;REGISTER BIT DEFINITIONS - MP FOR READ HEADER
120      000100      HDHSEL   =100       ;SECTOR MASK
121
122
123      000007      STAMSK  =7         ;HEAD SELECT MASK
124      000010      BHSTAT  =10       ;REGISTER BIT DEFINITIONS - MP FOR GET STATUS
125      000020      HOSTAT  =20       ;STATE MASK
126      000040      COSTAT  =40       ;BRUSH HOME STATUS
127      000100      HSSTAT   =100      ;HEADS OUT STATUS
128      000400      DSESTAT  =400      ;COVER OPEN STATUS
129      001000      VCSTAT   =1000     ;HEAD SELECT STATUS
130      002000      WGESTAT  =2000     ;DRIVE SELECT ERROR STATUS
131      004000      SPDSTAT  =4000     ;VOLUME CHECK STATUS
132      010000      STOSTAT  =10000    ;WRITE GATE ERROR STATUS
133      020000      WLSTAT   =20000    ;SPIN ERROR STATUS
134      040000      HCESTAT  =40000    ;SEEK TIMEOUT ERROR STATUS
135      100000      WDESTAT  =100000   ;WRITE LOCK STATUS
136
137
138      172540      CLKCSR   =172540    ;HEAD CURRENT ERROR STATUS
139      172542      CLKCSB   =172542    ;HEAD DATA ERROR STATUS
140      172544      CLKCTR   =172544    ;WRITE DATA ERROR STATUS
141
142
143
144      .SBTTL GLOBAL DATA SECTION
145
146
147
148      : TABLE OF OPERATION MESSAGES
149
150 002226 000000  OPMSGS: WORD 0      ;FILLER
151 002230 005775  WORD MMRCMK      ;MESSAGE FOR WRITE CHECK
152 002232 006020  WORD MGTSTA      ;GET STATUS
153 002234 005750  WORD MSEEK       ;SEEK
154 002236 005765  WORD MREACH      ;READ HEADER
155 002240 006006  WORD MMWRITE     ;WRITE DATA
156 002242 005754  WORD MREAD       ;READ DATA
157 002244 006103  WORD MWRSET      ;WITH RESET
158 002246 006032  WORD MDATCP      ;WITH DATA COMPARE
159 002250 006051  WORD MHDRCP      ;WITH HEADER COMPARE
160 002252 C06150  WORD MCYLUP      ;LOAD HEADS
161 002254 006137  WORD MULOAD      ;UNLOAD HEADS
162 002256 006177  WORD MINOUT      ;IN-OUT SEQ
163 002260 006160  WORD MGUTIN      ;OUT-IN SEQ
164 002262 006220  WORD MFOLWRT     ;FOLLOWING WRITE
165 002264 006240  WORD MREVSK      ;REV SEEK
166 002266 006271  WORD MFWDISK     ;FWD SEEK
167 002270 006356  WORD MRESKO      ;REV SEEK
168 002272 006322  WORD MFWSKO      ;FWD SEEK
169 002274 006412  WORD MBADAD      ;BAD DISK ADD FOR WRITE
170 002276 006067  WORD M4OHDR      ;40 HEADER OPERATION
171
172 002300 000000  T.DRIVE: WORD 0
173 002302 000000  JUNK: WORD 0
174 002304 000000  HLMTW: WORD 0

```

175	002306	000000	CLRBYT:	WORD	0
176	002310	000000	NXTHL:	WORD	0
177	002312	000000	G8ND:	WORD	0
178	002314	000000	CAMSK:	WORD	0
179	002316	C00000	DIRMSK:	WORD	0
180	002320	000000	HOCYL:	WORD	0
181					
182			RESTBL:	TABLE OF RESULT NAME MESSAGE ADDRESSES	
183	002322	010713		MCERR	:CONTROLLER ERROR
184	002324	011024		MDRERR	:DRIVE ERROR
185	002326	011242		MNEERR	:NON-EXISTANT MEMORY ERROR
186	002330	011214		MFLERR	:HEADER NOT FOUND-DATA LATE
187	002332	011177		MHDERR	:HEADER OR DATA ERROR
188	002334	011167		MOPERR	:OPERATION INCOMPLETE
189	002336	011274		MNRST	:NO DRIVE STATUS AVAILABLE
190	002340	000000		0	
191	002342	011152		MWDERR	:WRITE DATA ERROR
192	002344	011134		MHCERR	:HEAD CURRENT ERROR
193	002346	0C0000		0	
194	002350	011120		MSERR	:SEEK TIMEOUT ERROR
195	002352	011065		MSPERR	:SPINDLE ERROR
196	002354	011103		MWGERR	:WRITE GATE ERROR
197	002356	000000		0	
198	002360	011035		MDSERR	:DRIVE SELECT ERROR
199					
200			PATTBL:	PATTERN TABLE	
201	002362	005472		PAT1	
202	002364	005474		PAT2	
203	002366	005534		PAT3	
204	002370	005574		PAT4	
205	002372	005634		PAT5	
206	002374	005642		PAT6	
207	002376	005702		PAT7	
208	002400	005704		PAT8	
209	002402	005744		PAT9	
210	002404	005746		PAT10	
211					
212					
213			SUBSTK:	SUBROUTINE CALLING STACK	
214	002406	000000		0	;STACK IS 12 WORDS LONG
215	002410	000000		0	
216	002412	000000		0	
217	002414	000000		0	
218	002416	000000		0	
219	002420	000000		0	
220	002422	000000		0	
221	002424	000000		0	
222	002426	000000		0	
223	002430	000000		0	
224					
225			T25TBL:	RL01 TABLE OF CYLINDERS	
226	002432	000002		2	;TABLE OF DIFFERENCES
227	002434	000006		6	
228	002436	000011		9	
229	002440	000014		12	
230	002442	000021		17	
231	002444	000026		22	

232 002446 000033 .WORD 27.
233 002450 000042 .WORD 54.
234 002452 000051 .WORD 41.
235 002454 000200 .WORD 128.
236 002456 000377 .WORD 255.

237

238 ;RL02 TABLE OF CYLINDERS
239 002460 000004 t25TB2: .WORD 4
240 002462 000014 .WORD 12.
241 002464 000022 .WORD 18.
242 002466 000030 .WORD 24.
243 002470 000042 .WORD 34.
244 002472 000054 .WORD 44.
245 002474 000066 .WORD 54.
246 002476 000104 .WORD 68.
247 002500 000122 .WORD 82.
248 002502 000400 .WORD 256.
249 002504 000777 .WORD 511.

250

251 ; TABLE TO BE USED TO BUILD AND STORE THE CYLINDERS

252

253 002506 T33TBL: .BLKW 16.
254 002546 TBT: .BLKW 16.

255

256

257 002606 002 CYLTBL: .BYTE 2 ;TABLE OF DEFAULT CYLINDERS
258 002607 007 .BYTE 7.
259 002610 016 .BYTE 14.
260 002611 024 .BYTE 20.
261 002612 033 .BYTE 27.
262 002613 041 .BYTE 33.
263 002614 046 .BYTE 38.
264 002615 055 .BYTE 45.
265 002616 064 .BYTE 52.
266 002617 072 .BYTE 58.
267 002620 101 .BYTE 65.
268 002621 110 .BYTE 72.
269 002622 115 .BYTE 77.
270 002623 124 .BYTE 84.
271 002624 133 .BYTE 91.
272 002625 141 .BYTE 97.
273 002626 146 .BYTE 102.
274 002627 154 .BYTE 108.
275 002630 161 .BYTE 113.
276 002631 170 .BYTE 120.
277 002632 177 .BYTE 127.
278 002633 206 .BYTE 134.
279 002634 213 .BYTE 139.
280 002635 222 .BYTE 146.
281 002636 230 .BYTE 152.
282 002637 235 .BYTE 157.
283 002640 244 .BYTE 164.
284 002641 252 .BYTE 170.
285 002642 261 .BYTE 177.
286 002643 270 .BYTE 184.
287 002644 275 .BYTE 189.
288 002645 303 .BYTE 195.

289	002646	312	.BYTE	202.
290	002647	317	.BYTE	207.
291	002650	326	.BYTE	214.
292	002651	334	.BYTE	220.
293	002652	343	.BYTE	227.
294	002653	352	.BYTE	234.
295	002654	361	.BYTE	241.
296	002655	367	.BYTE	247.
297	002656	375	.BYTE	253.
298	002657	000	.BYTE	0
299	002660	000401	.WORD	257.
300	002662	000406	.WORD	262.
301	002664	000415	.WORD	269.
302	002666	000423	.WORD	275.
303	002670	000432	.WORD	282.
304	002672	000445	.WORD	293.
305	002674	000454	.WORD	300.
306	002676	000463	.WORD	307.
307	002700	000471	.WORD	313.
308	002702	000500	.WORD	320.
309	002704	000507	.WORD	327.
310	002706	000514	.WORD	332.
311	002710	000523	.WORD	339.
312	002712	000532	.WORD	346.
313	002714	000540	.WORD	352.
314	002716	000545	.WORD	357.
315	002720	000553	.WORD	363.
316	002722	000560	.WORD	368.
317	002724	000567	.WORD	375.
318	002726	000576	.WORD	382.
319	002730	000605	.WORD	389.
320	002732	000612	.WORD	394.
321	002734	000621	.WORD	401.
322	002736	000627	.WORD	407.
323	002740	000634	.WORD	412.
324	002742	000643	.WORD	419.
325	002744	000651	.WORD	425.
326	002746	000660	.WORD	432.
327	002750	000657	.WORD	439.
328	002752	000674	.WORD	444.
329	002754	000702	.WORD	450.
330	002756	000711	.WORD	457.
331	002760	000716	.WORD	462.
332	002762	000725	.WORD	469.
333	002764	000733	.WORD	475.
334	002766	000742	.WORD	482.
335	002770	000751	.WORD	489.
336	002772	000760	.WORD	496.
337	002774	000766	.WORD	502.
338	002776	000774	.WORD	508.
339	003000	000774	.WORD	508.
340	003002	000000	.WORD	0
341	003004	000000	SSINDEX:	.WORD 0 ;SUBROUTINE STACK INDEX POINTER
342				
343			OPFLAG:	.WORD 0 ;OPERATIONAL FLAGS
344	003006	000000	DONE:	.WORD 0 ;OPERATION FLAGS
345	003010	000000		;OPERATION COMPLETE FLAG

GLOBAL DATA SECTION

346 003012 000000	HADONE:	.WORD	0	;HEAD ALIGNMENT DONE FLAG
347 003014 000000	ERHEAD:	.WORD	0	;ADDRESS OF ERROR HEADER
348 003016 000000	MORECE:	.WORD	0	;MORE THAN 1 COMPARE ERROR
349 003020 000000	ERRSWI:	.WORD	0	;ERROR RETURN SWITCH
350 003022 000000	BSFLAG:	.WORD	0	;BAD SECTOR FLAGS
351 003024 000000	WRTSWI:	.WORD	0	;WRITE SWITCH
352 003026 000000	TBLSTR:	.WORD	0	;TABLE STORAGE
353				
354 003030 000000	RLBAS:	.WORD	0	;RL11 BASE ADDRESS
355 003032 000000	RLVEC:	.WORD	0	;RL11 VECTOR ADDRESS
356 C03034 000000	RLDRV:	.WORD	0	;DRIVE NUMBER UNDER TEST
357				
358 003036 000000	L.CS:	.WORD	0	;CONTROLLER REGISTER STORAGE
359 003040 000000	L.BA:	.WORD	0	;BEFORE OPERATION
360 003042 000000	L.DA:	.WORD	0	
361 003044 000000	L.MP:	.WORD	0	
362 003046 000000	T.CS:	.WORD	0	;CONTROLLER REGISTER STORAGE
363 003050 000000	T.BA:	.WORD	0	; AFTER OPERATION
364 003052 000000	T.DA:	.WORD	0	
365 003054	T.MP:			
366 003054 000000	HDWRD1:	.WORD	0	;HEADER WORD STORAGE
367 003056 000000	HDWRD2:	.WORD	0	
368 003060 000000	HDWRD3:	.WORD	0	
369				
370 003062 000000	T.STAT:	.WORD	0	;DRIVE STATE STORAGE
371				
372 003064 000000	RESPARM:	.WORD	0	;PARAM BLOCK FOR REASON REPORT
373 003066 000000		.WORD	0	
374 003070 000000		.WORD	0	
375 003072 000000		.WORD	0	
376 003074 000000		.WORD	0	
377				
378 003076 000000	DRVCNT:	.WORD	0	;DRIVE COUNT FOR DRIVES UNDER TEST
379 003100 000000	DIF AUG:	.WORD	0	;DIFFERENCE AUGMENT FOR SEEK
380 003102 000000	OLDCYL:	.WORD	0	;OLD CYLINDER
381 003104 000000	NEWCYL:	.WORD	0	;NEW CYLINDER
382 003106 000000	CURCYL:	.WORD	0	;CURRENT CYLINDER
383 003110 000000	OESDIF:	.WORD	0	;DESIRED DIFFERENCE
384 003112 000000	OESSGN:	.WORD	0	;DESIRED SIGN
385 003114 000000	DESHD:	.WORD	0	;DESIRED HEAD
386 J03116 000000	DESSEC:	.WORD	0	;DESIRED SECTOR
387 003120 000000	TEMPO:	.WORD	0	;TEMPORARY STORAGE
388 003122 000000	TEMP1:	.WORD	0	;TEMPORARY STARAGE
389 003124 000000	TEMP2:	.WORD	0	;TEMPORARY STORAGE
390 003126 000000	TEMP3:	.WORD	0	;TEMPORARY S TORAGE
391 003130	TEMP4:	.WORD	0	;TEMPORARY S TORAGE
392 003132 000000	TEMP5:	.WORD	0	;TEMPORARY S TORAGE
393 003134 000000	TEMP6:	.WORD	0	;TEMPORARY S TORAGE
394 003136 000000	TEMP7:	.WORD	0	;TEMPORARY S TORAGE
395 003140 000000	TEMP8:	.WORD	0	;TEMPORARY S TORAGE
397				
		TIMER	STORAGE	
398 C03142 000000	OFIN:	.WORD	0	;ONE CYLINDER FORWARD INNER
399 003144 000000	OFINU:	.WORD	0	;UPPER
400 003146 000000	OFMIO:	.WORD	0	;ONE CYLINDER FORWARD MIDDLE
401 003150 000000	OFMIDU:	.WORD	0	;UPPER
402 003152 000000	OFOUT:	.WORD	0	;ONE CYLINDER FORWARD OUTER
403 003154 000000	OFOUTU:	.WORD	0	;UPPER

404 003156 000000	ORIN:	.WORD	0	;ONE CYLINDER REVERSE INNER
405 003160 000000	ORINU:	.WORD	0	;UPPER
406 003162 000000	ORMID:	.WORD	0	;ONE CYLINDER REVERSE MIDDLE
407 003164 000000	ORMIDU:	.WORD	0	;UPPER
408 003166 000000	OROUT:	.WORD	0	;ONE CYLINDER REVERSE OUTER
409 003170 000000	OROUTU:	.WORD	0	;UPPER
410 003172 000000	HFIN:	.WORD	0	;128 CYLINDER FORWARD INNER
411 003174 000000	HFINU:	.WORD	0	;UPPER
412 003176 000000	HFOUT:	.WORD	0	;128 CYLINDER FORWARD OUTER
413 003200 000000	HFOUTU:	.WORD	0	;UPPER
414 003202 000000	HRIN:	.WORD	0	;128 CYLINDER REVERSE INNER
415 003204 000000	HRINU:	.WORD	0	;UPPER
416 003206 000000	HROUT:	.WORD	0	;128 CYLINDER REVERSE OUTER
417 003210 000000	HROUTU:	.WORD	0	;UPPER
418 003212 000000	AFMID:	.WORD	0	;256 CYLINDER FORWARD
419 003214 000000	AFMIDU:	.WORD	0	;UPPER
420 003216 000000	ARMID:	.WORD	0	;256 CYLINDER REVERSE
421 003220 000000	ARMIDU:	.WORD	0	;UPPER
422				
423 003222 000252	EXOCYL:	.WORD	170.	;EXPECTED TIME ONE CYLINDER
424 003224 001046	EXHCYL:	.WORD	550.	;EXPECTED TIME 128 CYLINDER
425 003226 001750	EXACYL:	.WORD	1000.	;EXPECTED TIME 256 CYLINDER
426 003230 000372	EXROT:	.WORD	250.	;EXPECTED ROTATION TIME
428 003232 000004	ERRVEC:	.WORD	4	;ERROR VECTOR
429				
430				MISCELLANEOUS COUNTERS
431 003234 000000	PASCNT:	.WORD	0	;PASS COUNTER (LOCAL TO A TEST)
432 003236 000000	COUNT:	.WORD	0	;A COUNTER (LOCAL TO A TEST)
433 003240 000000	TSTNM:	.WORD	0	CURRENT TEST NUMBER OF LOCAL TEST
434 003242 000000	ERRPOINT:	.WORD	0	;ERROR POINTER
435 003244	ERRCNT:	.BLKW	64.	;ERROR COUNTER FOR PROGRAM
436 003444 000000	PASNUM:	.WORD	0	;PASS NUMBER FOR PROGRAM
437 003446 000000	PSETNM:	.WORD	0	COUNTER FOR PARAMETER SET NUMBER IN USE
438 003450 000	LOCERR:	.BYTE	0	;LOCAL ERROR COUNTER
439 003451 000	NOERCT:	.BYTE	0	;INHIBIT ERROR COUNTING FLAG
440 003452 000000	TRPFLG:	.WORD	0	;HARDWARE TRAP OCCURANCE
441 003454 000000	PWRFLG:	.WORD	0	;POWER FAILURE OCCURANCE
442 003456 000000	XDELAY:	.WORD	0	
443 003460 000000	YDELAY:	.WORD	0	
444 003462 000000	MININC:	.WORD	0	
445 003464 000000	TEMP:	.WORD	0	
446 003466 000000	TIM.US:	.WORD	0	
447 003470 000000	TAG:	.WORD	0	
448 003472 000000	MAJINC:	.WORD	0	
449 003474 000000	CLKFLG:	.WORD	0	;FLAG INDICATING PRESENCE OF A P-CLOCK
450 003476 000000	CLKADR:	.WORD	0	;POINTER TO DIAGNOSTIC MONITOR CLOCK TABLE
451				
452				
453				BAD SECTOR TABLES AND POINTERS
454 003500 000000	BSFVAL:	.WORD	0	;BAD SECTORS FILES VALID FLAG: ;0=NOT READ, 1=VALID, -1=NOT VALID
455				
456 003502	FCTBSF:	.BLKW	125.	;FACTORY BAD SECTOR FILE STORAGE
457 004074 177777		.WORD	-1	;FULL TERMINATE
458 004076	FLDBSF:	.BLKW	125.	;FIELD BAD SECTOR FILE STORAGE
459 004470 177777		.WORD	-1	;FULL TERMINATE
460				
461 004472	IBUFF:	.BLKW	128.	;INPUT BUFFER (1 sector of data)

462 005072	OBUFF:	.BLKW	128.	;OUTPUT BUFFER	"
463					
464 005472 090000	PAT1:	.WORD	0	;PATTERN 1 (ALL ZEROS)	
465 005474 177772	PAT2:	.WORD	177772		
466 005476 177777		.WORD	177777		
467 005500 177777		.WORD	177777		
468 005502 052525		.WORD	052525		
469 005504 052525		.WORD	052525		
470 005506 052525		.WORD	052525		
471 005510 177777		.WORD	177777		
472 005512 177777		.WORD	177777		
473 005514 052525		.WORD	052525		
474 005516 052525		.WORD	052525		
475 005520 177777		.WORD	177777		
476 005522 052525		.WORD	052525		
477 005524 177252		.WORD	177252		
478 005526 177252		.WORD	177252		
479 005530 172765		.WORD	172765		
480 005532 172765		.WORD	172765		
481					
482 005534 000003	PAT3:	.WORD	000003		
483 005536 000000		.WORD	000000		
484 005540 0C0000		.WORD	000000		
485 005542 177777		.WORD	177777		
486 005544 177777		.WORD	177777		
487 005546 177777		.WORD	177777		
488 005550 000000		.WORD	000000		
489 005552 000000		.WORD	000000		
490 005554 177777		.WORD	177777		
491 005556 177777		.WORD	177777		
492 005560 000000		.WORD	000000		
493 005562 177777		.WORD	177777		
494 005564 000000		.WORD	000000		
495 005566 177777		.WORD	177777		
496 005570 000000		.WORD	000000		
497 005572 177777		.WORD	177777		
498					
499 005574 025252	PAT4:	.WORD	025252		
500 005576 052525		.WORD	052525		
501 005600 052525		.WORD	052525		
502 005602 125252		.WORD	125252		
503 005604 125252		.WORD	125252		
504 005606 125252		.WORD	125252		
505 005610 052525		.WORD	052525		
506 005612 052525		.WORD	052525		
507 005614 125252		.WORD	125252		
508 005616 125252		.WORD	125252		
509 005620 052525		.WORD	052525		
510 005622 125252		.WORD	125252		
511 005624 052525		.WORD	052525		
512 005626 125252		.WORD	125252		
513 005630 052525		.WORD	052525		
514 005632 125252		.WORD	125252		
515					
516 005634 155555	PAT5:	.WORD	155555		
517 005636 133333		.WORD	133333		
518 005640 066666		.WORD	066666		

GLOBAL DATA SECTION

519
 520 005642 121105 .WORD 121105
 521 005644 150442 .WORD 150442
 522 005646 064221 .WORD 064221
 523 005650 132110 .WORD 132110
 524 005652 055044 .WORD 055044
 525 005654 026442 .WORD 026442
 526 005656 013211 .WORD 013211
 527 005660 105504 .WORD 105504
 528 005662 042642 .WORD 042642
 529 005664 021321 .WORD 021321
 530 005666 110550 .WORD 110550
 531 005670 044264 .WORD 044264
 532 005672 022132 .WORD 022132
 533 005674 011055 .WORD 011055
 534 005676 104426 .WORD 104426
 535 005700 042213 .WORD 042213
 536

537 005702 177777 .WORD 177777
 538
 539 005704 045513 .WORD 045513
 540 005706 122645 .WORD 122645
 541 005710 151322 .WORD 151322
 542 005712 064551 .WORD 064551
 543 005714 132264 .WORD 132264
 544 005716 055132 .WORD 055132
 545 005720 026455 .WORD 026455
 546 005722 113226 .WORD 113226
 547 005724 045513 .WORD 045513
 548 005726 122645 .WORD 122645
 549 005730 151322 .WORD 151322
 550 005732 064551 .WORD 064551
 551 005734 132264 .WORD 132264
 552 005736 055132 .WORD 055132
 553 005740 026455 .WORD 026455
 554 005742 113226 .WORD 113226
 555

556 005744 125252 .WORD 125252
 557
 558 005746 155555 .WORD 155555
 559

560
 561
 562 .SBTTL GLOBAL MESSAGES
 563
 564
 565

566
 567
 568
 569 005750 123 113 040 MSEEK: .ASCIZ /SK /
 570 005754 122 104 040 MREAD: .ASCIZ /RD DATA /
 571 005765 122 104 040 MREADH: .ASCIZ /RD HDR /
 572 005775 127 122 124 MWRCHK: .ASCIZ /WRT CHCK/
 573 006006 127 122 124 MWRITE: .ASCIZ /WRT DATA /
 574 006020 107 105 124 MGTSTA: .ASCIZ /GET STAT /
 575 006032 127 111 124 MDATCP: .ASCIZ /WITH DATA CMP /
 576 006051 127 111 124 MHDRCP: .ASCIZ /WITH HDR CMP /
 577 006067 106 117 122 M40HDR: .ASCIZ /FOR 40 HOURS/
 578 006103 127 111 124 MWRSET: .ASCIZ /WITH RESET /
 579 006117 117 120 105 MOPER: .ASCIZ /OPER: /

580	006126	122	105	123	MRSLT:	.ASCIZ	/RE	/
581	006137	125	116	114	MULOAD:	.ASCIZ	/UNLU DRV/	
582	006150	114	104	040	MCYLUP:	.ASCIZ	/LD DRV /	
583	006160	106	117	114	MOUTIN:	.ASCIZ	/FOL 0 TO CC SK/	
584	006177	106	117	114	MINOUT:	.ASCIZ	/FOL 255 TO CC SK/	
585	006220	106	117	114	MFOLWRT:	.ASCIZ	/FOL WRT (ND SK)/	
586	006240	101	104	112	MREVSK:	.ASCIZ	/ADJ CYL WRTTN AFT REV SK/	
587	C06271	101	104	112	MFWDSK:	.ASCIZ	/ADJ CYL WRTTN AFT FWD SK/	
588	006322	123	113	040	MFWSKO:	.ASCIZ	/SK FWD,WRT - SK REV,OVERWRT,	
589	006356	123	113	040	MRESKO:	.ASCIZ	/SK REV,WRT - SK FWD,OVERWRT/	
590	006412	117	116	040	MBADAD:	.ASCIZ	/ON BAD SEC FILES/	
591	006433	103	101	116	MFBSF:	.ASCIZ	/CAN'T FIND GOOD COPY OF FACTORY BAD SEC FILE/	
592	006510	103	101	116	MUBSF:	.ASCIZ	/CAN'T FIND GOOD COPY OF FIELD BAD SEC FILE/	
593	006563	102	101	104	MFMTER:	.ASCIZ	/BAD SEC FILE FMT ERR/	
594	006610	102	125	123	BASADD:	.ASCIZ	/BUS ADD=/	
595	006621	104	122	126	DRVNAME:	.ASCIZ	/DRV=/	
596	006626	116	117	040	DRVNAV:	.ASCIZ	/NO DRV FOR TST/	
597	006645	104	122	126	NOPWP:	.ASCIZ	/DRV DID NOT REC'R FROM PWR FAIL/	
598	006705	122	114	103	CSNAME:	.ASCIZ	/RLCS/	
599	006712	122	114	102	BANAM:	.ASCIZ	/RLBA/	
600	006717	122	114	104	DANAM:	.ASCIZ	/RLDA/	
601	006724	122	114	115	MPNAM:	.ASCIZ	/RLMP/	
602	006731	117	120	040	LAB1:	.ASCIZ	/OP INIT = /	
603	006744	117	120	040	LAB2:	.ASCIZ	/OP DONE = /	
604	006757	127	117	122	MWORD:	.ASCIZ	/WORD /	
605	006765	111	116	124	MTOSLOW:	.ASCIZ	/INTRPT TOO LATE/	
606	007005	116	117	040	MDRRES:	.ASCIZ	/NO DRV RSPNSE/	
607	007023	116	117	040	MNPOINT:	.ASCIZ	/NO INTRPT ON CMND COMPLETE/	
608	007056	103	116	124	MCONHNG:	.ASCIZ	/CNTLR HUNG /	
609	007072	105	122	122	MNOCLR:	.ASCIZ	/ERR DID NOT CLR/	
610	007112	126	117	114	VCNRST:	.ASCIZ	/VOL CHK NOT RSET/	
611	007133	125	116	130	UNXERR:	.ASCIZ	/UNXPCTED ERR/	
612	007150	040	124	105	TSTLAB:	.ASCIZ	/ TEST/	
630	007156	117	125	124	P2T03E:	.ASCIZ	/OUT GRO BAND /	
631	007174	111	116	103	P2T04E:	.ASCIZ	/INC SK FWD HD 0/	
632	007214	111	116	103	P2T05E:	.ASCIZ	/INC SK REV HD 0/	
633	007234	111	116	103	P2T06E:	.ASCIZ	/INC SK FWD HD 1/	
634	007254	111	116	116	P2T07E:	.ASCIZ	/INC GRO BAND /	
635	007272	111	116	103	P2T08E:	.ASCIZ	/INC SK REV HD 1/	
636	007312	123	113	000	P2T09E:	.ASCIZ	/SK/	
637	007315	106	127	104	P2T10E:	.ASCIZ	/FWD DSC SK/	
638	007330	122	105	126	P2T11E:	.ASCIZ	/REV OSC SK/	
639	007343	123	113	040	P2T12E:	.ASCIZ	/SK TIMING/	
640	007355	102	101	104	P2T13E:	.ASCIZ	/BAD SEC FILE RD DATA/	
641	007402	127	122	124	P2T14E:	.ASCIZ	&WRT/RD DATA (P1)&	
642	007423	123	120	111	P2T15E:	.ASCIZ	/SPINDLE ROT TIMING/	
643	007446	127	122	124	P2T16E:	.ASCIZ	&WRT/RD DATA (P2)&	
644	007467	127	122	124	P2T17E:	.ASCIZ	/WRT LCK ERR AND DATA PROT/	
645	007521	101	104	112	P2T18E:	.ASCIZ	/ADJ CYL INTERFNCE/	
646	007543	117	126	105	P2T19E:	.ASCIZ	/OVERWRT/	
647	007553	123	113	040	SKTMES:	.ASCIZ	/SK TIMES /	
648	007565	123	120	111	SRTMES:	.ASCIZ	/SPINDLE ROT TIME /	
649	007607	050	111	116	VALDES:	.ASCIZ	/(IN 100'S OF U-SEC)/	
650	007633	101	120	120	MAPROX:	.ASCIZ	/APPROX /	
651	007643	111	116	116	LABIN:	.ASCIZ	/INNER/	
652	007651	115	111	104	LABMID:	.ASCIZ	/MIDDLE/	
653	007660	117	125	124	LABOUT:	.ASCIZ	/OUTER/	

654 007666 115 101 130 LABEXP: .ASCIZ /MAX TIME/
 655 007677 061 040 103 LABOCF: .ASCIZ /1 CYL FWD/
 656 007711 061 040 103 LABOCR: .ASCIZ /1 CYL REV/
 657 007723 115 111 104 LABHCF: .ASCIZ /MID CYL FWD/
 658 007737 115 111 104 LABHCR: .ASCIZ /MID CYL REV/
 659 007753 115 101 120 LABACF: .ASCIZ /MAX CYL FWD/
 660 007767 115 101 130 LABACR: .ASCIZ /MAX CYL REV/
 662 010003 110 104 123 HDMOVF: .ASCIZ /HDS FAILED TO MV IN 10 TRYS/
 680 010037 122 105 123 OPR12: .ASCIZ /RESET WRT LCK /
 681 010056 117 116 040 OPR1A: .ASCIZ /ON /
 682 010062 117 116 040 OPR1B: .ASCIZ /ON DRV /
 683 010072 125 116 104 UNDTST: .ASCIZ /UNDER TEST/
 684 010105 123 105 124 OPR004: .ASCIZ /SET WRT LCK /
 685 010122 104 111 106 DIFWD: .ASCIZ /DIFF /
 686 010130 123 107 116 SGNWD: .ASCIZ /SGN /
 687 010135 110 104 040 HDWD: .ASCIZ /HD /
 688 010141 123 105 103 SECWD: .ASCIZ /SEC /
 689 010146 103 131 114 CYLWD: .ASCIZ /CYL /
 690 010153 106 122 117 FRMWD: .ASCIZ /FROM /
 691 010161 040 102 131 BYPSNM: .ASCIZ / BYPASSED /
 692 010174 122 117 125 SEQMES: .ASCIZ /ROUTINE TRACE SEQ:/
 693 010217 104 122 126 STAMES: .ASCIZ /DRV STAT/
 694 010230 124 117 124 TCERR: .ASCIZ /TOTAL CMP ERRS: /
 695 010251 104 122 111 NOCTRL: .ASCIZ /DRIVE DROPPED - NO CONTROLLER/
 696 010307 104 122 111 NOTRDY: .ASCIZ /DRIVE DROPPED - DID NOT RESPOND WITH "READY"/
 697 010364 045 116 045 NOTST: .ASCIZ /*NATEST #D2#A CANNOT BE PERFORMED.. P-CLOCK NOT AVAILABLE<CR><LF>
 698 010461 045 116 045 NOHD: .ASCIZ /*NATEST #D2#A CANNOT READ BAD SEC FILE.. HD 1 DISABLED BY SW QUESTION/<CR>
 699 010572 045 116 045 BSFNOT: .ASCIZ /*N#A*WARNING* ALL SECTORS ASSUMED GOOD FOR TESTS REQUIRING BAD SEC DATA/
 700
 701
 702 010702 104 122 126 MDRDY: .ASCIZ /DRV RDY /
 703 010713 103 117 116 MCERR: .ASCIZ /CONT ERR /
 704 010725 110 104 122 MMCRC: .ASCIZ /HDR CRC/
 705 010735 104 101 124 MDCRC: .ASCIZ /DATA CRC/
 706 010746 110 104 122 MHNF: .ASCIZ /HDR NOT FND/
 707 010762 104 101 124 MDTL: .ASCIZ /DATA LATE/
 708 010774 110 104 122 MHFCRC: .ASCIZ /HDR NOT FND/HDR CRC/OPIE
 709 011024 104 122 126 MDRERR: .ASCIZ /DRV ERR /
 718 011035 104 122 126 MDSERR: .ASCIZ /DRV SEL ERR /
 719 011052 104 122 126 MDRVST: .ASCIZ /DRV STATE /
 720 011065 123 120 111 MSPERR: .ASCIZ /SPIN TIMEOUT /
 721 011103 127 122 124 MWGERR: .ASCIZ /WRT GAT ERR /
 722 011120 123 113 040 MSTERR: .ASCIZ /SK TIMEOUT /
 723 011134 110 105 101 MMCCR: .ASCIZ /HEAD CUR ERR /
 724 011152 127 122 124 MWDERR: .ASCIZ /WRT DAT ERR /
 725 011167 117 120 122 MOPERR: .ASCIZ /OPR-INC/
 726 011177 110 104 122 MHDERR: .ASCIZ /HDR/DAT ERR &
 727 011214 110 104 122 MFLERR: .ASCIZ /HDR NOT FND/DAT LATE &
 728 011242 116 117 116 MNEERR: .ASCIZ /NON-EXISTENT MEMORY /
 729 011267 103 131 114 MCYLOC: .ASCIZ /CYL /
 730 011274 103 101 116 MNDRST: .ASCIZ /CAN'T GET DRV STAT/
 731 011317 125 116 113 MUNDEF: .ASCIZ /UNKN DRV STATE-NO RDY,NO ERR,HDS OUT/
 732 011364 106 101 111 MRLFAL: .ASCIZ /FAIL TO RELD HDS AFTER ERR CLR/
 733 011423 127 122 124 MWRTAB: .ASCIZ /WRT ABRTD/
 734 011435 040 117 126 MEXERS: .ASCIZ / OVR ERR LIMIT - UNIT DRPPD /
 735 011472 040 105 122 MERRS: .ASCIZ / ERR/
 736 011477 207 377 377 BELL: .ASCIZ <207><377><377>

737
 738
 739 011503 111 123 040 RESE3: .ASCIZ /IS/
 740 011507 040 123 102 RESE4: .ASCIZ /SB/
 741
 742
 743 011514 040 111 116 RESE5: .ASCIZ /IN/
 744 011521 040 117 106 RESE6: .ASCIZ /OF/
 745 011526 123 124 101 STATE2: .ASCIZ /STATE 2/
 746 011536 123 124 101 STATE3: .ASCIZ /STATE 3/
 747 011546 123 124 101 STATES5: .ASCIZ /STATE 5/
 751 011556 061 123 124 C1'OMS: .ASCIZ /1ST 3 MS/
 752 011567 065 060 060 C500MS: .ASCIZ /500MS/
 753 011575 133 131 103 CCYLUP: .ASCIZ /CYC UP/
 754 011604 104 101 124 CAFDT: .ASCIZ /DATA XFR/
 755 011615 065 040 123 C5SEC: .ASCIZ /5 SEC/
 756
 757 C11623 045 116 000 CRLF:: .ASCIZ /\$N/
 758 011626 045 124 000 FMTXT:: .ASCIZ /\$T/
 759 011631 045 116 045 FMTOP1: .ASCIZ /\$N\$T\$N\$T\$06\$T\$01\$N/
 760 011660 045 116 045 FMTOP2: .ASCIZ /\$N\$T\$01\$S1\$T\$01\$N/
 761 011702 045 116 045 FMTOP3: .ASCIZ /\$N\$T\$01\$S1\$T\$T\$N/
 762 011723 045 124 045 FMT1: .ASCIZ /\$T\$T/
 763 011730 045 116 045 FMT2: .ASCIZ /\$N\$T\$T/
 764 011737 045 116 045 FMT3: ;unused
 765 011737 045 116 045 FMT4: .ASCIZ /\$N\$T\$T\$N/
 766 011750 045 116 045 FMT5: .ASCIZ /\$N\$T\$06\$S1\$T\$01/
 767 011770 045 116 045 FMT6: .ASCIZ /\$N\$S1\$T\$S4\$T\$S4\$T\$S4\$T\$S4\$T\$S2\$T/
 768 012032 045 116 045 FMT7: .ASCIZ /\$N\$T\$06\$S2\$06\$S2\$06\$S2\$06\$S3\$03\$S2\$01\$N/
 769 012102 045 116 045 FMT8: .ASCIZ /\$N\$T\$06\$S2\$06\$S2\$06\$S2\$06\$S2\$06/
 770 012134 045 116 045 FMT9: .ASCIZ /\$N\$T/
 771 012141 045 116 045 FMT10: ;unused
 772 012141 045 124 045 FMT11: .ASCIZ /\$T\$01/
 773 012147 045 124 045 FMT12: .ASCIZ /\$T\$03/
 774 012155 045 116 045 FMT13: .ASCIZ /\$N\$S1\$T\$03\$S1\$T\$03\$S1\$T\$01\$S1\$T\$01/
 775 012221 045 116 045 FMT14: .ASCIZ /\$N\$T\$T\$03\$S1\$T\$06\$S1\$T\$06/
 776 012253 045 116 045 FMT15: .ASCIZ /\$N\$S1\$T\$D3\$S1\$T\$06\$S1\$T\$06/
 777 012307 045 116 045 FMT16: .ASCIZ /\$N\$S5\$06/
 778 012320 045 123 061 FMT17: .ASCIZ /\$S1\$T\$N\$S1\$06\$N/
 779 012342 045 116 045 FMT18: .ASCIZ /\$N\$S1\$T\$N\$S5\$T\$S4\$T\$S5\$T\$N/
 780 012374 045 124 045 FMT19: .ASCIZ /\$T\$S4\$06\$S4\$06\$S4\$06\$S4\$06\$N/
 781 012431 045 124 045 FMT20: .ASCIZ /\$T\$S2\$06\$S1\$4\$06\$S4\$06\$N/
 782 012461 045 124 045 FMT21: .ASCIZ /\$T\$S1\$2\$06\$S1\$4\$06\$N/
 783 012504 045 116 045 FMT22: .ASCIZ /\$N\$S1\$T\$C3\$S1\$T\$01\$S1\$T\$02/
 784 012540 045 124 045 FMT23: .ASCIZ /\$T\$T\$T\$01\$N/
 785 012554 045 116 045 FMT24: .ASCIZ /\$N\$T/
 786 012561 045 116 045 FMT25: .ASCIZ /\$N\$D2\$T/
 787 012571 045 116 045 FMT26: .ASCIZ /\$N\$S1\$T\$04\$T\$T\$D3\$N/
 788 012615 045 116 045 FMT27: .ASCIZ /\$N\$T\$D3\$T\$D3\$N/
 789 012634 045 116 045 FMT28: .ASCIZ /\$N\$T\$T\$T/
 790
 792 .EVEN

1 .SBTTL ERROR MESSAGES
2
3 : ERR1 R3 POINTS TO RESULT MESSAGE
4 : RESULT: (R3)
5 :
6 : ERR2 R3 POINTS TO RESULT NAME
7 : .RESULT: (R3) IS 1 SB 0
8 :
9 : ERR3 R3 POINTS TO RESULT NAME
10 : RESULT: (R3) IS 0 SB 1
11 :
12 : ERR4 R3 POINTS TO RESULT NAME
13 : R4 POINTS TO RESULT CONDITIONS
14 : RESULT: (R3) IS 1 SB 0 (R4)
15 :
16 : ERR5 R3 POINTS TO RESULT NAME
17 : R4 POINTS TO RESULT CONDITIONS
18 : RESULT: (R3) IS 0 SB 1 (R4)
19 :
20 : ERR6 RESULT ROUTINE DETERMINES WHICH ERROR(S) ARE SET AND
21 : REPORTS ALL
22 : RESULT: "ERROR" IS 1 SB 0
23 :
24 : ERR7 DRIVE STATE ERROR REPORT
25 : R3 CONTAINS EXPECTED STATE
26 : T.STATUS CONTAINS BAD STATE
27 : RESULT: DRIVE STATE IS (T.STATUS) SR (R3)
28 :
29 : ERR8 HEAD POSITIONING ERROR REPORT
30 : NEWCYL CONTAINS EXPECTED CYLINDER
31 : HDWRD1 CONTAINS BAD CYLINDER
32 : RESULT: CYLINDER IS (HDWRD1) SB (NEWCYL)
33 :
34 : ERR9 UTILITY RESULT REPORT
35 : R3 POINTS TO RESULT NAME
36 : R4 POINTS TO VALUE 1
37 : R5 POINTS TO VALUE 2
38 : RESULT: (R3-NAME) IS (R4-VALUE 1) SB (R5-VALUE 2)
39 :
40 : ERR10 COMPARE ERROR REPORT
41 : R3 CONTAINS THE BAO WORD NUMBER
42 : R4 POINTS TO BAO WORD
43 : R5 POINTS TO GOOD WORD
44 : RESULT: WORD (R3) IS (R4) SB (R5)
45 :

ERROR MESSAGES

2 012646	105737	003451		TSTB	NOERCT	;TEST IF ERROR COUNTING INHIBITED
3 012652	001002			BNE	1;	;YES - SKIP
4 012654	005277	170362		INC	ERRPOINT	;ELSE BUMP ERROR COUNT
5 012660	010146		1\$:	MOV	R1,-(SP)	;STORE R1
6 012662	004737	026274		JSR	PC,RPTOP	;REPORT OPERATION
7 012666	012721	000001		MOV	#1.(R1)+	;SET PARAM NUMBER
8 012672	010321			MOV	R3,(R1)+	;INSERT MESSAGE ADDRESS POINTER
9 012674	0C4737	027062		JSR	PC,RPTRES	;REPORT RESULTS
10 012700	004737	027270		JSR	PC,RPTREM	;REPORT REMAINDER
11 012704	012601			MOV	(SP),+R1	;RESTORE R1
12 012706	004737	016616		JSR	PC,CKERLM	;GO CHECK IF ERROR COUNT EXCEEDED
13 012712	012712	104423	-10000:	TRAP	C\$MSG	
14 012714	005277	170322		INC	ERRPOINT	;BUMP ERROR COUNT
17 012720	010146			MOV	R1,-(SP)	;STORE R1
18 012722	004737	026274		JSR	PC,RPTOP	;REPORT OPERATION
19 012726	012721	000003		MOV	#3.(R1)+	;SET PARAM NUMBER
20 012732	010321			MOV	R3,(R1)+	;INSERT NAME ADD POINTER
21 012734	012721	000001		MOV	#1,(R1)+	;SET IS VALUE
22 012740	005021			CLR	(R1)+	;SET SB VALUE
23 012742	004737	027062		JSR	PC,RPTRES	;REPORT RESULTS
24 012746	004737	027270		JSR	PC,RPTREM	;REPORT REMAINDER
25 012752	012601			MOV	(SP),+R1	;RESTORE R1
26 012754	004737	016616		JSR	PC,CKERLM	;GO CHECK IF ERROR COUNT EXCEEDED
27 012760	012760	104423	L10001:	TRAP	C\$MSG	
28 012762	005277	170254		INC	ERRPOINT	;BUMP ERROR COUNT
31 012766	010146			MOV	R1,-(SP)	;STORE R1
32 012770	004737	026274		JMP	PC,RPTOP	;REPORT OPERATION
33 012774	012721	000003		MOV	#3,(R1)+	;SET PARAM NUMBER
34 013000	010321			MOV	R3,(R1)+	;INSERT NAME ADD POINTER
35 013002	005021			CLR	(R1)+	;SET IS VALUE
36 013004	012721	000001		MOV	#1,(R1)+	;SET SB VALUE
37 013010	004737	027062		JSR	PC,RPTRES	;REPORT RESULTS
38 013014	004737	027270		JSR	PC,RPTREM	;REPORT REMAINDER
39 013020	012601			MOV	(SP),+R1	;RESTORE R1
40 013022	004737	016616		JSR	PC,CKERLM	;GO CHECK IF ERROR COUNT EXCEEDED
41 013026	013026	104423	L10002:	TRAP	C\$MSG	
42 013030	005277	170206		INC	ERRPOINT	;BUMP ERROR COUNT
45 013034	010146			MOV	R1,-(SP)	;STORE R1
46 013036	004737	026274		JSR	PC,RPTOP	;REPORT OPERATION
47 013042	012721	000004		MOV	#4,(R1)+	;SET PARAM NUMBER
48 013046	010321			MOV	R3,(R1)+	;INSERT NAME ADD POINTER
49 013050	012721	000001		MOV	#1,(R1)+	;SET IS VALUE
50 013054	005021			CLR	(R1)+	;SET SB VALUE
51 013056	010411			MOV	R4,(R1)	;INSERT ADD OF CONDITION POINTER
52 013060	004737	027062		JSR	PC,RPTRES	;REPORT RESULTS
53 013064	004737	027270		JSR	PC,RPTREM	;REPORT REMAINDER
54 013070	012601			MOV	(SP),+R1	;RESTORE R1
55 013072	004737	016616		JSR	PC,CKERLM	;GO CHECK IF ERROR COUNT EXCEEDED
56 013076	013076	104423	L10003:	TRAP	C\$MSG	

59	013100	005277	170136		INC	#ERRPOINT	:BUMP ERROR COUNT
60	013104	010146			MOV	R1,-(SP)	:STORE R1
61	013106	004737	026274		JSR	PC,RPTOP	:REPORT OPERATION
62	013112	012721	000004		MOV	#4,(R1)+	:SET PARAM NUMBER
63	013115	010321			MOV	R3,(R1)+	:INSERT NAME ADD POINTER
64	013120	005021			CLR	(R1)+	:SET IS VALUE
65	013122	012721	000001		MOV	#1,(R1)+	:SET SB VALUE
66	013126	010411			MOV	R4,(R1)	:INSERT ADD OF CONDITION POINTER
67	013130	004737	027062		JSR	PC,RPTRES	:REPORT RESULTS
68	013134	004737	027270		JSR	PC,RPTREM	:REPORT REMAINDER
69	013140	012601			MOV	(SP), R1	:RESTORE R1
70	013142	004737	016616		JSR	PC,CKERLM	:GO CHECK IF ERROR COUNT EXCEEDED
71	013145			L10004:	TRAP	C\$MSG	
72	013146	104423			TSTB	NOERCT	:TEST IF ERROR COUNTING INHIBITED
74	013150	105737	003451		BNE	2\$:YES - SKIP
75	013154	001002			INC	#ERRPOINT	:ELSE BUMP ERROR COUNT
76	013156	005277	170060		MOV	R1,-(SP)	:STORE R1
77	013162	010146		2\$:	MOV	R3,-(SP)	:STORE R3
78	013164	010346			MOV	R4,-(SP)	:STORE R4
79	013166	010446			MOV	R5,-(SP)	:STORE R5
80	013170	010546			JSR	PC,RPTOP	:REPORT OPERATION
81	013172	004737	026274		MOV	#3,(R1)+	:SET PARAM NUMBER
82	013176	012721	000003		MOV	#1,2(R1)	:INSERT IS VALUE
83	013202	012761	000001	000002	CLR	TEMP3	:CLEAR FOR STATUS STORAGE
84	013210	005037	003126		MOV	T.CS,R3	:GET T.CS
85	013214	013703	003046		BIC	#177761,R3	:AND CLEAR ALL BUT FUNCTION
86	013220	042703	177761		CMP	#4,R3	:CHECK IF IT WAS GET STATUS
87	013224	022703	000004		BEQ	1\$:YES - STATUS IS IN T.MP, SKIP
88	013230	001434			MOV	#GETSTAT,RLDA(R2)	;ELSE DO GET STATUS
89	013232	012762	000003	000004	MOV	#4,R3	
90	013240	012703	000004		MOV	RLDRV,R3	
91	013244	053703	003034		MOV	R3,RLCS(R2)	
92	013250	010362	000000		MOV	#10,XDELAY	:SAVE ARGUMENT
93	013254	012737	000012	003456	JSR	PC,TIME	:CALL TIMING ROUTINE
94	013266	032762	000200	000000	BIT	#CRDYMMSK,RLCS(R2)	;TEST IF READY
95	013274	001003			BNE	4\$:YES - SKIP
96	013276	012703	001000	3\$:	MOV	#BIT9,R3	:ELSE SET NO DRIVE STATUS BIT
97	013302	000413			BR	6\$:IN MESSAGE WORD AND SKIP
98							
99	013304	016203	000006	4\$:	MOV	RLMP(R2),R3	:STORE STATUS FOR REPORT
100	013310	010337	003126		MOV	R3,TEMP3	
101	013314	113703	003127		MOVB	TEMP3+1,R3	:GET ERROR BITS IN PROPER POSITION
102	013320	000402			BR	5\$	
103							
104	013322	113703	003055	1\$:	MOVB	T.MP+1,R3	:GET ERROR BITS FROM MP REG
105	013326	042703	177442	5\$:	BIC	#177442,R3	:CLEAR UNUSED BITS
106	013332	013704	003046	6\$:	MOV	T.CS,R4	:GET ERROR BITS FROM CS REG
107	013336	042704	001777		BIC	#1777,R4	:CLEAR UNUSED BITS
108	013342	050403			BIS	R4,R3	:MAKE ONE WORD OF POSSIBLE ERRORS
109	013344	032703	002000		BIT	#OPIERR,R3	:TEST IF OPI SET
110	013350	001442			BEQ	J1\$:NO - SKIP
111	013352	032703	010000		BIT	#HNFERR,R3	:TEST IF HDR NOT FOUND ERROR
112	013356	001026			BNE	9\$:YES - SKIP
113	013360	032703	004000		BIT	#CRCERR,R3	:TEST IF HDR CRC ERR
114	013364	001020			BNC	8\$:YES - SKIP

115	013366	012704	011167		MOV	#MOPERR,R4	;SET OPI ALONE MESSAGE
116	013372			7\$:	MOV	#MERRS,-(SP)	
	013372	012746	011472		MOV	R4,-(SP)	
	013376	010446			MOV	#MRSLT,-(SP)	
	013400	012746	006126		MOV	#FMT28,-(SP)	
	013404	012746	012634		MOV	#4,-(SP)	
	013410	012746	000004		MOV	SP, R0	
	013414	010600			TRAP	C\$PNTB	
	013416	104414			ADD	#12,SP	
	013420	062706	000012		BR	13\$;SKIP
117	013424	000430					
118							
119	013426	012704	010725	8\$:	MOV	#MHCRC,R4	;HDR CRC MESSAGE
120	013432	000757			BR	7\$	
121							
122	013434	032703	004000	9\$:	BIT	#HCRCERR,R3	;TEST IF HCRC WITH HDR NOT FND
123	013440	001003			BNE	10\$;YES - SKIP
124	013442	012704	010746		MOV	#MHNF,R4	;MESSAGE HEADER NOT FOUND
125	013446	C00751			BR	7\$	
126							
127	013450	012704	010774	10\$:	MOV	#MHFCRC,R4	;HNF AND HCRC MESSAGE
128	013454	000746			BR	7\$;SKIP
129							
130	C13456	032703	004000	11\$:	BIT	#DCKERR,R3	;TEST IF DATA CHECK SET, NOT OPI
131	013462	001403			BEQ	12\$;NO - SKIP
132	013464	012704	010735		MOV	#MDCRC,R4	;SET MESSAGE DATA CHECK
133	013470	000740			BR	7\$;SKIP
134							
135	013472	032703	010000	12\$:	BIT	#DLTERR,R3	;TEST IF DATA LATE ERROR
136	013476	001403			BEQ	13\$;NO - SKIP
137	013500	012704	010762		MOV	#MDLT,R4	;SET MESSAGE DATA LATE
138	013504	000732			BR	7\$;SKIP
139							
140	013506	012705	100000	13\$:	MOV	#BIT15,R5	;SET BIT POINTER FOR TEST
141	013512	005004			CLR	R4	;CLEAR R4 FOR TABLE COUNT
142	013514	030503		14\$:	BIT	R5,R3	;TEST IF BIT IS SET
143	013516	001005			BNE	16\$;YES - SKIP TO REPORT
144	013520	305724		15\$:	TST	(R4)+	;ELSE BUMP TABLE POINTER
145	013522	000241			CLC		;CLEAR CARRY
146	013524	006005			ROR	R5	;SHIFT BIT POINTER TO NEXT BIT
147	013526	001372			BNE	14\$;LOOP IF NOT 0
148	013530	000405			BR	17\$;ELSE REPORT REMAINDER
149							
150	013532	016411	002322	16\$:	MOV	RESTBL(R4),(R1)	;INSERT NAME ADDRESS
151	013536	004737	027062		JSR	PC,RPTRES	;REPORT RESULTS
152	013542	000766			BR	15\$;GET NEXT BIT
153							
154	013544	004737	027270	17\$:	JSR	PC,RPTREM	;REPORT REMAINDER
155	013550	005737	003126		TST	TEMP3	;TEST IF ANY NEW STATUS
156	013554	001414			BEQ	18\$;NO - SKIP
157	013556	013746	003126		MOV	TEMP3,-(SP)	
	013562	012746	010217		MOV	#STAMES,-(SP)	
	013566	012746	012320		MOV	#FMT17,-(SP)	
	013572	012746	000003		MOV	#3,-(SP)	
	013576	010600			MOV	SP, R0	
	013600	104414			TRAP	C\$PNTB	
	013602	062706	000010		ADD	#10,SP	

158	013606	032737	004000	003046	18\$:	BIT	#DCKERR,T.CS	:TEST IF DATA CHECK ERROR
159	013614	001453				BEQ	22\$:NO - SKIP
160	013616	032737	002000	003046		BIT	#OPIERR,T.CS	:TEST IF OPI SET
161	013624	001047				BNE	22\$:YES - SKIP
162	013626	005037	003016			CLR	MORECE	:CLEAR COMPARE ERROR COUNT
163	013632	012701	000200			MOV	#128,-R1	:SET COMPARE LENGTH
164	013636	012703	000001			MOV	#1,R3	:SET WORD COUNT
165	013642	012705	005072			MOV	#0BUFF,R5	:SET GOOD WORD POINTER
166	013646	012704	004472			MOV	#IBUFF,R4	:SET TEST WORD POINTER
167	013652	021514			19\$:	CMP	(R5),(R4)	:CHECK WORD
168	013654	001427				BEQ	21\$:GOOD - SKIP
169	013656	023727	003016	000012		CMP	MORECE,#10.	:TEST IF COMPARE LIMIT REACHED
170	013664	003021				BGT	20\$:YES - SKIP
171	013666	011546				MOV	(R5),-(SP)	
	013670	012746	011507			MOV	#RESE4,-(SP)	
	013674	011446				MOV	(R4),-(SP)	
	013676	012746	011503			MOV	#RESE3,-(SP)	
	013702	010346				MOV	R3,-(SP)	
	013704	012746	006757			MOV	#MWORD,-(SP)	
	013710	012746	012253			MOV	#FMT15,-(SP)	
	013714	012746	000007			MOV	#7,-(SP)	
	013720	010600				MOV	SP, R0	
	013722	104414				TRAP	C\$PNTB	
	013724	062706	000020			ADD	#20, SP	
172	013730	005237	003016		20\$:	INC	MORECE	:BUMP ERROR COUNTER
173	013734	022524			21\$:	CMP	(R5)+,(R4)+	:BUMP POINTERS
174	013736	005203				INC	R3	:BUMP COUNTER
175	013740	005301				DEC	R1	:DEC LENGTH COUNT
176	013742	001343				BNE	19\$:LOOP IF NOT DONE
177	013744	005737	003016		22\$:	TST	MORECE	:TEST IF ANY COMPARE ERRORS
178	013750	001421				BEQ	23\$:NO - SKIP
179	013752	012701	000200			MOV	#128, R1	:SET COMPARE LENGTH
180	013756	010146				MOV	R1,-(SP)	
	013760	012746	011521			MOV	#RESE6,-(SP)	
	013764	013746	003016			MOV	MORECE,-(SP)	
	013770	012746	010230			MOV	#TCERR,-(SP)	
	013774	012746	012615			MOV	#FMT27,-(SP)	
	014000	012746	000005			MOV	#5,-(SP)	
	014004	010600				MOV	SP, R0	
	014006	104414				TRAP	C\$PNTB	
	014010	062706	000014			ADD	#14, SP	
181	014014	012605			23\$:	MOV	(SP)+,R5	:RESTORE R5, 4, 3, 1
182	014016	012604				MOV	(SP)+,R4	
183	014020	012603				MOV	(SP)+,R3	
184	014022	012601				MOV	(SP)+,R1	
185	014024	004737	016616			JSR	PC,CKERLM	:GC CHECK IF ERROR COUNT EXCEEDED
186	014030	104423			L10005:	TRAP	C\$MSG	
187								
188	014032	005277	167204			INC	@ERRPOINT	:BUMP ERROR COUNT
189	014036	010146				MOV	R1,-(SP)	:STORE R1
190	014040	004737	026274			JSR	PC,RPTOP	:REPORT OPERATION
191	014044	012721	000003			MOV	#3,(R1)+	:SET PARAM NUMBER
192	014050	012721	011052			MOV	#MDRVST,(R1)+	:INSERT NAME ADD POINTER
193	014054	013721	003062			MOV	T,STAT,(R1)+	:INSERT IS VALUE
194	014060	010311				MOV	R3,(R1).INSERT	:SB VALUE
195	014062	004737	027062			JSR	PC,RPTRES	:REPORT RESULTS

197	014066	004737	027270		JSR	PC,RPTREM	:REPORT REMAINDER
198	014072	012601			MOV	(SP)+,R1	:RESTORE R1
199	014074	004737	016616	L10006:	JSR	PC,CKERLM	:GO CHECK IF ERROR COUNT EXCEEDED
200	014100				TRAP	C\$MSG	
201							
203	014102	305277	167134		INC	#ERRPOINT	:BUMP ERROR COUNT
204	014106	010146			MOV	R1,-(SP)	:STORE R1
205	014110	010346			MOV	R3,-(P)	:STORE R3
206	014112	004737	026274		JSR	PC,RPTOP	:REPORT OPERATION
207	014116	012721	000003		MOV	#3,(.1)+	:SET PARAM NUMBER
208	014122	012721	011267		MOV	#MCYLOC,(R1)+	:INSERT NAME ADD POINTER
209	014126	013711	003054		MOV	HDWRD1,(R1)	:GET HEADER WORD
210	014132	012703	000007		MOV	#7,R3	:SET SHIFT COUNT
211	014136	000241		14\$:	CLC		
212	014140	006011			ROR	(R1)	:ALIGN CHAR FOR PRINTING
213	014142	005303			DEC	R3	: AS IS VALUE
214	014144	001374			BNE	14\$	
215	014146	005721			TST	(R1)+	:BUMP PARAM POINTER
216	014150	013711	003104		MOV	NEWCYL,(R1)	:INSERT SB VALUE
217	014154	004737	027062		JSR	PC,RPTRES	:REPORT RESULTS
218	014160	004737	027270		JSR	PC,RPTREM	:REPORT REMAINDER
219	014164	012603			MOV	(SP)+,R3	:RESTORE R3
220	014166	012601			MOV	(SP)+,R1	:RESTORE R1
221	014170	004737	016616	L10007:	JSR	PC,CKERLM	:GO CHECK IF ERPOR COUNT EXCEEDED
222	014174				TRAP	C\$MSG	
223							
225	014176	005277	167040		INC	#ERRPOINT	:BUMP ERROR COUNT
226	014202	010146			MOV	R1,-(SP)	:STORE R1
227	014204	004737	026274		JSR	PC,RPTOP	:REPORT OPERATION
228	014210	012721	000003		MOV	#3,(R1)+	:SET PARAM NUMBER
229	014214	010321			MOV	R3,(R1)+	:INSERT NAME ADD POINTER
230	014216	C10421			MOV	R4,(R1)+	:SET IS VALUE
231	014220	010521			MOV	R5,(R1)+	:SET SB VALUE
232	014222	004737	027062		JSR	PC,RPTRES	:REPORT RESULTS
233	014226	004737	027270		JSR	PC,RPTREM	:REPORT REMAINDER
234	014232	012601			MOV	(SP)+,R1	:RESTORE R1
235	014234	004737	016616	L10010:	JSR	PC,CKERLM	:GO CHECK IF ERROR COUNT EXCEEDED
236	014240				TRAP	C\$MSG	
237							
239	014242	010146			MOV	R1,-(SP)	:STORE R1
240	014244	005737	003016		TST	MORECE	:TEST IF 2ND BAD LINE
241	014250	001051			BNE	14\$:YES - SKIP
242	014252	005277	166764		INC	#ERRPOINT	:BUMP ERROR COUNT
243	014256	004737	026274		JSR	PC,RPTOP	:REPORT OPERATION
244	014262	005046			CLR	-(SP)	
	014264	153716	003035		BISB	RLDRV+1,(SP)	
	014270	012746	006621		MOV	#DRVNAME,-(SP)	
	014274	013746	003030		MOV	RLBAS,-(SP)	
	014300	012746	006610		MOV	#BASAOD,-(SP)	
	014304	012746	011750		MOV	#FMT5,-(SP)	
	014310	012746	000005		MOV	#5,-(SP)	
	014314	010600			MOV	SP,RO	
	014316	104414			TRAP	C\$PNTB	
	014320	062706	000014		ADD	#14,SP	

245	014324	011546		MOV	(R5),-(SP)		
	014326	012746	011507	MOV	#RESE4,-(SP)		
	014332	011446		MOV	(R4),-(SP)		
	014334	012746	011503	MOV	#RESE3,-(SP)		
	014340	010346		MOV	R3,-(SP)		
	014342	012746	006757	MOV	#MWORD,-(SP)		
	014346	012746	006126	MOV	#MRSLT,-(SP)		
	014352	012746	012221	MOV	#FMT14,-(SP)		
	014356	012746	000010	MOV	#10,-(SP)		
	014362	010600		MOV	SP, R0		
	014364	104414		TRAP	C\$PNTB		
	014366	062706	000022	ADD	\$22, SP		
246	014372	000421		BR	15\$		
247							
248	014374		14\$:				
	014374	011546		MOV	(R5),-(SP)		
	014376	012746	011507	MOV	#RESE4,-(SP)		
	014402	011446		MOV	(R4),-(SP)		
	014404	012746	011503	MOV	#RESE3,-(SP)		
	014410	010346		MOV	R3,-(SP)		
	014412	012746	006757	MOV	#MWORD,-(SP)		
	014416	012746	012253	MOV	#FMT15,-(SP)		
	014422	012746	000007	MOV	#7,-(SP)		
	014426	010600		MOV	SP, R0		
	014430	104414		TRAP	C\$PNTB		
	014432	062706	000020	ADD	\$20, SP		
249	014436	005237	003016	15\$:	INC	MORECE	; INC COMPARE ERROR COUNT
250	014442	012601			MOV	(SP)+, R1	; RESTORE R1
251	014444	004737	016616		JSR	PC, CKERLM	; GO CHECK IF ERROR COUNT EXCEEDED
252	014450			L10011:	TRAP	C\$MSG	
	014450	104423					

ERROR MESSAGES

1
 2 :LOAD PROTECTION TABLE
 3
 4 014452 000000 .WORD 0 :OFFSET OF CSR IN P-TABLE
 5 014454 177777 .WORD -1 :NOT A MASS-BUS DRIVE
 6 014456 000011 .WORD DRSB+1 :OFFSET OF DRIVE IN P-TABLE
 7
 8
 9
 10
 11 014460 000006 .WORD L10013-L\$HW/2
 12 014462 174400 .WORD 174400 :CSR BASE ADDRESS DEFAULT
 13 014464 000160 .WORD 160 :VECTOR DEFAULT
 14 014466 000240 .WORD 240 :PRIORITY DEFAULT
 15 014470 000001 .WORD 1 :TYPE OF DRIVE
 16 014472 000000 .WORD 0 :DRIVE NUMBER DEFAULT
 17 014474 000001 .WORD 1 :RL11 CONTROLLER
 18 014476 L10013:
 19
 20
 21 014476 000007 MISWIW: .WORD L10014-L\$SW/2 :BIT 0 = USE ALL CYLINDERS
 22 014500 000000 .WORD 0 :BIT 1 = USE ALL SECTORS
 23
 24
 25
 26
 27
 28
 29
 30
 31 014502 000000 LOLIMW: .WORD 0 :BIT 2 = EXECUTE DRIVE SELECT TEST
 32 014504 000377 HILIMW: .WORD 255. :BIT 3 = EXECUTE HEAD ALIGNMENT
 33 014506 000000 HEADW: .WORD 0 :BIT 12 = HEAD SELECT SUPPLIED FLAG
 34 014510 000024 ERLIMW: .WORD 20. :BIT 13 = HILIMIT SPECIFIED FLAG
 35 014512 000012 DCLIMW: .WORD 10. :BIT 14 = LO LIMIT SPECIFIED FLAG
 36 014514 000000 BSERRS: .WORD 0 :BIT 15 = DO MANUAL INTERVENTION
 37 014516 L10014:
 38
 39
 40
 41 014516 000010 .WORD 8
 42 014520 027554 .WORD T1
 43 014522 031474 .WORD T2
 44 014524 031526 .WORD T3
 45 014526 031750 .WORD T4
 46 014530 032564 .WORD T5
 47 014532 033702 .WORD T6
 48 014534 034726 .WORD T7
 49 014536 036150 .WORD T8

```

1          .SBTTL INITIALIZATION SECTION
2
3
4
5
6          ;CHECK FOR PRESENCE OF A P-CLOCK
7
8 014540 005037 003474      CLR    CLKFLG      ;CLEAR CLOCK FLAG
9 014544 012700 000120      MOV    #P, R0
10 014550 104462            TRAP   C$CLK
11 014552 010037 003476      MOV    R0, CLKADR
12 014556 103002            BCC   1$
13 014560 005237 003474      INC    CLKFLG      ;INDICATE PRESENCE OF A P-CLOCK
14 014564 012700 000340      1$:   MOV    #340, R0
15 014570 104441            TRAP   C$SPRI
16 014572 104433            TRAP   C$RESET
17 014574 104450            TRAP   C$MANI
18 014600 042737 100014 014500      BCS   2$
19 014606 005037 003004      BIC    #MITEST!DRSELT!HOALIGN,MISWIW ;CLEAR ALL MANUAL
20 014612 012700 000034      CLR    SSINOX      ;INTERVENTION FLAGS
21 014616 104447            TRAP   #EF.PWR, R0
22 014620 103005            BCC   3$
23 014622 013737 002012 003454      MOV    L$UNIT,PWRFLG ;SET POWER FAIL FLAG
24 014630 000137 015246      JMP    PWCON       ;GO SERVICE POWER FAIL
25 014634 012700 000040      3$:   MOV    #EF.START, R0
26 014640 104447            TRAP   C$REFG
27 014642 103034            BCC   RESTART
28
29
30 014644 013737 002012 003076      ;ON START INITIALIZE TO START AT FIRST DRIVE, CLEAR INTERNAL
31 014652 005037 003444      RSTRT: MOV    L$UNIT,DRV_CNT ;PASS COUNT, AND ERROR COUNT.
32 014656 012700 003244      CLR    PASNUM      ;CLEAR PASS NUMBER
33 014662 012701 000100      MOV    #ERRCNT, R0
34 014666 005020            1$:   MOV    #64, R1      ;GET A COUNT
35 014670 005301            CLR    (R0)+       ;CLEAR AN ERROR COUNTER STORAGE AREA
36 014672 001375            DEC    R1
37 014674 012737 003242 003242      BNE   1$         ;LOOP TILL ALL CLEARED
38 014702 012737 177777 003446      MOV    #ERRCNT-2, ERRPOINT ;INIT ERROR POINTER
39 014710 012737 177777 003012      MOV    #-1, PSETNM ;SET PARAM SELECT TO INITIAL VALUE
40 014716 032737 040000 014500      MOV    #-1, HADONE ;PRESET HEAD ALIGN DONE FLAG
41 014724 001002            BIT    #LOCYL,MISWIW ;TEST IF LO LIMIT SET
42 014726 005037 014502            BNE   2$         ;YES - SKIP
43 014732 000432            CLR    LOLIMW      ;ELSE CLEAR LO LIMIT
44
45 014734 012700 000037      2$:   BR    SETOON
46 014734 012700 000037      RESTART: MOV    #EF.RESTART, R0
47 014740 104447            TRAP   C$REFG
48 014742 103743            BCS   RSTRT
49 014744 012700 000036      CONTINUE: MOV    #EF.CONTINUE, R0
50 014750 104447            TRAP   C$REFG
51 014752 103535            BCS   PWCON

```

```

51
52
53 014754 012700 000035 : ON CONTINUE PICK UP UNIT LAST UNDER TEST
54 014760 104447 MOV #EF.NEW,RO
54 014762 103403 TRAP C$REFG
55 BCS PASNEW
56 014764 005737 003076 NXTPAS: TST DRVCNT ;TEST IF ALL UNITS CHECKED
57 014770 001013 BNE SETDON ;NO - SKIP
58
59 014772 005237 003444 PASNEW: INC PASNUM ;ELSE BUMP PASS COUNT
60 014776 012737 003242 MOV #ERRCNT-2,ERRPOINT ;INIT ERROR POINTER
61 015004 013737 002012 MOV L$UNIT,DRVCNT ;GET ALL DRIVES
62 015012 012737 177777 003446 MOV #1,PSETNM ;SET PARAM SELECT TO INITIAL
63
64 015020 005037 003500 SETOON: CLR BSFVAL ;ENABLE BAD SEC FILE READ
65 015024 005237 003446 INC PSETNM ;NEXT SET OF PARAMETERS
66 015030 005337 003076 DEC ORVCNT ;DOWN COUNT DRIVE TOTAL
67 015034 062737 000002 003242 ADO #2,ERRPOINT ;UPDATE THE ERROR POINTER
68 015042 013700 003446 MOV PSETNM,RO ;SET UP TO GET PARAMETERS
69 015046 012702 003030 MOV #RLBAS,R2
70 015052 104442 TRAP C$GPHRD
71 015054 010001 MOV RO,R1
72 015056 103406 BCS 1$
73 015060 005737 003454 IST PWRFLG ;RECENT POWER FAILURE
74 015064 001737 BEQ NXTPAS ;NO
75 015066 005337 003454 DEC PWRFLG ;ACCOUNT FOR DRIVE
76
77 015072 000734 BR NXTPAS
77 015074 012122 1$: MOV {R1}+, {R2}+ ;STORE PARAMETERS CSR
78 015076 012122 MOV {R1}+, {R2}+ ;VECTOR
79 015100 005721 TST {R1}+ ;BUMP PAST PRIORITY
80 015102 012137 002300 MOV {R1}+, T.DRIVE
81 015106 012122 MOV {R1}+, {R2}+
82 015110 022737 000001 002300 CMP #1,T.DRIVE ;IS THIS AN RL01 TYPE DRIVE?
83 015116 001426 BEQ 2$ ;BRANCH IF YES, ELSE
84 015120 012737 000776 002310 MOV #510..,NXTHL ;SETUP PARAMETERS FOR AN RL02 DRIVE
85 015126 012737 000777 002304 MOV #511..,HLMTW
86 015134 012737 001000 002312 MOV #512..,GBND
87 015142 012737 177600 002314 MOV #177600,CAMSK
88 015150 012737 177600 002316 MOV #177600,DIRMSK
89 015156 012737 177600 002320 MOV #177600,HDCYL
90 015164 012737 177000 002306 MOV #177000,CLRBYT
91 015172 000425 BR PWCON
92
93 015174 012737 000377 002304 2$: MOV #255..,HLMTW ;SETUP PARAMETERS FOR AN RL01 DRIVE
94 015202 012737 000400 002312 MOV #256..,GBND
95 015210 012737 077600 002314 MOV #77600,CAMSK
96 015216 012737 077600 002316 MOV #77600,DIRMSK
97 015224 012737 077600 002320 MOV #77600,HDCYL
98 015232 012737 000376 002310 MOV #254..,NXTHL
99 015240 012737 177400 002306 MOV #177400,CLRBYT
100
101 015246 032737 020000 014500 PWCON: BIT #HICYL,MISWIW ;SELECT HI CYLINDER ENABLED?
102 015254 001003 BNE 1$ ;BRANCH IF NO
103 015256 013737 002304 014504 MOV HLMTW,HILIWW ;SETUP HI CYLINDER LIMIT WORD
104 015264 012746 000340 1$: MOV #340..-(SP)

```

```

015270 012746 016536      MOV    #INTHLR -(SP)
015274 013746 003032      MOV    RLVEC -(SP)
015300 012746 000003      MOV    #3 -(SP)
015304 104437              TRAP   C$SVEC
015306 062706 000010      ADD    #10, SP
105 015312 012700 000000      MOV    #0, R0
015316 104441              TRAP   C$SPRI
106 015320 013702 003030      MOV    RLBAS,R2      ;SET RL11 BASE ADDRESS POINTER
116
117          :CHECK IF POWER FAILURE WAIT IS NEEDED
118
119 015324 005737 003454      TST    PWRFLG
120 015330 001472              BEQ    3$      ;NEEDED???
121
122 015332 013705 003034      MOV    RLCRV,R5      ;DRIVE SELECT
123 015336 052705 000200      BIS    #CRDYMSK,R5      ;SET CRDY
124 015342 010562 000000      MOV    R5,RLCS(R2)      ;SELECT DRIVE
125 015346 012701 000170      MOV    #120, R1      ;INITIALIZE WAIT COUNT
126 015352 032762 000001      000000  2$:     BIT    #ORDYMSK,RLCS(R2)      ;DRIVE UP YET?
127 015360 001056              BNE    3$      ;YES START TEST
128
129 015362 012737 000012      MOV    #10, YDELAY      ;SAVE ARGUMENT
015370 004737 016354              JSR    PC,X+TIME      ;CALL TIMING ROUTINE
130 015374 005301              DEC    R1      ;SIXTY GONE BY
131 015376 001365              BNE    2$      ;NO
132 015400 012746 006645      MDV    #NOPWR,-(SP)
015404 012746 012554      MDV    #FMT24,-(SP)
015410 012746 000002      MDV    #2,-(SP)
015414 010600              MDV    SP,R0
015416 104417              TRAP   C$PNTF
015420 062706 000006      ADD    #6, SP
133 015424 005046              CLR    -(SP)
015426 153716 003035      BISB   RLDRV+1,(SP)
015432 012746 006621      MOV    #DRVNAME,-(SP)
015436 013746 003030      MOV    RLBAS,-(SP)
015442 012746 006610      MOV    #BASADD,-(SP)
015446 012746 011750      MOV    #FMT5,-(SP)
015452 012746 000005      MOV    #5,-(SP)
015456 010600              MOV    SP,R0
015460 104417              TRAP   C$PNTF
015462 062706 000014      ADD    #14, SP
134 015466 012746 011623      MOV    #CRLF,-(SP)
015472 012746 000001              MOV    #1,-(SP)
015476 010600              MOV    SP,R0
015500 104417              TRAP   C$PNTF
015502 062706 000004      ADD    #4, SP
135 015506 013700 003446      MOV    PSETNM,R0
015512 104451              TRAP   C$DODU
136 015514 104444              TRAP   C$DCLN
137 015516              3$:
138
139 015516 104411              L10015: TRAP   C$INIT

```

```

1          .SBTTL AUTO DROP SECTION
2
3          ;THE AUTO DROP SECTION IS INVOKED BY THE DIAGNOSTIC SUPERVISOR WHENEVER THE
4          ;"ADR" FLAG IS SET BY THE OPERATOR. IT IS EXECUTED AFTER THE INITIALIZATION
5          ;CODE AND CHECKS THE DRIVE TO DETERMINE IF IT IS READY TO RECEIVE A COMMAND
6          ;IF THE DRIVE IS NOT READY IT IS DROPPED FROM THE TEST CYCLE AND THE NEXT
7          ;DRIVE IS ACCESSED. IF THE DRIVE IS READY THE HARDWARE TESTS ARE PERFORMED
8          ;AFTER WHICH THE NEXT DRIVE IS ACCESSED.
9
11 015520 005037 003452      CLR    TRPFLG      ;CLEAR TRIP FLAG
12 015524 012746 000340      MOV    #340,-(SP)
13 015530 012746 016530      MOV    #TRPHAN,-(SP)
14 015534 012746 003232      MOV    ERRVEC,-(SP)
15 015540 012746 000003      MOV    #3,-(SP)
16 015544 104437             TRAP   C$VEC
17 015546 062706 000010      ADD    #10,SP
18 015552 013702 003030      MOV    RLBAS,R2      ;/NON-EXISTENT CONTROLLER
19 015556 005762 000000      TST    RLCSC(R2)    ;GET RL11 BASE ADDRESS
20 015562 005737 003452      TST    TRPFLG      ;ACCESS DRIVE CONTROLLER ADDRESS
21 015566 001447             BEQ    1$          ;DID TRAP OCCUR?
22 015570 012746 010251      MOV    #NOCTLR,-(SP) ;BRANCH TO CHECK DRIVE IF TRAP DID NOT OCCUR
23 015574 012746 012554      MOV    #FMT24,-(SP)
24 015600 012746 000002      MOV    #2,-(SP)
25 015604 010600             MOV    SP,RO
26 015606 104417             TRAP   C$PNTF
27 015610 062706 000006      ADD    #6,SP
28 015614 005046             CLR    -(SP)
29 015616 153716 003035      BISB   RLDdrv+1,(SP)
30 015622 012746 006621      MOV    #DRVNAME,-(SP)
31 015626 013746 003030      MOV    RLBAS,-(SP)
32 015632 012746 006610      MOV    #BASADD,-(SP)
33 015636 012746 011750      MOV    #FMT5,-(SP)
34 015642 012746 000005      MOV    #5,-(SP)
35 015646 010600             MCV    SP,RO
36 015650 104417             TRAP   C$PNTF
37 015652 062706 000014      ADD    #14,SP
38 015656 012746 011623      MOV    #CRLF,-(SP) ;PRINT DRIVE INFORMATION
39 015662 012746 000001      MOV    #1,-(SP)
40 015666 010600             MOV    SP,RO
41 015670 104417             TRAP   C$PNTF
42 015672 062706 000004      ADD    #4,SP
43 015676 013700 003446      MOV    PSETNM,RO
44 015702 104451             TRAP   C$DODU
45 015704 000460             BR    2$          ;BRANCH TO EXIT
46 015706 013705 003034      1$:   MOV    RLDdrv,R5      ;ELSE, GET DRIVE NUMBER
47 015712 052705 000200      BIS    #CRDYMSK,R5    ;SET CONTROLLER READY
48 015716 010562 000000      MOV    R5,RLCSC(R2)  ;LOAD IN THE DRIVE NUMBER
49 015722 032762 000001      BIT    #RDYMSK,RLCSC(R2) ;IS DRIVE READY?
50 015730 001046             BNE    2$          ;BRANCH TO PERFORM TESTS IF DRIVE IS READY
51 015732 012746 010307      MOV    #NOTRDY,-(SP)
52 015736 012746 012554      MOV    #FMT24,-(SP)
53 015742 012745 000002      MOV    #2,-(SP)
54 015746 010600             MOV    SP,RO

```

015750	104417		TRAP	C\$PNTF	
015752	062706	000006	ADD	#6.SP	
32					;WITH 'READY'
33	015756	005046	CLR	- (SP)	
	015760	153716	BISB	RLDRV+1,(SP)	
	015764	012746	MOV	#DRVNAME,-(SP)	
	015770	013746	MOV	RLBAS,-(SP)	
	015774	012746	MOV	#BASADD,-(SP)	
	016000	012746	MOV	#FMTS,-(SP)	
	016004	012746	MOV	#5,-(SP)	
	016010	010600	MOV	SP, R0	
	016012	104417	TRAP	C\$PNTF	
	016014	062706	ADD	#14, SP	
34					;PRINT DRIVE INFORMATION
35	016020	012746	MOV	#CRLF,-(SP)	
	016024	012746	MOV	#1,-(SP)	
	016030	010600	MOV	SP, R0	
	016032	104417	TRAP	C\$PNTF	
	C16034	062706	ADD	#4, SP	
36	016040	013700	MOV	PSETNM,R0	
	016044	104451	TRAP	C\$DODU	
37	016046		2\$:		
	016046	013700	MOV	ERRVEC,R0	
	016052	104436	TRAP	C\$CVEC	
38	016054		L10016:		
	016054	104461	TRAP	C\$AUTO	

CLEANUP CODE SECTION

```

1          .SBTTL CLEANUP CODE SECTION
2
5 016056 012746 000340      MOV    $340,-(SP)
6 016062 012746 016530      MOV    @TRPHAN,-(SP)
7 016066 013746 003232      MOV    ERRVEC,-(SP)
8 016072 012746 000003      MOV    @3 -(SP)
9 016076 104437              TRAP   C$SVEC
10 016100 062706 000010     ADD    $10,SP
11 016104 012700 000007     MOV    #7, R0
12 016110 104441              TRAP   C$SPRI
13 016112 032752 000200 000000 1$:  BIT    $CRDYMSK,RLCS(R2) ;TEST IF CONTROLLER READY
14 016120 001407              BEQ    2$ ;NO LOOP UNTIL READY
15 016122 053762 003034 000000  BIS    RLDRV,RLCS(R2) ;SET DRIVE NUMBER
16 016130 032762 000001 000000  BIT    $DRDYMSK,RLCS(R2) ;TEST IF DRIVE BUS/
17 016136 001005              BNE    3$ ;NO - SKIP
18 016140 012737 000003 003460 2$:  MOV    #3,YDELAY ;SAVE ARGUMENT
19 016146 004737 016354              JSR    PC,XTIME ;CALL TIMING ROUTINE
20 016152 013700 003032              MOV    RLVEC,R0
21 016156 104436              TRAP   C$CVEC
22 016160 005737 003454              TST    PWRFLG ;PWR FAIL SET
23 016164 001402              BEQ    4$ ;NO
24 016166 005337 003454              DEC    PWRFLG
25 016172 013700 003232 4$:  MOV    ERRVEC,R0
26 016176 104436              TRAP   C$CVEC
27 016200 104433              TRAP   C$RESET
28 016202 104412              L10017: TRAP   C$CLEAN
29 016204 000240              L10020: NOP
30 016206 104453              TRAP   C$DU

```

GLOBAL SUBROUTINES

```

1          .SBTTL GLOBAL SUBROUTINES
2
4
5 016210 012737 000160 002116 TIME: MOV #160,L$DLY      ;GET OUTER DELAY LOOP
6 016216 005237 003466           INC TIM.US        ;US-WAIT ROUTINE INDICATOR
7 016222 013737 003456 003462   MOV XDELAY,MININC ;SAVE ORIGINAL US WAIT
8 016230 005437 003456           NEG XDELAY       ;GET NEGATIVE OF FACTOR
9 016234 104407                 TRAP C$RDBU
10 016236 103420                BCS 2$           ;GET TIME
11 016240 012727 000001          1$: MOV #1..,(PC)+    ;WAIT FACTOR EXPIRED?
12 016244 000000                 WORD 0             ;BRANCH - IF NO
13 016246 013727 002116          MOV L$DLY,(PC)+  ;GET TIME
14 016252 000000                 WORD 0
15 016254 005367 177772          DEC -6(PC)
16 016260 001375                 BNE -.4
17 016262 0C5367 177756          DEC -22(PC)
18 016266 001367                 BNE -.20
19 016270 005237 003456          INC XDELAY       ;WAIT FACTOR EXPIRED?
20 016274 002761                 BLT 1$           ;BRANCH - IF NO
21 016276 000422                 BR 4$            ;GET TIME
22
23 016300 012737 000065 002116 2$: MOV #65,L$DLY      ;GET OUTER DELAY LOOP
24 016306 012727 000001          3$: MOV #1..,(PC)+    ;WAIT FACTOR EXPIRED?
25 016312 000000                 WORD 0             ;BRANCH - IF NO
26 016314 013727 002116          MOV L$DLY,(PC)+  ;GET TIME EXPIRED
27 016320 000000                 WORD 0
28 016322 005367 177772          DEC -6(PC)
29 016326 001375                 BNE -.4
30 016330 005367 177756          DEC -22(PC)
31 016334 001367                 BNE -.20
32 016336 005237 003456          INC XDELAY       ;WAIT FACTOR EXPIRED?
33 016342 002761                 BLT 3$           ;BRANCH - IF NO
34 016344 063737 003462 003120 4$: ADD MININC,TEMPO ;GET TIME EXPIRED
35 016352 000207                 RTS PC            ;RETURN
36
37 016354 012737 000160 002116 XTIME: MOV #160,L$DLY      ;GET OUTER DELAY LOOP
38 016362 005037 003466           CLR TIM.US        ;MS. WAIT INDICATOR
39 016366 013737 003460 003472   MOV YDELAY,MAJINC ;SAVE ORIGINAL WAIT MS
40 016374 006337 003460           ASL YDELAY       ;MULTIPLY BY FACTOR 4
41 016400 006337 003460           ASL YDELAY
42 016404 005437 003460           NEG YDELAY       ;GET NEGATIVE OF RESULT
43 016410 104407                 TRAP C$RDBU
44 016412 103023                 BCC 2$           ;GET TIME
45 016414 012737 000150 002116 1$: MOV #150,L$DLY      ;GET OUTER DELAY LOOP
46 016422 012727 000020          2$: MOV #20..,(PC)+    ;WAIT FACTOR EXPIRED?
47 016426 000000                 WORD 0             ;BRANCH - IF NO
48 016430 013727 002116          MOV L$DLY,(PC)+  ;GET TIME
49 016434 000000                 WORD 0
50 016436 005367 177772          DEC -6(PC)
51 016442 001375                 BNE -.4
52 016444 005367 177756          DEC -22(PC)
53 016450 001367                 BNE -.20
54 016452 005237 003460          INC YDELAY       ;WAIT FACTOR EXPIRED?
55 016456 002761                 BLT 1$           ;BRANCH - IF NO

```

35 016460 000417 BR 3\$;GET TIME
36
37 016462 012727 000010 2\$: MOV #10,(PC)+
016462 000000 WORD 0
016466 000000 MOV L\$DLY,(PC)+
016470 013727 002116 WORD 0
016474 000000 DEC -6(PC)
016476 005367 177772 BNE .-4
016502 001375 DEC -22(PC)
016504 005367 177756 BNE .-20
016510 001367 INC YDELAY
38 016512 005237 003460 :WAIT FACTOR EXPIRED?
39 016516 002761 BLT 2\$:BRANCH - IF NO
40 016520 063737 003472 003464 3\$: ADD MAJINC,TEMP :GET EXPIRED TIME
41 016526 000207 RTS PC :RETURN
42
44
45 ;TRAP HANDLER INDICATES OCCURRENCE OF A TRAP.
46
47 016530 005237 003452 TRPHAN: INC TRPFLG
48
49 016534 000002 L10021: RTI
51
52 ;INTERRUPT HANDLER. ABORTS WAIT TIMER AND STORES RL11 REGISTERS.
53
54 016536 012237 003046 INTHLR: MOV (R2)+,T.CS ;STORE RL REGISTERS
55 016542 012237 003050 MOV (R2)+,T.BA
56 016546 012237 003052 MOV (R2)+,T.DA
57 016552 011237 003054 MOV (R2)+,T.MP
58 016556 012737 177777 003010 MOV #-1,DONE ;SET DONE FLAG
59 016564 013702 003030 MOV RLBAS,R2 ;RESTORE R2
60 016570 013737 003456 003120 MOV XDELAY,TEMPO ;SAVE MICRO-SEC RUN TIME
016576 013737 003460 003464 MOV YDELAY,TEMP ;SAVE MILLI-SEC RUN TIME
016604 005037 003456 CLR XDELAY ;ABORT MICRO-SEC WAIT
016610 005037 003460 CLR YDELAY ;ABORT MILLI-SEC WAIT
61 016614 000002 L10022: RTI

GLOBAL SUBROUTINES

```

1          :      ERROR LIMIT CHECKING ROUTINE
2          :      DROPS DRIVE IF ERROR LIMIT EXCEEDED
3
4 016616 027737 164420 014510 CKERLM: CMP    #ERRPOINT,ERLIMW
5 016624 002453           BLT    1$:                ;TEST IF ERROR LIMIT EXCEEDED
6 016626 104420           TRAP   C$INLP
7 016630 103451           BCS    1$
8 016632 012746 011435     MOV    #MEXERS,-(SP)
9 016636 013746 014510     MOV    ERLIMW,-(SP)
10 016642 Ci2746 012561    MOV    #FMT25,-(SP)
11 016646 012746 000003    MOV    #3,-(SP)
12 016652 010600           MOV    SP, R0
13 016654 104417           TRAP   C$PNTF
14 016656 062706 000010    ADD    #10, SP
15 016662 005046           CLR    -(SP)
16 016664 153716 003035    BISB   RLDRV+1,(SP)
17 016670 012746 006621    MOV    #DRVNAME,-(SP)
18 016674 013746 003030    MOV    RLBAS,-(SP)
19 016700 012746 006610    MOV    #BASADD,-(SP)
20 016704 012746 C11750    MOV    #FMT5,-(SP)
21 016710 012746 000005    MOV    #5,-(SP)
22 016714 010600           MOV    SP, R0
23 016716 104417           TRAP   C$PNTF
24 016720 062706 000014    ADD    #14, SP
25 016724 012746 011623    MOV    #CRLF,-(SP)
26 016730 012746 000001    MOV    #1,-(SP)
27 016734 010600           MOV    SP, R0
28 016736 104417           TRAP   C$PNTF
29 016740 062706 000004    ADD    #4, SP
30 016744 013700 003446    MOV    PSETNM, R0
31 016750 104451           TRAP   C$DODU
32 016752 104444           TRAP   C$DCLN
33 016754 000207           RTS    PC
34
35          :      READ AND STORE ALL RL11 REGISTERS
36
37 016756 016237 000000 003046 READRL: MOV    RLCSR(R2),T.CS :GET CS REG
38 016764 016237 000002 003050           MOV    RLBA(R2),T.BA :GET BUS ADDRESS REG
39 016772 016237 000004 003052           MOV    RLDA(R2),T.DA :GET DISK ADDRESS
40 017000 016237 000006 003054           MOV    RLMP(R2),T.MP :GET MULTI-PURPOSE REG
41 017006 000207           RTS    PC :RETURN

```

GLOBAL SUBROUTINES

5

```

1 ; WAIT FOR CONTROLLER TIMEOUT TO FORCE INTERRUPT ROUTINE
2
3 017010 011646          WAITIN: MDV (SP),-(SP)      ;MAKE ROOM FOR ERROR POINTER
4 017012 005066 000002    CLR 2(SP)                  ;CLEAR FOR POINTER
5 017016 032762 000200 000000    BIT #CRDYMSK,RLCSR(R2) ;TEST IF CONTROLLER READY
6 017024 001420          BEQ 3$                      ;NO SKIP TO WAIT
7 017026 004737 016756          JSR PC,READRL      ;READ ALL RL REGS
8 017032 005737 003010          TST DONE           ;TEST IF INTERRUPT OCCURRED
9 017036 001435          BEQ 5$                      ;NO - GO SET NO INTERRUPT ERR FLAG
10 017040 012766 006765 000002 1$: MDV #MTOSLOW,2(SP) ;ELSE SET TOO SLOW ERROR POINTER
11 017046 032737 002000 003046    BIT #OPIERR,T.CS ;TEST IF OPI SET
12 017054 001403          BEQ 2$                      ;NO - SKIP
13 017056 012766 007005 000002    MDV #MDRRES,2(SP) ;SET MESSAGE FOR NO DRIVE RESPONSE
14 017064 000207          RTS PC                      ;RETURN
15 017066 012737 000003 003460          MDV #3,YDELAY   ;SAVE ARGUMENT
16 017074 004737 016354          JSR PC,XTIME      ;CALL TIMING ROUTINE
17 017100 032762 000200 000000    BIT #CRDYMSK,RLCS(R2) ;TEST IF READY NOW SET
18 017106 001006          BNE 4$                      ;YES - SKIP
19 017110 004737 016756          JSR PC,READRL      ;READ RL REGS
20 017114 012766 007056 000002    MDV #MCONHNG,2(SP) ;SET MESSAGE FOR CONTROLLER HUNG
21 017122 000760          BR 2$                      ;SKIP
22 017124 005737 003010          4$: TST DONE        ;ELSE CHECK IF INTERRUPT OCCURRED
23 017130 001343          BNE 1$                      ;YES - SKIP TO SET TOO SLOW
24 017132 004737 016756          5$: JSR PC,READRL      ;READ RL REGS
25 017136 012766 007023 000002    MDV #MNPOINT,2(SP) ;ELSE SET NO INTERRUPT FLAG
26 017144 000747          BR 2$                      ;GO TO RETURN
27
28 ; OPERATION AND TEST INITIALIZE ROUTINE
29
30 017146 005037 003006          TSTINT: CLR OPFLAG      ;CLEAR OPERATION FLAGS
31 017152 105037 003451          CLR CLRBT          ;RESET INHIBIT ERROR COUNTING
32 017156 005037 003016          CLR MORECE        ;RESET MORE COMPARE ERRORS
33 017162 000207          RTS PC                      ;RETURN

```

```

1          : GET STATUS AND GET STATUS WITH RESET ROUTINE
2
3 017164 013746 003130      : GSTATR: MOV TEMP4,-(SP) ;STORE TEMP4
4 017170 012737 000013 003130    MOV #GETSTAT!DRSET,TEMP4 ;SET FOR RESET
5 017176 000412      BR GSTATG
6
7 017200 013746 003130      : GSTATC: MOV TEMP4,-(SP) ;STORE TEMP4
8 017204 012737 000003 003130    MOV #GETSTAT,TEMP4 ;SET FOR NO RESET
9 017212 000404      BR GSTATG
10
11 017214 013746 003130      : GSTAT: MOV TEMP4,-(SP) ;STORE TEMP4
12 017220 005037 003130      CLR TEMP4 ;SET FOR SAVE L. AND T. REGS
13 017224 010346      : GSTATG: MOV R3,-(SP) ;STORE R3
14 017226 013703 003004      MOV SS$INDX,R3 ;GET SUBROUTINE INDEX
15 017232 005723      TST (R3)+ ;BUMP IT FOR NEXT ENTRY
16 017234 016663 000004 002406  MOV 4(SP),SUBSTK(R3) ;INSERT THIS CALL
17 017242 162763 000004 002406  SUB #4, SUBSTK(R3) ;ADJUST IT TO CALLING LOCATION
18 017250 010337 003004      MOV R3,SS$INDX ;STORE IT BACK
19 017254 010046      MOV R0,-(SP) ;STORE R0
20 017256 010146      MOV R1,-(SP) ;STORE R1
21 017260 012737 000002 003020  MOV #2,ERRSWI ;SET FOR NO ERROR RETURN
22 017266 032737 000010 003130  BIT #DRSET,TEMP4 ;TEST IF DRIVE RESET
23 C17274 001460      BEQ 4$ ;NO - SKIP
24 017276 032762 040000 000000  BIT #DRVERR,RLCS(R2) ;TEST IF DRIVE ERROR SET
25 017304 001405      BEQ 1$ ;NO - SKIP
26 017306 012737 000003 003460  MOV #3,YDELAY ;SAVE ARGUMENT
27 017314 004737 016354      JSR PC,XTIME ;CALL TIMING ROUTINE
28 017320 012701 000062      MOV #50,R1 ;INITIALIZE WAIT COUNT
29 017324 004737 017214      1$: JSR PC,GSTAT ;GET DRIVE STATUS
30 017330 020014      2$: 16$ ;UNLOAD
31 017332 032737 000001 003046  BIT #DRDYMSK,T.CS ;TEST IF DRIVE READY
32 017340 001054      BNE 6$ ;YES - GO DO CLEAR
33 017342 032737 000020 003054  BIT #HOSTAT,T.MP ;ELSE TEST IF HEADS OUT
34 017350 001010      BNE 3$ ;YES - BYPASS RELOAD WAIT FLAG SETTING
35 017352 032737 144000 003054  BIT #SPDSTAT!HCESTAT!WDESTAT,T.MP ;TEST IF DRIVE HAS ERROR
36                                         ;THAT CAUSED HEADS TO
                                         ;UNLOAD
37 017360 001444      BEQ 6$ ;NO - SKIP
38 017362 052737 040000 003006  BIS #RELDWT,OPFLAG ;ELSE SET WAIT FLAG
39 017370 000440      BR 6$ ;SKIP TO CLEAR
40
41 017372 032737 040000 003046  3$: BIT #DRVERR,T.CS ;TEST IF DRIVE ERROR NOW
42 017400 001034      BNE 6$ ;YES - SKIP TO CLEAR
43 017402 012737 000001 003460  MOV #1,YDELAY ;SAVE ARGUMENT
44 017410 004737 016354      JSR PC,XTIME ;CALL TIMING ROUTINE
45 017414 005301      DEC R1 ;DEC WAIT COUNTER
46 017416 001342      BNE 2$ ;IF NOT DONE, LOOP
47 017420 012703 011317      MOV #MUNDEF,R3 ;MESSAGE FOR UNDEFINED STATE
48 017424 104456      TRAP C$ERHRD
49 017426 023421      WORD 10001
50 017430 000000      WORD 0
51 017432 012646      WORD ERR1
52 017434 000565      BR 15$ ;EXIT
53
54 017436 005737 003130      4$: TST TEMP4 ;TEST IF SAVE REGISTERS
55 017442 001013      BNE 6$ ;NO SKIP
56 017444 012701 000004  MOV #4,R1 ;SET SAVE COUNT

```

```

53 017450 012703 003046      5$:    MOV    #L.MP+2,R3      ;SET ADDRESS OF FIRST SAVE
54 017454 014346                MOV    (R3),-(SP)    ;PUT REG ON STACK
55 017456 005301                DEC    R1           ;DEC COUNT
56 017460 001375                BNE    5$          ;LOOP UNTIL ALL SAVED
57 017462 012737 000003 003042  MOV    #GETSTAT,L.DA   ;SET FOR GET STATUS
58 017470 000403                BR     7$          ;SKIP
59
60 017472 013737 003130 003042 6$:    MOV    TEMP4,L.DA   ;INSERT PRESET FOR STATUS
61 017500                7$:    CLR    DONE         ;CLEAR INTERRUPT FLAG
62 017500 005037 003010                MOV    RLDRV,L.CS  ;SET UP TO GET STATUS
63 017504 013737 003034 003036  BIC    #BIT10,L.CS  ;CLEAR FOR DRIVE 4 7 SPEC'D
64 017512 042737 002000 003036  BIS    #GTSTAT,L.CS
65 017520 052737 000104 003036  MOV    L.DA,RLDA(R2)
66 017526 013762 003042 000004  MOV    L.CS,RLCSR(R2)
67 017534 013762 003C36 000000  MOV    #1,XDELAY
68 017542 012737 000001 003456  JSR    PC,TIME
017550 004737 016210                TST    DONE         ;CALL TIMING ROUTINE
69 017554 005737 003010                BEQ    13$        ;CHECK IF INTERRUPT OCCURRED
70 017560 001504                TST    13$        ;NO - SKIP
71 017562 013737 003054 003062 8$:    MOV    T.MP,T.STAT
72 017570 042737 177770 003062  BIC    #!C<STAMSK>,T.STAT
73 017576 032737 000010 003042  BIT    #DRSET,L.DA
74 017604 001503                BEQ    16$        ;TEST IF RESET WAS SPECIFIED
75 017606 032737 040000 003006  BIT    #RELDWT,OPFLAG
76 017614 001427                BEQ    10$        ;TEST IF RELOAD WAIT FLAG SET
77 017616 012701 001130                MOV    #600,R1
78 017622 032762 000001 000000 9$:    BIT    #DRDYMSK,RLCS(R2)
79 017630 001021                BNE    10$        ;TEST IF DRIVE NOW READY
80 017632 012737 000001 003460  MOV    #1,YDELAY
017640 004737 016354                JSR    PC,XTIME
81 017644 005301                DEC    R1           ;SAVE ARGUMENT
82 017646 001365                BNE    9$          ;DEC COUNT
83 017650 004737 017214                JSR    PC,GSTAT
84 017654 020014                16$        ;LOOP IF NOT 0
85 017656 012703 011364                MOV    #MRLFAL,R3
86 017662 104456                TRAP   C$ERHRD
017664 023423                .WORD   10003
017666 000000                .WORD   0
017670 012646                .WORD   ERR1
87 017672 000446                BR     15$        ;SET RESULT MESSAGE POINTER
88
89 017674 012737 000012 003456 10$:    MOV    #10,XDELAY
017702 004737 016210                JSR    PC,TIME
90 017706 004737 017214                JSR    PC,GSTAT
91 017712 020014                16$        ;GET DRIVE STATUS
92 017714 032737 100000 003046  BIT    #ANYERR,T.CS
93 017722 001434                BEQ    16$        ;TEST IF ANY ERROR
94 017724 032737 001000 003054  BIT    #VCSTAT,T.MP
95 017732 J01403                 BEQ    11$        ;NO - SKIP
96 017734 012703 007112                MOV    #VCNRST,R3
97 017740 000417                BR     14$        ;YES SKIP
98
99 017742 032737 040000 003046 11$:    BIT    #DRVERR,T.CS
100 017750 001405                BEQ    12$        ;CHECK IF DRIVE ERROR
101 017752 104456                TRAP   C$ERHRD
017754 023424                .WORD   10004

```

017756	000000		.WORD	0	
017760	013150		.WORD	ERR6	
102	017762	000412	BR	15\$:EXIT
103					
104	017764	012703	007133	12\$:	MOV #UNXERR,R3 ;SET REASON POINTER
105	017770	000403		BR 14\$;EXIT
106					
107	017772	004737	017010	13\$:	JSR PC_WAITIN ;WAIT FOR INTERRUPT
108	017776	012603		MOV (SP)+,R3	;STORE REASON POINTER FOR RETURN
109	020000		14\$:		
	020000	104456		TRAP C\$ERHRD	
	020002	023422		.WORD 10002	
	020004	000000		.WORD 0	
	020006	012646		.WORD ERR1	
110	020010	005037	003020	15\$:	CLR ERRSWI ;CLEAR FOR ERROR RETURN
111	020014	005737	003130	16\$:	TST TEMP4 ;TEST IF REGISTERS WERE SAVED
112	020020	001007		BNE 18\$:NO - SKIP
113	020022	012703	003036	MOV #L.CS,R3	;SET POINTER TO RESTORE
114	020026	012701	000004	MOV #4,R1	;SET REGISTER COUNT
115	020032	012623		MOV (SP)+,(R3)+	;RESTORE REG
116	020034	005301		DEC R1	;DEC COUNT
117	020036	001375		BNE 17\$;LOOP UNTIL ALL ARE RESTORED
118	020040	162737	00000?	SUB #2,SSINDX	;REMOVE ENTRY FROM SUBROUT STACK
119	020046	012601		MOV (SP)+,R1	;RESTORE R1
120	020050	012600		MOV (SP),R0	;RESTORE R0
121	020052	012603		MOV (SP)+,R3	;RESTORE R3
122	020054	012637	003130	MOV (SP)+,TEMP4	;RESTORE TEMP4
123	020060	005737	003020	TST ERRSWI	;TEST IF ERROR RETURN
124	020064	001403		BEQ 19\$:YES - SKIP
125	020066	063716	003020	ADD ERRSWI,(SP)	;ADD IN ERROR RETURN
126	020072	000207		RTS PC	
127	020074	017616	000000	19\$:	MOV @SP),(SP) ;SET ERROR RETURN ADDRESS
128	020100	000207		RTS PC	

```

1 ; SEEK ROUTINE
2
3 020102 012737 177777 003122 XSEEKT: MOV #1,TEMP1 ;SET SPECIAL TIMING SEEK FLAG
4 020110 000402 BR XSEEK1
5
6 020112 005037 003122 XSEEK: CLR TEMP1 ;CLEAR SPECIAL SEEK FOR TIMING FLAG
7 020116 010346 003004 XSEEK1: MDV R3,-(SP) ;STORE R3
8 020120 013703 003004 MOV SSINDX,R3 ;GET SUBROUTINE INDEX
9 020124 005723 TST (R3)+ ;BUMP IT FOR NEXT ENTRY
10 020126 016663 000002 002406 MOV 2(SP),SUBSTK(R3) ;INSERT THIS CALL
11 020134 162763 000004 002406 SUB #4,SUBSTK(R3) ;ADJUST IT TO CALLING LOCATION
12 020142 010337 003004 MOV R3,SSINDX ;STORE IT BACK
13 020146 010046 MOV R0,-(SP)
14 020150 010146 MOV R1,-(SP)
15 020152 010546 MOV R5,-(SP)
16 020154 012737 000002 003020 MOV #2,ERRSWI ;STORE REG
17 020162 005037 003100 CLR DIF AUG ;SET FOR NO ERROR RETURN
18 ;CLEAR DIFFERENCE AUGMENT (FOR SEEKING
19 020166 004737 024054 JSR PC,GETPOS ;PAST GUARD BAND)
20 020172 020624 12$ ;GET PRESENT POSITION
21 020174 013737 003106 003102 MOV CURCYL,OLDCYL ;MOVE CURRENT TO OLD CYLINDER
22 020202 023737 003104 002304 CMP NEWCYL,HLMTW ;TEST IF NEW IS GREATER THAN MAX CYL
23 020210 003427 BLE 1$ ;NO - SKIP
24 C20212 163737 002304 003104 SUB HLMTW,NEWCYL ;ELSE SUBTRACT MAX CYL.
25 020220 013737 003104 003100 MDV NEWCYL,DIF AUG ;STORE DIFFERENCE AS AUGMENT
26 020225 013737 002304 003104 MOV HLMTW,NEWCYL ;SET NEWCYL AS MAX CYL.
27 020234 022737 000001 002300 CMP #1,T.DRIVE
28 020242 001424 BEQ 2$ ;NO - SKIP
29 020244 162737 000001 003104 SUB #1,NEWCYL ;TEST IF NEWCYL HAS NEGATIVE VALUE
30 020252 012737 000001 003112 MOV #1,DESSGN ;NO - SKIP
31 020260 012737 000001 003110 MOV #1,DESDIF ;ELSE MAKE IT POSITIVE
32 020266 000451 BR 6$ ;AND STORE IT AS AUGMENT
33
34 020270 005737 003104 1$: TST NEWCYL ;AND SET NEWCYL TO 0
35 020274 100007 BPL 2$ ;COMPUTE DIFFERENCE AND NEW CYLINDER
36 020276 005437 003104 NEG NEWCYL ;SUB NEWCYL FROM CURCYL
37 020302 013737 003104 0,3100 MOV NEWCYL,DIF AUG ;IF DIFF IS POSITIVE - SKIP(REV SEEK)
38 020310 005037 003104 CLR NEWCYL ;ELSE SET SIGN FOR FORWARD
39 020314 013705 003106 003104 MDV CURCYL,R5 ;MAKE DIFFERENCE POSITIVE
40 020320 163705 003104 SUB NEWCYL,R5 ;SKIP
41 020324 100005 BPL 3$ ;SET SIGN FOR REVERSE
42 020326 012737 000001 003112 MOV #1,DESSGN ;STORE DIFFERENCE
43 020334 005405 NEG R5 ;IS THERE A DIFFERENCE AUGMENT
44 020336 000402 BR 4$ ;NO - SKIP
45
46 020340 005037 003112 3$: CLR DESSGN ;CHECK IF NEW CYL IS MAX CYL.
47 020344 010537 003110 4$: MOV R5,DESDIF ;NO - SKIP
48 020350 005737 003100 TST DIF AUG ;ELSE FORCE SIGN FOR FORWARD
49 020354 001416 BEQ 6$ ;(INNER GUARD BAND)
50 020356 023737 003104 002304 CMP NEWCYL,HLMTW
51 020364 001007 BNE 5$ ;NO - SKIP
52 020366 012737 000C01 003112 MOV #1,DESSGN ;ELSE FORCE SIGN FOR FORWARD
53
54 020374 022737 000001 002300 CMP #1,T.DRIVE
55 020402 001003 BNE 6$ ;GET L REG ADDRESS
56 020404 063737 003100 003110 5$: ADD DIF AUG,DESDIF ;#L.CS,R5
57 020412 012705 003036 6$: MOV

```

58	020416	012715	000106		MOV	#SEEK,(R5)	;SET FOR SEEK
59	020422	053715	003034		BIS	RLDRV,(R5)	;INSERT DRIVE NUMBER
60	020426	042725	002000		BIC	#BIT10,(R5)+	;CLEAR IF DRIVE 4 7 SPEC'D
61	020432	005025			CLR	(R5)+	;CLEAR BUS ADDRESS
62	020434	013715	003110		MOV	DESDIF,(R5)	;LOAD DIFFERENCE
63	020440	012700	000007		MOV	#7,R0	;SET TO SHIFT DIFFERENCE
64	020444	006315		7\$:	ASL	(R5)	
65	020446	005300			DEC	R0	
66	020450	001375			BNE	7\$;LOOP UNTIL ALIGNED
67	020452	005737	003112		TST	DESSGN	;TEST SIGN
68	020456	001402			BEQ	8\$;SKIP IF 0
69	020460	052715	000004		BIS	#DIRBIT,(R5)	;ELSE INSERT SIGN
70	020464	005737	003114	8\$:	TST	DESHD	;TEST IF HEAD 0
71	020470	001402			BEQ	9\$;YES - SKIP
72	020472	052715	000020		BIS	#HDSEL,(R5)	;ELSE SET HEAD BIT
73	020476	052725	000001		BIS	#MBSET0,(R5)+	;INSERT MARKER BIT
74	020502	004737	021230		JSR	PC, RDYCHK	;CHECK IF DRIVE READY
75	020506	020624			12\$		
76	020510	005037	003010		CLR	DONE	;CLEAR INTERRUPT FLAG
77	020514	005737	003122		TST	TEMP1	;CHECK IF SPECIAL SEEK FLAG SET
78	020520	001041			BNE	12\$;YES - SKIP, DO NOT START SEEK
79	020522	014562	000004		MOV	-(R5), RLDA(R2)	;LOAD RL REGISTERS
80	020526	014562	000002		MOV	-(R5), RLBA(R2)	
81	020532	014562	000000		MDV	-(R5), RLCS(R2)	
82	020536	012737	000012	003456	10\$:		
	020536	012737	000012		MDV	#10,XOELAY	;SAVE ARGUMENT
	020544	004737	016210		JSR	PC, TIME	;CALL TIMING ROUTINE
83	020550	005737	003010		TST	DONE	;TEST IF INTERRUPT DONE
84	020554	001012			BNE	11\$;YES - SKIP
85	020556	004737	017010		JSR	PC, WAITIN	;GO WAIT FOR INTERRUPT
86	020562	012603			MOV	(SP), R3	;GET RESULT MESSAGE POINTER
87	020564	104456			TRAP	C\$ERHHD	
	020566	023425			.WORD	10005	
	020570	000000			.WORD	0	
	020572	012646			.WORD	ERR1	
88	020574	005037	003020		CLR	ERRSWI	;CLEAR FOR ERROR RETURN
89	020600	000411			BR	12\$	
90							
91	020602	005737	003046		11\$:	TST	
92	020606	100006				BPL	12\$
93	020610	104456				TRAP	C\$ERHHD
	020612	023426				.WORD	10006
	020614	000000				.WORD	0
	020616	013150				.WORD	ERR6
94	020620	005037	003020			CLR	ERRSWI
95	020624	162737	000002	003004	12\$:	SUB	#2, SSINOX
96	020632	012605				MOV	(SP), R5
97	020634	012601				MOV	(SP), R1
98	020636	012600				MOV	(SP), R0
99	020640	012603				MOV	(SP), R3
100	020642	005737	003020			TST	ERRSWI
101	020646	001403				BEQ	13\$
102	020650	063716	003020			ADD	ERRSWI, (SP)
103	020654	000207				RTS	PC
104	020656	017616	000000		13\$:	MOV	a(SP),(SP)
105	020662	000207				RTS	PC
							;SET ERROR RETURN ADDRESS

GLOBAL SUBROUTINES

```

1 ; POSITION HEADS ROUTINE. POSITIONS HEADS USING 1 CYLINDER SEEKS
2 ; TO CYLINDER SPECIFIED IN R5 BY THE CALLING ROUTINE
3
4 020664 010346      POSHDS: MOV    R3,-(SP)      :SAVE REGS
5 020666 013703 003004      MOV    SS$IDX,R3      :GET SUBROUTINE INDEX
6 020672 005723      TST    (R3)+      :BUMP IT FOR NEXT ENTRY
7 020674 016663 000002 002406      MOV    2(SP),SUBSTK(R3)  :INSERT THIS CALL
8 020702 162763 000004 002406      SUB    #4,SUBSTK(R3)  :ADJUST IT TO CALLING LOCATION
9 020710 010337 003004      MOV    R3,SS$IDX      :STORE IT BACK
10 020714 010346      MOV    R3,-(SP)
11 020716 010446      MOV    R4,-(SP)
12 020720 012737 000002 003020      MOV    #2,ERRSWI      :SET FOR NO ERROR RETURN
13 020726 004737 024054      JSR    PC,GETPOS      :GET CURRENT POSITION
14 020732 021172      PH65$      MOV    #10,,R4      :SET RETRY COUNT
15 020734 012704 000012      TRAP   C$BSEG
16 020740 104404
17
18 020742 104420      1$:      TRAP   C$INLP
19 020744 103012      BCC   2$
20 020746 004737 024054      JSR    PC,GETPOS      :ELSE GET POSITION
21 020752 021170      10$      BNE   4$          :CHECK IF AT INTENDED POSITION
22 020754 023737 003106 003104      CMP    CURCYL,NEWCYL
23 020762 001017      BNE   4$          :NO - SKIP
24 020764 004737 021570      JSR    PC,ONSWAP      :SWAP OLDCYL AND NEWCYL
25 020770 000414      BR    4$          :SKIP
26
27 020772 013737 003106 003102 2$:      MOV    CURCYL,OLDCYL      :IN NOT LOOPING, STORE CURCYL AS OLDCYL
28 021000 023705 003106      CMP    CURCYL,RS      :CHECK IF HDS AT FINAL POSITION
29 021004 001471      BEQ   10$          :YES - GO TO EXIT
30 021006 003003      BGT   3$          :IF CURCYL > FINAL POSITION - SKIP
31 021010 005237 003104      INC    NEWCYL      :ELSE BUMP NEWCYL (MOVE HDS IN)
32 021014 000402      BR    4$          :SKIP
33
34 021016 005337 003104      3$:      DEC    NEWCYL      :DEC NEWCYL (MOVE HDS OUT)
35 021022 004737 020112      4$:      JSR    PC,XSEEK      :DO SEEK
36 021026 021170      10$      BPL   C$ERHRD
37 021030 012701 005670      MOV    #3000,,R1      :SET WAIT COUNT 300 MS
38 021034 004737 023570      JSR    PC,RDYWAIT      :WAIT FOR DRIVE READY
39 021040 021170      10$      TCS
40 021042 005737 003046      TST    5$          :TEST IF ANY ERROR
41 021046 100007      BPL   10008
42 021050 104456      TRAP   C$ERHRD
43 021052 023430      .WORD  0
44 021054 000000      .WORD  0
45 021056 013150      .WORD  ERR6
46 021060 005037 003020      CLR    ERRSWI      :CLEAR FOR ERROR ERROR RETURN
47 021064 000441      BR    10$          :NO - SKIP
48
49 021066 004737 024054      5$:      JSR    PC,GETPOS      :GET POSITION
50 021072 021170      10$      CMP    CURCYL,NEWCYL
51 021074 023737 003106 003104      BNE   7$          :CHECK IF ARRIVED AT DESIRED PLACE
52 021102 001003      BNE   7$          :NO - SKIP
53 021104 012704 000012      6$:      MOV    #10,,R4      :ELSE INIT RETRY COUNT
54 021110 000714      BR    1$          :GO DO NEXT SEEK
55
56 021112 005737 003112      7$:      TST    DESSGN      :TEST IF GOING IN

```

```

54 021116 001017      BNE    9$      ;YES - SKIP
55 021120 023737 003106 003104      CMP    CURCYL, NEWCYL ;CHECK IF HEADS DID NOT MOVE IN
56 021126 003366      BGT    6$      ;YES - SKIP
57 021130 005304      DEC    R4      ;DEC RETRY COUNT
58 021132 001333      BNE    4$      ;DO ANOTHER SEEK IF NOT 0
59 021134 012703 010003      MOV    #HDMOVF, R3 ;ELSE SET RESULT MESSAGE POINTER
60 021140 104456      TRAP   C$ERHRD
61 021142 023431      .WORD  10009
62 021144 000000      .WORD  0
63 021146 012646      .WORD  ERR1
64 021150 005037 003020      CLR    ERRSWI ;CLEAR FOR ERROR ERROR RETURN
65 021154 000405      BR     10$
66 021156 023737 003106 003104 9$:      CMP    CURCYL, NEWCYL ;HDS SHOULD MOVE OUT, CHK THEY DID
67 021164 002747      BLT    6$      ;YES - SKIP
68 021166 000760      BR     8$      ;ELSE GO DEC AND RETRY
69 021170 104405      10$:      TRAP   C$ESEG
70 021172 162737 000002 003004 PH65$:      SUB    #2_SSINDX ;REMOVE ENTRY FROM SUBROUT STACK
71 021200 012604      MOV    (SP)+, R4 ;RESTORE REGISTERS
72 021202 012600      MOV    (SP)+, R0
73 021204 012603      MOV    (SP)+, R3
74 021206 005737 003020      TST    ERRSWI ;TEST IF ERROR RETURN
75 021212 001403      BEQ    1$      ;YES - SKIP
76 021214 063716 003020      ADD    ERRSWI, (SP) ;ADD IN ERROR RETURN
77 021220 000207      RTS    PC
78 021222 017616 000000      1$:      MOV    B(SP), (SP) ;SET ERROR RETURN ADDRESS
79 021226 000207      RTS    PC

```

```

1 : DRIVE READY TEST ROUTINE. CHECKS DRIVE IS READY. IF NOT, WAIT
2 : 500MS FOR READY TO SET.
3
4 021230 010346      RODYCHK: MOV    R3,-(SP)      ;STORE REGS
5 021232 013703 003004      MOV    SSINDEX,R3   ;GET SUBROUTINE INDEX
6 021236 005723      TST    (R3),+        ;BUMP IT FOR NEXT ENTRY
7 021240 016663 000002 002406      MOV    2(SP),SUBSTK(R3) ;INSERT THIS CALL
8 021246 162763 000004 002406      SUB    #4, SUBSTK(R3) ;ADJUST IT TO CALLING LOCATION
9 021254 010337 003004      MOV    R3,SSINDEX ;STORE IT BACK
10 021260 010046      MDV    R0,-(SP)
11 021262 010146      MOV    R1,-(SP)
12 021264 010446      MOV    R4,-(SP)
13 021266 012737 000002 003020      MOV    #2,ERRSWI ;SET FOR NO ERROR RETURN
14 021274 012701 011610      MOV    #5000, R1   ;SET WAIT COUNT
15 021300 004737 017214      JSR    PC,GSTAT ;GET DRIVE STATUS
16 021304 021440      4$:               BIT    #DRDYMSK,T.CS ;TEST IF DRIVE READY
17 021306 032737 000001 003046      BNE    5$           ;YES - EXIT
18 021314 001053      MOV    #1,XDELAY ;SAVE ARGUMENT
19 021316 012737 000001 003456      JSR    PC,TIME  ;CALL TIMING ROUTINE
20 021324 004737 016210      DEC    R1           ;DEC WAIT COUNT
21 021332 001362      BNE    1$           ;LOOP IF NOT 0
22 021334 012703 010702      MOV    #MDRDY,R3  ;SET RESULT MESSAGE POINTER
23 021340 012704 011567      MOV    #C500MS,R4 ;SET CONDITION MESSAGE POINTER
24 021344 104456      TRAP   C$ERHMD ;WORD 10010
021346 023432      .WORD   0
021350 000000      .WORD   0
021352 013100      WORD    ERR5
25 021354 012701 000062      MOV    #50, R1   ;SET WAIT COUNT FOR 5 SECONDS
26 021360 004737 017214      2$:               JSR    PC,GSTAT ;GET DRIVE STATUS
27 021364 021440      4$:               BIT    #DRDYMSK,T.CS ;TEST IF DRIVE READY
28 021366 032737 000001 003046      BNE    3$           ;YES - SKIP
29 021374 001007      MOV    #1,YDELAY ;SAVE ARGUMENT
30 021376 012737 000001 003460      JSR    PC,XTIME ;CALL TIMING ROUTINE
31 021404 004737 016354      DEC    R1           ;DEC WAIT COUNTER
32 021410 005301      BNE    2$           ;LOOP UNTIL TIME DONE
33 021412 001362      3$:               BIT    #ANYERR,T.CS ;TEST IF ANYERR SET
34 021422 001406      BEQ    4$           ;NO - SKIP
35 021424 104456      TRAP   C$ERHMD ;WORD 10011
021426 023433      .WORD   0
021430 000000      .WORD   0
021432 013150      WORD    ERR6
36 021434 005337 003244      DEC    ERRCNT ;REDUCE ERROR COUNT FOR DUAL ERRORS
37 021440 005037 003020      CLR    ERRSWI ;CLEAR FOR ERROR RETURN
38 021444 162737 000002 003004      4$:               SUB    #2,SSINDEX ;REMOVE ENTRY FROM SUBROUTINE STACK
39 021444 012604      MOV    (SP)+,R4 ;RESTORE REGS
40 021452 012601      MOV    (SP)+,R1
41 021456 012600      MOV    (SP)+,R0
42 021460 012603      MOV    (SP)+,R3
43 021462 005737 003020      TST    ERRSWI ;TEST IF ERROR RETURN
44 021466 001403      BEQ    6$           ;YES - SKIP
45 021470 063716 003020      ADD    ERRSWI,(SP) ;ADD IN ERROR RETURN
46 021474 000207      RTS    PC
47 021476 017616 000000      6$:               RTS    (SP),(SP) ;SET ERROR RETURN ADDRESS
48 021502 000207      RTS    PC
49

```

50 : CHOOSE HEAD ROUTINE. PICKS HEAD 0 UNLESS SPECIFIC HEAD IS
51 : SELECTED BY SOFTWARE PARAMETER.
52
53 021504 005037 0C3114 : CHOSHD: CLR DESHD :CLEAR TO HEAD 0
54 021510 032737 010000 014500 BIT #HEADLM,MISWIW ;TEST IF HEAD SPECIFIED
55 021516 001403 BEQ 1\$;NO - SKIP
56 021520 013737 014506 003114 MOV HEADW,DESHD ;INSERT SPECIFIED HEAD
57 021526 000207 1\$: RTS PC
58
59 : SWAP HEAD ROUTINE. CHANGES SELECTED HEAD TO HEAD 1
60 : UNLESS HEAD 0 SPECIFICALLY SELECTED BY SOFTWARE PARAMETER.
61
62 021530 032737 010000 014500 SWAPHD: BIT #HEADLM,MISWIW ;TEST IF HEAD SPECIFIED
63 021536 001011 BNE 1\$;YES - TAKE ABRTR EXIT
64 021540 C05737 003114 TST DESHD ;TEST IF HEAD ONE USED
65 021544 001006 BNE 1\$;YES - TAKE ABORT EXIT
66 021545 012737 000001 003114 MOV #1,DESHD ;ELSE SET FOR HEAD ONE
67 021554 062716 000002 ADD #2,(SP) ;BUMP PAST ABORT RETURN
68 021560 000207 RTS PC ;RETURN
69 021562 017616 000000 1\$: MOV 8(SP),(SP) ;GET ABORT DESTINATION
70 021566 000207 2\$: RTS PC
71
72 : SWAP OLD CYLINDER AND NEW CYLINDER ROUTINE.
73 021570 010046 003102 003102 ONSWAP: MOV R0,-(SP) ;STORE R0
74 021572 013700 003102 MOV OLDCYL,R0 ;MOVE OLD TO R0
75 021576 013737 003104 003104 MOV NEWCYL,OLDCYL ;MOVE NEW TO OLD
76 021604 010037 003104 MOV R0,NEWCYL ;PUT OLD IN NEW
77 021610 012600 MOV (SP)+,R0 ;RESTORE R0
78 021612 000207 RTS PC

```

1 ; BAD SECTOR FILES VALID CHECK ROUTINE. CHECKS IF BAD SECTOR
2 ; FILES HAVE BEEN READ AND STORED. IF NOT, READ BAD SECTOR
3 ; FILES, ELSE EXIT ROUTINE.
4
5 021614 005737 003500      CKBSVD: TST     BSFVAL      ;TEST STATUS OF BAD SECTOR FILE
6 021620 001C02          BNE     I$:                   ;BR IF READ WITH ERRORS OR
7                                     ;VALID
8 021622 004737 021630      JSR     PC,RDBSF      ;READ BAD SECTOR FILE
9 021626 000207          RTS     PC
10
11 : READ BAD SECTOR FILE ROUTINE
12
13 021630 012737 007355 003014  RDBSF: MOV     #P2T13E,ERHEAD ;SET ERROR HEADER
14 021636 012737 000001 003114  MOV     #1,DESHD    ;SET TO HEAD 1
15 021644 032737 010000 014500  BIT     #HEADLM,MISWIW ;TEST IF HEAD SPEC'D
16 021652 001417          BEQ     I$:                   ;NO - SKIP
17 021654 005737 014506      TST     HEADW      ;TEST IF HEAD 0
18 021660 001014          BNE     I$:                   ;NO - SKIP, ELSE
19 021662 013746 003240      MOV     TSTNM,-(SP)   ;SET ERROR HEADER
20 021666 012746 010461      MOV     #NOHD1,-(SP)  ;SET TO HEAD 1
21 021672 012746 000C02      MOV     #2,-(SP)    ;TEST IF HEAD SPEC'D
22 021676 010600          HQV     SP,RO      ;NO - SKIP
23 021700 104417          TRAP    C$PNTF      ;TEST IF HEAD 0
24 021702 062706 000006      ADD     #6,SP      ;NO - SKIP, ELSE
25 021706 000137 022402      JMP     16$       ;EXIT
26
27 021712 013737 002304 003104  1$:      MOV     HLMTW,NEWCYL ;POSITION HEADS AT LAST CYLINDER (BSF)
28 021720 004737 020112          JSR     PC,XSEEK    ;DO SEEK
29 021724 022350          14$     ;ERROR RETURN ADDRESS
30 021726 012701 005670      MOV     #3000,,R1    ;SET WAIT COUNT FOR 300 MS
31 021732 004737 023570      JSR     PC,RDYWAIT ;WAIT FOR INTERRUPT
32 021736 022350          14$     ;ERROR RETURN ADDRESS
33 021740 004737 024202      JSR     PC,VERPOS   ;VERIFY POSITION
34 021744 022350          14$     ;ERROR RETURN ADDRESS
35 021746 005037 003116      CLR     DESSEC     ;SET FOR SECTOR 0
36 021752 012737 003502 003132      MOV     #FCTBSF,TEMP5 ;SET TEMP STORAGE FOR FACTORY BS FILE
37 021760 012737 000020 003134      MOV     #16,,TEMP6   ;SET MAX SECTOR COUNT
38 021766 112737 000001 003451      MOVB    #1,NOERCT  ;SET FOR NO ERROR COUNTING
39 021774 105037 003450          CI,RB    LOCERR    ;CLEAR LOCAL ERROR COUNTER
40 022000 005037 003126          2$:      CLR     TEMP3     ;CLEAR ONES DETECTED FLAG
41 022004 013701 003132      MOV     TEMP5,R1    ;INIT POINTERS
42 022010 013700 003134      MOV     TEMP6,RO
43 022014 012703 00472         MOV     #IBUFF,R3
44 022020 012737 000C02          003020      MOV     #2,ERRSWI  ;SETUP NO ERROR SWITCH
45 022026 004737 025362          JSR     PC,XREAD   ;DO READ
46 022032 022242          10$     ;ERROR RETURN ADDRESS
47 022034 005723          TST     (R3)+    ;TEST IF WORD 0 NOT NEG
48 022036 100470          BMI     9$       ;YES - BAD FMT ERROR
49 022040 005723          TST     (R3)+    ;ELSE TEST WORD 1 NOT NEG
50 022042 100466          BMI     9$       ;YES - BAD FMT ERRCX REPORT
51 022044 005723          TST     (R3)+    ;TEST WORD 2 IS 0
52 022046 001064          BNE     9$       ;NO - SKIP TO FMT ERROR RPT
53 022050 005723          TST     (R3)+    ;TEST WORD 3 IS 0
54 022052 001062          BNE     9$       ;NO - SKIP TO FMT ERROR RPT
55 022054 026327 000764 177777      CMP     764(R3),#-1 ;TEST IF NEXT TO LAST WORD IS ALL 1'S
56 022062 001056          BNE     9$       ;NO - SKIP
57 022064 026327 000766 177777      CMP     766(R3),#-1 ;TEST IF LAST WORD IS ALL 1'S

```

53	022072	001052			BNE	9\$;NO - SKIP	
54	022074	021327	177777	3\$:	CMP	(R3),#-1	;TEST IF NEXT WORD IS ALL 1'S	
55	022100	001005			BNE	4\$;NO SKIP	
56	022102	012737	000001	003126	MOV	#1, TEMP3	;ELSE SET 1'S DETECTED FLAG	
57	022110	022313			CMP	(R3)+,(R3)	;ADJUST POINTER	
58	022112	001420			BEQ	7\$;BR IF THE SAME	
59	022114	005737	003126	4\$:	TST	TEMP3	;TEST IF ONES HAVE BEEN DETECTED	
60	022120	001037			BNE	9\$;YES - SKIP TO FMT ERROR RPT	
61	022122	012311			MOV	(R3),,(R1)	;STORE CYLINDER WORD	
62	022124	012705	000007	5\$:	MOV	#7, R5	;ALIGN IT TO LOOK LIKE HEADER	
63	022130	006311			ASL	(R1)		
64	022132	005305			DEC	R5		
65	022134	001375			BNE	5\$		
66	022136	032713	000400		BIT	#BIT8,(R3)	;TEST IF HEAD 1	
67	022142	001402			BEQ	6\$;NO - SKIP	
68	022144	052711	000100		BIS	#BIT6,(R1)	;INSERT HEAD BIT	
69	022150	042713	177400	6\$:	BIC	#177400,(R3)	;CLEAR ALL BUT SECTOR	
70	022154	052321		7\$:	BIS	(R3)+,(R1)+	;INSERT SECTOR NUMBER	
71	022156	020327	005466		CMP	R3,#IBUFF+508.	;CHECK IF IBUFF EMPTY	
72	022162	001344			BNE	3\$;NO GET NEXT CYLINDER	
73	022164	022737	000044	003134	CMP	#36., TEMP6	;DONE CHECKING ALL BSF's YET?	
74	022172	001470			BEQ	15\$;BRANCH IF YES, ELSE	
75	022174	012737	004076	003132	8\$:	MOV	#FLDBSF, TEMPS	;CHANGE POINTERS TO FIELD BS FILE
76	022202	012737	000044	003134	MOV	#36., TEMP6	;MAX SECTOR NUMBER	
77	022210	012737	000024	003116	MOV	#20., DESSEC	;SECTOR NUMBER START	
78	022216	000670			BR	2\$;DO READ	
79								
80	022220	005737	014514	9\$:	TST	BSERRS	;OUTPUT ALL BSF ERRORS?	
81	022224	001413			BEQ	11\$;BRANCH IF NO	
82	022226	012703	006563		MOV	#MFMTER,R3	;SET RESULT MESSAGE POINTER	
83	022232	104456			TRAP	C\$ERHRD		
	022234	002426			.WORD	1302		
	022236	000000			.WORD	0		
	022240	012646			.WORD	ERR1		
84	022242	005737	014514	10\$:	TST	BSERRS	;OUTPUT ALL BSF ERRORS?	
85	022246	001402			BEQ	11\$;BRANCH IF NO	
86	022250	104420			TRAP	C\$INLP		
87	022252	103652			BCS	2\$		
88								
89	022254	023737	003116	003134	11\$:	CMP	DESSEC, TEMP6	;CHECK IF ALL SECTORS READ
90	022262	001026			BNE	13\$;NO - SKIP	
91	022264	105237	003450		INC B	LOCERR	;BUMP LOCAL ERROR COUNTER	
92	022270	012703	006433		MOV	#MFBSF,R3	;SET ERROR MESSAGE POINTER	
93	022274	022737	004076	003132	CMP	#FLDBSF, TEMPS	;IS THIS FIELD BS FILE?	
94	022302	001002			BNE	12\$;BRANCH IF NO	
95	022304	012703	006510		MOV	#MUBSF,R3	;SET ERROR MESSAGE POINTER	
96	022310	012777	177777	160614	12\$:	MOV	#-1,@TEMPS	;TERMINATE FILE STORAGE
97	022316	104456			TRAP	C\$ERHRD		
	022320	002425			.WORD	1301		
	022322	000000			.WORD	0		
	022324	012646			.WORD	ERR1		
98	022326	022737	004076	003132	CMP	#FLDBSF, TEMPS	;DID WE CHECK FIELD BS FILE YET?	
99	022334	001407			BEQ	15\$;BRANCH IF YES, ELSE	
100	022336	000716			BR	8\$;GO CHECK FIELD BSF	
101								
102	022340	062737	000004	003116	13\$:	ADD	#4, DESSEC	;BUMP TO NEXT SECTOR
103	022346	000614			BR	2\$;GO DO READ	

104
105 022350 105237 003450 14\$: INC8 LOCERR ;INC LOCAL ERROR COUNT
106 022354 012737 000002 003020 15\$: MOV #2,ERRSHI ;SETUP FOR NO ERROR RETURN
107 022362 012737 000001 003500 MOV #1,BSFVAL ;SET BAD SEC FILE VALID FLAG
108 022370 105737 003450 TSTB LOCERR ;TEST IF LOCAL ERRORS
109 022374 001454 BEQ 17\$;NO - SKIP
110 022376 005237 003244 INC ERRCNT ;BUMP ERROR COUNT
111 022402 012737 177777 003500 16\$: MOV #1,BSFVAL ;SET BAD READ OR INVALID BAD SEC FILE
112 022410 012746 010572 MOV #BSFNOT,-(SP)
022414 012746 000001 MOV #1,-(SP)
022420 010600 MOV SP,RO
022422 104417 TRAP C\$PNTF
022424 062706 000004 ADD #4,SP
113 022430 005046 CLR -(SP)
022432 153716 003035 BIS8 RLORV+1,(SP)
022436 012746 006621 MOV #DRVNAME,-(SP)
022442 013746 003030 MOV RLBAS,-(SP)
022446 012746 006610 MCV #BASADD,-(SP)
022452 012746 011750 MOV #FMT5,-(SP)
022456 012746 000005 MOV #5,-(SP)
022462 010600 MOV SP,RO
022464 104417 TRAP C\$PNTF
022466 062706 000014 ADD #14,SP
114 022472 012746 011623 MOV #CRLF,-(SP)
022476 012746 000001 MOV #1,-(SP)
022502 010600 MOV SP,RO
022504 104417 TRAP C\$PNTF
022506 062706 000004 ADD #4,SP
115 022512 012737 177777 003502 MOV #-1,FCTBSF ;TERMINATE FACTORY BSF LIST
116 022520 012737 177777 004076 MOV #-1,FLDBSF ;TERMINATE FIELD BSF LIST
117 022526 000207 RTS PC ;RETURN

GLOBAL SUBROUTINES

```

1 ; READ HEADERS ROUTINE.
2
3 022530 012737 000001 003130 XROHOC: MOV #1, TEMP4 ;SET FLAG TO BYPASS REG STORAGE
4 022536 000402 BR XR0HDG ;GO DO IT
5
6 022540 005037 003131 XROHO: CLR TEMP4 ;SET FLAG TO SAVE T. AMO L. REGS
7 022544 010346 XROHDG: MOV R3,-(SP) ;STORE REGISTERS
8 022546 013703 003004 MOV SSINDX,R3 ;GET SUBROUTINE INDEX
9 022552 005723 TST (R3)+ ;BUMP IT FOR NEXT ENTRY
10 022554 016663 000002 002406 MOV 2(SP),SUBSTK(R3) ;INSERT THIS CALL
11 022562 162763 000004 002406 SUB #4, SUBSTK(R3) ;ADJUST IT TO CALLING LOCATION
12 022570 010337 003004 MOV R3,SSINDX ;STORE IT BACK
13 022574 010046 MOV R0,-(SP)
14 022576 010146 MOV R1,-(SP)
15 022600 010446 MOV R4,-(SP)
16 022602 012737 000002 003020 MOV #2,ERRSWI ;SET FOR NO ERROR RETURN
17 022610 005737 003130 TST TEMP4 ;TEST IF REGISTERS TO BE SAVED
18 022614 001007 BNE 2$ ;NO - SKIP
19 022616 012703 003046 MOV #L.MP+2,R3 ;SET POINTER FOR REGS
20 022622 012701 000004 MOV #4,R1 ;SET COUNT
21 022626 014346 1$: MOV -(R3),-(SP) ;SAVE REGISTER
22 022630 005301 DEC R1 ;DEC COUNT
23 022632 001375 BNE 1$ ;LOOP UNTIL ALL ARE SAVED
24 022634 004737 021230 2$: JSR PC, RDYC IK ;CHECK DRIVE READY
25 022640 023110 11$ ;CLEAR INTERRUPT FLAG
26 022642 005037 003010 CLR DONE ;GET ADDRESS OF LOAD REGS
27 022646 012701 003036 MOV #L.CS,R1 ;LOAD DRIVE NUMBER
28 022652 013711 003034 MOV RLDRV(R1)
29 022656 042711 002000 BIC #BIT10,(R1) ;CLEAR FOR DRIVE 4 - ? SPEC'D
30 022662 052721 000110 BIS #RDHEAD,(R1)+ ;INSERT COMMAND
31 022666 005021 CLR (R1)+ ;CLEAR BA
32 022670 005021 CLR (R1)+ ;CLEAR DA
33 022672 014162 000004 MOV -(R1),RLDA(R2) ;LOAD RL11 REGS
34 022676 014162 000002 MOV -(R1),RLBA(R2)
35 022702 014162 000000 MOV -(R1),RLCSR(R2)
36 022706 3$: ;TEST IN INTERRUPT FLAG SET
37 022720 005737 003010 TST DONE ;NO - SKIP
38 022724 001460 BEQ 9$ ;TEST IF DRIVE READY
39 022726 032737 000001 003046 4$: BIT #ORDYMSK,T.CS ;YES - SKIP
40 022734 001035 BNE 7$ ;SET NO READY MESSAGE
41 022736 012703 010702 MOV #MDRDY,R3 ;CONDITION OF AFTER DATA XFER
42 022742 012704 011604 MOV #CAFDT,R4
43 022746 104456 TRAP C$ERHRD
44 022750 023441 WORD 10017
45 022752 000000 WORD 0
46 022754 013100 WORD ERR5
47 022756 012701 000062 MOV #50,R1 ;SET WAIT COUNT FOR 5 SECONDS
48 022762 004737 017214 5$: JSR PC,G$STAT ;GET STATUS
49 022766 023104 10$ ;NO - SKIP
50 022770 032737 000001 003046 BIT #ORDYMSK,T.CS ;TEST IF DRIVE HAS COME READY
51 022776 001403 BEQ 6$ ;CLEAR ERROR SWITCH
52 023000 005037 003020 CLR ERRSWI ;SKIP
53 023004 000411 BR 7$ ;DEC WAIT COUNT
54 023006 005301 6$: DEC R1 ;LOOP UNTIL TIME DONE
55 023010 001364 BNE 5$ ;SET CONDITION AFTER 5 SECONDS
56 023012 012704 011615 MOV #C5SEC,R4

```

GLOBAL SUBROUTINES

```

55 023016 104456           TRAP    C$ERHRD
      023020 023436          .WORD   10014
      023022 000000          .WORD   0
      023024 013100          .WORD   ERR5
56 023026 000426          BR     10$      ;EXIT
57
58 023030 005737 003046    7$:   TST     T, CS
59 023034 100005          BPL    8$      ;CHECK FOR ANY ERRORS
60 023036 104456          TRAP   C$ERHRD
      023040 023440          .WORD   10016
      023042 000000          .WORD   0
      023044 013150          .WORD   ERR6
61 023046 000416          BR     10$      ;EXIT
62
63 023050 012701 003056    8$:   MOV     #HDWRD2,R1
64 023054 016221 000006          MOV     RLMP(R2),(R1)+ ;STORE LAST TWO HEADER WORDS
65 023060 016221 000006          MOV     RLMP(R2),(R1)+ ;NO - SKIP
66 023064 000411          BR     11$      ;EXIT
67
68 023066 004737 017010    9$:   JSR     PC_WAITIN ;WAIT FOR INTERRUPT
69 023072 012603          MOV     (SP)+,R3 ;GET RESULTS
70 023074 104456          TRAP   C$ERHRD
      023076 023437          .WORD   10015
      023100 000000          .WORD   0
      023102 012646          .WORD   ERR1
71 023104 005037 003020    10$:  CLR    ERRSWI ;CLEAR FOR ERROR RETURN
72 023110 005737 003130    11$:  TST    TEMP4 ;TEST IF REGISTERS WERE SAVED
73 023114 001007          BNE    13$      ;NO - SKIP
74 023116 012703 003036    MOV    #L_CS R3 ;SET POINTER TO RESTORE REGS
75 023122 012701 000004    MOV    #4_R1 ;SET COUNT
76 023126 012623          MOV    (SP)+,(R3)+ ;RESTORE REGISTER
77 023130 005301          DEC    R1 ;DEC COUNT
78 023132 001375          BNE    12$      ;LOOP UNTIL ALL ARE RESTORED
79 023134 162737 000002 003004 13$:  SUB    #2_SSINDX ;REMOVE ENTRY FROM SUBROUT STACK
80 023142 012604          MOV    (SP)+,R4 ;RESTORE REGS
81 023144 012601          MOV    (SP)+,R1
82 023146 012600          MOV    (SP)+,R0
83 023150 012603          MOV    (SP)+,R3
84 023152 005737 003020    TST    ERRSWI ;TEST IF ERROR RETURN
85 023156 001403          BEQ    14$      ;YES - SKIP
86 023160 063716 003020    ADD    ERRSWI,(SP) ;ADD IN ERROR RETURN
87 023164 000207          RTS    PC
88 023166 017616 000000    MOV    @(SP),(SP) ;SET ERROR RETURN ADDRESS
89 023172 000207          RTS    PC

```

```

1          ; VERIFY HEADERS ROUTINE. COMPARES 40 HEADERS FOR CONTENT AND
2          ; SEQUENCE.
3
4 023174 010346      VERHDR: MOV    R3,-(SP)      ; STORE REGS
5 023176 013703 003004      MOV    SSINDX,R3      ; GET SUBROUTINE INDEX
6 023202 005723      TST    (R3)+      ; BUMP IT FOR NEXT ENTRY
7 023204 016663 000002 002406      MOV    2(SP),SUBSTK(R3)  ; INSERT THIS CALL
8 023212 162763 000004 002406      SUB    #4,SUBSTK(R3)  ; ADJUST IT TO CALLING LOCATION
9 023220 010337 003004      MOV    R3,SSINDX      ; STORE IT BACK
10 023224 010446      MOV    R0,-(SP)
11 023226 010146      MOV    R1,-(SP)
12 023230 010446      MOV    R4,-(SP)
13 023232 010546      MOV    R5,-(SP)
14 023234 012737 000002 003020      MOV    #2,ERRSWI      ; SET FOR NO ERROR RETURN
15 023242 052737 000002 003006      BIS    #HDRCMP,OPFLAG  ; SET HEADER COMPARE FLAG
16 023250 005037 003016      CLR    MORECE      ; CLEAR MORE ERRORS FLAG
17 023254 012704 004472      MOV    #IBUFF,R4      ; SET POINTER TO HEADERS
18 023260 012705 003120      MOV    #TEMPO,R5      ; SET POINTER TO WORK AREA
19 023264 005003      CLR    R3      ; CLEAR FOR WORD COUNTER
20 023266 011415      MOV    (R4),(R5)      ; MOVE HDR WORD TO WORK AREA
21 023270 011401      MOV    (R4),R1      ; PUT WORD IN REG 1
22 023272 042701 000177      BIC    #177,R1      ; CLEAR ALL BUT CYLINDER
23 023276 012700 000007      MOV    $7,R0      ; SET SHIFT COUNT
24 023302 J06201      1$: ASR    R1      ; SHIFT
25 023304 005300      DEC    R0      ; DEC
26 023306 001375      BNE    1$      ; LOOP
27 023310 020137 003104      CMP    R1,NEWCYL      ; CHECK IF CYLINDER PART GOOD
28 023314 001407      BEQ    2$      ; YES - SKIP
29 023316 104456      TRAP   C$ERHRD
30 023320 023442      .WORD   10018
31 023322 000000      .WORD   0
32 023324 014242      .WORD   ERR10
33 023326 005037 003020      CLR    ERRSWI      ; CLEAR FOR ERROR RETURN
34 023332 000456      BR     8$      ;
35 023334 012701 000050      2$: MOV    #40,R1      ; SET HEADER COUNT
36 023340 042715 000100      BIC    #HDHSEL,(R5)  ; CLEAR HEAD SELECT AND 0 BIT
37 023344 005737 003114      TST    DESHD      ; ARE WE USING HD 0?
38 023350 001402      BEQ    3$      ; YES - SKIP
39 023352 052715 000100      BIS    #HDHSEL,(R5)  ; INSERT HEAD BIT
40 023356 005065 000002      CLR    2(R5)      ; CLEAR 2ND WORD OF WORK AREA
41 023362 021524      CMP    (R5),(R4)+  ; TEST FIRST WORD OK
42 023364 001410      BEQ    5$      ; YES - SKIP
43 023366 005744      TST    -(R4)      ; ELSE SET POINTER FOR ERROR
44 023370 104456      TRAP   C$ERHRD
45 023372 023442      .WORD   10018
46 023374 000000      .WORD   0
47 023376 014242      .WORD   ERR10
48 023400 005037 003020      CLR    ERRSWI      ; CLEAR FOR ERROR RETURN
49 023404 005724      TST    (R4)+  ; RESET POINTER
50 023406 005203      INC    R3      ; BUMP WORD COUNTER
51 023410 005724      TST    (R4)+  ; TEST 2ND WORD IS 0
52 023412 001410      BEQ    6$      ; YES - SKIP
53 023414 022544      CMP    (R5)+,-(R4)  ; ADJUST POINTERS FOR REPORT
54 023416 104456      TRAP   C$ERHRD
55 023420 023442      .WORD   10018
56 023422 000000      .WORD   0

```

023424	014242			WORD	ERR10		
50 023426	005037	003020		CLR	ERRSWI	;CLEAR FOR ERROR RETURN	
51 023432	024524			CMP	-(R5),(R4)+	;RESET POINTERS	
52 023434	005724		6\$:	TST	(R4)+	;BUMP PAST ECC WORD	
53 023436	005203			INC	R3	;BUMP WORD COUNTER	
54 023440	005215			INC	(R5)	;BUMP SECTOR OF EXPECTED HEADER	
55 023442	011500			MOV	(R5),R0	;MOVE EXPECTED HDR TO R0	
56 023444	042700	177700		BIC	#tCHDSEC,R0	;CLEAR ALL BUT SECTOR	
57 023450	022700	000050		CMP	#40.,R0	;TEST IF AT SECTOR 40	
58 023454	001002			BNE	7\$;NO - SKIP	
59 023456	042715	000077		BIC	#HDSEC,(R5)	;CLEAR SECTOR TO 0	
60 023462	005203		7\$:	INC	R3	;BUMP HDR WORD COUNTER	
61 023464	005301			DEC	R1	;DEC HEADER COUNT	
62 023466	001335			BNE	4\$;LOOP IF NOT YET DONE	
63 023470	162737	000002	003004	8\$:	SUB	#2,SSINDX	;REMOVE ENTRY FROM SUBROUT STACK
64 023476	012605			MOV	(SP)+,R5	;RESTORE REGISTERS	
65 023500	012604			MOV	(SP)+,R4		
66 023502	012601			MOV	(SP)+,R1		
67 023504	012600			MOV	(SP)+,R0		
68 023506	012603			MOV	(SP)+,R3		
69 023510	005737	003020		TST	ERRSWI	;TEST IF ERROR RETURN	
70 023514	001403			BEQ	9\$;YES - SKIP	
71 023516	063716	003020		ADD	ERRSWI,(SP)	;ADD IN ERROR RETURN	
72 023522	000207			RTS	PC		
73 023524	017616	000000	9\$:	MOV	a(SP),(SP)	;SET ERROR RETURN ADDRESS	
74 023530	000207			RTS	PC		
75				:	POSITION HEAD BIT FROM HEADER OR MULTIPURPOSE REGISTER TO LSB.		
76							
77 023532	013705	003054		POSHW1:	MOV HDWRD1,R5	;START FOR POSITION HD BIT IN WD 1	
78 023536	000402			BR	POSHD0	;SKIP	
79							
80 023544	010146			POSHSB:	MOV T,MP,R5	;START FOR POSITION HD BIT IN MP	
81 023546	042705	177677		POSHD0:	MOV R1,-(SP)	;STORE R1	
82 023552	012701	000006			BIC #tCHSSTAT,R5	;CLEAR ALL BUT HEAD SEL BIT	
83 023556	006205		1\$:	MOV	#6,R1	;SET SHIFT COUNT	
84 023560	005301			ASR	R5	;SHIFT FOR RIGHT JUSTIFY	
85 023562	001375			DEC	R1		
86 023564	012601			BNE	1\$		
87 023566	000207			MOV	(SP)+,R1	;RESTORE R1	
88				RTS	PC	;RETURN	
89							
90							
91							
92							
93 023570	010346			RDYWAIT:	MOV R3,-(SP)	;WAIT FOR READY ROUTINE. DURATION OF WAIT PASSED TO THE ROUTINE	
94 023572	013703	003004			MOV SSINDX,R3	FROM THE CALLING ROUTINE IN R1.	
95 023576	005723			TST	(R3)+	;SET SUBROUTINE INDEX	
96 023600	016663	000002	002406	MOV	2(SP),SUBSTK(R3)	;BUMP IT FOR NEXT ENTRY	
97 023606	162763	000004	002406	SUB	#4,SUBSTK(R3)	;INSERT THIS CALL	
98 023614	010337	003004		MOV	R3,SSINDX	;ADJUST IT TO CALLING LOCATION	
99						;STORE IT BACK	
100 023620	010046			MOV	R0,-(SP)		
101 023622	010146			MOV	R1,-(SP)		
102 023624	010446			MOV	R4,-(SP)		
103 023626	012737	000002	003020	1\$:	MOV #2,ERRSWI	;SET FOR NO ERROR RETURN	
104 023634	004737	017214		JSR	PC,GSTAT	;GET DRIVE STATUS	
105 023640	024010			6\$			
106 023642	032737	000001	003046	BIT	#ORDYMSK,T.CS	;CHECK IF READY	

107	023650	001061			BNE	7\$:YES - SKIP
108	023652	005301			DEC	R1	:DEC WAIT COUNT
109	023654	001406			BEQ	2\$:SKIP IF 0
110	023656	012737	000001	003456	MDV	#1,XDELAY	:SAVE ARGUMENT
	023664	004737	016210		JSR	PC,TIME	:CALL TIMING ROUTINE
111	023670	000761			BR	1\$	
112							
113	023672	012703	010702		2\$:	MOV #MDRDY,R3	:SET NAME MESSAGE PTR
114	023676	104456			TRAP C\$ERHRD		
	023700	023444			.WORD 10020		
	023702	000000			.WORD 0		
	023704	012762			.WORD ERR3		
115	023706	012701	000062		MDV #50,R1		:SET WAIT COUNT FOR 5 SECONDS
116	023712	004737	017214		3\$:	JSR PC,GSTAT	:GET DRIVE STATUS
117	023716	024010			6\$		
118	023720	032737	000001	003046	BIT #DRDYMSK,T.CS		:TEST IF DRIVE READY
119	023726	001016			BNE 4\$:YES - SKIP
120	023730	012737	000001	003460	MOV #1,YDELAY		:SAVE ARGUMENT
	023736	004737	016354		JSR PC,XTIME		:CALL TIMING ROUTINE
121	023742	005301			DEC R1		:DEC WAIT COUNT
122	023744	001362			BNE 3\$:LOOP UNTIL TIME DONE
123	023746	012704	011615		MDV #C5SEC,R4		:SET CONDITION AFTER 5 SECDS
124	023752	104456			TRAP C\$ERHRD		
	023754	023445			.WORD 10021		
	023756	000000			.WORD 0		
	023760	013100			.WORD ERR5		
125	023762	000410			BR 5\$:EXIT
126							
127	023764	032737	100000	003046	4\$:	BIT #ANYERR,T.CS	:TEST IF ANY ERROR SET
128	023772	001406			BEQ 6\$:NO - SKIP
129	023774	104456			TRAP C\$ERHRD		
	023776	023446			.WORD 10022		
	024000	000000			.WORD 0		
	024002	013150			.WORD ERR6		
130	024004	005337	003244		5\$:	DEC ERRCNT	:DEC FOR DOUBLE ERROR REPORT
131	024010	005037	003020		6\$:	CLR ERRSWI	:CLEAR FOR ERROR ERROR RETURN
132	024014	162737	000002	003004	7\$:	SUB #2,SSINDX	:REMOVE ENTRY FROM SUBROUT STACK
133	024022	012604			MOV (SP)+,R4		:RESTORE REGISTERS
134	024024	012601			MOV (SP)+,R1		
135	024026	012600			MOV (SP)+,R0		
136	024030	012603			MOV (SP)+,R3		:RESTORE R3
137	024032	005737	003020		TST ERRSWI		:TEST IF ERROR RETURN
138	024036	001403			BEQ 8\$:YES - SKIP
139	024040	063716	003020		ADD ERRSWI,(SP)		:ADD IN ERROR RETURN
140	024044	000207			RTS PC		
141	024046	017616	000000		8\$:	MOV a(SP),(SP)	:SET ERROR RETURN ADDRESS
142	024052	000207			RTS PC		
143							
144					:	GET POSITION ROUTINE. READS A HEADER FROM CURRENT CYLINDER	
145					:	(WHERE IT IS PRESENTLY POSITIONED) AND STORES CYLINDER	
146						NUMBER IN CURCYL.	
147	024054	010346			GETPOS: MOV R3,-(SP)		:STORE REGISTERS
148	024056	013703	003004		MOV SSINDX,R3		:GET SUBROUTINE INDEX
149	024062	005723			TST (R3)+		:BUMP IT FOR NEXT ENTRY
150	024064	016663	000002	002406	MOV 2(SP),SUBSTK(R3)		:INSERT THIS CALL
151	024072	162763	000004	002406	SUB #4,SUBSTK(R3)		:ADJUST IT TO CALLING LOCATION
152	024100	010337	003004		MOV R3,SSINDX		:STORE IT BACK

```

153 024104 010046      MOV    R0,-(SP)
154 024106 010546      MOV    R5,-(SP)
155 024110 004737 022540 JSR    PC,XRDHD   ;DO READ HEADER
156 024114 024144      2$    
157 024116 013703 003054 MOV    HDWRD1,R3  ;GET HEADER WORD
158 024122 012705 000007 MOV    #7,R5    ;SET SHIFT COUNT
159 024126 006203      ASR    R3      ;SHIFT TO RIGHT JUSTIFY
160 024130 005305      DEC    R5
161 024132 001375      BNE    1$    
162 024134 042703 177000 BIC    #177000,R3
163 024140 010337 003106 MOV    R3,CURCYL
164 024144 162737 000002 003004 2$:   SUB    #2,SSINDX ;STORE AS CURRENT CYLINDER
165 024152 012605      MOV    (SP)+,R5 ;REMOVE ENTRY FROM SUBROUT STACK
166 024154 012600      MOV    (SP)+,R0 ;RESTORE REGISTERS
167 024156 012603      MOV    (SP)+,R3
168 024160 005737 003020 TST    ERRSWI   ;TEST IF ERROR RETURN
169 024164 001403      BEQ    3$    ;YES - SKIP
170 024166 063716 003020 ADD    ERRSWI,(SP) ;ADD IN ERROR RETURN
171 024172 000207      RTS    PC
172 024174 017616 000000 3$:   MOV    0(SP),(SP) ;SET ERROR RETURN ADDRESS
173 024200 000207      RTS    PC

175
176 :     VERIFY POSITION ROUTINE. READS A HEADER (USING GETPOS) AND
177 :     CHECKS HEADS ARE POSITIONED AT NEW CYLINDER (CURCYL = NEWCYL).
178

179 024202 010346      VERPOS: MOV    R3,-(SP) ;STORE R3
180 024204 013703 003004 MOV    SSINDX,R3 ;GET SUBROUTINE INDEX
181 024210 005723      TST    (R3)+ ;BUMP IT FOR NEXT ENTRY
182 024212 016663 000002 002406 MOV    2(SP),SUBSTK(R3) ;INSERT THIS CALL
183 024220 162763 000004 002406 SUB    #4,SUBSTK(R3) ;ADJUST IT TO CALLING LOCATION
184 024226 010337 003004 MOV    R3,SSINDX ;STORE IT BACK
185
186 024232 012737 000002 003020 MOV    #2,ERRSWI ;SET FOR NO ERROR RETURN
187 024240 004737 024054 JSR    PC,GETPOS ;GET POSITION
188 024244 024272      1$    
189 024246 023737 003104 003106 CMP    NEWCYL,CURCYL ;CHECK IF CURRENT CYL IS NEW CYL
190 024254 001406      BEQ    1$    ;YES - SKIP
191 024256 104456      TRAP   C$ERHRD
192 024260 023446      .WORD   10022
193 024262 000000      .WORD   0
194 024264 014102      .WORD   ERR8
195 024266 005037 003020 CLR    ERRSWI ;CLEAR FOR ERROR RETURN
196 024272 162737 000002 003004 1$:   SUB    #2,SSINDX ;REMOVE ENTRY FROM SUBROUT STACK
197 024300 012603      MOV    (SP)+,R3 ;RESTORE R3
198 024302 005737 003020 TST    ERRSWI   ;TEST IF ERROR RETURN
199 024306 001403      BEQ    2$    ;YES - SKIP
200 024310 063716 003020 ADD    ERRSWI,(SP) ;ADD IN ERROR RETURN
201 024314 000207      RTS    PC
202 024316 017616 000000 2$:   MOV    0(SP),(SP) ;SET ERROR RETURN ADDRESS
203 024322 000207      RTS    PC

204 :     READ ALL HEADERS ROUTINE. 40 HEADERS ARE READ AND STORED
205 :     IN IBUFF.
206 024324 010346      RDALHD: MOV    R3,-(SP) ;STORE REGISTERS
207 024326 013703 003004 MOV    SSINDX,R3 ;GET SUBROUTINE INDEX
208 024332 005723      TST    (R3)+ ;BUMP IT FOR NEXT ENTRY

```

GLOBAL SUBROUTINES

```

209 024334 016663 000002 002406      MOV    2(SP),SUBSTK(R3) ;INSERT THIS CALL
210 024342 162763 000004 002406      SUB    #4, SUBSTK(R3) ;ADJUST IT TO CALLING LOCATION
211 024350 010337 003004               MDV    R3,SSINDEX ;STORE IT BACK
212 024354 010046                   MDV    R0,-(SP)
213 024356 010146                   MOV    R1,-(SP)
214 024360 010446                   MOV    R4,-(SP)
215 024362 012737 000002 003020      MOV    #2,ERRSWI ;SET FOR NO ERROR RETURN
216 024370 012701 000050               MOV    #40,R1 ;SET HEADER COUNT
217 024374 052737 100000 003006      BIS    #HDR0,OPFLAG ;SET 40 HDR OP FLAG
218 024402 012703 004472               MOV    #IBUFF,R3 ;SET POINTER TO STORE HDRS
219 024406 013704 003030               MOV    RLBAS,R4 ;GET BASE ADDRESS
220 024412 062704 000006               ADD    #RLMP,R4 ;MAKE IT POINT TO MP REG
221 024416 012737 000010 003036      MOV    #10,L.CS ;LOAD FOR READ HEADER. NO INTERRUPT
222 024424 053737 003034 003036      BIS    RLDRV,L.CS ;INSERT DRIVE NUMBER
223 024432 042737 002000 003036      BIC    #BIT10,L.CS ;CLEAR FOR DRIVE 4 7 SPEC'D
224 024440 005037 003040               CLR    L.BA ;CLEAR BA
225 024444 005037 003042               CLR    L.DA ;CLEAR DA
226 024450 005737 003114               TST    DESHD ;TEST IF HEAD 0
227 024454 001403               BEQ    1$ ;YES SKIP
228 024456 052737 000020 003042      BIS    #HDSEL,L.DA ;ELSE INSERT HEAD 0
229 024464 013762 003042 000004      MOV    L.DA,RLDA(R2) ;LOAD RLDA REG
230 024472 013762 003040 000002      MOV    L.BA,RLBA(R2) ;LOAD RLBA
231 024500 032762 000200 000000      BIT    #CRDYMSK,RLCS(R2) ;TEST IF CONTROLLER READY
232 024506 001003               BNE    2$ ;YES - SKIP
233 024510 004737 021230               JSR    PC,RDYCHK ;ELSE CHECK READY
234 024514 024632               6$ ;LOAD RLCS REG
235 024516 013762 003036 000000      MOV    #77777,R0 ;SET COUNT FOR WAIT
236 024524 012700 077777               MOV    #CRDYMSK,RLCS(R2) ;CHECK THAT OPERATION COMPLETED
237 024530 032762 000200 000000      BIT    3$ ;YES - SKIP
238 024536 001016               BNE    4$ ;DEC COUNT
239 024540 005300               DEC    R0 ;DEC COUNT
240 024542 001372               BNE    3$ ;SKIP IF NOT YET 0
241 024544 004737 016756               JSR    PC,READRL ;ELSE GET ALL REGISTERS
242 024550 004737 017010               JSR    PC,WAITIN ;ELSE WAIT FOR TIMEOUT
243 024554 012603               MOV    (SP)+,R3 ;GET RESULT MESSAGE POINTER
244 024556 104456               TRAP   C$ERHWD
245 024560 023451               .WORD  10025
246 024562 000000               .WORD  0
247 024564 012646               .WORD  ERR1
248 024566 005037 003020               CLR    ERRSWI ;CLEAR FOR ERROR RETURN
249 024572 000417               BR    6$ ;NO - SKIP
250 024574 005737 003046               4$: TST    T.CS ;TEST FOR ANY ERRORS
251 024600 100007               BPL    5$ ;NO - SKIP
252 024602 104456               TRAP   C$ERHWD
253 024604 023452               .WORD  10026
254 024610 000000               .WORD  0
255 024612 013150               .WORD  ERR6
256 024615 000405               CLR    ERRSWI ;CLEAR FOR ERROR RETURN
257 024620 011423               BR    6$ ;STORE HEADER WORDS
258 024622 011423               MOV    (R4),(R3)+ ;DEC HEADER COUNT
259 024624 011423               MOV    (R4),(R3)+ ;REMOVE ENTRY FROM SUBROUT STACK
260 024626 005301               DEC    R1
261 024630 001332               BNE    2$ ;BNE
262 024632 162737 000002 003004      SUB    #2,SSINDEX ;SUB

```

```

260 024640 012604      MOV  (SP)+,R4      ;RESTORE REGISTERS
261 024642 012601      MOV  (SP)+,R1
262 024644 012600      MOV  (SP)+,R0
263 024646 012603      MOV  (SP)+,R3
264 024650 005737 003020 TST  ERRSWI      ;TEST IF ERROR RETURN
265 024654 001403      BEQ  7$          ;YES - SKIP
266 024656 063716 003020 ADD   ERRSWI,(SP) ;ADO IN ERROR RETURN
267 024662 000207      RTS   PC
268 024664 017616 000000 7$:   MOV  @(SP),(SP) ;SET ERROR RETURN ADDRESS
269 024670 000207      RTS   PC

271
272 : GENERATE DATA ROUTINE. PATTERN TO BE GENERATED IS GIVEN
273 : IN THE WORD FOLLOWING THE CALL. 128 WORDS ARE GENERATED
274 :
275

276 024672 010146      DATGEN: MOV  R1,-(SP) ;STORE REGISTERS
277 024674 010346      MOV  R3,-(SP)
278 024676 010446      MOV  R4,-(SP)
279 024700 012701 005072 MOV  #OBUFF,R1      ;SET POINTER TO OBUFF
280 024704 012504      MOV  (R5)+,R4      ;GET DATA PATTERN SELECTOR
281 024706 006304      ASL   R4          ;ADJUST IT FOR INDEXING
282 024710 016403 002362 MOV  PATTBL(R4),R3 ;GET ADDRESS OF PATTERN
283 024714 011321      MOV  (R3),(R1)+ ;MOVE FIRST PATTERN WORD
284 024716 001421      BEQ  5$          ;SKIP IF PATTERN IS 0
285 024720 021327 177777 LDC  (R3),#-1   ;CHECK IF PATTERN IS ALL 1'S
286 024724 001416      LDC  5$          ;YES - SKIP
287 024726 020427 000010 LDC  R4,#8.     ;TEST IF PATTERN 5
288 024732 001403      BNE  3$          ;YES - SKIP
289 024734 020427 000020 CMP  R4,#16.    ;CHECK IF PATTERN 9 OR 10
290 024740 002413      BLT  6$          ;NO - SKIP
291 024742 005723      ST   (R3)+      ;BUMP SOURCE POINTER
292 024744 012321      MOV  (R3)+,(R1)+ ;MOVE TWO MORE WORDS FORM SOURCE
293 024746 012321      MOV  (R3)+,(R1)+ ;SET COUNT
294 024750 012704 000015 MOV  #13.,R4
295 024754 012703 005072 MOV  #OBUFF,R3      ;RESET POINTER
296 024760 030406      BR   8$          ;GO TO FILL

297
298 024762 012703 005072 5$:   MOV  #OBUFF,R3      ;ELSE SET OBUFF AS PATTERN SOURCE
299 024766 000401      BR   7$          ;GO TO FILL

300

301 024770 005723      6$:   TST  (R3)+      ;BUMP SOURCE POINTER
302 024772 012704 000017 7$:   MOV  #15.,R4      ;SET MOVE COUNT
303 024776 012321      8$:   MOV  (R3)+,(R1)+ ;MOVE 15 WORDS INTO BUFFER
304 025000 005304      DEC  R4
305 025002 001375      BNE  8$          ;REPEAT PATTERN IN BUFFER
306 025004 012703 005072 MOV  #OBUFF,R3      ;SET SOURCE TO TOP OF OBUFF
307 025010 012704 000160 MOV  #112.,R4      ;SET COUNT FOR REST OF BUFFER
308 025014 012321      10$:  MOV  (R3)+,(R1)+ ;RESTORE REGISTERS
309 025016 005304      DEC  R4
310 025020 001375      BNE  10$         ;RETURN
311 025022 012604      MOV  (SP)+,R4
312 025024 012603      MOV  (SP)+,R3
313 025026 012601      MOV  (SP)+,R1
314 025030 000205      RTS   R5

```

```

1          ; DATA COMPARE ROUTINE. COMPARES THE CONTENTS OF IBUFF AND OBUFF.
2          ; ERROR REPORTING IS LIMITED BY SOFTWARE PARAMETER.
3
4 025032 010346          DATCOM: MOV    R3,-(SP)      ;STORE R3
5 025034 013703 003004    MOV    SSINDEX,R3      ;GET SUBROUTINE STACK INDEX
6 025040 005723          TST    (R3),        ;BUMP INDEX TO NEXT ENTRY
7 025042 016663 000002 002406    MOV    2(SP),SUBSTK(R3) ;INSERT THIS CALL
8 025050 162763 000004 002406    SUB    #4,SUBSTK(R3) ;ADJUST IT TO CALLING LOCATION
9 025056 010337 003004          MOV    R3,SSINDEX ;STORE IT BACK
10 025062 010146          MOV    R1,-(SP)      ;STORE OTHER REGISTERS
11 025064 010446          MOV    R4,-(SP)
12 025066 010546          MOV    R5,-(SP)
13 025070 052737 000001 003006    BIS    #DATACMP,OPFLAG ;SET DATA COMPARE FLAG
14 025076 005037 003016          CLR    MORECE       ;CLEAR MORE ERROR FLAG
15 025102 012705 005072          MOV    #OBUFF,R5      ;SET POINTERS TO DATA FOR COMPARE
16 025106 012704 004472          MOV    #IBUFF,R4
17 025112 012703 000001          MJV    #1,R3        ;SET WORD COUNTER
18 025116 012701 000200          MOV    #128,,R1      ;SET COMPARE COUNT
19 025122 022425          1$:   CMP    (R4),,(R5),+ ;COMPARE DATA
20 025124 001052          BNE    6$           ;ERROR - SKIP TO REPORT
21 025126 005203          INC    R3            ;BUMP WORD COUNT
22 025130 005301          DEC    R1            ;DEC COMPARE COUNT
23 025132 001373          BNE    1$           ;LOOP IF NOT 0
24 025134 042737 000001 003006    BIC    #DATACMP,OPFLAG ;CLEAR DATA COMPARE FLAG
25 025142 005737 003020          TST    ERRSWI      ;TEST IF ANY COMPARE ERRORS
26 025146 001021          BNE    4$           ;NO - SKIP
27 025150 012701 000200          MOV    #128,,R1      ;SET REPORT VALUE
28 025154 010146          MOV    R1,-(SP)
29 025156 012746 011521          MOV    #RESET,-(SP)
30 025162 013746 003016          MOV    MOREC,E,-(SP)
31 025166 012746 010230          MOV    #TCERR,-(SP)
32 025172 012746 012615          MOV    #FMT27,-(8)
33 025176 012746 000005          MOV    #5,-(SP)
34 025202 010600          MOV    SP,RO
35 025204 104414          TRAP   C$PNTB
36 025206 062706 000014          ADD    #14,SP
37 025212 162737 000002 003004    SUB    #2,SSINDEX ;REMOVE ENTRY FROM SUBROUTINE STACK
38 025220 012605          MOV    (SP),+,R5      ;RESTORE REGS
39 025222 012604          MOV    (SP),+,R4
40 025224 012601          MOV    (SP),+,R1
41 025226 012603          MOV    (SP),+,R3
42 025230 005737 003020          TST    ERRSWI      ;TEST IF ERROR RETURN
43 025234 001403          BEQ    5$           ;YES - SKIP
44 025236 063716 003020          ADD    ERRSWI,(SP) ;ADD IN ERROR RETURN
45 025242 000207          RTS    PC
46 025244 017616 000000          5$:   MOV    @(SP),(SP) ;SET ERROR RETURN ADDRESS
47 025250 000207          RTS    PC
48 025252 023737 003016 014512    CMP    MORECE,DCLIMW ;TEST IF COMPARE ERRORS LIMIT EXCEEDED
49 025260 002011          BGE    7$           ;YES - SKIP
50 025262 024445          CMP    -(R4),-(R5) ;SET PTRS BACK TO ERROR WORDS
51 025264 104456          TRAP   C$ERHWD
52 025266 023463          .WORD  10035
53 025270 000000          .WORD  0
54 025272 014242          .WORD  ERR10
55 025274 005037 003020          CLR    ERRSWI      ;CLEAR ERROR SWITCH
56 025300 022425          CMP    (R4),,(R5),+ ;BUMP PTRS PAST ERROR WORDS
57 025302 000711          BR    . 2$           ;DO NEXT COMPARE

```

DC

CZRL4C0 RL01/02 DRIVE TEST 3 MACRO V05.03b Monday 06-Jan-86 00:23 Page 56-1
GLOBAL SUBROUTINES

SEQ 0092

47
48 025304 005237 003016 7\$: INC MORECE
49 025310 000706 BR 2\$;BUMP ERROR COUNTER
 ;DO NEXT COMPARE

GLOBAL SUBROUTINES

```

1          ,      WRITE AND READ DATA ROUTINE.
2
3 025312 012737 177777 003122 XWRITT: MOV    @-1 TEMP1      ;SET SPECIAL WRITE FOR TIMING FLAG
4 025320 000402           BR     XWRITI: CLR    TEMP1
5                                     ;CLEAR SPECIAL WRITE FLAG
6 025322 005037 003122 XWRITE:  MOV    @WTDATA,TEMP7 ;SET FOR WRITE
7 02532F 012737 000112 003136 XWRIT1: MOV    HLMWT,CURCYL ;TEST IF CYLINDER MAX (BAD SEC)
8 025334 023737 002304 003106 CMP    BNE   1$       ;NO SKIP
9 025342 001006           TST    DESHD
10 025344 005737 003114           BEQ   1$       ;TEST IF HEAD 1 (BAD SECTOR FILES)
11 025350 001403           BIS    #BADADD,OPFLAG ;NO SKIP
12 025352 052737 004000 003006 BIS    #BADADD,OPFLAG ;SET BAD ADDRESS FLAG
13 025360 000403           1$:  BR     XREADG
14                                     ;SKIP TO EXECUTE
15 025362 012737 000114 003136 XREAD:  MOV    #RDATA,TEMP7 ;SET FOR READ
16 025370 010346           XREADG: MDV    R3,-(SP) ;STORE R3
17 025372 013703 003004           MOV    SSINDX,R3 ;SET SUBROUTINE INDEX
18 025376 005723           TST    (R3)+ ;BUMP TO NEXT STACK ENTRY
19 025400 016663 000002 002406 MOV    2(SP),SUBSTK(R3) ;INSERT THIS CALL
20 025406 162763 000004 002406 SUB    #4,SUBSTK(R3) ;ADJUST TO POINT TO CALL
21 025414 010337 003004           MOV    R3,SSINDX ;STORE IT BACK
22 025420 010446           MOV    R0,-(SP)
23 025422 010146           MOV    R1,-(SP) ;STORE OTHER REGISTERS
24 025424 010446           MOV    R4,-(SP)
25 025426 004731 021230 JSR    PC,RDYCHK ;CHECK IF DRIVE READY
26 025432 026064           14$: 
27 025434 012703 003036           MOV    #L.CS,R3 ;GET ADDRESS OF LOAD REGS
28 025440 013713 003136           MOV    TEMP7,(R3) ;SET COMMAND
29 025444 053713 003034           BIS    RLDRV,(R3) ;INSERT DRIVE NUMBER
30 025450 042713 002000           BIC    #BIT10,(R3) ;CLEAR FOR DRIVE 4 - 7 SPEC'D
31 025454 032723 000004           BIT    #BIT2,(R3)+ ;TEST IF WRITE DATA
32 025460 001403           BEQ   1$       ;YES - SKIP
33 025462 012723 004472           MOV    #IBUFF,(R3)+ ;ELSE SET BA FOR READ
34 025466 000402           BR     2$       ;SET BA FOR WRITE
35
36 025470 012723 005072           1$:  MOV    #OBUFF,(R3)+ ;GET CURRENT CYLINDER
37 025474 013713 003106           2$:  MOV    CURCYL,(R3) ;ALIGN IT IN DA
38 025500 012704 000007           3$:  MOV    #7,R4
39 025504 006313           ASL    (R3)
40 025506 005304           DEC    R4
41 025510 001375           BNE   3$       ;TEST IF HEAD 0
42 025512 005737 003114           TST    DESHD ;YES - SKIP
43 025516 001402           BEQ   4$       ;SET FOR HEAD 1
44 025520 052713 000100           BIS    #HSMRK,(R3) ;INSERT DESIRED SECTOR
45 025524 053723 003116           BIS    DESSEC,(R3)+ ;INSERT WORD COUNT
46 025530 012713 177600           MOV    #177600,(R3) ;IS THIS BSF CYLINDER?
47 025534 023737 003106 002304 CMP    CURCYL,HLMWT ;NO - SKIP
48 025542 001004           BNE   5$       ;TEST IF HEAD 1
49 025544 005737 003114           TST    DESHD ;NO - SKIP
50 025550 001401           BEQ   5$       ;MAKE WORD COUNT 2 SECTORS
51 025552 006313           ASL    (R3) ;CHECK IF SPECIAL WRITE FOR TIMING
52 025554 005737 003122           5$:  TST    TEMP1 ;NO - SKIP
53 025560 001402           BEQ   6$       ;ELSE SET FOR 1 WORD TRANSFER
54 025562 012713 177777           MOV    #177777,(R3) ;TEST IF BAD ADDRESS FLAG SET
55 025566 032737 004000 003006 6$:  BIT    #BADADD,OPFLAG ;NO - SKIP
56 025574 001414           BEQ   7$       ;CLEAR ALL BUT THIS FLAG
57 025576 042737 173777 003006           BIC    #CBADADD,OPFLAG

```

58	025604	012703	011423		MOV	\$MWRTAB,R3	;SET RESULT MESSAGE POINTER	
59	025610	'04456			TRAP	C\$ERHRO		
	025612	023460			.WORD	10032		
	025614	000000			.WORD	0		
	025616	012646			.WORD	ERR1		
60	025620	005037	003006		CLR	OPFLAG	;CLEAR ALL FLAGS	
61	025624	000515			BR	13\$		
62								
63	025626	005037	003010	7\$:	CLR	DONE	;CLEAR INTERRUPT FLAG	
64	025632	005737	003122		TST	TEMP1	;CHECK IF SPECIAL WRITE FLAG SET	
65	025636	001112			BNE	14\$;YES - DO NOT START WRITE	
66	025640	011362	000006		MOV	(R3),RLMP(R2)	;LOAD RL REGS	
67	025644	014362	000004		MOV	-(R3),RLDA(R2)		
68	025650	014362	000002		MOV	-(R3),RLBA(R2)		
69	025654	014362	000000		MOV	-(R3),RLCS(R2)		
70	025660	012737	005670	003456	8\$:	MOV	#3000,XDELAY	;SAVE ARGUMENT
	025666	004737	016210		JSR	PC,TIME	;CALL TIMING ROUTINE	
71	025672	005737	003010		TST	DONE	;CHECK IF INTERRUPT	
72	025676	001010			BNE	9\$;YES - SKIP	
73	025700	004737	017010		JSR	PC,WAITIN	;WAIT FOR INTERRUPT	
74	025704	012603			MOV	(SP)+,R3	;GET RESULT MESSAGE	
75	025716	104456			TRAP	C\$ERHRO		
	025710	023456			.WORD	10030		
	025712	000000			.WORD	0		
	025714	012646			.WORD	ERR1		
76	025716	000460			BR	13\$		
77								
78	025720	032737	000001	003046	9\$:	BIT	\$ORDYMSK,T.CS	;TEST IF DRIVE READY
79	025726	001033			BNE	11\$;YES - SKIP	
80	025730	012703	010702		MOV	\$MDRDY,R3	;SET RESULT MESSAGE	
81	025734	012704	011604		MOV	#CAFDT,R4	;CONDITION AFTER DATA XFER	
82	025740	104456			TRAP	C\$ERHRO		
	025742	023460			.WORD	10032		
	025744	000000			.WORD	0		
	025746	013100			.WORD	ERR5		
83	025750	012701	000062		MOV	#50,R1	;SET WAIT COUNT FOR 5 SECDS	
84	025754	004737	017214	10\$:	JSR	PC,G\$STAT	;GET DRIVE STATUS	
85	025760	026060			13\$			
86	025762	032737	000001	003046	BIT	\$DROYMSK,T.CS	;TEST IF DRIVE READY NOW	
87	025770	001012			BNE	11\$;YES - SKIP	
88	025772	005301			DEC	R1	;DEC WAIT COUNT	
89	025774	001367			BNE	10\$;LOOP IF NOT TIME DONE	
90	025776	012704	011615		MOV	#C5SEC,R4	;SET CONDITION 5 SECONDS	
91	026002	104456			TRAP	C\$ERHRO		
	026004	023461			.WORD	10033		
	026006	000000			.WORD	0		
	026010	013100			.WORD	ERR5		
92	026012	005037	003020		CLR	ERRSWI	;CLEAR ERROR SWITCH	
93	026016	005737	003046	11\$:	TST	T.CS	;CHECK IF ANY ERROR	
94	026022	100020			BPL	14\$;NO - SKIP	
95	026024	023737	003106	002304	CMP	CURCYL,HLMTW	;IS THIS BSF CYLINDER?	
96	026032	001006			BNE	12\$;NO - SKIP	
97	026034	005737	003114		TST	DESHD	;TEST IF HEAD 1	
98	026040	001403			BEQ	12\$;NO - SKIP	
99	026042	005737	014514		TST	BSERRS	;OUTPUT ALL BSF ERRORS?	
100	026046	001404			BEQ	13\$;NO - SKIP	

```

101 026050      12$:    TRAP    C$ERHRO
102 026050 104456 .WORD   10031
103 026052 023457 .WORD   0
104 026054 000000 .WORD   ERR6
105 026056 013150 .WORD
106 026060 005037 003020 13$:    CLR     ERRSWI
107 026064 162737 000002 003004 14$:    SUB    $2,SSINDEX ;CLEAR ERROR SWITCH
108 026072 012604  MOV    (SP)+,R4 ;REMOVE ENTRY FROM SUBROUT STACK
109 026074 012601  MDV    (SP)+,R1 ;RESTORE REGISTERS
110 026076 012600  MDV    (SP)+,R0
111 026100 012603  MOV    (SP)+,R3
112 026102 005737 003020  TST     ERRSWI ;TEST IF ERROR RETURN
113 026106 001403  BEQ    15$ ;YES - SKIP
114 026110 063716 003020  ADD    ERRSWI,(SP) ;ELSE ADD IN ERROR RETURN
115 026114 000207  RTS    PC
116 026116 017616 000000 15$:    MOV    @(SP),(SP) ;ADJUST FOR ERROR RETURN
117 026122 000207  RTS    PC
118 026124 010046      ;BAO SECTOR CHECK ROUTINE. CHECKS IF SECTOR SPECIFIED IN CURCYL,
119 026126 010146      ;DESHD, AND DESSEC IS LISTED AS BAO IN THE BAO SECTOR FILES.
120 026130 010346      ;:
121 026132 005037 003022 BSCHK: MOV    R0,-(SP) ;STORE REGISTERS
122 026136 012703 003502 MOV    R1,-(SP)
123 026142 022713 177777 MOV    R3,-(SP)
124 026146 001005      CLR    BSFLAG ;CLEAR FLAG
125 026150 012703 004076 1$:    MOV    #FLDBSF,R3 ;GET POINTER TO FACTORY FILE
126 026154 022713 177777 CMP    #-1,(R3) ;CHECK IF ALL ONES
127 026160 001431      BNE    2$ ;NO SKIP TO TEST
128 026162 013700 003104 1$:    MOV    #FLDBSF,R3 ;ELSE SET POINTER TO FIELD BS FILE
129 026166 012701 000007 2$:    CMP    #-1,(R3) ;CHECK IF ALL ONES
130 026172 006300      BEQ    8$ ;YES - EXIT
131 026174 005301      MOV    NEWCYL,R0 ;BUILD HEADER OF ADDRESS IN QUESTION
132 026176 001375      MOV    #7,R1 ;# OF POSITIONS TO SHIFT CYLINDER
133 026200 005737 003114 3$:    ASL    R0 ;SHIFT NUMBER
134 026204 001402      DEC    R1 ;DONE YET?
135 026206 052700 000100 BNE    3$ ;NO, ANOTHER SHIFT PLEASE
136 026212 053700 003116 TST    DESHO ;CHECK IF HEAD 0
137 026216 022300      BEQ    4$ ;YES - SKIP
138 026220 001402      BIS    #BIT6,R0 ;INSERT HEAD 1
139 026222 101005      BIS    DESSEC,R0 ;INSERT SECTOR
140 026224 000774      4$:    CMP    (R3)+,R0 ;DID WE FIND AN ENTRY MATCH?
141 026226 012737 000001 003022 5$:    BEQ    6$ ;YES - EXIT
142 026234 000403      BR    5$ ;NO - FOUND FILE TERMINATOR
143 026236 020327 004076 6$:    MOV    #1,BSFLAG ;NEITHER TRY NEXT ENTRY...
144 026242 003742      BR    8$ ;SET ERROR FLAG
145 026244 012603      7$:    CMP    R3,#FLDBSF ;GO TO EXIT
146 026246 012601      BLE    1$ ;DONE BOTH FILES?
147 026250 012600      MOV    (SP)+,R3 ;NO, GO DO FIELD FILE
148 026252 005737 003022 8$:    MOV    (SP)+,R1 ;ELSE RESTORE REGISTERS
149 026256 001003      MOV    (SP)+,R0
150 026260 062716 000002 TST    BSFLAG ;CHECK IF ERROR
151 026264 000207      BNE    9$ ;YES - SKIP
152 026264 000207      ADD    #2,(SP) ;ELSE BUMP ERROR RETURN
153 026264 000207      RTS    PC

```

```

154 026266 017616 000000      9$:    MOV    @($P),($P)      ;SET FOR ERROR RETURN
155 026272 000207      RTS    PC
157
158      ; REPORT OPERATION ROUTINE. PRINTS SUBROUTINE TRACE SEQUENCE AND
159      ; OPERATION BEING PERFORMED PORTION OF ALL
160      ; ERROR MESSAGES.
161
162 026274 010446      RPTOP. MOV    R4,-($P)
163 026276 005737 003004      TST    SS$INDX      ;TEST SUBROUTINE INDEX 0
164 026302 001433      BEQ    2$          ;SKIP IF 0
165 026304 012704 000002      MOV    #2,R4        ;SET INDEXER TO FIRST ENTRY
166 026310 012746 010174      MOV    #SEQMES,-($P)
167 026314 012746 012134      MOV    #FMT9,-($P)
168 026320 012746 000002      MOV    #2,-($P)
169 026324 010600      MOV    SP,RO
170 026326 104414      TRAP   C$PNTB
171 026330 062706 000006      ADD    #6,SP
172 026334 016446 002406      1$:    MOV    SUBSTK(R4),-($P)
173 026340 012746 012307      MOV    #FMT16,-($P)
174 026344 012746 000002      MOV    #2,($P)
175 026350 010600      MOV    SP,RO
176 026352 104414      TRAP   C$PNTB
177 026354 062706 000006      ADD    #6,SP
178 026360 062704 000002      ADD    #2,R4        ;BUMP INDEX
179 026364 020437 003004      CMP    R4,SS$INDX      ;CHECK IF ALL PRINTED
180 026370 003761      BLE    1$          ;LOOP IF NOT ALL PRINTED YET
181
182 026372 012746 007150      2$:    MOV    #TSTLAB,-($P)
183 026376 013746 003014      MOV    ERHEAD,-($P)
184 026402 012746 011737      MOV    #FMT4,-($P)
185 026406 012746 000003      MOV    #3,-($P)
186 026412 010600      MOV    SP,RO
187 026414 104414      TRAP   C$PNTB
188 026416 062706 000010      ADD    #10,SP
189 026422 042737 030000 003006      BIC    #SEEKOP|RORWOP,OPFLAG      ;CLEAR SK & RD OR WRT FLAG
190 026430 013701 003036      MOV    L,CS,R1      ;GET COMMAND EXECUTED
191 026434 042701 177741      BIC    #177741,R1      ;STRIP ALL BUT FUNCTION CODE
192 026440 022701 000006      CMP    #6,R1        ;TEST IF SEEK OPERATION
193 026444 001003      BNE    3$          ;NO - SKIP
194 026446 052737 010000 003006      BIS    #SEEKOP,OPFLAG      ;ELSE SET SEEK FLAG
195 026454 022701 000012      3$:    CMP    #12,R1        ;TEST IF WRITE
196 026460 001003      BNE    4$          ;NO - SKIP
197 026462 052737 020000 003006      BIS    #RORWOP,OPFLAG      ;SET RD OR WRT FLAG
198 026470 022701 000014      4$:    CMP    #14,R1        ;TEST IF READ
199 026474 001003      BNE    5$          ;NO - SKIP
200 026476 052737 020000 003006      BIS    #RORWOP,OPFLAG      ;SET RD OR WRT FLAG
201 026504 016146 002226      5$:    MOV    OPMMSG(R1),-($P)
202 026510 012746 006117      MOV    #MOPER,-($P)
203 026514 012746 011723      MOV    #FMT1,-($P)
204 026520 012746 000003      MOV    #3,-($P)
205 026524 010600      MOV    SP,RO
206 026526 104414      TRAP   C$PNTB
207 026530 062706 000010      ADD    #10,SP
208 026534 020127 000004      CMP    R1,#4        ;CHECK IF GET STATUS
209 026540 001007      BNE    6$          ;NO - SKIP

```

```

187 026542 032737 000010 003042      BIT    #ORSET,L.DA   ;TEST IF RESET INCLUDED
188 026550 001403 000016      BEQ    6$      ;NO - SKIP
189 026552 012701 000016      MOV    #16,R1    ;SET TO PRINT WITH RESET
190 026556 000436      BR     10$    
191
192 026560 032737 007777 003006 6$:   BIT    #CCMPOP,OPFLAG ;TEST IF ANY OTHER OPERATION
193 026566 001424 003006      BEQ    9$      ;NO - SKIP
194 026570 013704 000020      MOV    OPFLAG,R4  ;SET UP TO DETERMINE WHICH ONE
195 026574 012701 000020      MOV    #20,R1    ;PRESET THE POINTER
196 026600 032704 000001      BIT    #BIT00,R4  ;CHECK THE BIT
197 026604 001003      BNE    8$      ;IF SET - SKIP
198 026606 005721      TST    (R1)+   ;BUMP POINTER
199 026610 006204      ASR    R4    
200 026612 000772      BR     7$    
201
202 026614 016146 002226      MOV    OPMMSG(S(R1))-(SP)
203 026620 012746 011626      MOV    #FMTXT,-(SP)
204 026624 012746 000002      MOV    #2,-(SP)
205 026630 010600      MOV    SP,RO
206 026632 104414      TRAP   CPSPNTB
207 026634 062706 000006      ADD    #6,SP
208 026640 032737 100000 003006 9$:   BIT    #HDR40,OPFLAG ;TEST IF 40 HEADER OPERATION
209 026646 001415 000050      BEQ    11$      ;NO - SKIP
210 026650 012701      MOV    #50,R1    ;ELSE PRINT IT
211 026654 016146 002226      MOV    OPMMSG(S(R1))-(SP)
212 026660 012746 011626      MOV    #FMTXT,-(SP)
213 026664 012746 000002      MOV    #2,-(SP)
214 026670 010600      MOV    SP,RO
215 026672 104414      TRAP   CPSPNTB
216 026674 062706 000006      ADD    #6,SP
217 026700 000434      BR     12$      ;SKIP
218
219 026702 032737 010000 003006 11$:   BIT    #SEEKOP,OPFLAG ;TEST IF SEEK
220 026710 001430      BEQ    12$      ;NO - SKIP
221 026712 013746 003114      MOV    DESHD,-(SP)
222 026716 012746 010135      MOV    #HDWD,-(SP)
223 026722 013746 003112      MOV    DESSGN,-(SP)
224 026726 012746 010130      MOV    #SGNWID,-(SP)
225 026732 013746 003110      MOV    DESDIF,-(SP)
226 026736 012746 010122      MOV    #DIFWD,-(SP)
227 026742 013746 003102      MOV    OLDCYL,-(SP)
228 026746 012746 010153      MOV    #FRMWD,-(SP)
229 026752 012746 012155      MOV    #FMT13,-(SP)
230 026756 012746 000011      MOV    #11,-(SP)
231 026762 010600      MOV    SP,RO
232 026764 104414      TRAP   CPSPNTB
233 026766 062706 000024      ADD    #24,SP
234
235 026772 032737 020000 003006 12$:   BIT    #RORWOP,OPFLAG ;TEST IF READ OR WRITE SET
236 027000 001424 003116      BEQ    13$      ;NO - SKIP
237 027002 013746 003116      MOV    DESSEC,-(SP)
238 027006 012746 010141      MOV    #SECWD,-(SP)
239 027012 013746 003114      MOV    DESHD,-(SP)
240 027016 012746 010135      MOV    #HDWD,-(SP)
241 027022 013746 003106      MOV    CURCYL,-(SP)
242 027026 012746 010146      MOV    #CYLWD,-(SP)

```

```

027032 012746 012504      MOV    #FMT22,-(SP)
027036 012746 000007      MOV    #7,-(SP)
027042 010600              MOV    SP,R0
027044 104414              TRAP   C$PNTB
027046 062706 000020      ADD    #20,SP
215 027052 004737 027524    13$:   JSR    PC,CLRPARM      ;CLEAR PARAM TABLE
216 027056 012604          MOV    (SP)+,R4      ;RESTORE R4
217 027060 000207          RTS    PC
218
219
220
221 027062 010146          ; REPORT REASON ROUTINE
222 027064 010346          RPTRES: PRINTS REASON PORTION FOR ALL ERROR REPORTS.
223 027066 010446          MOV    R1,-(SP)      ;STORE R1
224 027070 012701 003064      MOV    R3,-(SP)      ;STORE R3
225 027074 012103          MOV    R4,-(SP)      ;STORE R4
226 027076 011146          MOV    #RESPARM,R1      ;GET START OF PARAM
027100 012746 006126      MOV    (R1)+,R3      ;GET NUMBER OF PARAM
027104 012746 011730      MOV    (R1),-(SP)
027110 012746 000003      MOV    #MRSLT,-(SP)
027114 010600              MOV    #FMT2,-(SP)
027116 104414              MOV    #3,-(SP)
027120 062706 000010      MOV    SP,R0
027124 021127 011274      TRAP   C$PNTB
027130 001453              ADD    #10,SP
027132 012704 012141      CMP    (R1),#MNDRST      ;TEST IF MESSAGE IS NO DRV STATUS
027136 022127 011267      BEQ    2$      ;YES - SKIP REST OF REPORT
027142 001002              MOV    #FMT11,R4      ;PRISET FOR FORMAT 11
027144 012704 012147      CMP    (R1)+,#MCYLOC      ;CHECK IF REPORTING CYLINDER LOC
027150 005303              BNE    1$      ;NO - SKIP
027152 001442              MOV    #FMT12,R4      ;ELSE CHANGE TO FORMAT 12
027154 012146              DEC    R3      ;DEC PARAM COUNT
027156 012746 011503      BEQ    2$      ;IF 0 - EXIT
027162 010446              MOV    (R1)+,-(SP)
027164 012746 000003      MOV    #RESE3,-(SP)
027170 010600              MOV    R4,-(SP)
027172 104414              MOV    #3,-(SP)
027174 062706 000010      MOV    SP,R0
027200 012146              TRAP   C$PNTB
027202 012746 011507      ADD    #10,SP
027206 010446              MOV    (R1)+,-(SP)
027210 012746 000003      MOV    #RESE4,-(SP)
027214 010600              MOV    R4,-(SP)
027216 104414              MOV    #3,-(SP)
027220 062706 000010      MOV    SP,R0
027224 162703 000002      TRAP   C$PNTB
027230 001413              ADD    #10,SP
027232 012146              SUB    #2,R3      ;DEC PARAM COUNT
027234 012746 011514      BEQ    2$      ;IF 0 - EXIT
027240 012746 011723      MOV    (R1)+,-(SP)
027244 012746 000003      MOV    #RESE5,-(SP)
027250 010600              MOV    #FMT1,-(SP)
027252 104414              MOV    #3,-(SP)
027254 062706 000010      MOV    SP,R0
0240 027260 012604          TRAP   C$PNTB
0241 027262 012603          ADD    #10,SP      ;RESTORE REGS
0242 027264 012601          MOV    (SP)+,R4
                                MOV    (SP)+,R3
                                MOV    (SP)+,R1

```

GLOBAL SUBROUTINES

```

243 027266 000207           RTS      PC      ;RETURN
244
245
246
247 027270           : REPORT PHYSICAL ADDRESS OF DEVICE UNDER TEST
248 027270 005046          RPTREM: AND ALL REGISTER CONTENTS.
249 027272 153716 003035
250 027276 012746 006621
251 027302 013746 003030
252 027306 012746 006610
253 027312 012746 011750
254 027316 012746 000005
255 027322 010600          CLR     -(SP)
256 027324 104414          BISB    RLDdrv+1,(SP)
257 027326 062706 000014          MOV     #DRVNAME,-(SP)
258
259
250 027332 012746 010135          MOV     RLBAS,-(SP)
251 027336 012746 010146          MOV     #BASADD,-(SP)
252 027342 012746 006724          MOV     #FMT5,-(SP)
253 027346 012746 006712          MOV     #5,-(SP)
250 027352 012746 006717          MOV     SP, R0
251 027356 012746 006705          TRAP    C$PNTB
252 027362 012746 011770          ADD     #14,SP
253
250 027372 010600          : REPORT RL111 REGISTERS
251 027374 104414          MOV     #HDWD,-(SP)
252 027376 062706 000020          MOV     #CYLWD,-(SP)
253
250 027402 013746 003044          MOV     #MPNAM,-(SP)
251 027406 013746 003040          MOV     #BANAM,-(SP)
252 027412 013746 003042          MOV     #DANAM,-(SP)
253 027416 013746 003036          MOV     #CSNAM,-(SP)
250 027422 012746 006731          MOV     #FMT6,-(SP)
251 027426 012746 012102          MOV     #7,-(SP)
252 027432 012746 000006          MOV     SP, R0
253 027436 010600          TRAP    C$PNTB
250 027440 104414          ADD     #20,SP
251 027442 062706 000016          MOV     L.MP,-(SP)
252 027446 013746 003114          MOV     L.BA,-(SP)
253 027452 013746 003106          MOV     L.DA,-(SP)
250 027456 013746 003054          MOV     L.CS,-(SP)
251 027462 013746 003050          MOV     #LAB1,-(SP)
252 027466 013746 003052          MOV     #FMT8,-(SP)
253 027472 013746 003046          MOV     #6,-(SP)
250 027476 012746 006744          MOV     SP, R0
251 027502 012746 012032          TRAP    C$PNTB
252 027506 012746 000010          ADD     #16,SP
253 027512 010600          MOV     DSHD,-(SP)
250 027514 104414          ADD     #10,-(SP)
251 027516 062706 000022          MOV     SP, R0
252 027522 000207          TRAP    C$PNTB
253
254
255 027524 010546          CLRPARM: CLR     R5,-(SP)      ;STORE R5
256 027526 012701 003064          MOV     #RESPARM,R1   ;GET ADDRESS OF BLOCK
257 027532 012705 000005          MOV     #5,R5        ;SET COUNT
258 027536 005021          1$:    CLR     (R1)+       ;CLEAR WORD

```

J6

CZRLNCO RL01/02 DRIVE TEST 3 MACRO V05.03b Monday 06 Jan 86 00:23 Page 57 7
GLOBAL SUBROUTINES

SEQ 0100

260 027540 005305	DEC R5	;DEC COUNT
261 027542 001375	BNE 1\$;LOOP UNTIL 0
262 027544 012701 003064	MOV #RESPARM,R1	;RESET POINTER
263 027550 012605	MOV (SP)+,R5	;RESTORE RS
264 027552 000207	RTS PC	
265		

GLOBAL SUBROUTINES

```

1          .TITLE CZRLNCO RL01/02 DRIVE TEST 3
2
3
4          .SBttl *TEST 1      **SEEK TIMING
5
6
7 027554          T1:::    MOV    #1,TSTNM   ;SAVE TEST NUMBER
8 027554          MOV    #P2T12E,ERHEAD ;SET ERROR HEADER
9
10 027562         ;CHECK FOR PRESENCE OF A P-CLOCK.. BYPASS TEST IF NOT AVAILABLE
11 027562         TST    CLKFLG    ;P-CLOCK?
12 027562         BNE    1$        ;BRANCH TO PERFORM TEST IF P-CLOCK IS PRESENT
13 027570         005737  003474
14 027570         TST    CLKFLG    ;P-CLOCK?
15 027574         001014  003240
16 027576         013746  003240
17 027602         012746  010364
18 027606         012746  000002
19 027612         010600  000000
20 027614         104417  000000
21 027616         062706  000006
22 027622         000137  031472
23 027626         004737  017146
24 027632         004737  017164
25 027636         031472
26 027640         012700  003142
27 027644         012701  000030
28 027650         005020
29 027652         005301
30 027654         001375
31 027656         005037  003234
32 027662         005037  003104
33 027666         004737  020112
34 027672         031472
35 027674         012701  005670
36 027700         004737  023570
37 027704         031472
38 027706         004737  024202
39 027712         031472
40 027714         004737  021504
41 027720         012700  003152
42 027724         012701  003154
43 027730         012703  003166
44 027734         012704  003170
45 027740         012737  000001  003104
46 027746         012737  000200  003236
47 027754         012737  000110  003140
48 027762         053737  003034  003140
49 027770         042737  002000  003140
50 027776         004737  020102
51 030002         031472
52 030004         013762  003042  000004
53 030012         013762  003036  000000
54 030020         010046
55 030034         005737  003010
56 030040         001011
57 030042         004737  017010
58 030046         012603

          1$:    JSR    PC,TSTINT  ;INITIALIZE TEST
          1$:    JSR    PC,GSTATR ;CLEAR DRIVE
          20$   MOV    #OFIN,R0   ;GET ADDRESS OF 1ST TIME VALUE
          20$   MOV    #24,,R1   ;SET COUNT FOR CLEAR
          20$   CLR    (R0)+    ;CLEAR TIMER STORAGE
          20$   DEC    R1
          20$   BNE    2$        ;CLEAR PASS COUNTER
          20$   CLR    PASCNT   ;POSITION HEADS AT 0
          20$   JSR    PC,XSEEK  ;DO SEEK
          20$   MOV    #3000,,R1  ;SET WAIT FOR 300 MS
          20$   JSR    PC,RDYWAIT ;WAIT FOR READY
          20$   JSR    PC,VERPOS  ;VERIFY POSITION
          20$   JSR    PC,CHOSHD  ;GO CHOSE HEAD
          20$   MOV    #OFOUT,,R0  ;SET PTRS FOR 1 CYL FWD OUTER TIMER
          20$   MOV    #OFOUTU,,R1
          20$   MOV    #OROUT,,R3
          20$   MOV    #OROUTU,,R4
          20$   MOV    #1,NEWCYL  ;SET NEWCYL TO CYL 1
          20$   MOV    #128,COUNT ;SET COUNTER FOR SEEK LOOP
          20$   MOV    #RDHEAD,TEMP8 ;BUILD READ HEADER COMMAND
          20$   BIS    RLDdrv,TEMP8
          20$   BIC    #BIT10,TEMP8
          20$   JSR    PC,XSEEKT  ;DO SEEK BUILD BUT DO NOT START
          20$   MOV    L.DA,RLDA(R2) ;LOAD RL REGISTERS
          20$   MOV    L.CS,RLCS(R2)
          20$   MOV    R0,-(SP)    ;STORE R0
          20$   TST    DONE      ;TEST IF INTERRUPT
          20$   BNE    5$        ;YES - SKIP
          20$   JSR    PC,WAITIN  ;WAIT FOR INTERRUPT
          20$   MOV    (SP)+,R3    ;GET MESSAGE POINTER

```

57	030050	104456		TRAP	C\$ERHRD		
	030052	002261		.WORD	1201		
	030054	000000		.WORD	0		
	030056	012646		.WORD	ERR1		
58	030060	000137	031472	JMP	20\$		
59							
60	030064	005737	003046	5\$:	TST	T,CS	:CHECK IF ANY ERRORS
61	030070	100006		BPL	6\$:NO - SKIP
62	030072	104456		TRAP	C\$ERHRD		
	030074	002262		.WORD	1202		
	030076	000000		.WORD	0		
	030100	013150		.WORD	ERR6		
63	030102	000137	031472	JMP	20\$		
64							
65	030106	005037	003010	6\$:	CLR	DONE	:CLEAR INTERRUPT FLAG
66	030112	005037	172542	CLR	\$@CLKCSB		:CLEAR CLOCK COUNT SET BUFFER
	030116	005037	172544	CLR	\$@CLKCTR		:CLEAR CLOCK COUNTER
	030122	012737	000023	MOV	\$23,\$@CLKCSR		:INITIALIZE CLOCK FOR COUNT-UP MODE,
67			172540				:/OF TIME INTERVAL
68	030130	013762	003140	000000	MOV	TEMP8,RLCS(R2)	:LOAD RL11 CONTROL AND STATUS REGISTER
69							:/TO INITIATE SEEK OPERATION
70	030136	012737	003720	003456	MOV	#2000,XDELAY	:SAVE ARGUMENT
	030144	004737	016210	JSR	PC,TIME		:CALL TIMING ROUTINE
71	030150	013705	172544	MOV	\$@CLKCTR,R5		:STORE CLOCK COUNTER CONTENTS
	030154	005037	172540	CLR	\$@CLKCSR		:EVENT FINISHED, STOP CLOCK
72	030160	012600		MOV	(SP)+,R0		:RESTORE R0
73	030162	013737	003140	003036	MOV	TEMP8,L,CS	:SET IF ERROR TO REPORT
74	030170	004737	024202	JSR	PC,VERPOS		:VERIFY POSITION
75	030174	031472			20\$		
76	030176	005737	003112	TST	DESSGN		:CHECK WHICH SEEK DIRECTION
77	030202	001403		BEQ	7\$:REVERSE - SKIP
78	030204	060510		ADD	R5,(R0)		:ADD TO FORWARD TOTAL
79	030206	005511		AOC	(R1)		:ADD IN OVERFLOW
80	030210	000402		BR	8\$:SKIP
81							
82	030212	060513		7\$:	ADD	R5,(R3)	:ADD TO REVERSE TOTAL
83	030214	005514		AOC	(R4)		:ADD IN OVERFLOW
84	030216	005337	003236	8\$:	DEC	COUNT	:DEC SEEK COUNT
85	030222	001403		BEQ	9\$:SKIP IF 0
86	030224	004737	021570	JSR	PC,ONSWAP		:ELSE SWAP OLD AND NEW CYL
87	030230	000662		BR	4\$:REDO SEEK LOOP
88							
89	030232	162710	000470	9\$:	SUB	#312.,(R0)	:SUB CONSTANT FOR READ HEADER TIME
90	030236	162713	000470		SUB	#312.,(R3)	
91	030242	012705	000006		MOV	#6,R5	:SET SHIFT COUNT TO DIVIDE BY 64
92	030246	000241		10\$:	CLC		:DIVIDE BOTH TOTALS BY 64
93	030250	006011			ROR	(R1)	
94	030252	006010			ROR	(R0)	
95	030254	000241			CLC		
96	030256	006014			ROR	(R4)	
97	030260	006013			ROR	(R3)	
98	030262	005305			DEC	R5	
99	030264	001370			BNE	10\$	
100	030266	005237	003234		INC	PASCNT	:BUMP PASS COUNT
101	030272	022737	000001	003234	CMP	\$1,PASCNT	:TEST IF PASS 1
102	030300	001051			BNE	13\$:NO - SKIP
103	030302	012737	000177	003104	MOV	\$127.,NEWCYL	:ELSE SET TO POSITION HDS TO 127

*TEST 1 **SEEK TIMING

SEQ 0104

161	030632	022737	000004	003234	15\$:	CMP	#4, PASCNT	; TEST IF PASS 4
162	030640	001041				BNE	17\$; NO - SKIP
163	030642	012737	000252	003104		MOV	#170, NEWCYL	; ELSE SET UP TO TIME 85 CYL SEEK
164	030650	022737	000001	002300		CMP	#1, T,DRIVE	; RL01?
165	030656	001403				BEQ	16\$; YES
165	030660	012737	000525	003104	16\$:	MOV	#341, NEWCYL	; NO - SET FOR RL02
167	030666	004737	020112			JSR	PC,XSEEK	; AT INNER LIMIT
168	030672	031472				20\$		
169	030674	012701	005670			MOV	#3000, R1	; SET WAIT COUNT FOR 300 MS
170	030700	004737	023570			JSR	PC,RDYWAIT	; WAIT FOR READY
171	030704	031472				20\$		
172	030706	004737	024202			JSR	PC,VERPOS	; VERIFY POSITION
173	030712	031472				20\$		
174	030714	012700	003172			MOV	#HFIN, R0	; SET POINTERS
175	030720	012701	003174			MOV	#HFINU, R1	
176	030724	012703	003202			MOV	#HRIN, R3	
177	030730	012704	003204			MOV	#HRINU, R4	
178	030734	013737	002304	003104		MOV	HLMTW, NEWCYL	; SET NEWCYL TO MAX CYL
179	030742	000434				BR	18\$; DO TIMING LOOP
180								
181	030744	022737	000005	003234	17\$:	CMP	#5, PASCNT	; TEST IF PASS 5
182	030752	001032				BNE	19\$; NO - SKIP
183	030754	005037	003104			CLR	NEWCYL	; ELSE SET UP TO TIME 256/512 CYL SEEK
184	030760	004737	020112			JSR	PC,XSEEK	; OVER ALL SURFACE
185	030764	031472				20\$		
186	030766	012701	005670			MOV	#3000, R1	; SET WAIT COUNT FOR 300 MS
187	030772	004737	023570			JSR	PC, RDYWAIT	; WAIT FOR DRIVE READY
188	030776	031472				20\$		
189	031000	004737	024202			JSR	PC,VERPOS	; VERIFY POSITION
190	031004	031472				20\$		
191	031006	012700	003212			MOV	#AFMID, R0	; SET POINTERS
192	031012	012701	003214			MOV	#AFMIDU, R1	
193	031016	012703	003216			MOV	#ARMID, R3	
194	031022	012704	003220			MOV	#ARMIDU, R4	
195	031026	013737	002304	003104		MOV	HLMTW, NEWCYL	; SET NEWCYL
196	031034	000137	027746		18\$:	JMP	3\$	
197								
198	031040				19\$:			
	031040	012746	007607			MOV	#VALDES, -(SP)	
	031044	012746	007553			MOV	#SKTMES, -(SP)	
	031050	012746	011730			MOV	#FMT2, -(SP)	
	031054	012746	000003			MOV	#3, -(SP)	
	031060	010600				MOV	SP, R0	
	031062	104417				TRAP	C\$PNTF	
	031064	062706	000010			ADD	#10, SP	
199	031070	005046				CLR	-(SP)	
	031072	153716	003035			SISB	RLDRV+1, (SP)	
	031076	012746	006621			MOV	#DRVNAME, -(SP)	
	031102	013746	003030			MOV	RLBAS, -(SP)	
	031106	012746	006610			MOV	#BASADD, -(SP)	
	031112	012746	011750			MOV	#FMT5, -(SP)	
	031116	012746	000005			MOV	#5, -(SP)	
	031122	010600				MOV	SP, R0	
	031124	104417				TRAP	C\$PNTF	
	031126	062706	000014			ADD	#14, SP	
200	031132	012746	007666			MOV	#LABEXP, -(SP)	
	031136	012746	007660			MOV	#ABOUT, -(SP)	

031142	012746	007651	MOV	\$LABMID,-(SP)
031146	012746	007643	MOV	\$LABIN,-(SP)
031152	012746	012342	MOV	#FMT18,-(SP)
031156	012746	000005	MOV	\$5,-(SP)
031162	010600		MOV	SP RO
031164	104417		TRAP	C\$PNTF
031166	062706	000014	ADD	\$14 SP
201 031172	013746	003222	MOV	EXOCYL,-(SP)
031176	013746	003152	MOV	OFOUT,-(SP)
031202	013746	003146	MOV	OFMID,-(SP)
031206	013746	003142	MOV	OFIN,-(SP)
031212	012745	007677	MOV	\$LABOCF,-(SP)
031216	012746	012374	MOV	#FMT19,-(SP)
031222	012746	000006	MOV	\$6,-(SP)
031226	010600		MOV	SP RO
031230	104417		TRAP	C\$PNTF
031232	062706	000016	ADD	\$16 SP
202 031236	013746	003222	MOV	EXOCYL,-(SP)
031242	013746	003166	MOV	OROUT,-(SP)
031246	013746	003162	MOV	ORMID,-(SP)
031252	013746	003156	MOV	ORIN,-(SP)
031256	012746	007711	MOV	\$LABOCR,-(SP)
031262	012746	012374	MOV	#FMT19,-(SP)
031266	012746	000006	MOV	\$6,-(SP)
031272	010600		MOV	SP RO
031274	104417		TRAP	C\$PNTF
031276	062706	000016	ADD	\$16 SP
203 031302	013746	003224	MOV	EXHCYL,-(SP)
031306	013746	003176	MOV	HFOUT,-(SP)
031312	013746	003172	MOV	HFIN,-(SP)
031316	012746	007723	MOV	\$LABHCF,-(SP)
031322	012746	012431	MOV	#FMT20,-(SP)
031326	012746	000005	MOV	\$5,-(SP)
031332	010600		MOV	SP RO
031334	104417		TRAP	C\$PNTF
031336	062706	000014	ADD	\$14 SP
204 031342	013746	003224	MOV	EXHCYL,-(SP)
031346	013746	003206	MOV	HRDUT,-(SP)
031352	013746	003202	MOV	HRIN,-(SP)
031356	012746	007737	MOV	\$LABHCR,-(SP)
031362	012746	012431	MOV	#FMT20,-(SP)
031366	012746	000005	MOV	\$5,-(SP)
031372	010600		MOV	SP RO
031374	104417		TRAP	C\$PNTF
031376	062706	000014	ADD	\$14 SP
205 031402	013746	003226	MOV	EXACYL,-(SP)
031406	013746	003212	MOV	AFMID,-(SP)
031412	012746	007753	MOV	\$LABACF,-(SP)
031416	012746	012461	MOV	#FMT21,-(SP)
031422	012746	000004	MOV	\$4,-(SP)
031426	010600		MOV	SP RO
031430	104417		TRAP	C\$PNTF
031432	062706	000012	ADD	\$12 SP
206 031436	013746	003226	MOV	EXACYL,-(SP)
031442	013746	003216	MOV	ARMID,-(SP)
031446	012746	007767	MOV	\$LABACR,-(SP)
031452	012746	012461	MOV	#FMT21,(SP)

• FST 1 ••SEEK TIMING

031456	012746	000004	MOV	4,-(SP)
031462	010600		MOV	SP, R0
031464	104417		TRAP	C\$BNTF
031466	062706	000012	ADD	#12, SP
207	031472		20\$:	
208	031472		L10023:	
031472	104401		TRAP	C\$ETST

Dg

CZRLNCO RL01/02 DRIVE TEST 3 MACRO V05.03b Monday 06-Jan-86 00:23 Page 59
*TEST 2 **BASIC READ DATA (BAD SECTOR FILE)

SEQ 0107

```

1          .SBTTL *TEST 2           **BASIC READ DATA (BAD SECTOR FILE)
2
3 031474
4 031474 012737 000002 003240 T2:::      MOV      #2,TSTNM      ;SAVE TEST NUMBER
5 031502 004737 017146          JSR      PC,TSTINT    ;INITIALIZE TEST
6 031506 004737 017164          JSR      PC,GSTATR   ;CLEAR DRIVE
7 031512 031524               1$                  ;ERROR RETURN ADDRESS
8 031514 005037 003500          CLR      BSFVAL      ;ENABLE BAD SEC FILE READ
9 031520 004737 021630          JSR      PC,RDBSF    ;READ BAD SECTOR FILE
10
11
12 031524               1$:                L10024:      TRAP     C$ETST
13 031524
14 031524

```

```

1           .SBTTL *TEST 3      **WRITE/READ DATA (PART 1)
2
3 031526   012737  000003  003210
4
5 031526   012737  007402  003014
6           T3:::          MOV    #3,TSTNM   ;SAVE TEST NUMBER
7           MOV    #P2T14E,ERHEAD ;SET ERROR HEADER
8           JSR    PC,TSTINT  ;INITIALIZE TEST
9           JSR    PC,GSTATR  ;CLEAR DRIVE
10          031522  031746
11          T3065$:
12          JSR    PC,CKBSVD ;GO CHECK IF BAD SECTOR FILES VALID
13          JSR    PC,CHOSHD ;GO CHOOSE HEAD
14          031560  004737  021614
15          CLR    DE$SEC   ;SECTOR 0
16          CLR    NEWCYL   ;CYLINDER 0
17          CLR    T310$    ;CLEAR PATTERN SELECT
18          JSR    PC,XSEEK  ;POSITION HEADS
19          031604  031746
20          T3065$:
21          JSR    PC,RDYWAIT;SET WAIT COUNT FOR 300 MS
22          031612  004737  023570 ;WAIT FOR READY
23          031616  031746
24          031620  004737  024202
25          031624  031746
26          031626  005037  031640
27          T3065$:
28          031632
29          031632
30          031632  104402  024672
31          031634  004537  024672
32          031640  000000  025322
33          031642  004737  025322
34          031646  031664
35          031650  004737  025362
36          031654  031664
37          031656  004737  025032
38          031662  031664
39          031664  012737  000002  003020
40          031672  104403
41          031672
42          031672
43          031672
44          031674  104410
45          031676  000050
46          031700  022737  000010  031640
47          031706  001403
48          031710  005237  031640
49          031714  000746
50          031716  004737  021530
51          031722  031746
52          031724  005037  031640
53          031730  004737  026124
54          031734  031740
55          031736  000720
56          031740  005237  003104
57          031744  000771
58          031746  104401
59
60          T310$:
61          L10026:          TRAP   C$BSUB
62          JSR    R5,DATGEN;GENERATE DATA
63          WORD  0          ;PATTERN SELECT WORD
64          JSR    PC,XWRITE;DO WRITE DATA
65          1$:             JSR    PC,XREAD ;DO READ DATA
66          1$:             JSR    PC,DATCOM;COMPARE DATA
67          1$:             MOV    #2,ERRSWI;INIT ERROR SWITCH
68          L10026:          TRAP   C$ESUB
69          TRAP   C$ESCAPE
70          WORD  L10025-;
71          CMP    #8.,T310$;WAS DATA PAT 8 USED?
72          BEQ    2$        ;YES - SKIP
73          INC    T310$    ;ELSE BUMP TO NEXT PATTERN
74          BR     T307$    ;DO TEST WITH NEW PATTERN
75          2$:             JSR    PC,SWAPHD;GO SWAP TO HEAD 1 OR END TEST
76          T3065$:
77          JSR    T310$    ;ABORT RETURN
78          CLR    PC,BSCHK ;SET PATTERN SELECT TO 0
79          JSR    PC,BSCHK ;CHECK IF SECTOR BAD
80          4$:             JSR    T306$    ;YES RETURN - SKIP TO 4$
81          BR     T306$    ;NO RETURN - DO TEST THIS SECTOR
82          4$:             INC    NEWCYL  ;BUMP TO NEXT CYLINDER
83          BR     3$        ;CHECK IF THIS ONE BAD
84
85          T3065$:
86          L10025:          TRAP   C$ETST

```

```
1 .SBTTL *TEST 4      **ROTATIONAL TIMING
2
3 031750 012737 000004 003240      T4::      MOV #4,TSTNM
4 031750 012737 007423 003014      MOV #P2T15E,ERHEAD ;SAVE TEST NUMBER
5
6 031756 012737          ;SET ERROR HEADER
7
8
9 031764 005737 003474      ;CHECK FOR PRESENCE OF A P-CLOCK.. BYPASS TEST IF NOT AVAILABLE
10 031770 001014          TST CLKFLG ;P-CLOCK?
11 031772 013746 003240      BNE 1$ ;BRANCH TO PERFORM TEST IF P-CLOCK IS PRESENT
12 031776 012746 010364      MOV TSTNM,-(SP)
13 032002 012746 003002      MOV #NOTST,-(SP)
14 032006 010600          MOV #2,-(SP)
15 032010 104417          MOV SP,R0
16 032012 062706 000006      TRAP C$PNTF
17
18 032016 104432          ADD #6,SP
19 032020 000542          ;/P-CLOCK IS NOT AVAILABLE"
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
```

032022 005003 1\$: CLR R3 ;CLEAR FOR TIMING STORAGE
032024 005004 CLR R4
032026 004737 017146 JSR PC,TSTINT ;INITIALIZE TEST
032032 004737 017164 JSR PC,GSTATR ;CLEAR DRIVE
032036 032554 8\$;CLEAR FOR TIMING STORAGE
032040 004537 024672 JSR R5,DATGEN ;GENERATE DATA
032044 000000 0 ;PATTERN 0
032046 005037 003116 CLR DESSEC ;CLEAR TO SECTOR 0
032052 004737 021504 JSR PC,CHOSHD ;GO SELECT HEAD
032056 013737 014502 003104 MOV LO1IMW,NEWCYL ;SET FOR CYLINDER
032064 004737 020112 JSR PC,XSEEK ;DO SEEK
032070 032554 8\$;CLEAR FOR TIMING STORAGE
032072 012701 005670 MOV #3000,,R1 ;SET WAIT FOR 300 MS
032076 004737 023570 JSR PC,RDYWAIT ;WAIT FOR READY
032102 032554 8\$;CLEAR FOR TIMING STORAGE
032104 004737 024202 JSR PC,VERPOS ;VERIFY POSITION
032110 032554 8\$;CLEAR FOR TIMING STORAGE
032112 012701 000100 MOV #64,,R1 ;SET LOOP COUNTER
032116 012705 003044 2\$: MOV #L,MP,R5 ;SET A POINTER
032122 004737 025312 JSR PC,XWRITT ;DO FIRST WRITE
032126 032554 8\$;CLEAR FOR TIMING STORAGE
032130 011562 000006 MOV (R5),RLMP(R2) ;LOAD RL REGISTERS
032134 014562 000004 MOV -(R5),RLDA(R2)
032140 014562 000002 MOV -(R5),RLBA(R2)
032144 014562 000000 MOV -(R5),RLCS(R2)
032162 005737 003010 TST DONE ;TEST IF INTERRUPT
032166 001011 BNE 3\$;YES - SKIP
032170 004737 017010 JSR PC,WAITIN ;ELSE WAIT FOR TIMEOUT
032174 012603 MOV (SP)+,R3 ;GET MESSAGE POINTER
032176 104456 TRAP C\$ERHRD
032200 002735 .WORD 1501
032202 000000 .WORD 0
032204 012646 .WORD ERR1
032206 000137 032554 47 JMP 8\$;TEST IF ANY ERRORS
032212 005737 003046 48 TST T,CS ;TEST IF ANY ERRORS
032216 100006 BPL 4\$;NO - SKIP
032220 104456 TRAP C\$ERHRD

*TEST 4 **ROTATIONAL TIMING

032222	002736		.WORD	1502	
032224	000000		.WORD	0	
032226	013150		.WORD	ERR6	
52	032230	000137	032554	JMP	8\$
53					
54	032234	012705	003044	4\$:	MOV #L_MP,R5 ;SET POINTER TO RL LOAD REGS
55	032240	005037	003010	CLR	DONE ;CLEAR INTERRUPT INDICATOR
56	032244	005037	172542	CLR	#CLKCSB ;CLEAR CLOCK COUNT SET BUFFER
	032250	005037	172544	CLR	#CLKCTR ;CLEAR CLOCK COUNTER
	032254	012737	000023	MOV	#23,#CLKCSR ;INITIALIZE CLOCK FOR COUNT-UP MODE, ;/OF TIME INTERVAL
57			172540		
58	032262	011562	000006	MOV	(R5),RLMP(R2) ;LOAD RL REGISTERS FOR 2ND WRITE
59	032266	014562	000004	MOV	-(R5),RLDA(R2)
60	032272	014562	000002	MOV	-(R5),RLBA(R2)
61	032276	014562	000000	MOV	-(R5),RLCS(R2)
62	032302	01237	005670	MOV	#3000,XDELAY ;SAVE ARGUMENT
	032310	004737	016210	JSR	PC,TIME ;CALL TIMING ROUTINE
63	032314	013700	172544	MOV	#CLKCTR,RO ;STORE CLOCK COUNTER CONTENTS
	032320	005037	172540	CLR	#CLKCSR ;EVENT FINISHED, STOP CLOCK
64	032324	005737	003010	TST	DONE ;TEST IF INTERRUPT OCCURRED
65	032330	001010		BNE	5\$;YES - SKIP
66	032332	004737	017010	JSR	PC,WAITIN ;GO WAIT FOR INTERRUPT
67	032336	012603		MOV	(SP)+,R3 ;GET MESSAGE POINTER
68	032340	104456		TRAP	C\$ERHARD
	032342	002737		.WORD	1503
	032344	000000		.WORD	0
	032346	012646		.WORD	ERR1
69	032350	000501		BR	8\$
70					
71	032352	005737	003046	5\$:	TST T_CS ;TEST IF ANY ERROR
72	032356	100005		BPL	6\$;NO - SKIP
73	032360	104456		TRAP	C\$ERHARD
	032362	002740		.WORD	1504
	032364	000000		.WORD	0
	032366	013150		.WORD	ERR6
74	032370	000471		BR	8\$
75					
76	032372	060003		6\$:	ADD R0,R3 ;ADD IN TIME USED
77	032374	005504		ADC	R4 ;DOUBLE PRECISION
78	032376	005301		DEC	R1 ;DEC LOOP COUNTER
79	032400	001246		BNE	2\$;LOOP UNTIL 0
80	032402	012701	000006	MOV	#6,R1 ;SET DIVIDE COUNT
81	032406	000241		7\$:	CLC ;CLEAR CARRY FOR DIVIDE
82	032410	006004		ROR	R4 ;DIVIDE SUM BY 100(8)
83	032412	006003		ROR	R3
84	032414	005301		DEC	R1 ;DEC DIVIDE COUNT
85	032416	001373		BNE	7\$;LOOP UNTIL DONE
86	032420	012746	007607	MOV	#VALDES,-(SP)
	032424	012746	007565	MOV	#SRTMES,-(SP)
	032430	012746	011730	MOV	#FMT2,-(SP)
	032434	012746	000003	MOV	#3,-(SP)
	032440	010600		MOV	SP, R0
	032442	104417		TRAP	C\$PNTF
	032444	062706	000010	ADD	#10,SP
87	032450	005046		CLR	-(SP)
	032452	153716	003035	BISB	RLDRV+1,(SP)
	032456	012746	006621	MOV	#DRVNAME,-(SP)

H4

CZRLNCO RL01/02 DRIVE TEST 3 MACRO V05.03b Monday 06 Jan 86 00:23 Page 61 2
*TEST 4 **ROTATIONAL TIMING

SFC 011.

032462	013746	003030		MOV	RLBAS,-(SP)	
032466	012746	006610		MOV	#BASADD,-(SP)	
032472	012746	011750		MOV	#FMT5,-(SP)	
032476	012746	000005		MOV	#5,-(SP)	
032502	010600			MOV	SP, R0	
032504	104417			TRAP	C\$PNTF	
032506	062706	000014		ADD	#14, SP	
88 032512	013746	003230		MOV	EXR0T,-(SP)	
032516	012746	007633		MOV	#MAPR0X,-(SP)	
032522	012746	011507		MOV	#RESE4,-(SP)	
032526	010346			MOV	R3,-(SP)	
032530	012746	011503		MOV	#RESE3,-(SP)	
032534	012746	012571		MOV	#FMT26,-(SP)	
032540	012746	000006		MOV	#6,-(SP)	
032544	010600			MOV	SP, R0	
032546	104417			TRAP	C\$PNTF	
89 032550	062706	000016	003020 8\$: L10027:	ADD	#16, SP	
90 032554	012737	000002		MOV	#2, ERSWI	;INITIALIZE ERROR SWITCH
032562	104401			TRAP	C\$EST	

```

1 .SBTTL *TEST 5      **WRITE/READ DATA (PART 2)
2
3 032564          T5::      MOV #5,TSTNM   ;SAVE TEST NUMBER
4 032564 012737 000005 003240  MOV #P2T16E,ERHEAD ;SET ERROR HEADER
5 032572 012737 007446 003014  JSR PC,TSTINT  ;INITIALIZE TEST
6 032600 004737 017146          JSR PC,GSTATR ;CLEAR DRIVE
7 032604 004737 017164          T3165$:
8 032610 033700          MOV PC,CKBSVD ;GO CHECK IF BAD SECTOR FILES VALID
9 032612 004737 021614          CLR PASCNT  ;CLEAR PASS TO 0
10 032616 005037 003234         MOV #-2,R5   ;SET
11 032622 012705 177776         TST PASNUM  ;TEST IF FIRST PASS (QUICK VERIFY)
12 032626 005737 003444         BNE 1$      ;NO - SKIP
13 032632 001006          BIT #ALLCYL,MISWIW ;TEST IF USE ALL CYLINDERS
14 032634 032737 000001 014500  BNE 1$      ;YES - SKIP
15 032642 001002          BNE #16..,R5   ;ELSE SET PEOPLE TO NEG 8
16 032644 012705 177760          MOV #16..,R5
17
18 032650          1$:        MOV #T33TBL,R1 ;GET ADDRESS OF WORK TABLE
19 032654 012701 002506          MOV #10,JUNK ;SET CLEAR COUNT
20 032662 013721 C14502         MOV LOLIMW,(R1)+;CLEAR LOCATIONS TO LO LIMIT
21 032666 005337 002302         DEC JUNK    ;DEC COUNT
22 032672 001373          BNE 2$      ;LOOP UNTIL 0
23 032674 013737 014504 002512  MOV HILIMW,T33TBL+4;INSERT HILIMIT
24 032702 013737 014504 002514  MOV HILIMW,T33TBL+6;INTO APPROPRIATE LOCATIONS
25 032710 013737 014504 002516  MOV HILIMW,T33TBL+10
26
27 032716 062705 000002          T3100$: ADD #2,R5   ;BUMP R5 BY 2
28 032722 032737 000001 014500  BIT #ALLCYL,MISWIW ;TEST IF USE ALL CYLINDERS
29 032730 001031          BNE 5$      ;YES - SKIP
30 032732 005737 003444         TST PASNUM  ;TEST IF FIRST PASS (QUICK VERIFY)
31 032736 001002          BNE 1$      ;NO - S'!?
32 032740 062705 000016         ADD #16,R5  ;ELSE BUMP CYLINDER POINTER BY 7
33 032744 022737 000001 002300  CMP #1,†.DRIVE ;RL01 OR RL02? THAT IS THE Q
34 032752 001404          BEQ 2$      ;ANS IS RL01
35 032754 020527 000244         CMP R5,#164.
36 032760 103013          BHIS 4$      ;TEST PAST TABLE-YES EXIT
37 032762 000403          BR 3$      ;TEST PAST TABLE-YES EXIT
38
39 032764 020527 000122         2$:  CMP R5,#82. ;TEST PAST THE TABLE
40 032770 103007          BHIS 4$      ;GET NEXT TABLE ENTRY
41 032772 016537 002606 002302  MOV CYLTBL(R5),JUNK ;CLEAR UPPER BYTE
42 033000 043737 002306 002302  BIC CLRBYT,JUNK
43 033006 001007          BNE 6$      ;TEST IF ALL CYLINDERS USED
44 033010 000137 033700         4$:  JMP T3165$ ;YES - EXIT TEST
45
46 033014 023705 014504         5$:  CMP HILIMW,R5 ;USE R5 AS NEXT CYLINDER
47 033020 001773          BEQ 4$      ;CHECK IF LOWER THAN LOLIMIT
48 033022 010537 002302         MOV R5,JUNK ;YES - SKIP
49 033026 023737 002302 014502  CMP JUNK,LOLIMW ;CHECK IF HIGHER THAN HILIMIT
50 033034 103730          BLO T3100$ ;YES - SKIP
51 033036 023737 002302 014504  CMP JUNK,HILIMW
52 033044 101324          BHI T3100$ ;TEST IF ALL CYLINDERS USED
53 033046 012703 002546         MOV #TBT,R3 ;YES - EXIT TEST
54 033052 013713 002302         MOV JUNK,(R3)
55 033056 013763 002302 000002  MOV JUNK,2(R3)
56 033064 013763 002302 000004  MOV JUNK,4(R3)
57 033072 013763 002302 000006  MOV JUNK,6(R3)

```

62 033100 013763 002302 000010 MOV JUNK,10(R3)
 63 033106 013763 002302 000012 MOV JUNK,12(R3)
 64 033114 010337 003026 MOV R3,TBLSTR
 65 033120 004737 021504 JSR PC,CHOSHD ;STORE TABLE ADDRESS
 66
 67 033124 T3101\$: ;GO CHOSE HEAD
 033124 T5.1:
 033124 104402 TRAP C\$BSUB
 68 033126 042737 003760 003006 BIC #MQUALS,OPFLAG ;CLEAR ALL MESSAGE QUALIFIERS
 69 033134 005737 003234 TST PASCNT ;TEST IF PASS 0
 70 033140 001414 BEQ 2\$;YES - SKIP
 71 033142 023727 003234 000003 CMP PASCNT,#3 ;TEST IF PASS 3
 72 033150 001404 BEQ 1\$;YES - SKIP
 73 033152 002407 BLT 2\$;CHECK IF LESS THAN 3, IF YES CLEAR TO 0
 74 033154 012737 000003 003234 MOV #3,PASCNT ;ELSE SET TO 3
 75 033162 052737 000020 003006 BIS \$INOUTS,OPFLAG ;SET MESSAGE QUAL
 76 033170 000405 BR 3\$;SKIP
 77
 78 033172 005037 003234 2\$: CLR PASCNT ;SET PASS COUNT TO 0
 79 033176 052737 000040 003006 BIS #OUTINS,OPFLAG ;SET MESSAGE QUAL
 80 033204 012737 000003 003024 3\$: MOV #3,WRTSWI ;SET READ AND WRITE SWITCH
 81 033212 013703 003026 MOV TBLSTR,R3 ;GET STORED TABLE ADDRESS
 82 033216 012701 002506 MOV #T33TBL,R1
 83 033222 012703 002546 MOV #TBT,R3
 84 033226 005037 003116 4\$: CLR DESSÉC ;CLEAR TO SECTOR 0
 85 033232 012137 003104 MOV (R1)+,NEWCYL ;GET NEXT TABLE ENTRY
 86 033236 004737 020112 JSR PC,XSEEK ;DO SEEK
 87 033242 033606 15\$
 88 033244 012701 005670 MDV #3000,,R1 ;SET WAIT COUNT FOR 300 MS
 89 033250 004737 023570 JSR PC,RDYWAIT ;WAIT FOR READY
 90 033254 033606 15\$
 91 033256 012337 003104 MOV (R1)+,NEWCYL ;GET NEXT TABLE ENTRY
 92 033262 004737 020112 JSR PC,XSEEK ;DO SEEK
 93 033266 033606 15\$
 94 033270 012701 005670 MOV #3000,,R1 ;SET WAIT COUNT FOR 300 MS
 95 033274 004737 023570 JSR PC,RDYWAIT ;WAIT FOR READY
 96 033300 033606 15\$
 97 033302 004737 024202 JSR PC,VERPOS ;VERIFY POSITION
 98 033306 033606 15\$
 99 033310 004737 026124 5\$: JSR PC,BSCHK ;CHECK FOR BAD SECTOR
 100 033314 033446 9\$: ;"YES" RETURN
 101 033316 013737 003116 033336 MOV DESSEC,6\$;SET DATA PATTERN = TO SECTOR NUMBER
 102 033324 042737 177770 033336 BIC #177770,6\$;CLEAR ALL BUT LSD
 103 033332 004537 024672 JSR R5,DATGEN ;GO GENERATE DATA
 104 033336 000000 .WORD 0
 105 033340 032737 000001 003024 6\$: BIT #BIT0,WRTSWI ;TEST IF WRITE THIS PASS
 106 033346 001425 BEQ 7\$;NO - SKIP
 107 033350 004737 025322 JSR PC,XWRITE ;DO WRITE
 108 033354 033606 15\$
 109 033356 005237 003116 INC DESSEC ;INC SECTOR
 110 033362 022737 000050 003116 CMP #40.,DESSEC ;TEST IF ALL SECTORS USED
 111 033370 001347 BNE 5\$;NO - SKIP
 112 033372 042737 000060 003006 BIC \$INOUTS!OUTINS,OPFLAG ;CLEAR QUALIFIERS
 113 033400 042737 000001 003024 BIC #BIT0,WRTSWI ;CLEAR WRITE REQUIRED SWITCH
 114 033406 052737 000100 003006 BIS #FOLWRT,OPFLAG ;SET FOLLOWING WRITE QUALIFIER
 115 033414 005037 003116 CLR DESSEC ;CLEAR TO SECTOR 0
 116 033420 000733 BR 5\$;SKIP

```

117
118 033422 032737 000002 003024 7$: BIT #BIT1,WRTSWI ;TEST IF READ THIS PASS
119 033430 001414 BEQ 10$ :NO - SKIP
120 033432 004737 025362 8$: JSR PC,XREAD ;ELSE DO READ
121 033436 033606 15$ :
122 033440 0C1737 025032 15$ :
123 033444 033606 15$ :
124 033446 005237 003116 9$: INC DESSEC ;BUMP SECTOR
125 033452 022737 000050 003116 CMP #40.,DESSEC ;TEST IF ALL SECTORS USED
126 033460 001313 BNE 5$ :NO - LOOP
127 033462 005037 003116 10$: CLR DESSEC ;CLEAR DESIRED SECTOR
128 033466 005037 003024 CLR WRTSWI ;CLEAR WRITE/READ SWITCH
129 033472 005237 003234 INC PASCNT ;BUMP PASS COUNT
130 033476 042737 003760 003006 BIC #MQUALS,OPFLAG ;CLEAR ALL QUALIFIERS
131 033504 023727 003234 000003 CMP PASCNT,#3 ;TEST IS PASS 3
132 033512 001435 BEQ 15$ :YES SKIP
133 033514 023727 003234 000006 CMP PASCNT,#6 ;TEST IF PASS 6
134 033522 001431 BEQ 15$ :YES - SKIP
135 033524 012737 000002 003024 MOV #BIT1,WRTSWI ;SET READ REQUIRED BIT
136 033532 023727 003234 000001 CMP PASCNT,#1 ;TEST IF PASS 1
137 033540 001415 BEQ 13$ :YES - SKIP
138 033542 023727 003234 000005 CMP PASCNT,#5 ;TEST IF PASS 4
139 033550 001411 BEQ 13$ :YES - SKIP
140 033552 000404 BR 12$ :SKIP
141
142 033554 052737 002000 003006 11$: BIS #FWDSKO,OPFLAG ;SET FWD QUALIFIER
143 033562 000407 BR 14$ ;GO DO NEXT PASS
144
145 033564 052737 000020 003006 12$: BIS #INOUTS,OPFLAG ;SET QUALIFIER
146 033572 000403 BR 14$ ;SKIP
147
148 033574 052737 000040 003006 13$: BIS #OUTINS,OPFLAG ;SET MESSAGE QUALIFIER
149 033602 000137 033226 14$: JMP 4$ ;GO DO NEXT PASS
150
151 033606 012737 000002 003020 15$: MOV #2,ERRSWI ;INIT ERROR SWITCH
152 033614 033614 L10031: TRAP C$ESUB
153
154 033616 104410 TRAP C$ESCAPE
033620 000060 .WORD L10030-
155 033622 012737 000003 003024 MOV #3,WRTSWI ;SET FOR READ AND WRITE REQ.
156 033630 023727 003234 000003 CMP PASCNT,#3 ;TEST IF PASS 3
157 033636 001004 BNE 16$ :NO - SKIP
158 033640 012737 002514 003026 MOV #T33TBL+6,TBLSTR ;STORE MID POINT IN TABLE
159 033646 000410 BR 17$ ;GO START PASS 4
160
161 033650 005037 003234 16$: CLR PASCNT ;CLEAR TO PASS 0
162 033654 004737 021530 JSR PC,SWAPHD ;GO SWAP TO HEAD 1 OR END TEST
163 033660 032716 T3100$ ;ABORT RETURN
164 033662 012737 002506 003026 MOV #T33TBL,TBLSTR ;STORE START OF TABLE
165 033670 062703 000006 17$: ADD #6,R3
166 033674 000137 033124 JMP T3101$ ;TRAP C$ETST
167
168 033700 T3165$: ;L10030:
033700 033700 104401 TRAP C$ETST

```

```

1           .SBTTL *TEST 6      **WRITE LOCK ERROR AND DATA PROTECTION
2
3 033702   012737  000006  003240    T6:::          MOV    #6,TSTNM   ;SAVE TEST NUMBER
4 033702   012737  003444          TST    PA$NUM   ;TEST IF FIRST PASS
5 033710   005737          BNE    1$       ;NO - SKIP
6 033714   001003          TST    MISWIW   ;TEST IF RUN MANUAL INTERVENTION
7 033716   005737  014500          BMI    2$       ;YES - SKIP
8 033722   100402          JMP    T3265$  ;EXIT TST
9 033724   000137  034724          1$:          MOV    #6,TSTNM   ;SAVE TEST NUMBER
10          T6.1:          TST    PA$NUM   ;TEST IF FIRST PASS
11          2$:          BNE    1$       ;NO - SKIP
12          T6.1:          JSR    PC,TSTINT ;INITIALIZE TEST
13 033730   033730  104402          JSR    PC,GSTATR;CLEAR DRIVE
14 033732   012737  007467  003014          TRAP   C$BSUB
15 033740   004737  017146          MOV    #P2T17E,ERHEAD;SET ERROR HEADER
16 033744   004737  017164          JSR    PC,TSTINT ;INITIALIZE TEST
17 033750   034572          JSR    PC,GSTATR;CLEAR DRIVE
18 033752   005037  003114          CLR    DESHD    ;SET TO HEAD 0
19 033756   005037  003116          CLR    DESSEC   ;SET TO SECTOR 0
20 033762   005037  003104          CLR    NEWCYL   ;CLEAR TO CYLINDER 0
21 033766   004737  020112          JSR    PC,XSEEK ;DO SEEK
22 033772   034572          11$:          MOV    #6000.,R1   ;INITIALIZE WAIT COUNT
23 033774   012701  013560          JSR    PC,ROYWAIT;WAIT FOR READY
24 034000   004737  023570          11$:          JSR    PC,VERPOS  ;VERIFY POSITION
25 034004   034572          JSR    PC,VERPOS  ;VERIFY POSITION
26 034006   004737  024202          JSR    PC,VERPOS  ;VERIFY POSITION
27 034012   034572          JSR    PC,VERPOS  ;VERIFY POSITION
28 034014   032737  020000  003054          BIT    #WLSTAT,T.MP ;TEST IF WRITE LOCK SET
29 034022   001116          BNE    4$       ;YES - SKIP
30 034024   004537  024672          JSR    R5,DATGEN ;GENERATE DATA
31 034030   000007          7          JSR    PC,XWRITE  ;PATTERN 7
32 034032   004737  025322          JSR    PC,XWRITE  ;WRITE DATA
33 034036   034572          JSR    PC,XREAD   ;READ DATA
34 034040   004737  025362          JSR    PC,XREAD   ;READ DATA
35 034044   034572          JSR    PC,DATCOM  ;CHECK DATA
36 034046   004737  025032          JSR    PC,DATCOM  ;CHECK DATA
37 034052   034572          JSR    PC,DATCOM  ;CHECK DATA
38 034054   005046          CLR    -(SP)    ;(SP)
39 034056   153716  003035          BISB   RLDRV+1,(SP)
40 034062   012746  006621          MOV    #DRVNAME,-(SP)
41 034066   013746  003030          MOV    RLBAS,-(SP)
42 034072   012746  006610          MOV    #BASADD,-(SP)
43 034076   012746  010056          MOV    #OPR1A,-(SP)
44 034102   012746  010105          MOV    #OPR004,-(SP)
45 034106   012746  011631          MOV    #FMTOP1,-(SP)
46 034112   012746  000007          MOV    #7,-(SP)
47 034116   010600          MOV    SP,R0
48 034120   104417          TRAP   C$PNTF
49 034122   062706  000020          ADD    #20,SP
50 034126   012701  000024          MOV    #20.,R1   ;INITIALIZE WAIT COUNT
51 034132          3$:          JSR    PC,GSTATR ;GET STATUS
52 034144   004737  017164          11$:          BIT    #WLSTAT,T.MP ;CHECK IF WRITE LOCK SET
53 034150   034572          BNE    4$       ;YES - SKIP
54 034152   032737  020000  003054          MOV    #BELL,-(SP)
55 034160   001037          MOV    #FMTXT,-(SP)
56 034166   012746  011626          MOV    #FMTXT,-(SP)

```

034172	012746	000002		MOV	#2,-(SP)		
034176	010600			MOV	SP, R0		
034200	104417			TRAP	C\$PNTF		
034202	062706	000006		ADD	#6, SP		
46 034206	005301			DEC	R1	:DEC COUNT	
47 034210	001350			BNE	3\$:SKIP IF NOT 0	
48 034212	005046			CLR	-(SP)		
034214	153716	003035		BISB	RLDRV+1,(SP)		
034220	012746	010056		MOV	#OPR1A,-(SP)		
034224	012746	010..51		MOV	#BYPSONM,-(SP)		
034230	012746	007467		MOV	#P2T17E,-(SP)		
034234	012746	012540		MOV	#FMT23,-(SP)		
034240	012746	000005		MOV	#5,-(SP)		
034244	010600			MOV	SP, R0		
034246	104417			TRAP	C\$PNTF		
034250	062706	000014		ADD	#14, SP		
49 034254	104432			TRAP	C\$EXIT		
034256	000446			.WORD	L10032-		
50							
51 034260	004537	024672	4\$:	JSR	R5,DATGEN	:GENERATE DATA	
52 034264	000001				1	:PATTERN 1	
53 034266	012705	003036		MOV	#L.CS,R5	:GET ADDRESS OF L REGS	
54 034272	012715	000112		MOV	#WTDATA,(R5)	:LOAD WRITE COMMAND	
55 034276	053715	003034		BIS	RLDRV,(R5)	:INSERT DRIVE NUMBER	
56 034302	042725	002000		BIC	#BIT10,(R5)+	:CLEAR FOR DRIVE 4 - 7 SPEC'D	
57 034306	012725	005072		MOV	#OBUFF,(R5)+	:LOAD BUS ADDRESS	
58 034312	005025			CLR	(R5)+	:CYL 0 HD 0, SECTOR 0	
59 034314	012725	177600		MOV	#177600,(R5)+	:128 WORDS	
60 034320	012701	000454		MOV	#300..R1	:SET WAIT COUNT FOR 30 MS	
61 034324	005037	003010		CLR	DONE	:CLEAR INTERRUPT FLAG	
62 034330	014562	000006		MOV	-((R5),RLMP(R2))	:LOAD RL REGS	
63 034334	014562	000004		MOV	-((R5),RLDA(R2))		
64 034340	014562	000002		MOV	-((R5),RLBA(R2))		
65 034344	014562	000000		MOV	-((R5),RLCS(R2))		
66 034350			5\$:				
67 034362	005737	003010		TST	DONE	:CHECK IF INTERRUPT	
68 034366	001013			BNE	6\$:YES - SKIP	
69 034370	005301			DEC	R1	:DEC WAIT COUNT	
70 034372	001366			BNE	5\$:LOOP IF NOT 0	
71 034374	004737	017010		JSR	PC,WAITIN	:WAIT FOR INTERRUPT	
72 034400	012603			MOV	(SP)+,R3	:GET RESULT MESSAGE	
73 034402	104456			TRAP	C\$ERHRD		
034404	003245			.WORD	1701		
034406	000000			.WORD	0		
034410	012646			.WORD	ERR1		
74 034412	104432			TRAP	C\$EXIT		
034414	000164			.WORD	L10033-		
75 034416	004737	017214	6\$:	JSR	PC,GSTAT	:GET STATUS	
76 034422	034572				11\$		
77 034424	032737	040000	003046	BIT	#DRVERR,T.CS	:TEST IF ANY ERROR SET	
78 034432	001006			BNE	7\$:YES - SKIP	
79 034434	012703	011024		MOV	#MDRERR,R3	:SET RESULT MESSAGE POINTER	
80 034440	104456			TRAP	C\$ERHRD		
034442	003246			.WORD	1702		
034444	000000			.WORD	0		
034446	012762			.WORD	ERR3		
81 034450	032737	002000	003054	7\$:	BIT	#WGESTAT,T.MP	:TEST IF WGE SET

```

82 034456 001006          BNE    8$:      ;YES - SKIP
83 034460 012703 011103    MOV    #WGEERR,R3   ;SET MESSAGE FOR WGE NOT SET
84 034464 104456          TRAP   C$ERHRD
85 034466 003250          .WORD  1704
86 034470 000000          .WORD  0
87 034472 012762          .WORD  ERP3
88 034474 042737 040000 003046 8$:  BIC    #ORVERR,T.CS  ;CLEAR DRIVE ERROR BIT
89 034502 042737 002000 003054    BIC    #WGESTAT,T.MP ;CLEAR WGE BIT
90 034510 032737 157400 003054    BIT    #157400,T.MP ;TEST IF ANY OTHER ERRORS
91 034516 001004          BNE    9$      ;YES - GO REPORT
92 034520 032737 036000 003046    BIT    #36000,T.CS  ;TEST ANY ERRORS IN CS REG
93 034526 001405          BEQ    10$     ;NO - SKIP
94 034530 104...          9$:      TRAP   C$ERHRO
95 034532 003247          .WORD  1703
96 034534 000000          .WORD  0
97 034536 013150          .WORD  ERR6
98 034540 000414          BR     11$     ;EXIT TEST
99 034542 004737 017164          10$:   JSR    PC,GSTATR ;GET STATUS AND RESET ERROR
100 034546 034572          JSR    R5,DATGEN ;GO GENERATE DATA
101 034550 004537 024672          JSR    7          ;PATTERN 7
102 034554 000007          JSR    PC,XREAD  ;READ DATA
103 034556 004737 025362          JSR    PC,DATCOM ;COMPARE DATA
104 034562 034572          JSR    PC,ERRSWI ;INIT ERROR SWITCH
105 034564 004737 025032          11$:   MOV    #2,ERRSWI
106 034570 034572          L10033:  TRAP   C$ESUB
107 034572 012737 000002 003020 T3204$: MOV    #2,ERRSWI ;INIT ERROR SWITCH
108 034600 104403          CLR    -(SP)
109 034610 005046          BISB   RLDRV+1,(SP)
110 034612 153716 003035          MOV    #0RVNAME-(SP)
111 034616 012746 006621          MOV    RLBAS,-(SP)
112 034622 013746 003030          MOV    #BASADD,-(SP)
113 034626 012746 006610          MOV    #OPR1A,-(SP)
114 034632 012746 010056          MOV    #OPR12,-(SP)
115 034636 012746 010037          MOV    #FMTOP1,-(SP)
116 034642 012746 011631          MOV    #7,-(SP)
117 034646 012746 000007          MOV    SP,RO
118 034652 010600          ADD    #20,SP
119 034654 104417          TRAP   C$PNTF
120 034656 062706 000020          ADD    #20,SP
121 034662 012701 001274          MOV    #700.,R1   ;INITIALIZE WAIT COUNT
122 034666 104417          1$:      JSR    PC,GSTATR ;GET STATUS
123 034700 004737 017164          T3204$: JSR    PC,GSTATR ;GET STATUS
124 034704 034602          BIT    #WLSTAT,T.MP ;CHECK IF WRITE LOCK RESET
125 034706 032737 020000 003054    BEQ    T3265$
126 034714 001403          DEC    R1
127 034716 005301          BNE    1$      ;DEC WAIT COUNT
128 034720 001362          BEQ    T3204$  ;LOOP IF NOT 0
129 034722 000727          BR     T3265$  ;ELSE REPEAT MESSAGE
130 034724 104401          T3265$:
131 034724 104401          L10032:  TRAP   C$ETST

```

```

1 .SBTTL *TEST 7      **ADJACENT CYLINDER INTERFERENCE
2
3 034726          T7:::                                ;SAVE TEST NUMBER
4 034726 012737 000007 003240      MOV #7,TSTNM
5 034734 012737 007521 003014      MOV #P2T18E,ERHEAD
6 034742 004737 017146          JSR PC,TSTINT
7 034746 004737 017164          JSR PC,GSTATR
8
9 034752 036146          T3365$                                ;CLEAR DRIVE
10 034754 004737 021614          JSR PC,CKBSVD
11 034760 005037 003234          CLR PASCNT
12 034764 012705 177776          MOV #-2,R5
13 034770 005737 003444          TST PASNUM
14 034774 001007          BNE 1$                                ;TEST IF FIRST PASS (QUICK VERIFY)
15 034776 032737 000C01 014500      BIT #ALLCYL,MISWIW
16 035004 001003          BNE 1$                                ;TEST IF USE ALL CYLINDERS
17 035006 012705 177730          MDV #-40,,R5
18 035012 000402          BR 2$                                ;ELSE SET R5 TO NEG 20
19
20
21
22
23 035014 012705 177770          1$: MOV #-10,R5
24 035020 012701 002506          2$: MOV #T33TBL,R1
25 035024 012737 000010 002302      MOV #10,JUNK
26 035032 013721 014502          3$: MOV LOLIMW,(R1)+,JUNK
27 035036 005337 002302          DEC JUNK
28 035042 001373          BNE 3$                                ;DEC COUNT
29 035044 004537 024672          JSR R5,DATGEN
30 035050 000011          9.                                ;PATTERN 9
31 035052 013737 014504 002510      MOV HILIMW,T33TBL+2
32 035060 013737 014504 002512      MOV HILIMW,T33TBL+4
33 035066 013737 014504 002516      MOV HILIMW,T33TBL+10
34 035074 013737 014504 002524      MOV HILIMW,T33TBL+16
35
36 035102 062705 000002          T3300$: ADD #2,R5
37 035106 032737 000001 014500      BIT #ALLCYL,MISWIW
38 035114 001034          BNE 6$                                ;TEST IF USE ALL CYLINDERS
39 035116 005737 003444          TST PASNUM
40 035122 001403          BEQ 1$                                ;YES - SKIP
41 035124 062705 000006          ADD #6,R5
42 035130 000402          BR 2$                                ;ELSE BUMP CYLINDER POINTER BY 3
43
44 035132 062705 000044          1$: ADD #36,,R5
45 035136 022737 000001 002300      2$: CMP #1,T,DRIVE
46 035144 001404          BEQ 3$                                ;BUMP TO NEXT ENTRY
47 035146 020537 000244          CMP R5,164.
48 035152 103013          BHIS 5$
49 035154 000403          BR 4$                                ;NO - SKIP
50
51 035156 020527 000122          3$: CMP R5,#82.
52 035162 103007          BHIS 5$                                ;TEST IF R5 0
53 035164 016537 002606 002302      4$: MOV CYLTBL(R5),JUNK
54 035172 043737 002306 002302      BIC CLRBYT,JUNK
55 035200 001013          BNE 8$                                ;NO - SKIP
56 035202 000137 033700          5$: JMP T3165$                                ;TEST IF ALL CYLINDERS USED
57
58 035206 005705          6$: TST R5
59 035210 001002          BNE 7$                                ;TEST IF R5 0
60 035212 062705 000002          ADD #2,R5
61 035216 023705 002304          7$: CMP HLMTW,R5

```

*TEST **ADJACENT CYLINDER INTERFERENCE

62	035222	001767		BEQ	5\$:YES - EXIT TEST
63	035224	010537	002302	MOV	R5,JUNK	:USE PS AS NEXT CYLINDER
64	035230	023737	002302	CMP	JUNK,LOLIMIT	:CHECK IF LOWER THAN LOLIMIT
65	035236	103721		BLC	T3300\$:YES - SKIP
66	035240	023737	002302	CMP	JUNK,HILIMIT	:CHECK IF HIGHER THAN HILIMIT
67	035246	101315		BHI	T3300\$:YES SKIP
68	035250	012703	002546	MOV	#BTB,R3	
69	035254	013713	002302	MOV	JUNK,(R3)	
70	035260	013763	002302	MOV	JUNK,6(R3)	
71	035266	013763	000006	MOV	JUNK,10(R3)	
72	035274	013763	000010	MOV	JUNK,12(R3)	
73	035302	013763	002302	MOV	JUNK,16(R3)	
74	035310	162737	000001	SUB	#1,JUNK	
75	035316	013763	002302	MCV	JUNK,2(R3)	
76	035324	013763	000012	MOV	JUNK,12(R3)	
77	035332	062737	000002	ADD	#2,JUNK	
78	035340	013763	002302	MOV	JUNK,4(R3)	
79	035346	013763	000004	MOV	JUNK,14(R3)	
80	035354	010337	003026	MOV	R3,TBLSTR	
81	035360	004737	021504	JSR	PC,CHOSHD	;GO CHOSE HEAD
82						
83	035364				T3301\$:	
	035364				T7.1:	
	035364	104402		TRAP	C\$BSUB	
84	035366	042737	003760	BIC	#EQUALS,OPFLAG	:CLEAR ALL MESSAGE QUALIFIERS
85	035374	005737	003234	TST	PASCNT	:TEST IF PASS 0
86	C35400	001414		BEQ	2\$:YES - SKIP
87	035402	023727	003234	CMP	PASCNT,#4	:TEST IF PASS 4
88	035410	001404		BEQ	1\$:YES - SKIP
89	035412	002407		BLT	2\$:CHECK IF LESS THAN 4, IF YES CLEAR TO 0
90	035414	012737	000004	MOV	#4,PASCNT	:ELSE SET TO 4
91	035422	052737	000020	003234	1\$:	SET MESSAGE QUAL
92	035430	000405	003006	BIS	#INOUTS,OPFLAG	
				BR	3\$:SKIP
93	035432	005037	003234	CLR	PASCNT	:SET PASS COUNT TO 0
95	035436	052737	000040	BJS	#OUTINS,OPFLAG	:SET MESSAGE QUAL
96	035444	012737	000003	003024	3\$:	MOV #3 WRTSWI
97	035452	012701	002506	MOV	#T33TBL,R1	:SET READ AND WRITE SWITCH
98	035456	012703	002546	MOV	#BTB,R3	
99	035462	005037	003116	4\$:	CLR DESEC	:CLEAR TO SECTOR 0
100	035466	012137	003104	MOV	(R1)+,NEWCYL	:GET NEXT TABLE ENTRY
101	035472	004737	020112	JSR	PC,XSEEK	:DO SEEK
102	035476	036054			15\$	
103	035500	012701	005670	MOV	#3000,,R1	:SET WAIT COUNT FOR 300 MS
104	035504	004737	023570	JSR	PC,RDYWAIT	:WAIT FOR READY
105	035510	036054			15\$	
106	035512	012337	003104	MOV	(R3)+,NEWCYL	:GET NEXT TABLE ENTRY
107	035516	004737	020112	JSR	PC,XSEEK	:DO SEEK
108	035522	036054			15\$	
109	035524	012701	005670	MOV	#3000,,R1	:SET WAIT COUNT FOR 300 MS
110	035530	004737	023570	JSR	PC,RDYWAIT	:WAIT FOR READY
111	035534	036054			15\$	
112	035536	004737	024202	JSR	PC,VERPOS	:VERIFY POSITION
113	035542	036054			15\$	
114	035544	004737	026124	5\$:	JSR PC,BSCCHK	:CHECK FOR BAD SECTOR
115	035550	035660			8\$;"YES" RETURN
116	035552	032737	000001	003024	BIT #BIT0,WRTSWI	:TEST IF WRITE THIS PASS

*TEST 7 **ADJACENT CYLINDER INTERFERENCE

```

117 035560 001425 BEQ   6$      :NO - SKIP
118 035562 004737 025322 JSR    PC,XWRITE 15$      :DO WRITE
119 035566 036054           INC    DESSEC :INC SECTOR
120 035570 005237 003116 CMP    #0.,DESSEC :TEST IF ALL SECTORS USED
121 035574 022737 000050 003116 BNE    5$      :NO - SKIP
122 035602 001360           BIC    #INOUTS!OUTINS,OPFLAG :CLEAR QUALIFIERS
123 035604 042737 000060 003006 BIC    #BIT0,WRTSWI :CLEAR WRITE REQUIRED SWITCH
124 035612 042737 000001 003024 BIC    #FOLWRT,OPFLAG :SET FOLLOWING WRITE QUALIFIER
125 035620 052737 000100 003006 BIS    DESSEC :CLEAR TO SECTOR 0
126 035626 005037 003116 CLR    BR      5$      :SKIP
127 035632 000744           BR
128
129 035634 032737 000002 003024 6$: BIT    #BIT1,WRTSWI :TEST IF READ THIS PASS
130 035642 001414           BEQ    9$      :NO - SKIP
131 035644 004737 025362           JSR    PC,XREAD 15$      :ELSE DO READ
132 035650 036054           JSR    PC,OATCOM :COMPARE DATA
133 035652 004737 025032           JSR
134 035656 036054           15$      :BUMP SECTOR
135 035660 005237 003116 003116 8$: INC    DESSEC :TEST IF ALL SECTORS USED
136 035664 022737 000050           CMP    #40.,DESSEC :NO - LOOP
137 035672 001324           BNE    5$      :CLEAR DESIRED SECTOR
138 035674 005037 003116           CLR    DESSEC :CLEAR WRITE/READ SWITCH
139 035700 005037 003024           CLR    WRTSWI :BUMP PASS COUNT
140 035704 005237 003234           INC    PASCNT :CLEAR ALL QUALIFIERS
141 035710 042737 003760 003006 BIC    #MQUALS,OPFLAG :TEST IS PASS 4
142 035716 023727 003234 000004 CMP    PASCNT,#4 :YES - SKIP
143 035724 001453           BEQ    15$      :TEST IF PASS 8.
144 035726 023727 003234 000010 CMP    PASCNT,#8. :YES - SKIP
145 035734 001447           BEQ    15$      :TEST IF PASS 3
146 035736 023727 003234 000003 CMP    PASCNT,#3 :YES - SKIP
147 035744 001430           BEQ    12$      :TEST IF PASS 7
148 035746 023727 003234 000007 CMP    PASCNT,#7 :YES - SKIP
149 035754 001430           BEQ    13$      :SET WRITE REQUIRED
150 035756 012737 000001 003024 MOV    #BIT0,WRTSWI :TEST IF PASS 1
151 035764 023727 003234 000001 CMP    PASCNT,#1 :YES - SKIP
152 035772 001411           BEQ    11$      :TEST IF PASS 2
153 035774 023727 003234 000002 CMP    PASCNT,#2 :YES - SKIP
154 036002 001405           BEQ    11$      :SET MESSAGE QUALIFIER
155 036004 052737 000040 003006 BIS    #INOUTS,OPFLAG :GO DO NEXT PASS
156 036012 000137 035462           JMP    4$      :
157
158 036016 052737 000020 003006 11$: BIS    #INOUTS,OPFLAG :SET MESSAGE QUALIFIER
159 036024 000772           BR    10$      :
160
161 036026 052737 000200 003006 12$: BIS    #REVSKS,OPFLAG :SET MESSAGE QUALIFIER
162 036034 000403           BR    14$      :
163
164 036036 052737 000400 003006 13$: BIS    #FDWSKS,OPFLAG :SET MESSAGE QUALIFIER
165 036044 012737 000002 003024 14$: MOV    #BIT1,WRTSWI :SET READ REQUIRED
166 036052 000757           BR    10$      :
167
168 036054 012737 000002 003020 15$: L10035: MOV    #2,ERRSWI :INIT ERROR SWITCH
169 036062 104403           TRAP   C$ESUB
170
171 036064 104410           TRAP   C$ESCAPE
036066 000060           WORD   L10034-.

```

172	036070	012737	000003	003024	MOV	#3	WRTSWI	:SET FDR READ AND WRITE REQ.
173	036076	023727	003234	000004	CMP	PASCNT,	#4	:TEST IF PASS 4
174	036104	001004			BNE	16\$:NO - SKIP
175	036106	012737	002516	003026	MOV	#T33TBL+10,TBLSTR		:STORE MID POINT IN TABLE
176	036114	000410			BR	17\$:GO START PASS 4
177								
178	036116	005037	003234		16\$:	CLR	PASCNT	:CLEAR TO PASS 0
179	036122	004737	021530			JSR	PC,SWAPHD	:GO SWAP TO HEAD 1 OR END TEST
180	036126	035102				T3300\$:ABORT RETURN
181	036130	012737	002506	003026	MOV	#T33TBL,TBLSTR		:STORE START OF TABLE
182								
183	036136	062703	000010		17\$:	ADD	#10,R3	
184	036142	000137	035364			JMP	T3301\$	
185								
186	036146					T3365\$:		
	036146					L10034:		
	036146	104401			TRAP	C\$ETST		

```

1 .SBTTL *TEST 8      **OVERWRITE
2
3 036150 012737 000010 003240      T8:::          MOV    #10,TSTNM   ;SAVE TEST NUMBER
4 036156 012737 007543 003014      MOV    #P2119E,ERHEAD ;SET ERROR HEADER
5 036164 004737 017146          JSR    PC,TSTINT  ;INITIALIZE TEST
6 036170 004737 017164          JSR    PC,GSTATR ;CLEAR DRIVE
7
8 036174 037346          T3465$          MOV    PC,CKBSVD ;GO CHECK IF BAD SECTOR FILES VALID
9
10 036176 004737 021614          JSR    PASCNT   ;CLEAR PASS TO 0
11 036202 005037 003234          CLR    R5        ;SET R5
12 036206 012705 177776          MOV    #-2,R5    ;TEST IF FIRST PASS (QUICK VERIFY)
13 036212 C05737 003444          TST    PASNUM   ;NO - SKIP
14 036216 001007          BNE    1$       ;TEST IF USE ALL CYLINDERS
15 036220 032737 000001 014500      BIT    #ALLCYL,MISWIW ;YES - SKIP
16 036226 001003          BNE    1$       ;ELSE SET R5 TO NEG 20
17 036230 012705 177730          MOV    #-40.,R5  ;SKIP
18 036234 000402          BR     2$       ;SET FOR NEXT ENTRY
19
20 036236 012705 177770          1$:   MOV    #-10,R5   ;GET ADDRESS OF WORK TABLE
21 036242 012701 002506          2$:   MOV    #T33TBL,R1 ;SET CLEAR COUNT
22 036246 012737 000010 002302      3$:   MOV    #10,JUNK  ;CLEAR LOCATIONS TO LOLIMIT
23 036254 013721 014502          MOV    LOLIMW,(R1)+;DEC COUNT
24 036260 005337 002302          DEC    JUNK    ;LOOP UNTIL 0
25 036264 001373          BNE    3$       ;INSERT HILIMIT
26 036266 013737 014504 002510      MOV    HILIMW,T33TBL+2 ;INTO APPROPRIATE LOCATIONS
27 036274 013737 014504 002514      MOV    HILIMW,T33TBL+6
28 036302 013737 014504 002520      MOV    HILIMW,T33TBL+12
29
30 036310 062705 000002          T3400$: ADD   #2,R5    ;TEST IF USE ALL CYLINDERS
31 036314 032737 000001 014500      BIT    #ALLCYL,MISWIW ;YES - SKIP
32 036322 001034          BNE    6$       ;TEST IF FIRST PASS (QUICK VERIFY)
33 036324 005737 003444          TST    PASNUM   ;NO - SKIP
34 036330 001003          BNE    1$       ;ELSE BUMP CYLINDER POINTER BY 19
35 036332 062705 000046          ADD   #38.,R5  ;SKIP
36 036336 000402          BR     2$       ;BUMP CYLINDER POINTER BY 3
37
38 036340 062705 000006          1$:   ADD   #6,R5    ;CMP #1,T.DRIVE
39 036344 022737 000001 002300      2$:   CMP   #1,T.DRIVE ;BEQ 3$
40 036352 001404          BEQ    3$       ;CMP R5,#164.
41 036354 020527 000244          CMP   BHIS    ;BHIS 5$
42 036360 103013          BR     4$       ;BR 4$
43
44 036362 000403          BR     4$       ;BUMP CYLINDER POINTER BY 3
45
46 036364 020527 000122          3$:   CMP   BHIS    ;5$ R5,#82.
47 036370 103007          BHIS   5$       ;5$ CYLTBL(R5),JUNK
48 036372 016537 002606 002302      4$:   MOV   CYLTBL(R5),JUNK ;CLRBYT,JUNK
49 036400 043737 002306 002302      5$:   BIC   CLRBYT,JUNK
50 036406 001013          BNE    8$       ;BNE T3465$ ;EXIT TEST
51 036410 000137 037346          JMP   T3465$  ;TEST IF R5 0
52
53 036414 005705          6$:   TST   R5     ;NO - SKIP
54 036416 001002          BNE    7$       ;TEST IF ALL CYLINDERS USED
55 036420 062705 000002          ADD   #2,R5  ;YES - EXIT TEST
56 036424 022705 002304          7$:   CMP   #HLMTW,R5 ;MOV R5,JUNK ;USE R5 AS NEXT CYLINDER
57 036430 001767          BEQ    5$       ;CMP JUNK,LOLIMW ;TEST IF PAST LO LIMIT
58
59 036432 010537 002302          MOV   R5,JUNK
60 036436 023737 002302 014502      CMP   JUNK,LOLIMW

```

```

62 036444 103721      BLO   T3400$    ;YES - SKIP
63 036446 023737 002302 014504      CMP   JUNK,HILIMW ;TEST IF PAST HILIMIT
64 036454 101315      BHI   T3400$,R3 ;YES - SKIP
65 036456 012703 002546      MOV   #TBT,R3
66 036462 013713 002302      MOV   JUNK,(R3)
67 036466 013763 002302 000002      MOV   JUNK,2(R3)
68 036474 013763 002302 000004      MOV   JUNK,4(R3)
69 036502 013763 002302 000006      MOV   JUNK,6(R3)
70 036510 013763 002302 000010      MOV   JUNK,10(R3)
71 036516 013763 002302 000012      MOV   JUNK,12(R3)
72 036524 010337 003026      MOV   R3,TBLSTR
73 036530 004737 021504      JSR   PC,CHOSHD ;GO CHOSE HEAD
74
75 036534      T3401$:          T8.1:
036534 104402      TRAP  C$BSUB
76 036536 042737 003760 003006      BIC   #MQUALS,OPFLAG ;CLEAR ALL MESSAGE QUALIFIERS
77 036544 005737 003234      TST   PASCNT ;TEST IF PASS 0
78 036550 001414      BEQ   2$ ;YES - SKIP
79 036552 023727 003234 000003      CMP   PASCNT,#3 ;TEST IF PASS 3
80 036560 001404      BEQ   1$ ;YES - SKIP
81 036562 002407      BLT   2$ ;CHECK IF LESS THAN 3, IF YES CLEAR TO 0
82 036564 012737 000003 003234      MOV   #3,PASCNT ;ELSE SET TO 3
83 036572 052737 000020 003006 1$:  BIS   #INOUTS,OPFLAG ;SET MESSAGE QUAL
84 036600 000405      BR    3$ ;SKIP
85
86 036602 005037 003234      2$:  CLR   PASCNT ;SET PASS COUNT TO 0
87 036606 052737 000040 003006      BIS   #OUTINS,OPFLAG ;SET MESSAGE QUAL
88 036614 012737 000003 003024 3$:  MOV   #3,WRTSWI ;SET READ AND WRITE SWITCH
89 036622 012701 002506      MOV   #T33TBL,R1
90 036626 012703 002546      MOV   #TBT,R3
91 036632 005037 003116      4$:  CLR   DESSÉC
92 036636 012137 003104      MOV   (R1)+,NEWCYL ;GET NEXT TABLE ENTRY
93 036642 004737 020112      JSR   PC,XSEEK ;DO SEEK
94 036646 037254      18$ ;DO SEEK
95 036650 012701 005670      MOV   #3000.,R1 ;SET WAIT COUNT FOR 300 MS
96 036654 004737 023570      JSR   PC,RDYWAIT ;WAIT FOR READY
97 036660 037254      18$ ;DO SEEK
98 036662 012337 003104      MOV   (R3)+,NEWCYL ;GET NEXT TABLE ENTRY
99 036666 004737 020112      JSR   PC,XSEEK ;DO SEEK
100 036672 037254      18$ ;DO SEEK
101 036674 012701 005670      MOV   #3000.,R1 ;SET WAIT COUNT FOR 300 MS
102 036700 004737 023570      JSR   PC,RDYWAIT ;WAIT FOR READY
103 036704 037254      18$ ;DO SEEK
104 036706 004737 024202      JSR   PC,VERPOS ;VERIFY POSITION
105 036712 037254      18$ ;DO SEEK
106 036714 004737 026124 5$:  JSR   PC,BSCHK ;CHECK FOR BAD SECTOR
107 036720 037070      11$ ;"YES" RETURN
108 036722 005737 003234      TST   PASCNT ;TEST IF PASS 0
109 036726 001407      BEQ   6$ ;YES - SKIP
110 036730 022737 000003 003234      CMP   #3,PASCNT ;TEST IF PASS 3
111 036736 001403      BEQ   6$ ;YES - SKIP
112 036740 005037 036760      CLR   8$ ;ELSE CLEAR DATA PATTERN SELECTOR
113 036744 000403      BR    7$ ;DO SEEK
114
115 036746 012737 000010 036760 6$:  MOV   #8,8$ ;SET DATA PATTERN SELECTOR TO 8
116 036754 004537 024672 7$:  JSR   R5,DATGEN ;GO GENERATE DATA

```

```

117 036760 000000 8$: WORD 0
118 035762 032737 000001 003024 BIT #BIT0,WRTSWI ;TEST IF WRITE THIS PASS
119 036770 001425 BEQ 9$ ;NO - SKIP
120 036772 004737 025322 JSR PC,XWRITE ;DO WRITE
121 036776 037254 18$ INC DESSEC ;INC SECTOR
122 037000 005237 003116 CMP #40.,DESSEC ;TEST IF ALL SECTORS USED
123 037004 022737 000050 003116 BNE 5$ ;NO - SKIP
124 037012 001340 BIC #INOUTS!OUTINS,OPFLAG ;CLEAR QUALIFIERS
125 037014 042737 000060 003006 BIC #BIT0,WRTSWI ;CLEAR WRITE REQUIRED SWITCH
126 037022 042737 000001 003024 BIS #FOLWRT,OPFLAG ;SET FOLLOWING WRITE QUALIFIER
127 037030 052737 000100 003006 CLR DESSEC ;CLEAR TO SECTOR 0
128 037036 005037 003116 BR 5$ ;SKIP
129 037042 000724
130
131 037044 032737 000002 003024 9$: BIT #BIT1,WRTSWI ;TEST IF READ THIS PASS
132 037052 001414 BEQ 12$ ;NO - SKIP
133 037054 004737 025362 10$: JSR PC,XREAD ;ELSE DO READ
134 037060 037254 18$ JSR PC,DATCOM ;COMPARE DATA
135 037062 004737 025032 18$ INC DESSEC ;BUMP SECTOR
136 037066 037254 11$: CMP #40.,DESSEC ;TEST IF ALL SECTORS USED
137 037070 005237 003116 BNE 5$ ;NO - LOOP
138 037074 022737 000050 003116 12$: CLR DESSEC ;CLEAR DESIRED SECTOR
139 037102 001304 CLR WRTSWI ;CLEAR WRITE/READ SWITCH
140 037104 005037 003116 INC PASCNT ;BUMP PASS COUNT
141 037110 005037 003024 CMP #MQUALS,OPFLAG ;CLEAR ALL QUALIFIERS
142 037114 005237 003234 14$: CLR PASCNT,#3 ;TEST IS PASS 3
143 037120 042737 003760 003006 BEQ 18$ ;YES - SKIP
144 037126 023727 003234 000003 CMP PASCNT,#6 ;TEST IF PASS 6
145 037134 001447 BEQ 18$ ;YES - SKIP
146 037136 023727 003234 000006 CMP PASCNT,#1 ;TEST IF PASS 1
147 037144 001443 BEQ 15$ ;YES - SKIP
148 037146 023727 003234 000001 CMP PASCNT,#4 ;TEST IF PASS 2
149 037154 001424 BEQ 16$ ;YES - SKIP
150 037156 023727 003234 000004 CMP MOV #BIT1,WRTSWI ;SET WRITE REQUIRED BIT
151 037164 001424 BEQ 16$ ;YES - SKIP
152 037166 012737 000002 003024 CMP PASCNT,#2 ;TEST IF PASS 2
153 037174 023727 003234 000002 BEQ 14$ ;YES - SKIP
154 037202 001405 155 037204 052737 001000 003006 BIS #REVSKO,OPFLAG ;SET REVERSE QUALIFIER
156 037212 000137 036632 13$: JMP 4$ ;GO DO NEXT PASS
157
158 037216 052737 002000 003006 14$: BIS #FWDSCO,OPFLAG ;SET FWD QUALIFIER
159 037224 000772 BR 13$ ;GO DO NEXT PASS
160
161 037226 052737 000020 003006 15$: BIS #INOUTS,OPFLAG ;SET QUALIFIER
162 037234 000403 BR 17$ ;SKIP
163
164 037236 052737 000040 003006 16$: BIS #OUTINS,OPFLAG ;SET MESSAGE QUALIFIER
165 037244 012737 000001 003024 17$: MOV #BIT0,WRTSWI ;SET WRITE REQUIRED BIT
166 037252 000757 BR 13$ ;GO DO NEXT PASS
167
168 037254 012737 000002 003020 18$: MDV #2,ERRSWI ;INIT ERROR SWITCH
169 037262 104403 L10037: TRAP C$ESUB
170
171 037264 104410 TRAP C$ESCAPE
037266 000060 .WORD L10036-.

```

*TEST 8 **OVERWRITE

172 037270 C12737 000003 003024 MOV #3,WRTSWI ;SET FOR READ AND WRITE REQ.
173 037276 023727 003234 000003 CMP PASCNT,#3 ;TEST IF PASS 3
174 037304 001004 BNE 19\$;NO - SKIP
175 037306 012737 002514 003026 MOV #T33TBL+6,TBLSTR ;STORE MID POINT IN TABLE
176 037314 000410 BR 20\$;GO START PASS 4
177
178 037316 005037 003234 19\$: CLR PASCNT ;CLEAR TO PASS 0
179 037322 004737 021530 JSR PC,SWAPHD ;GO SWAP TO HEAD ONE OR ABORT TEST
180 037326 036310 T3400\$ T3400\$;ABORT RETURN
181 037330 012737 002506 003026 MOV #T33TBL,TBLSTR ;STORE START OF TABLE
182 037336 062703 000006 20\$: ADD #6,R3
183 037342 000137 036534 JMP T3401\$
184
185 037346 T3465\$: L10036:
037346 TRAP C\$ETST
037346 104401

```

1          .SBTTL PARAMETER CODING
2
4 037350 000030      .WORD L10040 L$HARD/2
5 037352 005130      .WORD T$CODE
6 037354 037516      .WORD CNTYPE
7 037356 000001      .WORD 1
8 037360 000031      .WORD T$CODE
9 037362 037432      .WORD CSRMSG
10 037364 160000     .WORD T$LOLIM
11 037366 177776     .WORD T$HILIM
12 037370 001031     .WORD T$CODE
13 037372 037446     .WORD VECMSG
14 037374 000000     .WORD T$LOLIM
15 037376 000776     .WORD T$HILIM
16 037400 004032     .WORD T$CODE
17 037402 037510     .WORD DRMSG
18 037404 003400     .WORD 3400
19 037406 000000     .WORD T$LOLIM
20 037410 000007     .WORD T$HILIM
21 037412 003130     .WORD T$CODE
22 037414 037466     .WORD DRTYPE
23 037416 000001     .WORD 1
24 037420 002032     .WORD T$CODE
25 037422 037455     .WORD BRMSG
26 037424 000340     .WORD 340
27 037426 000000     .WORD T$LOLIM
28 037430 000007     .WORD T$HILIM
29
30 037432           .EVEN
31
32 037432           L10040:
33 037432           .ASCIZ /BUS ADDRESS/
34 037432           .ASCIZ /VECTOR/
35 037432           .ASCIZ /BR LEVEL/
36 037432           .ASCIZ /DRIVE TYPE = RL01/
37 037432           .ASCIZ /DRIVE/
38 037432           .ASCIZ /RL11/
39
40 037524 000061     .EVEN
41 037526 000130     .WORD L10041-L$SOFT/2
42 037530 037670     .WORD T$CODE
43 037532 000001     .WORD CYLQ
44
45 037534 000130     .WORD 1
46 037536 037704     .WORD T$CODE
47 037538 000002     .WORD SECQ
48
49 037542 000130     .WORD 2
50 037544 037721     .WORD T$CODE
51 037546 100000     .WORD MANQ
52 037548 000000     .WORD 100000
53
54 037550 000130     .WORD T$CODE
55 037552 037755     .WORD LOLIMQ
56 037554 040000     .WORD 40000
57
58 037556 006044     .WORD T$CODE
59 037560 001052     .WORD T$CODE
60 037562 037774     .WORD LIMVAL
61 037564 000777     .WORD 777
62 037566 000000     .WORD T$LOLIM
63 037570 000375     .WORD T$HILIM

```

10
37 037572 000130 .WORD T\$CODE
037572 040002 .WORD HILIMQ
037574 020000 .WORD 20000
38 037600 006044 .WORD T\$CODE
39 037602 002052 .WORD T\$CODE
037604 037774 .WORD LIMVAL
037606 000777 .WORD 777
037610 000000 .WORD T\$LOLIM
037612 000377 .WORD T\$HILIM
40 037614 000130 .WORD T\$CODE
037616 040023 .WORD HEADQ
037620 010000 .WORD 10000
41 037622 006044 .WORD T\$CODE
42 037624 003052 .WORD T\$CODE
037626 040045 .WORD HEADV
037630 000017 .WORD 17
037632 000000 .WORD T\$LOLIM
037634 000001 .WORD T\$HILIM
44 037636 004052 .WORD T\$CODE
037640 040070 .WORD ERLIMQ
037642 000377 .WORD 377
037644 000000 .WORD T\$LOLIM
037646 000377 .WORD T\$HILIM
46 037650 005052 .WORD T\$CODE
037652 040112 .WORD DCLIMQ
037654 000377 .WORD 377
037656 000001 .WORD T\$LOLIM
037660 000377 .WORD T\$HILIM
47 037662 006130 .WORD T\$CODE
037664 040133 .WORD BSOUTQ
037666 000001 .WORD 1
49 037670 .EVEN
51 L10041:
52 037670 125 123 105 CYLQ: .ASCIZ /USE ALL CYL/
53 037704 125 123 105 SECQ: .ASCIZ /USE ALL SECT/
60 037721 104 117 040 MANQ: .ASCIZ /DO MANUAL INTERVENTION TEST/
62 037755 114 117 127 LOLIMQ: .ASCIZ /LOW SEEK LIMIT/
63 037774 126 101 114 LIMVAL: .ASCIZ /VALUE/
64 040002 125 120 120 HILIMQ: .ASCIZ /UPPER SEEK LIMIT/
65 040023 125 123 105 HEADQ: .ASCIZ /USE ONLY ONE SURF/
66 040045 127 110 101 HEADV: .ASCIZ /WHAT SURF (0 OR 1)/
68 040070 111 116 120 ERLIMQ: .ASCIZ /INPUT ERROR LIMIT/
70 040112 104 101 124 DCLIMQ: .ASCIZ /DATA CMP ERR LMT/
71 040133 120 122 111 BSOUTQ: .ASCIZ /PRINT ERRORS DETECTED WHILE READING BAD SEC FILE/
74 .EVEN
040214 000000 .WORD 0
040216 000000 .WORD 0
040220 000001 .END
75 L\$LAST::

Symbol table

ADR = 000020 G	CKERLM 016616	C\$GETW= 000027	DRSET = 000010	FMT2 011730
AFMID 003212	CLKADR 003476	C\$GMAN= 000043	DRTYPE 037466	FMT20 012431
AFMIDU 003214	CLKCSB= 172542	C\$GPHR= 000042	DRVCNT 003076	FMT21 012461
ALLCYL - 000001	CLKCSR= 172540	C\$GPRI= 000040	DRVERR= 040000	FMT22 012504
ALLSEC = 000002	CLKCTR= 172544	C\$INIT= 000011	DRVNAV 006621	FMT23 012540
ANYERR = 100000	CLKFLG 003474	C\$INLP= 000020	DSESTA- 000400	FMT24 012554
ARMID 003216	CLNCOD 016056 G	C\$MANI= 000050	DSMSK - 001400	FMT25 012561
ARMIDU 003220	CLRBYT 002306	C\$MAP = 000102	DSPCOD 014516 G	FMT26 012571
ASSEMB= 000010	CLRPAR 027524	C\$MEM = 000031	EF.CDN= 000036 G	FMT27 012615
BADADD= 004000	CNT = 000012	C\$MMU = 000103	EF.NEW= 000035 G	FMT28 012634
BAMSK = 000060	CNTYPE 037516	C\$MSG = 000023	EF.PWR= 000034 G	FMT3 011737
BANAM 006712	COMPOP= 007777	C\$OPNR= 000034	EF.RES= 000037 G	FMT4 011737
BASADD 006610	CONHNG= 000004	C\$OPNW= 000104	EF.STA= 000040 G	FMT5 011750
BELL 011477	CONTIN 014744	C\$PNTB= 000014	EF.XM - 000033 G	FMT6 011770
BHSTAT - 000010	CDSTAT= 000040	C\$PNTF= 000017	ERHEAD 003014	FMT7 012032
BIT0 = 000001 G	COUNT 003236	C\$PNTS= 000016	ERLIM = 000010	FMT8 012102
BIT00 = 000001 G	CR = 000015	C\$PNTX= 000015	ERLIMQ 040070	FMT9 012134
BIT01 = 000002 G	CRDYSMS= 000200	C\$PUTB= 000072	ERLIMW 014510	FOLWRT= 000100
BIT02 = 000004 G	CRLF 011623 G	C\$PUTW= 000073	ERRCNT 003244	FRMWID 010153
BIT03 = 000010 G	CSNAM 006705	C\$QIO = 000377	ERRPDI 003242	FWDSKD= 002000
BIT04 = 000020 G	CSR = 000000	C\$RDBU= 000007	ERRSWI 003020	FWDSKS= 000400
BIT05 = 000040 G	CSRMSG 037432	C\$REFG= 000047	ERRVEC 003232	F\$AU = 000015
BIT06 = 000100 G	CURCYL 003106	C\$REL = 000077	ERR1 012646 G	F\$AUTO= 000020
BIT07 = 000200 G	CYLQ 037670	C\$RESE= 000033	ERR10 014242 G	F\$BGN = 000040
BIT08 = 000400 G	CYLTBL 002606	C\$REVI= 000004	ERR2 012714 G	F\$CLEA= 000007
BIT09 = 001000 G	CYLUP = 000004	C\$RFLA= 000021	ERR3 012762 G	F\$DU = 000016
BIT1 = 000002 G	CYLWD 010146	C\$RPT = 000025	ERR4 013030 G	F\$END = 000041
BIT10 = 002000 G	C\$AU = 000052	C\$SEFG= 000046	ERR5 013100 G	F\$HARD= 000004
BIT11 = 004000 G	C\$AUTD= 000061	C\$SPRI= 000041	ERR6 013150 G	F\$HW = 000013
BIT12 = 010000 G	C\$BRK = 000022	C\$SVEC= 000037	ERR7 014032 G	F\$INIT= 000006
BIT13 = 020000 G	C\$BSEG= 000004	C\$TOME= 000076	ERR8 014102 G	F\$JMP = 000050
BIT14 = 040000 G	C\$BSUB= 000002	C1OMS 011556	ERR9 014176 G	F\$MOO = 000000
BIT15 = 100000 G	C\$CLK= 000062	C\$SEC 011615	EVL = 000004 G	F\$MSG = 000011
BIT2 = 000004 G	C\$CLEA= 000012	C\$OOMS 011567	EXACYL 003226	F\$PRDT= 000021
BIT3 = 000010 G	C\$CLDS= 000035	DANAM 006717	EXHCYL 003224	F\$PWR = 000017
BIT4 = 000020 G	C\$CLP1= 000006	DATACM= 000C01	EXOCYL 003222	F\$RPT = 000012
BIT5 = 000040 G	C\$CPBF= 000074	DATCOM 025032	EXROT 003230	F\$SEG = 000003
BIT6 = 000100 G	C\$CPME= 000075	DATGEN 024672	E\$END = 002100	F\$SOFT= 000005
BIT7 = 000200 G	C\$CVEC= 000036	DCKERR= 004000	E\$LOAD= 000035	F\$SRV = 000010
BIT8 = 000400 G	C\$DCLN= 000044	DCLIM = 000012	FCTBSF 003502	F\$SUB = 000002
BIT9 = 001000 G	C\$DODU= 000051	DCLIMQ 040112	FLDBSF 004076	F\$SW = 000014
BDE = 000400 G	C\$DRPT= 000024	DCLIMW 014512	FMTOP1 011631	F\$TEST= 000001
BRMSG 037455	C\$DU = 000053	DESDIF 003110	FMTOP2 011660	GBND 002312
BSCHK 026124	C\$EDIT= 000001	DESHD 003114	FMTOP3 011702	GETPDS 024054
BSERR = 000014	C\$ERDF= 000055	DESSEC 003116	FMTXT 011626 G	GETSTA= 000003
BSERRS 014514	C\$ERHR= 000056	DESSGN 003112	FMT1 011723	GLBDAT 002226 G
BSFLAG 003022	C\$ERRO= 000060	DIAGMC= 000000	FMT10 012141	GLBEQA 002226 G
BSFNOT 010572	C\$ERSF= 000054	DIFAUG 003100	FMT11 012141	GLBERR 012646 G
BSFVAL 003500	C\$ERSD= 000057	DIFWD 010122	FMT12 012147	GLBSUB 016210 G
BSDUTQ 040133	C\$ESCA= 000010	DIRBIT= 000004	FMT13 012155	GLBXT 005750 G
BYPNSM 010161	C\$ESEG= 000005	DIRMSK 002316	FMT14 012221	GSTAT 017214
CAFDT 011604	C\$ESUB= 000003	DLTERR= 010000	FMT15 012253	GSTATC 017200
CAMSK 002314	C\$ETST= 000001	DONE 003010	FMT16 012307	GSTATG 017224
CCYLUP 011575	C\$EXIT= 000032	DRDYMS= 000001	FMT17 012320	GSTATR 017164
CHDSHD 021504	C\$FREQ= 000101	DRMSG 037510	FMT18 012342	GTSTAT= 000104
CKBSVD 021614	C\$FRME= 000100	DRSB = 000010	FMT19 012374	G\$CNTD= 000200
CKDATA= 000102	C\$GETB= 000026	DRSELT= 000004		G\$DELM= 000372

Symbol table

G\$DISP = 000003	H\$STAT = 000100	L\$CLEA 016056 G	L10007 014174	MHFCRC 010774
G\$EXCP = 000400	IBE = 010000 G	L\$CO 002032 G	L10010 014240	MHNF 010746
G\$HILI = 000002	IBUFF = 004472	L\$DEPO 002011 G	L10011 014450	MININC 003462
G\$LOLI = 000001	IDU = 000040 G	L\$DESC 002122 G	L10013 014476	MINOUT 006177
G\$NO = 000000	IER = 020000 G	L\$DESP 002076 G	L10014 014516	MISWI = 000000
G\$OFFS = 000400	INITCO 014540 G	L\$DEVP 002060 G	L10015 015516	MISWIW 014500
G\$OFSI = 000376	INOUTS = 000020	L\$DISP 014520 G	L10016 016054	MITEST = 100000
G\$PRMA = 000001	INTEBL = 000100	L\$DLY 002116 G	L10017 016202	MNDRST 011274
G\$PRMD = 000002	INTHLR 016536	L\$DTP 002040 G	L10020 016206	MNEERR 011242
G\$PRML = 000000	ISR = 000100 G	L\$DTYP 002034 G	L10021 016534	MNOCLR 007072
G\$RADA = 000140	IXE = 004000 G	L\$DU 016204 G	L10022 016614	MNOINT 007023
G\$RADB = 000000	I\$AU = 000041	L\$DUT 002072 G	L10023 031472	MOP F 006117
G\$RADB = 000040	I\$AUTO = 000041	L\$DVY 002214 G	L10024 031524	MOPEKR 011167
G\$RADL = 000120	I\$CLN = 000041	L\$EF 002052 G	L10025 031746	MORECE 003016
G\$RADO = 000020	I\$DU = 000041	L\$ENVI 002044 G	L10026 031672	MOUTIN 006160
G\$XFER = 000004	I\$HARD = 000041	L\$ETP 002102 G	L10027 032562	MPNAM 006724
G\$YES = 000010	I\$INIT = 000041	L\$EXP1 002046 G	L10030 033700	MQUALS = 003760
HADONE = 003012	I\$MOD = 000041	L\$EXP4 002064 G	L10031 033614	MREAD 005754
HCESTA = 040000	I\$MSG = 000041	L\$EXP5 002066 G	L10032 034724	MREADH 005765
HCRCER = 004000	I\$PROT = 000040	L\$HARD 037352 G	L10033 034600	MRESKO 006356
HDALIG = 000010	I\$PTAB = 000041	L\$HIME 002120 G	L10034 036146	MREVSK 006240
HDCYL = 002320	I\$PWR = 000041	L\$HPCP 002016 G	L10035 036062	MRLFAL 011364
HDHSEL = 000100	I\$RPT = 000041	L\$HPTP 002022 G	L10036 037346	MRSLT 006126
HDMOVF = 010003	I\$SEG = 000041	L\$HW 014462 G	L10037 037262	MSEEK 005750
HDRCMP = 000002	I\$SETU = 000041	L\$ICP 002104 G	L10040 037432	MSPERR 011065
HDR40 = 100000	I\$SFT = 000041	L\$INIT 014540 G	L10041 037670	MSTERR 011120
HDSEC = 000077	I\$SRV = 000041	L\$LDAP 002026 G	MAJINC 003472	MTOSLO 006765
HDSEL = 000020	I\$SUB = 000041	L\$LAST 040220 G	MANQ 037721	MUBSF 006510
HDWD = 010135	I\$TST = 000041	L\$LOAD 002100 G	MAPROX 007633	MULOAD 006137
HDWRD1 = 003054	JUNK = 002302	L\$LUN 002074 G	MBADAD 006412	MUNDEF 011317
HDWRD2 = 003056	J\$JMP = 000167	L\$MREV 002050 G	MBSETO = 000001	MWDERR 011152
HDWRD3 = 003060	LABACF 007753	L\$NAME 002000 G	MCERR 010713	MWGERR 011103
HEAD = 000006	LABACR 007767	L\$PRIO 002042 G	MCONHN 007056	MWORD 006757
HEADLM = 010000	LABEXP 007666	L\$PROT 014452 G	MCYLOC 011267	MWRCHK 005775
HEADQ = 040023	LABHCF 007723	L\$PRT 002112 G	MCYLUP 006150	MWRITE 006006
HEADV = 040045	LABHCR 007737	L\$REPP 002062 G	MDATCP 006032	MWRSET 006103
HEADW = 014506	LABIN 007643	L\$REV 002010 G	MDCRC 010735	MWRTAB 011423
HF_IN = 003172	LABMID 007651	L\$SOFT 037526 G	MDHEDR 002000 G	M40HDR 006067
HFINU = 003174	LABOCF 007677	L\$SPC 002056 G	MDT 010762	NEWCYL 003104
HFOUT = 003176	LABOCR 007711	L\$SPCP 002020 G	MDRDY 010702	NOCLR = 000010
HFOUTU = 003200	LABOUT 007660	L\$SPTP 002024 G	MDRERR 011024	NOCTLR 010251
HICYL = 020000	LAB1 = 006731	L\$STA 002030 G	MORRES 007005	NOERCT 003451
HILIM = 000004	LAB2 = 006744	L\$SW 014500 G	MDRVST 011052	NOHD1 010461
HILIMQ = 040002	LF = 000012	L\$TEST 002114 G	MDSERR 011035	NOIRPT = 000002
HILIMW = 014504	LIMVAL = 037774	L\$TML 002014 G	MERRS 011472	NOOP = 000100
HLMTW = 002304	LOCERR = 003450	L\$UNIT 002012 G	MEXERS 011435	NOPWR 006645
HNFERR = 010000	LOCYL = 040000	L.BA 003040	MFBF 006433	NOTROY 010307
HOE = 100000 G	LOE = 040000 G	L.CS 003036	MFLERR 011214	NOTST 010364
HOSTAT = 000020	LOLIM = 000002	L.DA 003042	MFMTER 006563	NXMERR = 020000
HPTCOD = 014460 G	LOLIMQ = 037755	L.MP 003044	MFOLWR 006220	NXTHL 002310
HRDPRM = 037350 G	LOLIMW = 014502	L10000 012712	MFWDSK 006271	NXTPAS 014764
HRDWTS = 027554 G	LOT = 000010 G	L10001 012760	MFWSKO 006322	OBUFF 005072
HRIN = 003202	L\$ACP = 002110 G	L10002 013026	MGTSTA 006020	OFIN 003142
HRINU = 003204	L\$APT = 002036 G	L10003 013076	MHCERR 011134	OFINU 003144
HROUT = 003206	L\$AUT = 002070 G	L10004 013146	MHCRC 010725	OFMID 003146
HROUTU = 003210	L\$AUTO = 015520 G	L10005 014030	MHDERR 011177	OFMIDU 003150
HSMSK = 000100	L\$CCP = 002106 G	L10006 014100	MHDRCP 006051	OFOUT 003152

Symbol table

OFOUTU	003154	PRI06	= 000300 G	SECWD	= 010141	T\$GMAN=	000000	T3265\$	034724
OLDCYI	003102	PRI07	= 000340 G	SEEK	= 000106	T\$HILI=	000377	T331BL	002506
ONSWAP	021570	PSETNM	= 003446	SEEKOP	= 010000	T\$LAST=	000001	T3300\$	035102
OPFLAG	003006	PWCON	= 015246	SEAMES	= 010174	T\$LOLI=	000001	T3301\$	035364
OPIERR	00200	PWRFLG	= 003454	SETDON	= 015020	T\$LSYM=	010000	3365\$	036146
OPMSGS	002226	P2T03E	= 007156	SFTPROM	= 037524 G	T\$LTNO=	000010	3400\$	036310
OPR004	010105	P2T04E	= 007174	SGNW0	= 010130	T\$NEST=	177777	13401\$	036534
OPR1A	010056	P2T05E	= 007214	SKTMES	= 007553	T\$NS0=	000000	T3465\$	037346
OPR1B	010062	P2T06E	= 007234	SPOSTA	= 004000	T\$NS1=	000005	T4	031750 G
OPR12	010037	P2T07E	= 007254	SPTCOD	= 014476 G	T\$NS2=	000002	T5	032564 G
ORIN	003156	P2T08E	= 007272	SRTMES	= 007565	T\$PTNU=	000000	T5.1	033124
ORINU	003160	P2T09E	= 007312	SSINDX	= 003004	T\$SAVL=	177777	T6	033702 G
ORMID	003162	P2T10E	= 007315	STAMES	= 010217	T\$SEGL=	177777	T6.1	033730
ORMIDU	003164	P2T11E	= 007330	STAMSK	= 000007	T\$SEKO=	010000	T7	034726 G
ROUT	003166	P2T12E	= 007343	STATE2	= 011526	T\$SUBN=	000001	T7.1	035364
ROUTU	003170	P2T13E	= 007355	STATE3	= 011536	T\$TAGL=	177777	T8	035150 G
OLTINS	- 000040	P2T14E	= 007402	STATE5	= 011546	T\$TAGN=	010042	T8.1	036534
O\$APTS	000000	P2T15E	= 007423	STCSTA	= 010000	T\$TEMP=	000000	UAM	= 000200 G
O\$AU	- 000000	P2T16E	= 007446	SUBSTK	= 002406	T\$TEST=	000010	ULOAD	= 000010
O\$BGNR	- 000000	P2T17E	= 007467	SVCBGL	= 000001	T\$TSTM=	177777	UNDTST	010072
O\$BGNNS	- 000001	P2T18E	= 007521	SVCGBL	= 000000	T\$TSTS=	000001	UNXERR	007133
O\$DJ	- 000001	P2T19E	= 007543	SVCINS	= 000000	T\$AUT=	010016	VALDES	007607
O\$ERPT	- 000000	RDALHD	= 024324	SVCSSUB	= 000000	T\$CLE=	010017	VCNRST	007112
O\$GNSW	- 000001	RDBSF	= 021630	SVCTAG	= 000000	T\$DU=	010020	VCSTAT	= 001000
O\$POIN	- 000001	RDDATA	= 000114	SVCTST	= 000000	T\$SHAR=	010040	VECMMSG	037446
O\$SL	- 000000	RDHEAD	= 000110	SWAPMD	= 021530	T\$SHW=	010013	VECT	= 000002
O\$SPAT	- 000001	RDNCHR	= 000116	S\$L_YM	= 010000	T\$INI=	010015	VERHDR	023174
O\$SPIN	- 000000	RDYCHK	= 021230	TAG	= 003470	T\$MSG=	010011	VERPOS	024202
O\$SPNW	01472	RDYWAI	= 023570	TBLSTR	= 003026	T\$PRO=	010012	WAITIN	017010
O\$SPNUM	003444	READRL	= 016756	TBT	= 002546	T\$SEG=	010000	WCMSK	= 017777
PATTBL	002362	RELDWT	= 040000	TCERR	= 010230	T\$SOF=	010041	WCRNG	= 160000
PAT1	005472	RESE3	= 011503	TEMP	= 003464	T\$SRV=	010022	WEESTA	= 100000
PAT10	005746	RESE4	= 011507	TEMPO	= 003120	T\$SUB=	010037	WGESTA	= 002000
PAT2	005474	RESE5	= 011514	TEMP1	= 003122	T\$\$SW=	010014	WLSTAT	= 020000
PAT3	005534	RESE6	= 011521	TEMP2	= 003124	T\$\$TES=	010036	WRTSWI	003024
PAT4	005574	RESPAR	= 003064	TEMP3	= 003126	T.BA	= 003050	WTDATA	= 000112
PAT5	005634	RESTAR	= 014734	TEMP4	= 003130	T.CS	= 003046	XDELAY	003456
PAT6	005642	RESTBL	= 002322	TEMP5	= 003132	T.DA	= 003052	XRDHD	022540
PAT7	005702	REVSKD	= 001000	TEMP6	= 003134	T.DRIV	= 002500	XRDHDC	022530
PAT8	005704	REVSKS	= 000200	TEMP7	= 003136	T.MP	= 003054	XRDHDG	022544
PAT9	005744	RLBA	= 000002	TEMP8	= 003140	TSTAT	= 003062	XREAD	025362
PH65\$	021172	RLBAS	= 003030	TIME	= 016210	T1	= 027554 G	XREADG	025370
PNT	- 001000 G	RLCS	= 000000	TIM.US	= 003466	T2	= 031474 G	XSEEK	020112
POSHDS	020664	RLCSR	= 000000	TOSLOW	= 000001	T25TBL	= 002432	XSEEKT	020102
POSHDO	023544	RLDA	= 000004	TRPFLG	= 003452	T25TB2	= 002460	XSEEK1	020116
POSHSB	023540	RLDRA	= 003034	TRPHAN	= 016530	T3	= 031526 G	XT1.IE	016354
POSHW1	023532	RLMP	= 000006	TSTINT	= 017146	T3.1	= 031632	XWRITE	025322
PRI	- 002000 G	RLVEC	= 003032	TSTLAB	= 007150	T306\$	= 031600	XWRITT	025312
PRIOR	- 000004	RORWOP	= 020000	TSTNM	= 003240	T3065\$	= 031746	XWRITI1	025326
PRI00	- 000000 G	RPTOP	= 026274	TYPDR	= 000006	T307\$	= 031632	X\$ALWA	= 000000
PRI01	- 000040 G	RPTREM	= 027270	T\$ARGC	= 000007	T310\$	= 031640	X\$FALS	= 000040
PRI02	- 000100 G	RPTRES	= 027062	T\$CODE	= 006130	T3100\$	= 032716	X\$OFFS	= 000400
PRI03	- 000140 G	RSTRT	= 014652	T\$ERRN	= 003247	T3101\$	= 033124	X\$TRUE	= 000020
PRI04	- 000200 G	SAMSK	= 000077	T\$EXCP	= 000000	T3165\$	= 033700	YDELAY	= 003460
PRI05	- 000240 G	SECQ	= 037704	T\$FLAG	= 000040	T3204\$	= 034602		

b11

CZRLNC0 RL01/02 DRIVE TEST 3 MACRO V05.03b Monday 06-Jan-88 00:23 Page 66-5
Symbol table

SEQ 0131

000000 001 (RW,I,LCL,REL,CON)
Errors detected: 0

*** Assembler statistics

Work file reads: 964
Work file writes: 775
Size of work file: 35176 Words { 138 Pages}
Size of core pool: 14080 Words { 55 Pages}
Operating system: RT-11 (Under RTEM-11)

Elapsed time: 00:05:26.00

CZRLNC.BIC,CZRLNC.LST/C=CZRLNC.DOC,CZRLNC.MAC,SVC41R.MLB/M

ADR	39-15*											
AFMID	39-418*	58-191	58-205									
AFMIDU	39-419*	58-192										
ALLCYL	39-34*	62-18	62-32	64-18	64-37	65-18	65-34					
ALLSEC	39-35*											
ANYERR	39-87*	49-92	52-33	55-127								
ARMIC	39-420*	58-193	58-206									
ARMIDU	39-421*	58-194										
ASSEMB	37-10	37-10										
BADADO	39-66*	57-12	57-55	57 57								
BAMSK	39-98*											
BANAM	39-599*	57-250										
BASADO	39-594*	41-264	43-133	44-19	44-33	47-9	53-113	57-247	58-199	61-87	63-38	63-106
BELL	39-736*	63-45										
BHSTAT	39-124*											
BIT0	39-15*	62-105	62-113	64-116	64-124	64-150	65-118	65-126	65-165			
BIT00	39-15	39-15*	39-34	39-56	39-75	57-196						
BIT01	39-15	39-15*	39-35	39-55	39-76							
BIT02	39-15	39-15*	39-36	39-57	39-77							
BIT03	39-15	39-15*	39-37	39-58	39-78							
BIT04	39-15	39-15*	39-59									
BIT05	39-15	39-15*	39-60									
BIT06	39-15	39-15*	39-61									
BIT07	39-15	39-15*	39-62									
BIT08	39-15	39-15*	39-63									
BIT09	39-15	39-15*	39-64									
BIT1	39-15*	62-118	62-135	64-129	64-165	65-131	65-152					
BIT10	39-15*	39-65	49-64	50-60	54-29	55-223	57-30	58-46	63-56			
BIT11	39-15*	39-66										
BIT12	39-15*	39-38	39-67									
BIT13	39-15*	39-39	39-68									
BIT14	39-15*	39-4	39-69									
BIT15	39-15*	39-41	39-70	41-140								
BIT2	39-15*	57-31										
BIT3	39-15*											
BIT4	39-15*											
BIT5	39-15*											
BIT6	39-15*	53-68	57-135									
BIT7	39-15*											
BIT8	39-15*	53-66										
BIT9	39-15*	41-96										
BOE	39-15*											
BRMSG	66-10	66-16*										
BSCHK	57-118*	60-48	62-99	64-114	65-106							
BSERR	39-31*	66-47	66-47	66-47								
BSERRS	42-36*	53-80	53-84	57-99								
BSFLAG	39-350*	57-121*	57-142*	57-150								
BSFNOT	39-699*	53-112										
BSFVAL	39-454*	43-64*	53-5	53-107*	53-111*	59-10*						
BSOUTQ	66-47	66-71*										
BYPSNM	39-691*	63-48										
C\$AU	37-10*											
C\$AUTO	37-10*	44-38										
C\$BRK	37-10*											
C\$BSEG	37-10*	51-16										
C\$BSUB	37-10*	60-27	62-67	63-13	64-83	65-75						

D1

C\$CLK	37-104	43-9
C\$CLEA	37-104	45-19
C\$CLOS	37-104	
C\$CLP1	37-104	
C\$CPBF	37-104	
C\$CPME	37-104	
C\$CVEC	37-104	44-37 45-13 45-17
C\$OCLN	37-104	43-136 47-12
C\$ODDU	37-104	43 135 44-23 44-36 47-11
C\$DRPT	37-104	
C\$DU	37-104	45-22
C\$EDIT	37-104	39-6
C\$ERDF	37-104	
C\$ERHR	37-104	49-47 49-86 49-101 49-109 50-87 50-93 51-42 51-60 52 24 52 35 53 83 53-97 54-43
	54 55	54 60 54-70 55-29 55-42 55-49 55-114 55-124 55-129 55 191 55-244 55-250 56-43 57-59
	57-75	57-82 57-91 57-101 58-57 58-62 61-46 61-51 61-68 61-73 63-73 63-80 63-84 63 91
C\$ERRO	37-104	
C\$ERSF	37-104	
C\$ERSO	37-104	
C\$ESCA	37-104	60 39 62-154 64-171 65-171
C\$ESEG	37-104	51-67
C\$ESUB	37-104	60-37 62-152 63-103 64-169 65-169
C\$ETST	37-104	58-208 59-12 60-55 61-90 62-168 63-117 64-186 65-185
C\$EXIT	37-104	61-14 63-4 63-74
C\$FREQ	37-104	
C\$FRME	37-104	
C\$GETB	37-104	
C\$GETW	37-104	
C\$GMAN	37-104	
C\$GPHP	37-104	43-70
C\$GPRI	37-104	
C\$INIT	37-104	43-139
C\$INLP	37-104	47-6 51-18 53-86
C\$MANI	37-104	43-14
C\$MAP	37-104	
C\$MEM	37-104	
C\$MMU	37-104	
C\$MSG	37-104	41-13 41-27 41-41 41-56 41-71 41-186 41-200 41-222 41-236 41-252
C\$OPNR	37-104	
C\$OPNW	37-104	
C\$PNTB	37-104	41-116 41-157 41-171 41-180 41-244 41-245 41-248 56-28 57-166 57-167 57-171 57-184 57-202
	57-206	57-211 57 214 57-226 57-235 57-236 57-239 57-247 57-250 57-251 57-252 57-251
C\$PNTF	37-104	43-132 43-133 43-134 44-18 44-19 44-21 44-31 44-33 44-35 47-8 47-9 47-10 53-19
	53-112	53-113 53-114 58-16 58-198 58-199 58-200 58-201 58-202 58-203 58-204 58-205 58-206 61-12
	61-86	61-87 61-88 63-38 63-45 63-48 63-106
C\$PNTS	37-104	
C\$PNTX	37-104	
C\$PUTB	37-104	
C\$PUTW	37-104	
C\$QIO	37-104	
C\$RDBU	37-104	46-9 46-29
C\$REFG	37-104	43-19 43-24 43-46 43-49 43-53
C\$REL	37-104	
C\$RESE	37-104	37-104 43-13 45-18
C\$REVI	37-104	39-6
C\$RFLA	37-104	

EXHCYL	39-424#	58-203	58-204
EXOCYL	39-423#	58-201	58-202
EXROT	39-426#	61-88	
F\$AU	37-10#		
F\$AUTO	37-10#	44-10	44-38
F\$BGN	37-10#	39-5	39-7
	41-58	41-73	41-188
	43-3	43-4	43-140
	58-7	58-208	59-3
	62-67	62-67	62-152
	64-83	64-165	64-171
	66-22	66-23	66-73
F\$CLEA	37-10#	45-4	45-19
F\$DU	37-10#	45-20	45-22
F\$END	37-10	37-10	37-10
	37-10	37-10#	39-5
	41-41	41-56	41-71
	42-47	43-3	43-139
	58-3	58-7	58-7
	60-27	60-27	60-37
	62-3	62-3	62-67
	63-49	63-74	63-103
	64-186	64-186	65-3
	66-12	66-20	66-22
F\$HARD	37-10#	66-4	66-12
F\$HW	37-10#	42-11	42-18
F\$INIT	37-10#	43-4	43-139
F\$JMP	37-10#	61-14	63-49
F\$MOD	37-10#	39-5	39-7
	42-38	42-40	42-47
	66-73		43-3
F\$MSG	37-10#	41-1	41-13
	41-200	41-202	41-222
F\$PROT	37-10#	42-3	42-7
F\$PWR	37-10#		
F\$RPT	37-10#		
F\$SEG	37-10#	51-16	51-67
F\$SOFT	37-10#	66-23	66-35
F\$SRV	37-10#	46-43	46-49
F\$SUB	37-10#	60-27	60-37
F\$SW	37-10#	42-22	42-37
F\$TEST	37-10#	58-7	58-208
	64-186	65-3	65-185
FCTBSF	39-456#	53-31	53-115*
FLDBSF	39-458#	53-75	53-93
FMT1	39-762#	57-184	57-239
FMT10	39-771#		
FMT11	39-772#	57-229	
FMT12	39-773#	57-232	
FMT13	39-774#	57-211	
FMT14	39-775#	41-245	
FMT15	39-776#	41-171	41-248
FMT16	39-777#	57-167	
FMT17	39-778#	41-157	
FMT18	39-779#	58-200	
FMT19	39-780#	58-201	58-202

FMT2	39-763*	57-226	58-198	61-86
FMT20	39-781*	58-203	58-204	
FMT21	39-782*	58-205	58-206	
FMT22	39-783*	57-214		
FMT23	39-784*	63-48		
FMT24	39-785*	43-132	44-18	44-31
FMT25	39-786*	47-8		
FMT26	39-787*	61-88		
FMT27	39-788*	41-180	56-28	
FMT28	39-789*	41-116		
FMT3	39-764*			
FMT4	39-765*	57-171		
FMT5	39-766*	41-244	43-133	44-19
FMT6	39-767*	57-250		
FMT7	39-768*	57-252		
FMT8	39-769*	57-251		
FMT9	39-770*	57-166		
FMTOP1	39-759*	63-38	63-106	
FMTOP2	39-760*			
FMTOP3	39-761*			
FMTXT	39-758*	57-202	57-206	63-45
FOLWRT	39-61*	39-71	62-114	64-125
FRMWD	39-690*	57-211		65-127
FWDSKO	39-65*	39-71	62-142	65-158
FWDSKS	39-63*	39-71	64-164	
G\$CNTO	37-10*			
G\$DELM	37-10*	46-11	46-17	46-32
G\$DISP	37-10*			46-37
G\$EXCP	37-10*			
G\$HILI	37-10*			
G\$LOLI	37-10*			
G\$NO	37-10*			
G\$OFFS	37-10*	66-5	66-6	66-7
	66-40	66-42	66-44	66-46
G\$OFSI	37-10*	66-5	66-6	66-7
	66-40	66-42	66-44	66-46
G\$PRMA	37-10*	66-6	66-7	
G\$PRMD	37-10*	66-8	66-10	66-36
G\$PRML	37-10*	66-5	66-9	66-25
G\$RADA	37-10*			66-39
G\$RADB	37-10*			
G\$RADD	37-10*	66-36	66-39	66-42
G\$RADL	37-10*	66-5	66-4	66-25
G\$RADO	37-10*	66-6	66-7	66-8
G\$XFER	37-10*	66-35	66-38	66-41
G\$YES	37-10*	66-5	66-6	66-7
	66-40	66-42	66-44	66-46
GBND	39-177*	43-86*	43-04*	
GETPOS	50-19	51-13	51-20	51-46
GETSTA	39-111*	41-89	49-4	49-8
GLBDAT	39-146*			
GLBEQA	39-13*			
GLBERR	40-3*			
GLBSUB	46-3*			
GLBTXT	39-564*			
GSTAT	49-11*	49-28	49-83	49-90
			52-15	52-26
				54-45
				55-104
				55-116
				57-84
				63-75

L\$EXP1	39-6*
L\$EXP4	39-6*
L\$EXP5	39-6*
L\$HARD	39-6 66-4 66-4*
L\$HIME	39-6*
L\$MPCP	39-6*
L\$MPTP	39-6*
L\$HW	39-6 42-11 42-11*
L\$ICP	39-6*
L\$INIT	39-6 43-4*
L\$LADP	39-6*
L\$LAST	39-6 66-74*
L\$LOAD	39-6*
L\$LUN	39-6*
L\$MREV	39-6*
L\$NAME	39-6*
L\$PRIO	39-6*
L\$PROT	39-6 42-3*
L\$PRT	39-6*
L\$REPP	39-6*
L\$REV	39-6*
L\$SOFT	39-6 66-23 66-23*
L\$SPC	39-6*
L\$SPCP	39-6*
L\$SPTP	39-6*
L\$STA	39-6*
L\$SW	39-6 42-22 42-22*
L\$TEST	39-6*
L\$TIML	39-6*
L\$UNIT	39-6 43-21 43-30 43-61
L.BA	39-359# 55-224* 55-230 57-251
L.CS	39-358# 49-63* 49-64* 49-65* 49-67 49-113 50-57 54-27 54-74 55-221* 55-222* 55-223* 55-235 57-27
	57-173 57-251 58-50 58-73* 63-53
L.DA	39-360# 49-57* 49-60* 49-66 49-73 55-225* 55-228* 55-229 57-187 57-251 58-49
L.MP	39-361# 49-53 54-19 57-251 61-34 61-54
L10000	41-13*
L10001	41-27*
L10002	41-41*
L10003	41-56*
L10004	41-71*
L10005	41-186*
L10006	41-200*
L10007	41-222*
L10010	41-236*
L10011	41-252*
L10013	42-11 42-18*
L10014	42-22 42-37*
L10015	43-139#
L10016	44-38*
L10017	45-19*
L10020	45-22*
L10021	46-49*
L10022	46-61*
L10023	58-208*
L10024	59-12*
L10025	60-39 60-55*

L10026	60-37*							
L10027	61-14	61-90*						
L10030	62-154	62-168*						
L10031	62-152*							
L10032	63-49	63-117*						
L10033	63-74	63-103*						
L10034	64-171	64-186*						
L10035	64-169*							
L10036	65-171	65-185*						
L10037	65-169*							
L10040	66-4	66-12*						
L10041	66-23	66-49*						
LAB1	39-602*	57-251						
LAB2	39-603*	57-252						
LABACF	39-659*	58-205						
LABACR	39-660*	58-206						
LABEXP	39-654*	58-200						
LABHCF	39-657*	58-203						
LABHCR	39-658*	58-204						
LABIN	39-651*	58-200						
LABMID	39-652*	58-200						
LABOCF	39-655*	58-201						
LABOCR	39-656*	58-202						
LABOUT	39-653*	58-200						
LF	39-567	39-697	39-698					
LIMVAL	66-36	66-39	66-63*					
LOCERR	39-438*	53-34*	53-91*	53-105*	53-108			
LOCYL	39-40*	43-40						
LOE	39-15*							
LOLIM	39-26*	66-36	66-36	66-36				
LOLIMQ	66-34	66-62*						
LOLIMW	42-31*	43-42*	61-25	62-24	62-53	64-26	64-64	65-26
LOT	39-15*							65-61
M40HDR	39-170	39-577*						
MAJINC	39-448*	46-25*	46-40					
MANQ	66-32	66-60*						
MAPROX	39-650*	61-88						
MBADAD	39-169	39-590*						
MBSETO	39-106*	50-73						
MCERR	39-183	39-703*						
MCONMN	39-608*	48-19						
MCYLOC	39-729*	41-208	57-230					
MCYLU	39-160	39-582*						
MDATCP	39-158	39-575*						
MDCRC	39-705*	41-132						
MDEHDR	39-5*							
MDLT	39-707*	41-137						
MORDY	39-702*	52-22	54-41	55-113	57-80			
MORERR	39-184	39-709*	63-79					
MORRES	39-606*	48-13						
MORVST	39-719*	41-193						
MSERR	39-198	39-718*						
MERRS	39-735*	41-116						
MEXERS	39-734*	47-8						
MFBSF	39-591*	53-92						
MFLERR	39-186	39-727*						

N1

CZRLNCO RL01/02 DRIVE TEST 3 MACRO V05.03b Monday 06-Jan-86 00:23 Page S-12
Cross reference table (CREF V05.03)

SEQ 01 3

Cross reference table (CREF V05.03)

T\$SOF	66 23	66-23#	66-49	46-61	63-13#	63 74	63-103	64 83#	64-169	65-75#	65-169			
T\$SRV	46-43#	46-49	46-50#	46-61	62-152	63-13#	63 74	63-103	64 83#	64-169	65-75#	65-169		
T\$SUB	60-27#	60-37	62-67#	62-152	63-13#	63 74	63-103	64 83#	64-169	65-75#	65-169			
T\$SW	42-22	42-22#	42-37											
T\$TES	58-7#	58-208	59-3#	59-12	60-3#	60-39	60-55	61-3#	61-14	61-90	62-3#	62-154	62-168	63-3#
T\$ARGC	63-49	63-117	64-3#	64-171	64-186	65-3#	65-171	65-185						
	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6#	39-6#	39-6#
	39-6#	39-6#	41-116	41-116	41-116	41-116	41-116	41-116#	41-116#	41-116#	41-116#	41-116#	41-157	41-157
	41-157	41-157	41-157#	41-157#	41-157#	41-157#	41-157#	41-157#	41-157#	41-157#	41-157#	41-157#	41-157	41-157
	41-171#	41-171#	41-171#	41-171#	41-171#	41-171#	41-171#	41-171#	41-171#	41-171#	41-171#	41-171#	41-171#	41-171#
	41-180#	41-180#	41-180#	41-244	41-244	41-244	41-244	41-244	41-244	41-244	41-244#	41-244#	41-244#	41-244#
	41-245	41-245	41-245	41-245	41-245	41-245	41-245	41-245	41-245	41-245	41-245#	41-245#	41-245#	41-245#
	41-245#	41-245#	41-248	41-248	41-248	41-248	41-248	41-248	41-248	41-248	41-248	41-248#	41-248#	41-248#
	41-248#	41-248#	41-248#	43-132	43-132	43-132	43-132#	43-132#	43-132#	43-133	43-133	43-133	43-133	43-133
	43-133	43-133#	43-133#	43-133#	43-133#	43-134	43-134	43-134#	44-18	44-18	44-18	44-18	44-18#	44-18#
44-19	44-19	44-19	44-19	44-19	44-19	44-19#	44-19#	44-19#	44-19#	44-19#	44-19#	44-21	44-21	44-21#
44-31	44-31	44-31	44-31#	44-31#	44-31#	44-33	44-33	44-33	44-33	44-33	44-33	44-33#	44-33#	44-33#
44-33#	44-33#	44-35	44-35	44-35#	47-8	47-8	47-8	47-8	47-8	47-8	47-8#	47-8#	47-9	47-9
47-9	47-9	47-9	47-9	47-9	47-9	47-9#	47-9#	47-9#	47-9#	47-9#	47-10	47-10#	53-19	53-19
53-19	53-19#	53-19#	53-112	53-112	53-112#	53-113	53-113	53-113	53-113	53-113	53-113	53-113	53-113#	53-113#
53-113#	53-113#	53-113#	53-114	53-114	53-114	56-28	56-28	56-28	56-28	56-28	56-28	56-28#	56-28#	56-28#
56-28#	56-28#	56-28#	57-166	57-166	57-166#	57-166#	57-167	57-167	57-167	57-167	57-167#	57-167#	57-171	57-171
57-171	57-171	57-171	57-171#	57-171#	57-171#	57-184	57-184	57-184	57-184	57-184	57-184#	57-184#	57-184#	57-202
57-202	57-202	57-202#	57-202#	57-206	57-206	57-206	57-206#	57-206#	57-211	57-211	57-211	57-211	57-211	57-211
57-211	57-211	57-211	57-211	57-211	57-211#	57-211#	57-211#	57-211#	57-211#	57-211#	57-211#	57-211#	57-211#	57-211#
57-214	57-214	57-214	57-214	57-214	57-214	57-214	57-214	57-214	57-214	57-214	57-214#	57-214#	57-214#	57-214#
57-214#	57-226	57-226	57-226	57-226	57-226#	57-226#	57-226#	57-226#	57-235	57-235	57-235	57-235	57-235	57-235#
57-235#	57-236	57-236	57-236	57-236	57-236#	57-236#	57-236#	57-236#	57-239	57-239	57-239	57-239	57-239	57-239#
57-239#	57-247	57-247	57-247	57-247	57-247	57-247	57-247#	57-247#	57-247#	57-247#	57-247#	57-247#	57-250	57-250
57-250	57-250	57-250	57-250	57-250	57-250	57-250#	57-250#	57-250#	57-250#	57-250#	57-250#	57-250#	57-250#	57-251
57-251	57-251	57-251	57-251	57-251	57-251	57-251#	57-251#	57-251#	57-251#	57-251#	57-251#	57-251#	57-251#	57-252
57-252	57-252	57-252	57-252	57-252	57-252	57-252	57-252#	57-252#	57-252#	57-252#	57-252#	57-252#	57-252#	57-252#
57-252#	58-16	58-16	58-16	58-16#	58-16#	58-198	58-198	58-198	58-198	58-198	58-198#	58-198#	58-198#	58-199
58-199	58-199	58-199	58-199	58-199	58-199#	58-199#	58-199#	58-199#	58-199#	58-199#	58-199#	58-200	58-200	58-200
58-200	58-200	58-200#	58-200#	58-200#	58-200#	58-200#	58-200#	58-201	58-201	58-201	58-201	58-201	58-201	58-201
58-201#	58-201#	58-201#	58-201#	58-201#	58-201#	58-202	58-202	58-202	58-202	58-202	58-202	58-202	58-202	58-202#
58-202#	58-202#	58-202#	58-202#	58-202#	58-203	58-203	58-203	58-203	58-203	58-203	58-203	58-203#	58-203#	58-203#
58-203#	58-203#	58-204	58-204	58-204	58-204	58-204	58-204	58-204#	58-204#	58-204#	58-204#	58-204#	58-204#	58-205
58-205	58-205	58-205	58-205	58-205#	58-205#	58-205#	58-205#	58-206	58-206	58-206	58-206	58-206	58-206	58-206#
58-206#	58-206#	61-87	61-87	61-87	61-87	61-87	61-87#	61-87#	61-87#	61-87#	61-87#	61-86	61-86	61-86#
61-86#	61-87	61-87	61-87	61-87	61-87	61-87	61-87#	61-87#	61-87#	61-87#	61-87#	61-87#	61-87#	61-88
61-88	61-88	61-88	61-88	61-88	61-88#	61-88#	61-88#	61-88#	61-88#	61-88#	61-88#	63-38	63-38	63-38
63-38	63-38	63-38	63-38	63-38	63-38#	63-38#	63-38#	63-38#	63-38#	63-38#	63-38#	63-38#	63-38#	63-45
63-45	63-45#	63-45#	63-48	63-48	63-48	63-48	63-48	63-48	63-48	63-48	63-48	63-48#	63-48#	63-48#
63-106	63-106	63-106	63-106	63-106	63-106	63-106	63-106	63-106#	63-106#	63-106#	63-106#	63-106#	63-106#	63-106#
T\$CODE	63-106#	66-5	66-5	66-5#	66-5#	66-6	66-6	66-6#	66-6#	66-6#	66-6#	66-6#	66-7	66-7
	66-7	66-7#	66-7#	66-8	66-8	66-8#	66-8#	66-8#	66-8#	66-8#	66-9	66-9	66-9#	66-9#
	66-9#	66-9#	66-10	66-10	66-10	66-10#	66-10#	66-10#	66-10#	66-10#	66-25	66-25	66-25#	66-25#
	66-26	66-26	66-26#	66-26#	66-26#	66-32	66-32	66-32	66-32	66-32	66-32#	66-32#	66-32#	66-34
	66-34	66-34#	66-34#	66-35	66-35	66-35	66-35	66-35	66-35	66-35	66-35#	66-35#	66-35#	66-35#
	66-36	66-36	66-36	66-36#	66-36#	66-37	66-37	66-37	66-37	66-37	66-37#	66-37#	66-38	66-38
	66-38	66-38	66-38	66-38#	66-38#	66-38#	66-38#	66-38#	66-38#	66-38#	66-39	66-39	66-39#	66-39#
	66-40	66-40	66-40#	66-40#	66-40#	66-41	66-41	66-41	66-41	66-41	66-41	66-41	66-41	66-41#
	66-41#	66-42	66-42	66-42	66-42#	66-42#	66-42#	66-42#	66-42#	66-44	66-44	66-44	66-44	66-44#
	66-46	66-46	66-46#	66-46#	66-46#	66-47	66-47	66-47	66-47	66-47	66-47#	66-47#	66-47#	66-47#

三五四

57-211#	57-211#	57-211#	57-211#	57-211#	57-211#	57-211#	57-211#	57-211#	57-211#	57-211#	57-214	57-214	57-214	57-214
57-214	57-214	57-214	57-214	57-214#	57-214#	57-214#	57-214#	57-214#	57-214#	57-214#	57-214#	57-214#	57-226	57-226
57-226	57-226	57-226#	57-226#	57-226#	57-226#	57-235	57-235	57-235	57-235	57-235	57-235#	57-235#	57-235#	57-235#
57-236	57-236	57-236	57-236	57-236#	57-236#	57-236#	57-236#	57-236#	57-236#	57-236#	57-239	57-239	57-239#	57-239#
57-239#	57-239#	57-247	57-247	57-247	57-247	57-247	57-247	57-247	57-247	57-247	57-247#	57-247#	57-247#	57-247#
57-250	57-250	57-250	57-250	57-250	57-250	57-250	57-250	57-250	57-250	57-250	57-250#	57-250#	57-250#	57-250#
57-250#	57-250#	57-251	57-251	57-251	57-251	57-251	57-251	57-251	57-251	57-251	57-251#	57-251#	57-251#	57-251#
57-251#	57-251#	57-252	57-252	57-252	57-252	57-252	57-252	57-252	57-252	57-252	57-252	57-252	57-252#	57-252#
57-252#	57-252#	57-252#	57-252#	57-252#	57-252#	58-16	58-16	58-16	58-16	58-16#	58-16#	58-198	58-198	
58-198	58-198	58-198#	58-198#	58-198#	58-198#	58-199	58-199	58-199	58-199	58-199	58-199	58-199#	58-199#	58-199#
58-199#	58-199#	58-199#	58-199#	58-199#	58-200	58-200	58-200	58-200	58-200	58-200#	58-200#	58-200#	58-200#	58-200#
58-200#	58-200#	58-201	58-201	58-201	58-201	58-201	58-201	58-201	58-201	58-201#	58-201#	58-201#	58-201#	58-201#
58-201#	58-201#	58-202	58-202	58-202	58-202	58-202	58-202	58-202	58-202	58-202#	58-202#	58-202#	58-202#	58-202#
58-202#	58-202#	58-203	58-203	58-203	58-203	58-203	58-203	58-203	58-203	58-203#	58-203#	58-203#	58-203#	58-203#
58-204	58-204	58-204	58-204	58-204	58-204	58-204#	58-204#	58-204#	58-204#	58-204#	58-204#	58-205	58-205	58-205
58-205	58-205	58-205	58-205#	58-205#	58-205#	58-205#	58-205#	58-205#	58-205#	58-206	58-206	58-206	58-206	58-206#
58-206#	58-206#	58-206#	58-206#	58-206#	61-12	61-12	61-12#	61-12#	61-12#	61-12#	61-12#	61-86	61-86	61-86
61-86#	61-86#	61-86#	61-86#	61-86#	61-87	61-87	61-87	61-87	61-87	61-87#	61-87#	61-87#	61-87#	61-87#
61-87#	61-87#	61-88	61-88	61-88	61-88	61-88	61-88	61-88	61-88	61-88#	61-88#	61-88#	61-88#	61-88#
61-88#	61-88#	63-38	63-38	63-38	63-38	63-38	63-38	63-38	63-38	63-38#	63-38#	63-38#	63-38#	63-38#
63-38#	63-38#	63-38#	63-38#	63-38#	63-45	63-45	63-45#	63-45#	63-45#	63-45#	63-45#	63-48	63-48	63-48
63-48	63-48	63-48#	63-48#	63-48#	63-48#	63-48#	63-48#	63-48#	63-48#	63-106	63-106	63-106	63-106	63-106
63-106	63-106	63-106#	63-106#	63-106#	63-106#	63-106#	63-106#	63-106#	63-106#	63-106#	63-106#	63-106	63-106	63-106
M\$RADI	66-5	66-5#	66-6	66-6#	66-7	66-7#	66-8	66-8#	66-9	66-9#	66-10	66-10#	66-25	66-25#
	66-26	66-26#	66-32	66-32#	66-34	66-34#	66-36	66-36#	66-37	66-37#	66-39	66-39#	66-40	66-40#
M\$RNRO	66-42	66-42#	66-44	66-44#	66-46	66-46#	66-47	66-47#						
M\$SETS	43-9	43-9#	43-70	43-70#										
	39-5#	39-13	39-13#	39-146	39-146#	39-564	39-564#	40-3	40-3#	41-1	41-1#	41-15	41-15#	
	41-29	41-29#	41-43	41-43#	41-58	41-58#	41-73	41-73#	41-188	41-188#	41-202	41-202#	41-224	41-224#
	41-238	41-238#	42-3	42-3#	42-10	42-10#	42-11	42-11#	42-21	42-21#	42-22	42-22#	42-40	42-40#
	43-3	43-3#	43-4	43-4#	44-10	44-10#	45-3	45-3#	45-4	45-4#	45-20	45-20#	46-3	46-3#
	46-43	46-43#	46-50	46-50#	51-16	51-16	51-16#	51-16#	58-3	58-3#	58-7	58-7#	59-3	59-3#
	60-3	60-3#	60-27	60-27#	61-3	61-3#	62-3	62-3#	62-67	62-67#	63-3	63-3#	63-13	63-13#
	64-3	64-3#	64-83	64-83#	65-3	65-3#	65-75	65-75#	66-3	66-3#	66-4	66-4#	66-22	66-22#
	66-23	66-23#												
M\$SVC	41-13	41-13#	41-27	41-27#	41-41	41-41#	41-56	41-56#	41-71	41-71#	41-116	41-116#	41-157	41-157#
	41-171	41-171#	41-180	41-180#	41-186	41-186#	41-200	41-200#	41-222	41-222#	41-236	41-236#	41-244	41-244#
	41-245	41-245#	41-248	41-248#	41-252	41-252#	43-9	43-9#	43-12	43-12#	43-13	43-13#	43-14	43-14#
	43-19	43-19#	43-24	43-24#	43-46	43-46#	43-49	43-49#	43-53	43-53#	43-70	43-70#	43-104	43-104#
	43-105	43-105#	43-132	43-132#	43-133	43-133#	43-134	43-134#	43-135	43-135#	43-136	43-136#	43-139	43-139#
	44-12	44-12#	44-18	44-18#	44-19	44-19#	44-21	44-21#	44-23	44-23#	44-31	44-31#	44-33	44-33#
	44-35	44-35#	44-36	44-36#	44-37	44-37#	44-38	44-38#	45-5	45-5#	45-6	45-6#	45-13	45-13#
	45-17	45-17#	45-18	45-18#	45-19	45-19#	45-22	45-22#	46-9	46-9#	46-29	46-29#	47-6	47-6#
	47-8	47-8#	47-9	47-9#	47-10	47-10#	47-11	47-11#	47-12	47-12#	49-47	49-86	49-101	49-109
	50-87	50-93	51-16	51-16#	51-18	51-18#	51-42	51-60	51-67	51-67#	52-24	52-35	53-19	53-19#
	53-83	53-86	53-86#	53-97	53-112	53-112#	53-113	53-113#	53-114	53-114#	54-43	54-55	54-60	54-70
	55-29	55-42	55-49	55-114	55-124	55-129	55-191	55-244	55-250	56-28	56-28#	56-43	57-59	57-75
	57-82	57-91	57-101	57-166	57-166#	57-167	57-171	57-171#	57-184	57-184#	57-202	57-202#	57-206	
	57-206#	57-211	57-211#	57-214	57-214#	57-226	57-226#	57-235	57-235#	57-236	57-236#	57-239	57-239#	57-247
	57-247#	57-250	57-250#	57-251	57-251#	57-252	57-252#	58-16	58-16#	58-57	58-62	58-198	58-198#	58-199
	58-199#	58-200	58-200#	58-201	58-201#	58-202	58-202#	58-203	58-203#	58-204	58-204#	58-205	58-205#	58-206
	58-206#	58-208	58-208#	59-12	59-12#	60-27	60-27#	60-37	60-37#	60-39	60-39#	60-55	60-55#	61-12
	61-12#	61-14	61-14#	61-46	61-51	61-68	61-73	61-86	61-86#	61-87	61-87#	61-88	61-88#	61-90
	61-90#	62-67	62-67#	62-152	62-152#	62-154	62-154#	62-168	62-168#	63-13	63-13#	63-38	63-38#	63-45
	63-45#	63-48	63-48#	63-49	63-49#	63-73	63-74	63-74#	63-80	63-84	63-91	63-103	63-103#	63-106
	63-106#	63-117	63-117#	64-83	64-83#	64-169	64-169#	64-171	64-171#	64-186	64-186#	65-75	65-75#	65-169

