

RL11,RLV11

RL01/02 PERF EXER
CZRLKBO

AH-F126B MC
FICHE 1 OF 1

MAR 1980
COPYRIGHT 1980
MADE IN USA



Microfilm frame containing a grid of 12 columns and 12 rows of data. Each cell contains a small table or chart with illegible text and numbers.



IDENTIFICATION

PRODUCT CODE: AC-F127B-MC
PRODUCT NAME: CZRLK80 RL01/02 PERFORMANCE EXERCISER
DATE CREATED: 5-JAN-79
REVISED: 7-DEC-79
MAINTAINER: DIAGNOSTIC ENGINEERING
AUTHORS: D. DEKNIS, C. CAMPBELL

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

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

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

COPYRIGHT (C) 1979, DIGITAL EQUIPMENT CORPORATION

TABLE OF CONTENTS

1.0	GENERAL INFORMATION
1.1	PROGRAM ABSTRACT
1.1.1	STRUCTURE OF PROGRAM
1.1.2	DIAGNOSTIC INFORMATION
1.2	SYSTEM REQUIREMENTS
1.2.1	HARDWARE REQUIREMENTS
1.2.2	SOFTWARE REQUIREMENTS
1.3	RELATED DOCUMENTS AND STANDARDS
1.4	DIAGNOSTIC HIERARCHY PREREQUISITES
1.5	ASSUMPTIONS
2.0	OPERATING INSTRUCTIONS
2.1	HOW TO RUN THIS DIAGNOSTIC
2.1.1	THE FIVE STEPS OF EXECUTION
2.1.2	SAMPLE RUN-THROUGH
2.2	CHAIN MODE OPERATION
2.3	DETAILS OF COMMANDS AND SYNTAX
2.3.1	TABLE OF COMMAND VALIDITY
2.3.2	COMMAND SYNTAX
2.4	EXTENDED P-TABLE DIALOGUE
2.5	HARDWARE PARAMETERS
2.6	SOFTWARE PARAMETERS
3.0	ERROR INFORMATION
3.1	ERROR REPORTING
3.2	ERROR HALTS
4.0	PERFORMANCE AND PROGRESS REPORTS
4.1	PERFORMANCE REPORTS
4.2	PROGRESS REPORTS
5.0	DEVICE INFORMATION TABLES
6.0	TEST SUMMARIES

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

THIS DIAGNOSTIC IS COMPATIBLE WITH BOTH XXDP+ AND ACT. IT CAN BE RUN STANDALONE UNDER XXDP+, AND CAN BE CHAINED UNDER XXDP+, ACT AND APT IN ACT MODE (SEE 2.2 "CHAIN MODE OPERATION" FOR DETAILS OF CHAINING PROCEDURE). IT IS A SINGLE PROGRAM FROM THE STANDPOINT OF THE DIAGNOSTIC USER, WHICH AT RUN TIME IS APPENDED TO A COMMON FRONT-END PIECE OF SUPERVISOR SOFTWARE THROUGH WHICH THE DIAGNOSTIC PROGRAM INTERFACES TO THE ENVIRONMENT AS IT EXECUTES.

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

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

1.1.2 DIAGNOSTIC INFORMATION

THE RL11/RLV11 RLO1/02 EXERCISER IS A PDP-11 (LSI-11) BASED PROGRAM. IT WILL RANDOMLY EXERCISE UP TO 2 CONTROLLERS AND 8 DRIVES. AFTER AN INITIAL WRITE OF EACH RLO1/02, THE DRIVES ARE RANDOMLY PICKED AND GIVEN A RANDOM STRING FUNCTION OF:

1. SEEK, WRITE, WRITE-CHECK
2. SEEK, READ DATA, DATA COMPARE
3. SEEK, READ HEADERS, READ 1 SECTOR W/NO HEADER COMPARE, GET STATUS
4. SEEK, READ, READ

1.2 SYSTEM REQUIREMENTS1.2.1 HARDWARE REQUIREMENTS

- * PDP-11/LSI-11 PROCESSOR WITH 16K OR MORE OF MEMORY
- * CONSOLE DEVICE (LA30,LA36,VT50,ETC.)
- * 1 OR 2 RL11/RLV11 CONTROLLER(S) WITH:
 - 1 - 8 RL01 DRIVES WITH RL01K CARTRIDGES CONTAINING A 'BAD SECTOR FILE'
 - 1 - 8 RL02 DRIVES WITH RL02K CARTRIDGES CONTAINING A 'BAD SECTOR FILE'
- * KW11-L OR KW11-P CLOCK
- * LINE PRINTER (OPTIONAL)

1.2.2 SOFTWARE REQUIREMENTS

CZRLKBO RL11/RLV11 RL01/RL02 PERFORMANCE EXERCISER
(FORMERLY CZRLEBO)

1.3 RELATED DOCUMENTS AND STANDARDS

RL01 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:

CVRLABO	RLV11 RL01 DISKLESS TEST (RLV11 ONLY)
CZRLGBO	RL11/RLV11 RL01/02 CONTROLLER TEST (PART 1)
CZRLHBO	RL11/RLV11 RL01/02 CONTROLLER TEST (PART 2)
CZRLIBO	RL01/02 DRIVE TEST (PART 1)
CZRLJBO	RL01/02 DRIVE TEST (PART 2)
CZRLNAO	RL01/02 DRIVE TEST (PART 3)

1.5 ASSUMPTIONS

THE HARDWARE OTHER THAN THE RL01 SUBSYSTEM IS ASSUMED TO WORK PROPERLY. FALSE ERRORS MAY BE REPORTED IF THE PROCESSOR, ETC., DO NOT FUNCTION PROPERLY.

2.0 OPERATING INSTRUCTIONS

2.1 HOW TO RUN THIS DIAGNOSTIC

2.1.1 THE FIVE STEPS OF EXECUTION

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

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

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

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

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

WHEN THIS DIAGNOSTIC IS STARTED THE FOLLOWING STEPS WILL OCCUR:

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

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

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

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

THINGS TO NOTE HERE:

1. ONLY THE FIRST THREE CHARACTERS OF THIS OR ANY COMMAND AT THE 'DR>' LEVEL NEED TO BE TYPED.

2. THE 'PASS' SWITCH SPECIFIES HOW MANY PASSES YOU DESIRE. A PASS CONSISTS OF RUNNING THE FULL DIAGNOSTIC AGAINST ALL UNITS BEING TESTED (THIS WILL BE EXPLAINED SHORTLY). ONE PASS IS SPECIFIED IN THE ABOVE EXAMPLE.
3. THE 'FLAGS' SWITCH MAY SPECIFY ANY OF A NUMBER OF FLAGS, BUT THE MAIN USEFUL ONES ARE:

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

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

* STEP 2 *

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

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

* STEP 3 *

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

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

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

* STEP 4 *

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

* STEP 5 *

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

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

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

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

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

2.1.2 SAMPLE RUN-THROUGH

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

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

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

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

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

THIS WILL DO THE FOLLOWING:

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

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

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

1. START
2. RESTART
3. CONTINUE

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

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

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

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

1. START
2. RESTART

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

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

	BY WHOM ENTERED: -----
.R CZRLKB	O
DRS LOADED	D
DIAG. RUN-TIME SERVICES REV. D APR-79	D
CZRLK-B-0	D
CZRLK RANDOMLY PERFORMS DRIVE SEEK, READ, AND WRITE FUNCTIONS	D
UNIT IS RL01, RL02	D
DR>STA/PASS:1/FLAGS:HOE	D,O
CHANGE HW (L) ? Y	D,O
# UNITS (D) ? 2	D,O
UNIT 0	D
RL11 (L) Y ?	D,O
BUS ADDRESS (O) 174400 ?	D,O
VECTOR (O) 160 ?	D,O
DRIVE (O) 0 ?	D,O
DRIVE TYPE = RL01 (L) Y ?	D,O
BR LEVEL (O) 5 ?	D,O
UNIT 1	D
RL11 (L) Y ?	D,O
BUS ADDRESS (O) 174400 ?	D,O
VECTOR (O) 160 ?	D,O
DRIVE (O) 0 ? 1	D,O
DRIVE TYPE = RL01 (L) ? N	D,O (N=RL02)
BR LEVEL (O) 5 ?	D,O
CHANGE SW (L) ? N	D,O
CZRLK HRD ERR 00004 TST 003 SUB 002 PC:004130 ERR HLT	
DR>PRO/FLAGS:IER:LOE:HOE=0	D,O

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

^C

O


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

2.2 CHAIN MODE OPERATION

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

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

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

TO EXECUTED A CHAIN FILE THE USER TYPES:

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

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

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

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

2.3 DETAILS OF COMMANDS AND SYNTAX

2.3.1 TABLE OF COMMAND VALIDITY

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

<u>HOW ENTERED</u>	<u>LEGAL COMMANDS</u>
1. OPERATOR ENTERED 'RUN DIAG'	START PRINT DISPLAY FLAGS ZFLAGS EXIT
2. DIAGNOSTIC HAS FINISHED ALL ITS REQUESTED PASSES	START RESTART PRINT DISPLAY FLAGS ZFLAGS EXIT
3. OPERATOR INTERRUPTED THE DIAGNOSTIC WITH CTRL/C	START RESTART CONTINUE PRINT DISPLAY FLAGS ZFLAGS EXIT
4. AN ERROR WAS ENCOUNTERED WITH THE HOE FLAG SET SET	START RESTART CONTINUE PROCEED PRINT DISPLAY FLAGS ZFLAGS EXIT

2.3.2 COMMAND SYNTAX

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

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

THE SWITCH ARGUMENTS ARE AS FOLLOWS:

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

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

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

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

IER INHIBIT ERROR REPORTING

IBE INHIBIT BASIC ERROR REPORTS

IXE INHIBIT EXTENDED ERROR REPORTS

PRI DIRECT ALL MESSAGES TO A LINE PRINTER

PNT PRINT NUMBER OF TEST BEING EXECUTED

BOE BELL ON ERROR
UAM RUN IN UNATTENDED MODE, BYPASSING MANUAL INTERVENTION TESTS
ISR INHIBIT STATISTICAL REPORTS
IDU INHIBIT DROPPING OF UNITS BY DIAGNOSTIC
ADR EXECUTE AUTODROP CODE
LOT LOOP ON TEST
EVL EVALUATE

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

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

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

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

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

THE SWITCH ARGUMENTS ARE AS IN THE START COMMAND EXCEPT:

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

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

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

THE SWITCH ARGUMENTS ARE AS IN THE START COMMAND EXCEPT:

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

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

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

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

1. UNSPECIFIED FLAG SETTINGS ARE UNCHANGED

EXIT

RETURN TO XXDP+ PROMPT MODE.

DRO(P)/UNITS:UNIT-LIST

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

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

ADD/UNITS:UNIT-LIST

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

PRI(NT)

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

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

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

FLA(GS)

THE CURRENT SETTINGS OF ALL FLAGS ARE PRINTED.

ZFL(AGS) .

ALL FLAGS ARE CLEARED.

2.4 EXTENDED P-TABLE DIALOGUE

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

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

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

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

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

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

UNITS (D) ? 8

UNIT 0

RL11 (L) Y ?

BUS ADDRESS (O) 174400 ?

VECTOR (O) 160 ?

DRIVE (O) 0 ? 0-3

DRIVE TYPE = RL01 (L) Y ?

BR LEVEL (O) 5 ?

UNIT 4

RL11 (L) Y ?

BUS ADDRESS (O) 174400 ? 175400

VECTOR (O) 160 ? 164

DRIVE (O) 0 ? 0-3

DRIVE TYPE = RL01 (L) Y ? N

BR LEVEL (O) 5 ?

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

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

2.5 HARDWARE PARAMETERS

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

RL11 (L) Y?

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

BUS ADDRESS (O) 174400?

ANSWER WITH THE BUS ADDRESS OF THE CONTROLLER.

VECTOR (O) 160?

ANSWER WITH THE INTERRUPT VECTOR OF THE CONTROLLER.

DRIVE (O) 0?

ANSWER WITH THE DRIVE(S) CONNECTED TO THE CONTROLLER

DRIVE TYPE = RL01 (L) ?

ANSWER NO (N) IF DRIVE IS AN RL02

BR LEVEL (O) 5?

ANSWER WITH THE INTERRUPT PRIORITY OF THE CONTROLLER.

2.6 SOFTWARE PARAMETERS

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

'CHANGE S.W. ?'

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

'RETRY LMT (D) 1 ?'

THIS IS THE NUMBER OF TIMES THE PROGRAM WILL ATTEMPT A COMMAND BEFORE IT QUILTS AND REPORTS A HARD ERROR. IF THE RETRY IS SUCCESSFUL BEFORE THE RETRY LIMIT IS EXCEEDED IT WILL PRINT AND LOG A SOFT ERROR.

LIMITS 0 - 65,535

"SEEK RETRY LMT (D) 1 ?"

THIS IS THE NUMBER OF RETRYS THAT WILL BE ATTEMPTED TO SEEK TO A CYLINDER ON A MIS-SEEK. AFTER RETRY IS EXHAUSTED, WE WILL NOT TRY FOR THAT CYLINDER BUT CONTINUE WITH A NEW CYLINDER.

LIMITS 0 - 65,535

"DATA DMP ON DCK ERR (L) Y ?"

GIVES THE ABILITY TO SEE THE 1 SECTOR BUFFER THAT HAD A DATA CRC ERROR. THE RESULTS OF THE PRINTOUT ARE ONE OF TWO POSSIBILITIES.

1. ONLY THOSE WORDS OF THE SECTOR THAT WERE BAD ARE PRINTED WITH WHAT WAS EXPECTED.
2. IF ONE OF THE 1ST TWO WORDS IS BAD (USED TO KEY) THE ENTIRE BUFFER IS DUMPED.

LIMITS Y OR N

"# OF ERR DUMPED (D) 128 ?"

THIS IS THE NUMBER OF MISCOMPARES THAT WILL BE PRINTED.

LIMITS 0 - 128

"TIME BETW REPORTS (MIN) (D) 240 ?"

THIS IS THE INTERVAL BETWEEN AUTOMATIC STATISTICAL REPORTS ON ALL DRIVES IF A CLOCK IS PRESENT AND WAS ANSWERED SO IN THE INITIAL DIALOGUE.

LIMITS 1 - 65,535

"DROP DR ON ERR LMTS REACHED (L) Y ?"

GIVES THE ABILITY TO AUTOMATICALLY STOP TESTING ON A DRIVE ONCE ONE OF THE ERROR LIMITS HAVE BEEN EXCEEDED (SEEK, DRIVE, HARD, SOFT). IF THE ANSWER IS YES THEN THE FOLLOWING FOUR QUESTIONS WILL BE ASKED, IF NO THEN THE NEXT QUESTION WILL BE 2.3.13.11.

LIMITS Y OR N

'HRD ERR LMT (D) 3 ?'

THIS IS THE LIMIT OF HARD ERRORS THAT A DRIVE WILL BE DROPPED ON.
A HARD ERROR IS ONE ON WHICH THE RETRY HAS BEEN EXHAUSTED.

LIMITS 1 - 65,535

'SFT ERR LMT (D) 10 ?'

THIS IS THE LIMIT OF SOFT ERRORS THAT A DRIVE WILL BE DROPPED ON.
A SOFT ERROR IS AN ERROR ON AN OPERATION THAT WAS SUCCESSFUL WITHIN
THE RETRY LIMIT.

LIMITS 1 - 65,535

'DATA MISCOMPARE LIMIT (D) 10 ?'

THIS IS THE LIMIT OF IN CORE MISCOMPARES THAT THE DRIVE WILL BE
DROPPED ON.

LIMITS 1 - 65,535

'SK ERR LMT (D) 3 ?'

THIS IS THE LIMIT OF MIS-SEEK AND TRACKING ERRORS THAT A DRIVE WILL
BE DROPPED.

LIMITS 1 - 65,535

'DR ERR LMT (D) 3 ?'

THIS IS THE LIMIT OF DRIVE ERRORS THAT A DRIVE WILL BE DROPPED ON.

LIMITS 1 - 65,535

'DROP DR ON OPER LMTS REACHED (L) N ?'

GIVES THE ABILITY TO STOP TESTING ON A DRIVE THAT HAS EXCEEDED
CERTAIN OPERATION LIMITS (SEEK, BITS TRANSFERRED). THE DRIVE WILL
BE DROPPED ONLY WHEN BOTH HAVE BEEN EXCEEDED. IF THE ANSWER IS YES
THEN THE NEXT TWO QUESTIONS WILL BE ASKED.

LIMITS Y OR N

'DATA XFER LMT (*10(10)) (D) 25000 ?'

THIS IS THE LIMIT OF COMBINED BITS READ/WITTEN (*10(10)) ON WHICH THE DRIVE WILL BE DROPPED.

LIMITS 1 - 65,535

'SK LMT (*10(3)) (D) 10000 ?'

THIS IS THE LIMIT OF SEEK OPERATIONS (*10(3)) ON WHICH THE DRIVE WILL BE DROPPED.

LIMITS 1 - 65,535 (*10(3))

'DO YOU WANT TO CHANGE SEEK, R/W PARAMETERS (L) N ?'

THE NORMAL OPERATION IS TO SEEK AND TRANSFER ON THE ENTIRE CARTRIDGE, CYLINDERS 0 - 255. (RL01) OR 511. (RL02), SECTORS 0 - 39 AND BOTH SURFACES. THE NORMAL TRANSFER IS RANDOM BETWEEN 3 AND 1280 WORDS.

THE NEXT 8 PARAMETERS WILL ALLOW THE USER TO CONFINE THE TESTING TO ANY CONTIGUOUS SECTION OF THE CARTRIDGE AND CONTROL THE SIZE OF THE TRANSFERS.

A YES ANSWER WILL ASK THE NEXT 13 QUESTIONS.

'STIPULATE R/W XFER SIZE (L) N ?'

THE PROGRAM WILL NORMALLY MAXIMIZE THE TRANSFER SIZE BY USING ALL OF MEMORY (<28K) AVAILABLE. THIS QUESTION IF ANSWERED YES WILL RESTRICT THE BUFFER TO THOSE VALUES GIVEN IN NEXT TWO QUESTIONS. QUESTION IS 2.3.13.19.

LIMITS Y OR N

'MAX XFER (D) 2560 ?'

REPRESENTS THE MAXIMUM AMOUNT OF WORDS TO READ OR WRITE

LIMITS 3 - 5120

'MIN XFER (D) 3 ?'

REPRESENTS THE MINIMUM AMOUNT OF WORDS TO READ OR WRITE

LIMITS 3 - 5120

'RD ONLY (L) N ?'

GIVES THE ABILITY TO INHIBIT WRITING THE PACK WHILE TESTING, THE INITIAL WRITE OF THE PACK FROM THE START COMMAND WILL STILL OCCUR.

LIMITS Y OR N

'RAN PAT (L) Y ?'

NORMAL OPERATION SHOULD BE YES, BUT THIS PARAMETER WILL ALLOW THE WRITING OF ONLY ONE PATTERN OF EIGHT NORMAL PATTERNS. THE PATTERNS IN NEXT QUESTION.

LIMITS Y OR N

'WHICH ONE (0) 4 ?'

IT IS NOW POSSIBLE TO CONTAIN THE EXERCISER IN WRITING ONLY ONE OF THE FOLLOWING EIGHT PATTERNS:

- 0 - ALL 0'S
- 1 - 177777,177777,177777,52525,52525,52525
177777,177777,52525,52525,177777,52525
177252,177252,172765,172765
- 2 - 0,0,0,177777,177777,177777
0,0,177777,177777,0,177777,0,177777
0,177777
- 3 - 25252,52525,52525,125252,125252,125252
52525,52525,125252,125252,52525,125252
52525,125252,52525,125252
- 4 - WORST CASE DATA
155555,133333,66666,155555,133333,66666
155555,133333,66666,155555,133333,66666
155555,133333,66666,155555
- 5 - 121105,150442,64221,132110,55044,26422
13211,105504,42642,21321,110550,44264
22132,11055,104426,42213
- 6 - ALL 1'S
- 7 - 45513,122645,151322,64551,132264,55132
26455,113226,45513,122645,151322,64551
132264,55132,26455,113226

LIMITS 0 - 7

'WORDS PER SECTOR COMPARED ON READ (D) 16 ?'

NORMAL TRANSFERS ARE RANDOM BETWEEN 3 AND 1280 WORDS, THIS PARAMETER WILL ALLOW YOU TO SPECIFY HOW MANY WORDS SHOULD BE COMPARED PER SECTOR IN CORE AFTER EACH READ. IF THE VALUE SPECIFIED IS GREATER THAN THAT READ IN ONLY THE NUMBER READ IN ARE COMPARED. THE FEWER WORDS COMPARED IN CORE ON EACH READ THE FASTER THROUGHPUT THE EXERCISER WILL HAVE.

LIMITS 0 - 128

'# OF DATA ERR RPT'D PER BUF (D) 3 ?'

THIS PARAMETER WILL LIMIT THE NUMBER OF IN CORE MISCOMPARES PRINTED. THE PROGRAM WILL CONTINUE TO COMPARE AS MANY WORDS AS SPECIFIED BUT WILL INHIBIT THE PRINTOUT ONCE THIS LIMIT IS REACHED. AFTER ALL WORDS ARE CHECKED A SUMMARY WILL BE PRINTED:

X WORDS BAD OUT OF 128 WORDS READ

LIMITS 0 - 126

'MAX HD (D) 1 ?'

REPRESENTS MAXIMUM HEAD TO USE IN SEEK OPERATIONS.

LIMITS 0 - 1

'MIN HD (D) 0 ?'

REPRESENTS MINIMUM HEAD TO USE IN SEEK OPERATIONS

LIMITS 0 - 1

'CHANGE VALUES OF MXCYL & MINCYL (L) Y ?'

IF NO THEN THE NEXT TWO QUESTIONS WILL BE SKIPPED

'MAX CYL (D) 511 ?'

MAXIMUM INNER CYLINDER TO BE USED IN SEEK OPERATIONS.

LIMITS 0 - 255. (RL01) OR 511. (RL02)

'MIN CYL (D) 0 ?'

MINIMUM OUTER CYLINDER TO BE USED IN SEEK OPERATIONS.

LIMITS 0 - 255. (RL01) OR 511. (RL02)

'MAX SEC (D) 0 ?'

MAXIMUM SECTOR TO START TRANSFER ON

LIMITS 0 - 39

'MIN SEC (D) 0 ?'

MINIMUM SECTOR TO START TRANSFER ON

LIMITS 0 - 39

AFTER ANSWERING THE LAST SOFTWARE PARAMETER THE PROGRAM WILL START THE TESTING.

3.0 . ERROR INFORMATION

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

3.1 ERROR REPORTING

THE FOLLOWING ARE ERROR HEADINGS THAT MAY BE ENCOUNTERED WHILE RUNNING. A BRIEF DESCRIPTION IS GIVEN.

'SFT ERROR'

AN ERROR WAS DISCOVERED, BUT ON RETRY THE ERROR DID NOT PERSIST. INFO GIVEN IS ERROR, RLCS, RLBA, AND RLDA

'EXH'D RETRY ON SEEK'

THE NUMBER OF RETRIES GIVEN HAVE FAILED TO POSITION DRIVE TO THE GIVEN TRACK. INFO GIVEN IS RLCS, RLDA, RLBA, LAST POSITION, PRESENT POSITION, AND DRIVE STATUS

'VOL CHK WILL NOT RESET' .

A DRIVE RESET WILL NOT RESET VOLUME CHECK BIT

'DID NOT REC'R FROM PWR UP'

DRIVE DID NOT COME BACK UP AFTER A POWER FAILURE

'DATA DMP - DATA CHECK/GARBBLED DATA'

THE PROGRAM ENCOUNTERED A DATA CHECK ERROR BUT WAS UNABLE TO MAKE SENSE OUT OF THE FIRST TWO WORDS, WHICH ARE USED TO KEY OFF OF. THEREFORE ALL WORDS OF SECTOR ARE DUMPED.(REFER TO SECTION 2.3.13.21)

'LIMITS EXCEEDED! HIGH - X LOW - Y'

ANSWER GIVEN IS NOT WITHIN LIMITS FOR QUESTION.

'NO DEFAULT PROVIDED!'

CANNOT <CR> TO THIS QUESTION

'ILLEGAL COMMAND'

START, RESTART, CONTINUE, PRINT TYPED IN WRONG FORM

'ILL ENTRY IN P-TABLE'

ANSWERS IN HARDWARE SECTION THAT ARE NOT LEGAL (I.E., MORE THAN TWO CONTROLLERS)

'CAN'T READ FACTORY BAD SECTOR FILE'

PROGRAM IS UNABLE TO READ ANY OF THE FACTORY FILES

'CAN'T READ FIELD BAD SECTOR FILE'

PROGRAM IS UNABLE TO READ ANY OF THE FIELD FILES

'MORE THAN 16 BAD SECTORS'

PROGRAM LIMITS EXERCISING CARTRIDGES TO THOSE WITH LESS THAN 16 BAD SECTORS.

'NO DRIVES ENTERED''

EITHER NO DRIVES WERE ENTERED OR ALL DRIVES THAT WERE ENTERED WERE DROPPED FOR ONE REASON OR ANOTHER. THE PROGRAM WILL LOOP AFTER PRINTING THE ERROR, WAITING FOR ^C. A START COMMAND IS NOW NECESSARY.

'DRV NOT RDY W/O DRV ERR''

ON COMPLETION OF A COMMAND, DRIVE READY IS CHECKED FOR A POSSIBLE TRACKING DRIFT PROBLEM. IF THERE IS NO DRIVE READY A GET STATUS IS DONE TO VERIFY THAT THE DRIVE IS NOT IN PROCESS OF SEEKING. IF IT IS SEEKING THE CONDITION IS LEGAL. THIS TIMEOUT IMPLIES THERE WERE NO DRIVE ERRORS WHICH MAY HAVE CAUSED DRIVE READY TO GO AWAY.

'TRCK ERR''

THIS ERROR MEANS THAT THE DRIVE IS NO LONGER ON THE TRACK SELECTED. ANY SUBSEQUENT READ HEADER, READ OR WRITE COMMANDS WILL PRINT THIS ERROR IF THE TRACK IS NOT CORRECT. THIS ERROR WILL PRINT THE POSITION BEFORE THE LAST SEEK, THE PRESENT POSITION AND THE EXPECTED POSITION.

'MIS-SK ERR''

AFTER A SEEK WAS DONE, READ HEADER IS DONE TO VERIFY THE SEEK. THE ERROR PRINTOUT WILL INCLUDE THE LAST POSITION BEFORE THE SEEK, THE PRESENT POSITION AND THE EXPECTED POSITION.

'DRV STAT ERR''

THE RESULT OF A GET STATUS OPERATION IS INCORRECT. EITHER A ERROR BIT IS SET OR THE STATE IS WRONG

'HRD ERR''

THE NUMBER OF RETRIES WERE EXHAUSTED WITH OUT SUCCESS THE ERROR PRINTOUT CONSISTS OF ALL REGISTERS BEFORE COMMAND AND AT TIME OF ERROR.

'INIT WR OF SEC BAD''

WHILE WRITING THE PACK INITIALLY, THE SECTOR INDICATED COULD NOT BE WRITTEN AND VERIFIED. THIS SECTOR WAS NOT IN THE BAD SECTOR FILE. ONE OF THE FOLLOWING STEPS SHOULD BE ISSUED: A) STOP THE EXERCISER AND CHANGE CARTRIDGE, B) STOP THE EXERCISER AND VERIFY THE CARTRIDGE (USE THE BAD SECTOR FILE TOOL - CZRLMA) OR C) IGNORE ALL ERRORS FROM THAT SECTOR.

3.2 ERROR HALTS

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

4.0 PERFORMANCE AND PROGRESS REPORTS4.1 PERFORMANCE REPORTS

PERFORMANCE REPORTS ARE GIVEN AUTOMATICALLY (PER SOFTWARE PARAMETERS), WHEN A DRIVE IS DROPPED, OR AT OPERATOR REQUEST (PRINT) THE FORMAT IS:

*** RL01 PERFORMANCE REPORT ***

TIME: HH:MM:SS RLCS: XXXXXX DRIVE: Y DRIVE TYPE = RLOX
*** RUNNING OR DROPPED DH:DM
PACK SERIAL #: DDDDDDDDD
TOTAL SEEKS: IIIII
WORDS READ: JJJJJJJJJ
WORDS WRITTEN: KKKKKKKKK

ERRORS
DRV-ER: N SEEK: N TRACK: N DATA: N
HARD: N SOFT: N
DCK: N HCRC: N NXM: N HNF: N
DLT: N OPI: N

WHERE:

HH IS HOURS SINCE START/RESTART
MM IS MINUTES SINCE START/RESTART
SS IS SECONDS SINCE START/RESTART
XXXXXX IS ADDRESS OF CONTROLLER
Y IS DRIVE NUMBER
DH IS HOUR AT WHICH DRIVE WAS DROPPED
DM IS MINUTE AT WHICH DRIVE WAS DROPPED
DDDDDDDDDD - IS 10 DIGIT OCTAL SERIAL NUMBER OF PACK
IIII IS TOTAL NUMBER OF SEEKS SINCE START TIME 0:00:00
JJJJ IS TOTAL NUMBER OF WORDS READ SINCE START TIME 0:00:00
KKKK IS TOTAL NUMBER OF WORDS WRITTEN SINCE START TIME 0:00:00
N IS NUMBER OF THAT TYPE ERROR SINCE START TIME 0:00:00

4.2 PROGRESS REPORTS

THE ONLY PROGRESS REPORT IS THE AUTOMATIC PERFORMANCE REPORT.

5.0 DEVICE INFORMATION TABLES

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

RLCS - CONTROL AND STATUS REGISTER (XXXXX0)

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

BIT 0 - DRIVE READY

RLBA - BUS ADDRESS REGISTER (XXXXX2)

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

RLDA - DISK ADDRESS REGISTER (XXXXX4)

FOR READ/WRITE FUNCTIONS

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

FOR SEEK FUNCTION

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

FOR GET STATUS FUNCTION

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

RLMP - MULTIPURPOSE REGISTER

FOR READ/WRITE FUNCTION

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

FOR READ HEADER FUNCTION

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

FOR GET STATUS FUNCTION

HAS DRIVE STATUS

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

- 3 - LOAD HEADS
- 4 - SEEK - TRACK COUNTING
- 5 - SEEK - LINEAR MODE
- 6 - UNLOAD HEADS
- 7 - SPIN DOWN

6.0 TEST SUMMARIES

PROGRAM DESCRIPTION

THE PROGRAM WILL TRY TO SIMULATE A USER ENVIRONMENT WITH RANDOM SELECTION OF DRIVES PERFORMING RANDOM OPERATIONS OF GET STATUS, SEEK, READ AND WRITE.

INITIALLY THE BAD SECTOR FILE IS RECOVERED FROM EACH DRIVE AND STORED, THEN EACH PACK IS ENTIRELY WRITTEN RANDOMLY WITH ONE OF EIGHT PREDETERMINED PATTERNS.

THE MAIN LOOP IS A CONTINUOUS LOOP OF THE FOLLOWING STEPS

1. RANDOMLY SELECT A DRIVE
2. CHECK CONTROLLER OF SELECTED DRIVE IS NOT BUSY;
3. THEN STEP 3; ELSE STEP 1
4. RANDOMLY SELECT FUNCTION FOR DRIVE
IF SEEK/WRITE/WRITE CHECK - THEN GO TO STEP 5
IF SEEK/READ - THEN GO TO STEP 11
IF SEEK/READ/READ - THEN GO TO STEP 15
IF SEEK/READ HDRS/READ W/NO HDR COMPARE/GET STATUS - THEN GO TO STEP 21
5. GET A RANDOM CYLINDER ADDRESS (NOT THE BAD SECTOR FILE)
6. SEEK TO THE SELECTED CYLINDER AND WAIT TILL COMPLETED
7. GET A RANDOM WORD COUNT FOR THE WRITE FUNCTION - MAKE SURE THAT IT WON'T OVERFLOW THE TRACK
8. GET A RANDOM DATA PATTERN TO WRITE ON THE TRACK POINTED TO
9. ISSUE THE WRITE FUNCTION AND WAIT TILL COMPLETED
10. ISSUE A WRITE CHECK FUNCTION ON THE SAME DISK ADDRESS TO COMPARE THE DATA JUST WRITTEN BY THE WRITE FUNCTION THEN GO TO STEP #1
11. GET A RANDOM CYLINDER # FOR THE SEEK
12. SEEK TO THE SELECTED CYLINDER AND WAIT TILL COMPLETED
13. GET A RANDOM WORD COUNT FOR THE READ FUNCTION - MAKE SURE IT WILL NOT OVERFLOW THE SELECTED TRACK

14. ISSUE THE READ FUNCTION AND WAIT TILL COMPLETED ...THE INTERRUPT SERVICE WILL INITIATE A DATA COMPARE ON THE DATA READ (IF THE FUNCTION IS ENABLED) THEN GO TO STEP #1
15. GET A RANDOM CYLINDER FOR THE SEEK
16. SEEK AND WAIT TILL COMPLETED
17. GET A RANDOM WORD COUNT FOR THE READ COMMAND
18. ISSUE A READ COMMAND AND WAIT TILL COMPLETED
19. GET ANOTHER RANDOM² WORD COUNT FOR SAME TRACK SELECTED
20. ISSUE A SECOND READ FUNCTION AND WAIT TILL COMPLETED. THEN GOTO STEP #1
21. ISSUE A SEEK TO A RANDOM CYLINDER AND WAIT TILL COMPLETED
22. ISSUE A READ HEADER FUNCTION AND WAIT TILL COMPLETED
23. ISSUE A READ DATA WITH NO HEADER COMPARE (1 SECTOR TO BE READ) AND WAIT TILL COMPLETED
24. ISSUE A GET STATUS FUNCTION THEN GO TO STEP #1

THE PROGRAM WILL STAY WITHIN THAT MAIN LOOP UNTIL INTERRUPTED OUT BY A FUNCTION FINISHING AT WHICH TIME THE INTERRUPT SERVICE ROUTINE WILL START EXECUTION.

1. READ ALL REGISTERS OF CONTROLLER THAT INTERRUPTED AND SAVE IMAGES
2. IF NO ERROR SET; THEN STEP 3; ELSE STEP 14
3. CHECK FUNCTION WHICH CAUSED INTERRUPT
IF WRITE CHECK; THEN STEP 3A
IF GET STATUS; THEN STEP 5
IF SEEK; THEN STEP 4A.
IF READ HEADER; THEN STEP 7
IF READ; THEN STEP 9
IF WRITE; THEN STEP 3B
- 3A. CLEAR WRITE CHECK NEEDED FLAG, THEN STEP 4
- 3B. SET WRITE CHECK NEEDED FLAG IF REQUESTED THEN STEP 4
4. IF RETRY > 0 THEN REPORT SOFT ERROR, ELSE STEP 4A
- 4A. EXIT TO MAIN PROGRAM
5. CHECK STATUS FOR.
NO ERRORS
COVER CLOSED
BRUSHES HOME
HEADS OUT

, SEEK LINEAR/TRACKING

IF THEN STEP 4; ELSE STEP 6

6. REPORT STATUS ERROR; GO TO STEP 4A
7. SET VERIFICATION DONE FLAG COMPARE PRESENT POSITION WITH
HEADER WORD IF THEN STEP 4A; ELSE STEP 8
8. REPORT MIS-SEEK, SET NEW POSITION; GO TO STEP 4
9. IF DATA TO BE COMPARED; THEN STEP 10; ELSE STEP 4
10. CHECK VALIDITY OF FIRST TWO WORDS; IF THEN STEP 12; ELSE
STEP 11.
11. REPORT GARBLED DATA; GO TO STEP 4
12. CHECK WORDS READ IN IF OKAY THEN STEP 4A ELSE STEP 13
13. REPORT DATA ERROR, GO TO STEP 4
14. IF DRIVE ERROR; THEN STEP 33; ELSE STEP 15
15. IF NXM; THEN STEP 18; ELSE STEP 16
16. IF OPI; THEN STEP 18; ELSE STEP 17
17. IF DLT; THEN STEP 18; ELSE STEP 20
18. IF RETRY < LIMIT THEN STEP 4A, ELSE STEP 19
19. REPORT HARD ERROR; CLEAR FLAGS; GO TO STEP 4A
20. IF HCRC; THEN STEP 24; ELSE STEP 21
21. IF DCRC, THEN STEP 29; ELSE STEP 22
22. IF HNF, THEN STEP 30; ELSE STEP 23
23. YOU SHOULD NEVER GET HERE
24. IF DOING READ/WRITE THEN STEP 25 IF DOING READ HEADER THEN
STEP 26
25. CHECK IF DA IS BAD SECTOR THEN STEP 4A; ELSE STEP 18.
26. READ 40 HEADERS, IF ALL GOOD THEN STEP 27; ELSE STEP 28
27. REPORT SOFT HEADER CRC; GO TO 4A
28. FIGURE OUT BAD HEADER IF IN FILE THEN STEP 4A; ELSE STEP
18
29. CHECK IF DA-1 IS IN FILE IF THEN STEP 4A; ELSE STEP 18

30. READ HEADER. IF ON CORRECT TRACK THEN STEP 31; ELSE STEP 32

31. CHECK IF DA IS IN FILE IF THEN STEP 4A, ELSE STEP 18

32. REPORT TRACKING; FIX POSITION, GO TO STEP 4

33. ACT UPON: VC
SKTO
SPE
WGE
WDE
CHE

34. GO TO STEP 4

@

29	BIT AND OFFSET DEFINITIONS
176	MACRO DEFINITIONS
222	GLOBAL DATA AND CONSTANTS
330	GLOBAL MESSAGES
449	ERROR MESSAGES
635	DEFAULT HARDWARE P-TABLE PARAMETERS
652	DEFAULT SOFTWARE P-TABLE PARAMETERS
701	STATISTICAL CODE
725	LOAD PROTECTION TABLE
733	INITIALIZATION CODE
964	AUTO DROP SECTION
1097	GLOBAL SUBROUTINES
1161	REPORT ROUTINE
1189	PROGRAM MAIN LOOP
1461	ROUTINES TO SETUP AND ISSUE GET STATUS & SEEK
1571	ROUTINE TO LOAD READ HEADER AND ISSUE IT
1576	ROUTINE TO LOAD WRITE DATA COMMAND
1597	ROUTINE TO LOAD READ DATA COMMAND
1614	SETUP CONTROLLER AND DRIVE INFO FOR INTERRUPT PROCESSING
1630	ROUTINE TO LOAD FUNCTION
1655	INTERRUPT SERVICE ROUTINES
1787	CONTROLLER ERROR CHECK ROUTINE
2014	COMMAND SERVICE ROUTINES
2046	SEEK INTERRUPT SERVICE
2057	READ INTERRUPT SERVICE
2076	READ HEADER INTERRUPT SERVICE
2110	GET STATUS INTERRUPT SERVICE
2135	WRITE INTERRUPT SERVICE
2160	EXIT FOR INTERRUPT SERVICE
2191	DRIVE ERROR INTERRUPT SERVICE
2292	BUFFER GENERATION ROUTINE FOR THE 'WRITE' FUNCTION
2349	RETRY LIMIT ROUTINE
2360	LIST OF FUNCTION ROUTINES
2373	BAD SECTOR FILE ROUTINE
2529	ROUTINE TO DROP DRIVE
2574	ROUTINE TO CHECK DATA
2660	ROUTINE TO WAIT FOR CONTROLLER READY
2682	GET STATUS/DRIVE RESET ROUTINE
2730	ROUTINE TO WRITE PACKS INITIALLY
2927	HEADS HOME ROUTINE
2949	RANDOM WC AND DA ROUTINE
3026	ROUTINE TO DUMP BUFFER ON DCK
3158	ROUTINE TO CHECK FOR BAD SECTOR
3376	DRIVE INFORMATION BUFFERS

```
1          .TITLE  CZRLKBO RL01/02 PERF EXER
2          .ENABLE AMA
3          .ENABLE ABS
4
5          002000          .=2000
6
7
8          .MCALL  SVC
9
10         002000          SVC
11         000000          SVCINS=0
12         000000          SVCTAG=0
13
14
15         002000          POINTER BGNRPT,BGNSW,BGNSFT,BGNAU,BGNDU
16
17
18         002000          BGNMOD  MDHEDR
19         002000          HEADER  CZRLK,B,0,0,1
(4) 002000          103          .ASCII /C/
(4) 002001          132          .ASCII /Z/
(4) 002002          122          .ASCII /R/
(4) 002003          114          .ASCII /L/
(4) 002004          113          .ASCII /K/
(6) 002005          000          .BYTE 0
(6) 002006          000          .BYTE 0
(5) 002007          000          .BYTE 0
(4) 002010          102          .ASCII /B/
(4) 002011          060          .ASCII /O/
(4) 002012          000000       .WORD 0
(4) 002014          000000       .WORD 0
(4) 002016          031716       .WORD L$HARD
(4) 002020          032072       .WORD L$SOFT
(4) 002022          010642       .WORD L$HW
(4) 002024          010660       .WORD L$SW
(4) 002026          033520       .WORD L$LAST
(4) 002030          000000       .WORD 0
(4) 002032          000000       .WORD 0
(4) 002034          000001       .WORD 1
(4) 002036          000000       .WORD 0
(4) 002040          010760       .WORD L$DISPATCH
(4) 002042          000000       .WORD 0
(4) 002044          000000       .WORD 0
(4) 002046          000000       .WORD 0
(4) 002050          003          .BYTE C$REVISION
(3) 002051          003          .BYTE C$EDIT
(4) 002052          000000       .WORD 0
(5) 002054          000000       .WORD 0
(4) 002056          000000       .WORD 0
(4) 002060          002230       .WORD L$DVTYP
(4) 002062          010762       .WORD L$RPT
(4) 002064          000000       .WORD 0
(4) 002066          000000       .WORD 0
(4) 002070          013346       .WORD L$AU
(4) 002072          013432       .WORD L$DU
(4) 002074          000000       .WORD 0
```


(4)	002076	002122			.WORD	L\$DESC
(4)	002100	104035			EMT	E\$LOAD
(4)	002102	000000			.WORD	0
(4)	002104	011046			.WORD	L\$INIT
(4)	002106	013150			.WORD	L\$CLEAN
(4)	002110	012676			.WORD	L\$AUTO
(4)	002112	011040			.WORD	L\$PROT
(4)	002114	000000			.WORD	0
(4)	002116	000000			.WORD	0
(4)	002120	000000			.WORD	0

20
21 002122 ENDMOD
22
23

24 002122 DESCRIPT <CZRLK PERFORMS RANDOM OPERATIONS OF GET STATUS, SEEK, READ, AND WRITE>
(3) 002122 055103 046122 020113 .ASCIZ /CZRLK PERFORMS RANDOM OPERATIONS OF GET STATUS, SEEK, READ, AND WRITE/
(3) 002130 042520 043122 051117
(3) 002136 051515 051040 047101
(3) 002144 047504 020115 050117
(3) 002152 051105 052101 047511
(3) 002160 051516 047440 020106
(3) 002166 042507 020124 052123
(3) 002174 052101 051525 020054
(3) 002202 042523 045505 020054
(3) 002210 042522 042101 020054
(3) 002216 047101 020104 051127
(3) 002224 052111 000105

(2) .EVEN
25

26
27 002230 DEVTYP <RL01,RL02>
(3) 002230 046122 030460 051054 .ASCIZ /RL01,RL02/
(3) 002236 030114 000062

(2) .EVEN
28
29 .SBTTL BIT AND OFFSET DEFINITIONS
30
31 ;DEFINITIONS
32

33 002242 BGNMOD GLBEQAT
34
35 002242 EQUALS
36
(1) ; BIT DIFINITIONS
(1) :
(1) :
(1) BIT15== 100000
(1) BIT14== 40000
(1) BIT13== 20000
(1) BIT12== 10000
(1) BIT11== 4000
(1) BIT10== 2000
(1) BIT09== 1000
(1) BIT08== 400
(1) BIT07== 200
(1) BIT06== 100

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


38	000000	CS=0	:CONTROL AND STATUS OFFSET
39	000002	BA=2	:BUSADDRESS OFFSET
40	000004	DA=4	:DISK ADDRESS OFFSET
41	000006	MP=6	:MULTI PURPOSE OFFSET
42		:CONSTANT OFFSETS FOR INDIVIDUAL DRIVE BUFFERS	
43		:THE ONLY POSITION THAT IS CRITICAL IS THAT OF	
44		: "PRPOS" IT M U S T (MUST) BE THE LAST ENTRY OF THE BUFFER	
45			
46	000000	SKCNT=0	:SEEK OPERATION COUNT
47	000002	RXFR1=2	:READ OPERATION COUNT (BITS) LOW ORDER
48	000004	RXFR2=4	:HIGH ORDER
49	000006	WXFR1=6	:WRITE OPERATION COUNT (BITS) LOW ORDER
50	000010	WXFR2=10	:HIGH ORDER
51	000012	ERRCNT=12	:ERROR COUNT - HARD
52	000014	SFTCNT=14	:ERROR COUNT - SOFT
53	000016	SKECNT=16	:SEEK ERROR COUNT
54	000020	DERCNT=20	:DRIVE ERROR COUNT
55	000022	DCRCER=22	:DATA CRC ERROR COUNT
56	000024	HRCRCER=24	:HEADER CRC ERROR COUNT
57	000026	DLTCNT=26	:DATA LATE ERROR COUNT
58	000030	OPICNT=30	:OPERATION INCOMPLETE ERROR COUNT
59	000032	HNFERR=32	:HEADER NOT FOUND ERROR COUNT
60	000034	NXMCNT=34	:NON EXISTENT MEMORY ERROR COUNT
61	000036	RETRY=36	:PRESENT RETRY NUMBER
62	000040	BDA=40	:DISK ADDRESS CONTENTS
63	000042	BMP=42	:PRESENT MULTIPURPOSE CONTENTS
64	000044	FUNC=44	:LAST FUNCTION LOADED
65	000046	BCSADR=46	:CSR IMAGE OF LAST COMMAND
66	000050	LSTHDR=50	:LAST POSITION ON DISK
67	000052	RTYPE=52	:ERROR ON WHICH RECOVERY IS BEING TRIED
68	000054	SKCNT1=54	:LOW SEEK COUNT
69	000056	PRFLGS=56	:INTERNAL FLAGS
70	000060	RXFR3=60	:THIRD ORDER READ COUNT
71	000062	WXFR3=62	:THIRD ORDER WRITE COUNT
72	000064	LSTDA=64	:DISK ADDRESS AT SOFT ERROR
73	000066	DIFWD=66	:LAST DIFFERENCE WORD OF SEEK
74	000070	DPHOUR=70	:HOUR OF DRIVE DROPPED
75	000071	DPMIN=71	:MINUTE OF DRIVE DROPPED
76	000072	TRERR=72	:TRACKING ERRORS COUNT
77	000074	DATCER=74	:DATA CMP ERRORS
78	000076	DOWCK=76	:PERFORM WRITE CHECK
79	000100	SERNM1=100	:SERIAL NUMBER OF CARTRIDGE
80	000102	SERNM2=102	:SERIAL NUMBER OF CARTRIDGE
81	000104	DCS=104	:CSR ADDRESS
82	000106	DRSEL=106	:DRIVE SELECT BITS(8,9,10)
83	000110	BBA=110	:PRESENT BUS ADDRESS CONTENTS
84	000112	BSECPT=112	:POINTER TO BAD SECTOR FILE
85	000114	RSEEK=114	:SEEK IN PROCESS OF RECOVERY
86	000116	SOFTCS=116	:CSR OF SOFT ERROR
87	000120	TDR=120	
88	000122	WRIPG=122	:WRITE IN PROGRESS FLAG
89	000124	PRPOS=124	:PRESENT POSITION ON DISK
90			
91	000001	SKDON=BIT0	
92	000001	DRDY=BIT0	:DRIVE READY
93	000100	INTEN=BIT6	:INTERRUPT ENABLE

94	100000	ERR=BIT15	:COMPOSITE ERROR
95	040000	DERR=BIT14	:DRIVE ERROR
96	100000	WDE=BIT15	:WRITE DATA ERROR
97	040000	HCE=BIT14	:HEAD CURRENT ERROR
98	020000	WL=BIT13	:WRITE LOCK
99	010000	SKTO=BIT12	:SEEK TIMEOUT ERROR
100	004000	SPE=BIT11	:SPINDLE TIMEOUT/UNDER/OVER SPEED
101	002000	WGE=BIT10	:WRITE GATE ERROR
102	001000	VC=BIT9	:VOLUME CHECK
103	000400	DSE=BIT8	:DRIVE SELECT ERROR
104	020000	NXM=BIT13	:NON-EXISTENT MEMORY ERROR
105	010000	DLT=BIT12	:DATA LATE
106	004000	DCRC=BIT11	:DATA CRC ERROR
107	004000	HCRC=BIT11	:HEADER CRC ERROR
108	010000	HNF=BIT12	:HEADER NOT FOUND ERROR
109	002000	OPI=BIT10	:OPERATION INCOMPLETE ERROR
110	000200	CRDY=BIT7	:CONTROLLER READY
111	000040	BA17=BIT5	:EXTENDED BUS ADDRESS BIT 17
112	000020	BA16=BIT4	:EXTENDED BUS ADDRESS BIT 16
113	000002	WRCHK=BIT1	:WRITE CHECK FUNCTION CODE
114	000004	GSTAT=BIT2	:GET DRIVE STATUS FUNCTION CODE
115	000006	SEEK=BIT1!BIT2	:SEEK FUNCTION CODE
116	000010	RDHDR=BIT3	:READ HEADER FUNCTION CODE
117	000012	WRITE=BIT3!BIT1	:WRITE FUNCTION CODE
118	000014	READ=BIT3!BIT2	:READ FUNCTION CODE
119	000013	DRST=BIT3!BIT1!BIT0	:DRIVE RESET COMMAND CODE FOR DRIVE COMMAND WORD
120	000003	GSR=BIT1!BIT0	:GET STATUS COMMAND CODE FOR DRIVE COMMAND WORD
121	000001	MK=BIT0	:MARKER BIT FOR DRIVE COMMAND WORD(SEEK,GET STATUS)
122	000004	SIGN=BIT2	:DIRECTION FOR SEEK(0=AWAY FROM SPINDLE)
123	000020	SKHS=BIT4	:HEAD SELECT FOR SEEK
124	000100	HEAD=BIT6	:HEAD SELECT FOR READ,WRITE,GET STATUS
125			
126		:OFFSET FOR HARDWARE P-TABLE	
127			
128	000000	CSR=0	
129	000002	VECT=2	
130	000004	PRIOR=4	
131	000006	TYPDR=6	
132	000010	DRBT=10	
133	000012	CNT=12	
134			
135		:OFFSET FOR SOFTWARE P-TABLE	
136			
137	000000	RLT=0	
138	000002	ELT=2	
139	000004	SET=4	
140	000006	DAT=6	
141	000010	SKT=10	
142	000012	TYT=12	
143	000014	RDT=14	
144	000016	DDT=16	
145	000020	CHFLG=20	
146	000022	MXB=22	
147	000024	MXH=24	
148	000026	MNH=26	
149	000030	MXC=30	

150	000032	MNC=32
151	000034	MXS=34
152	000036	MNS=36
153	000040	DCKFG=40
154	000042	DRFLG=42
155	000044	MNB=44
156	000046	SEL=46
157	000050	OPFLG=50
158	000052	DET=52
159	000054	ROF=54
160	000056	RAN=56
161	000060	PAT=60
162	000062	SRLT=62
163	000064	CLMT=64
164	000066	AUTO=66
165	000070	STIP=70
166	000072	WCK=72
167	000074	DCD=74
168	000076	ANS=76

170
171 002242
172
173

ENDMOD

```
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218
```

```
.SBTTL MACRO DEFINITIONS  
:DELAY EXECUTION OF PROGRAM A SPECIFIED NUMBER OF 100-MILLISECOND TIME COUNTS  
.MACRO WAITMS ARG,?WAIT  
MOV #ARG,DLYCNT ;INITIALIZE DELAY COUNTER  
ASL DLYCNT ;MULTIPLY ARGUMENT BY 2  
ASL DLYCNT ;MULTIPLY ARGUMENT BY 2 AGAIN  
WAIT: DELAY #250. ;IMPLEMENT 25-MS TIME DELAY  
DEC DLYCNT ;DECREMENT DELAY COUNT  
BNE WAIT ;BRANCH IF TIME DELAY NOT EXPIRED  
.ENDM  
:DELAY EXECUTION OF PROGRAM A SPECIFIED NUMBER OF 100-MICROSECOND TIME COUNTS  
.MACRO WAITUS ARG  
DELAY #ARG ;IMPLEMENT 100-US TIME DELAY, ARGUMENT SPECIFIES  
;/THE NUMBER OF 100-US TIME COUNTS  
.ENDM  
:ACTIVATE THE CLOCK TO INITIATE THE GENERATION OF CLOCK INTERRUPTS  
.MACRO CLKON  
JSR PC,CLKINI ;ACTIVATE CLOCK WITH 1-SEC INCREMENTS  
JSR PC,CLKST ;INITIALIZE CLOCK  
;START CLOCK  
.ENDM  
:DEACTIVATE THE CLOCK TO HALT THE GENERATION OF CLOCK INTERRUPTS  
.MACRO CLKOFF  
CLR CLKSON ;INDICATE "CLOCK OFF"  
CMP #1,CLKTYP ;P-CLOCK?  
BNE 11$ ;BRANCH TO CHECK FOR L-CLOCK  
CLR @#172540 ;CLEAR P-CLOCK  
11$: CMP #2,CLKTYP ;L-CLOCK?  
BNE 12$ ;BRANCH FOR NO CLOCK  
CLR @#177546 ;CLEAR L-CLOCK  
12$:  
.ENDM  
:REQUEST ELAPSED TIME IN SECONDS OCCURRING BETWEEN SUPERVISOR INITIATION  
:AND THE GENERATION OF THE REQUEST  
.MACRO REQTIM ARG  
MOV CLKACC,ARG  
.ENDM
```



```
220
221
222      .SBTTL  GLOBAL DATA AND CONSTANTS
223
224 002242      BGNMOD  GLBDAT
225
226 002242      000000      RECNT:  .WORD  0      :READ ERROR COUNT
227 002244      000000      RWCNT:  .WORD  0      :R/W ERROR COUNT
228 002246      000000      WHY:    .WORD  0      :REASON FOR DROPPING DRIVE
229 002250      000000      TSTDRV: .WORD  0      :COPY OF SELECTED DRIVE FOR TESTING
230 002252      000      DRUT:   .BYTE  0      :DRIVES UNDER TEST
231 002253      000      DRPRS:  .BYTE  0      :DRIVES PRESENT
232 002254      000000      T.DRIVE: .WORD  0      :TYPE OF DRIVE FROM P-TABLE
233 002256      000000      SYMSK:  .WORD  0      :MASK FOR 0-7 DRIVES
234 002260      176543      HINUM:  .WORD  176543 :PRIME FOR RANDOM
235 002262      123456      LONUM:  .WORD  123456 :NUMBER GENERATOR
236 002264      100177      CYLSK:  .WORD  100177 :MASK FOR CYLINDER ONLY
237 002266      100077      SECMSK: .WORD  100077 :MASK OUT SECTOR BITS
238 002270      000177      CMSK:   .WORD  000177
239 002272      000077      SMSK:   .WORD  000077
240 002274      000000      WRINIT: .WORD  0      :WRITE INIT FLAG
241 002276      000000      WRPOS:  .WORD  0      :WRITE UNIT FLAG
242 002300      000000      CYL:   .WORD  0      :CYLINDER #
243 002302      000000      SUR:   .WORD  0      :SURFACE #
244 002304      000000      SEC:   .WORD  0      :SECTOR #
245 002306      000000      REGEN: .WORD  0      :REGEN FLAG FOR BUFFERS
246 002310      000000      KILLDC: .WORD  0      :INHIBIT DATA COMP FLAG
247 002312      000000      CLKFRQ: .WORD  0      :CLOCK FREQUENCY FLAG, 1=60 HZ, 2=50 HZ
248 002314      000000      CLKTYP: .WORD  0      :CLOCK TYPE FLAG, 1=P-CLOCK, 2=L-CLOCK
249 002316      000000      CLKADR: .WORD  0      :POINTER TO ADDRESS OF SUPERVISOR CLOCK TABLE
250
251
252      ;THE FOLLOWING LOCATIONS ARE CLEARED AS A GROUP (DOWN TO 'STFLG')
253      ;THEREFORE DON'T INSERT ANY CONSTANTS
254
255 002320      174400      CNTLR1: .WORD  174400 :CSR OF CONTROLLER 1 (LUN 0-3)
256 002322      000000      CNTLR2: .WORD  0      :CSR OF CONTROLLER 2 (LUN 4-7)
257 002324      000000      LSTDR1: .WORD  0      :BUFFER POINTER OF DRIVE
258 002326      000000      LSTDR2: .WORD  0      :BUFFER POINTER OF DRIVE
259 002330      000000      BCSR:   .WORD  0      :CSR FROM P-TABLE
260 002332      000000      BVEC:   .WORD  0      :VECTOR " "
261 002334      000000      BPRIOR: .WORD  0      :PRIORITY " "
262 002336      000000      BDRSEL: .WORD  0      :DRIVE " "
263 002340      000000      HDRFND: .WORD  0      :FLAG TO INDICATE HDR IN BAD LIST
264 002342      000000      CHKSEC: .WORD  0      :SECTOR OF ERROR - USED BY BAD SECTOR LOCATION
265 002344      000000      DECNT:  .WORD  0      :DATA ERROR COUNT
266 002346      000000      TEMP0:  .WORD  0      :TEMP LOCATION
267 002350      000000      TEMP1:  .WORD  0      :TEMP LOCATION
268 002352      000000      TEMP2:  .WORD  0      :TEMP LOCATION
269 002354      000000      TEMP3:  .WORD  0      :" "
270 002356      000000      TEMP4:  .WORD  0      :" "
271 002360      000000      TEMP5:  .WORD  0      :" "
272 002362      000000      TEMP6:  .WORD  0      :" "
273 002364      000000      TEMP7:  .WORD  0      :" "
274 002366      000000      TEMP8:  .WORD  0      :" "
275 002370      000000      TEMP9:  .WORD  0      :" "
```

```
276 002372 000160      VECT1: .WORD 160      :VECTOR OF FIRST CONTROLLER
277 002374 000000      VECT2: .WORD 0        :VECTOR " 2ND
278 002376 000000      PRIOR1: .WORD 0
279 002400 000000      PRIOR2: .WORD 0
280 002402 000000      GDDAT: .WORD 0
281 002404 000000      RNTEMP: .WORD 0
282 002406 000000      INTERVAL: .WORD 0    :KEEPS TRACK OF TIME BETWEEN STATISTICAL REPORTS
283                                     :/(MINUTES RUNNING TIME)
284 002410 000000      TICK: .WORD 0        :STORAGE FOR TICK COUNT
285 002412 000000      SECOND: .WORD 0      :SECONDS OF SYSTEM CLOCK
286 002414 000000      MINUTE: .WORD 0     :MINUTES OF SYSTEM CLOCK
287 002416 000000      HOUR: .WORD 0       :HOURS OF SYSTEM CLOCK
288 002420 000000      E.CS: .WORD 0       :IMAGES OF REGISTERS
289 002422 000000      E.BA: .WORD 0       :ON INTERRUPT
290 002424 000000      E.DA: .WORD 0
291 002426 000000      E.MP: .WORD 0
292 002430 000000      E.MP1: .WORD 0
293 002432 000000      E.MP2: .WORD 0
294 002434 000000      C.HDR: .WORD 0     :CURRENT HEADER - FOR ERROR REPORT
295 002436 000000      BUF1: .WORD 0      :BUFFER FOR FIRST CONTROLLER
296 002440 000000      BUF2: .WORD 0      :BUFFER FOR SECOND CONTROLLER
297 002442 000000      MAXWC: .WORD 0     :MAX WORD COUNT DETERMINED BY CORE
298 002444 000000      UUT: .WORD 0       :NUMBER OF UNITS ON SYSTEM
299 002446 000000      PWRFLG: .WORD 0    :POWER FAIL INDICATOR
300 002450 000000      TRPFLG: .WORD 0    :INDICATES OCCURRENCE OF A TIME-OUT TRAP
301 002452 000000      STFLG: .WORD 0     :START FLAG
302                                     :
303                                     :END OF MASS CLEAR
304                                     :
305 002454 000000      CNTFLG: .WORD 0    :CONTINUE FLAG
306 002456 000000      FASCII: .WORD 0    :ASCII MESSAGE OF FUNCTION
307 002460 000000      FASPNT: .WORD 0    :POINTER
308 002462 000000      DWCNT: .WORD 0     :ERROR COUNT
309 002464 000000      DWCNT1: .WORD 0    :ERROR COUNT
310 002466 000004      ERRVEC: .WORD 4    :ERROR VECTOR
311 002470 000034      ST1: .WORD 34      :STATES ALLOWED
312 002472 000035      ST2: .WORD 35      :STATES ALLOWED
313 002474 000000      OPCALL: .WORD 0
314 002476 000000      INCALL: .WORD 0
315 002500 000000      DLYCNT: .WORD 0    :DELAY COUNTER FOR WAITMS TIMING MACRO
316 002502 000000      SYSCLK: .WORD 0    :FLAG INDICATING PRESENCE OF A SYSTEM CLOCK
317 002504 000000      CLKSON: .WORD 0    :"CLOCK ON" INDICATOR
318 002506 000000      CLKCNT: .WORD 0    :CLOCK COUNTER TO STORE TICK VALUE
319 002510 000000      CLKBFR: .WORD 0    :CLOCK BUFFER TO STORE CLOCK TICK COUNT
320 002512 000000      CLKACC: .WORD 0    :CLOCK ACCUMULATOR TO STORE ELAPSED TIME IN
321                                     :/SECONDS OF SUPERVISOR TIME
322 002514 000000      CLKFLD: .WORD 0    :CLOCK FIELD USED TO CHECK IF LSI-11 CLOCK
323                                     :/IS 'TICKING'
324
325 002516                                     ENDMOD
326
327
```



```
329
330 .SBTTL GLOBAL MESSAGES
331
332 002516 BGNMOD GLBTXT
333
334 ;GLOBAL TEXT
335
336
337
338
339 002516 044524 042515 020072 TIME: .ASCIZ 'TIME: ''
340 002525 040 046122 051503 MRLCS: .ASCIZ '' RLCS: ''
341 002535 040 051050 041514 CRLCS: .ASCIZ '' (RLCS): ''
342 002547 076 020076 052506 MFUNC: .ASCIZ '>> FUNCTION: ''
343 002565 040 051050 041114 CRLBA: .ASCIZ '' (RLBA): ''
344 002577 040 051050 042114 CRLDA: .ASCIZ '' (RLDA): ''
345 002611 040 051050 046514 CRLMP: .ASCIZ '' (RLMP): ''
346
347
348
349 002623 104 043111 053440 DIFMSG: .ASCIZ /DIF WD: /
350 002634 040520 045503 051440 CART: .ASCIZ /PACK SERIAL #: /
351 002654 047516 041440 042122 NOCRDY: .ASCIZ /NO CRDY/
352 002664 051104 053111 020105 DNRDY: .ASCIZ /DRIVE NOT READY/
353 002704 051104 047040 052117 NORDY: .ASCIZ %DR NOT RDY W/O DR ERR%
354 002732 052502 000107 PRGER: .ASCIZ /BUG/
355 002736 047111 052111 053440 NWRTS: .ASCIZ /INIT WR OF SEC BAD/
356 002761 040 042523 052103 SMSG: .ASCIZ / SECTOR: /
357 002773 116 020117 047507 EXHAUS: .ASCIZ /NO GOOD HDR/
358 003007 125 042116 040511 UDERR: .ASCIZ /UNDIAGNOSABLE ERR/
359 003031 123 042505 020113 MSKER: .ASCIZ /SEEK ERR/
360 003042 047523 052106 042440 MSFER: .ASCIZ /SOFT ERR ENC'D/
361 003061 104 020122 051105 DRVER: .ASCIZ /DR ERR/
362 003070 051104 042440 051122 MDERS: .ASCIZ /DR ERR WILL NOT RESET/
363 003116 051104 051440 040524 MDSER: .ASCIZ /DR STAT ERR/
364 003132 047526 020114 044103 MVCER: .ASCIZ /VOL CHK WILL NOT CLR/
365 003157 127 020122 040507 WGEST: .ASCIZ /WR GATE ERR WILL NOT RESET/
366 003212 051104 042440 051122 MRDER: .ASCIZ /DR ERR - RECOVERED/
367 003235 104 052101 020101 MDCER: .ASCIZ /DATA CMP ERR/
368 003252 040510 042122 042440 MHDER: .ASCIZ /HARD ERROR/
369 003265 104 052101 020101 DMPDCK: .ASCIZ /DATA DUMP - DCK/
370 003305 124 040522 045503 TRACK: .ASCIZ /TRACKING ERR/
371 003322 051110 020104 051105 ERLMTM: .ASCIZ /HRD ERR LMT EXC'D/
372 003344 045523 042440 051122 SERLMT: .ASCIZ /SK ERR LMT EXC'D/
373 003365 123 052106 042440 SFEMSG: .ASCIZ /SFT ERR LMT EXC'D/
374 003407 104 052101 020101 DCDMSG: .ASCIZ /DATA ERR LMT EXC'D/
375 003432 051104 042440 051122 DERMSG: .ASCIZ /DR ERR LMT EXC'D/
376 003453 102 043125 042506 OVER: .ASCIZ /BUFFER CHOSEN TOO BIG - WAS /
377 003510 042522 020121 054502 REQ: .ASCIZ /REQ BY OPR/
378 003523 105 044130 042047 SEXHAU: .ASCIZ /EXH'D RETRY ON SEEK/
379 003547 110 051504 047040 UNLOAD: .ASCIZ /HDS NOT UNLD ON ERR/
380 003573 104 020122 046127 NOLOAD: .ASCIZ /DR WLD NOT LD/
381 003611 117 042520 020122 SOPLMT: .ASCIZ /OPER LMTS-EXC'D/
382 003631 107 051101 046102 NOREV: .ASCIZ /GARBLED DATA - CAN'T CHECK IT/
383 003667 115 051117 020105 MBDMSC: .ASCIZ /MORE THAN 16 BAD SECTORS/
384 003720 047516 043040 041501 HWSEC: .ASCIZ /NO FACTORY FILE/
385 003740 047516 043040 042511 SWSEC: .ASCIZ /NO FIELD FILE/
386 003756 026520 040524 046102 MPT: .ASCIZ /P-TABLE: /
387 003770 046111 020114 026520 ILLEG: .ASCIZ /ILL P-TABLE/
```

388	004004	053040	041505	047524	MVEC:	.ASCIZ	/ VECTOR: /
389	004016	047516	042040	044522	NODRIV:	.ASCIZ	/NO DRIVES/
390	004030	042040	044522	042526	DRNM:	.ASCIZ	/ DRIVE: /
391	004041	040	051514	020124	LPS:	.ASCIZ	/ LST POS: /
392	004054	054105	020120	047520	EPS:	.ASCIZ	/EXP POS: /
393	004066	051040	041505	050040	RPS:	.ASCIZ	/ REC POS: /
394	004101	104	020122	044504	NOPWR:	.ASCIZ	/DR DID REC'R FROM PWR UP/
395	004132	052101	041040	051525	BUSAD:	.ASCIZ	/AT BUS ADDR: /
396	004150	042522	051124	051531	MRT:	.ASCIZ	/RETRYS: /
397	004161	040	051105	047522	ERT:	.ASCIZ	/ ERROR TYPE: /
398	004177	123	040524	052524	MST:	.ASCIZ	/STATUS WAS: /
399	004214	051440	047510	046125	MST1:	.ASCIZ	/ SHOULD BE: /
400	004231	040	042522	051124	RT1:	.ASCIZ	/ RETRIES ATTEMPTED/
401	004254	042440	050130	042047	EXP:	.ASCIZ	/ EXP'D: /
402	004265	040	042522	023503	RCD:	.ASCIZ	/ REC'D: /
403	004276	051104	053111	020105	DROP:	.ASCIZ	/DRIVE DROPPED/
404	004314	044040	043116	000	MTHNF:	.ASCIZ	/ HNF/
405	004321	040	041510	041522	MTHCRC:	.ASCIZ	/ HCRC/
406	004327	040	041504	000113	MTDCRC:	.ASCIZ	/ DCK/
407	004334	042040	052114	000	MTDLT:	.ASCIZ	/ DLT/
408	004341	040	050117	000111	MTOPI:	.ASCIZ	/ OPI/
409	004346	047040	046530	000	MTNXM:	.ASCIZ	/ NXM/
410	004353	040	051104	000126	MTDRV:	.ASCIZ	/ DRV/
411	004360	042524	052123	047111	MSTART:	.ASCIZ	/TESTING STARTED/
412	004400	051127	052111	047111	MSWRPK:	.ASCIZ	/WRITING PACK /
413	004416	040520	045503	047040	NORDDC:	.ASCIZ	/PACK NOT FULLY INIT'D...DATA COMPARE INHIBITED/
414	004476	052503	051122	047105	ERRHDR:	.ASCIZ	/CURRENT POSITION (HDR) = /
415	004530	054523	052123	046505	NOCLK:	.ASCIZ	/SYSTEM CLOCK IS NOT AVAILABLE/
416	004566	042520	043122	051117	NOREPT:	.ASCIZ	/PERFORMANCE REPORTS WILL NOT BE PRINTED/
417	004636	044504	020104	047516	NOTRDY:	.ASCIZ	/DID NOT RESPOND WITH 'READY'/
418	004673	116	020117	047503	NOCTLR:	.ASCIZ	/NO CONTROLLER/
419	004711	123	051531	042524	INSMEM:	.ASCIZ	/SYSTEM FATAL ERROR - INSUFFICIENT MEMORY BUFFER SPACE/

420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443

: THIS LIST OF ASCII TEXT IS USED AS A TABLE FOR PRINTING
: FUNCTIONS IN ERROR MESSAGES TABLE IS 'MTCR - MTRD',
: THE ORDER IS IMPORTANT AS WELL AS THE LENGTH OF EACH
: ASCII STRING. EACH STRING IS SEVEN(10) BYTES PLUS ZERO
: FILL BYTE (TOTAL 8(EIGHT) BYTES) LONG. USED IN LINE1
: SUBROUTINE.....
:.....

429	004777	040	051127	044103	MTCR:	.ASCIZ	/ WRCHK /
430	005007	040	052107	052123	MTGS:	.ASCIZ	/ GTSTAT/
431	005017	040	042523	045505	MTSK:	.ASCIZ	/ SEEK /
432	005027	040	042122	042110	MTRH:	.ASCIZ	/ RDHDR /
433	005037	040	051127	052111	MTWR:	.ASCIZ	/ WRITE /
434	005047	040	042522	042101	MTRD:	.ASCIZ	/ READ /
435	005057	040	042122	047055	MTRNH:	.ASCIZ	/ RD-NHD/

:.....
: END OF LIST - YOU CAN PUT ANYTHING YOU WANT HERE
:.....

.NLIST CND,MD,ME


```
444
445      005070      .EVEN
446
447 005070      ENDMOD
448
449      .SBTTL  ERROR MESSAGES
450
451 005070      BGNMOD  GLBERR
452
453                                     ;GENERAL ERROR REPORT
454
455 005070      BGNMSG  ERR1
456 005070 004737 006330      JSR      PC,LINE3
457 005074      ENDMSG
(3) 005074
(3) 005074 104423      L10000:  TRAP      C$MSG
458
459                                     ;MIS-SEEK ERROR REPORT
460
461 005076      BGNMSG  ERR2
462 005076 004737 006330      JSR      PC,LINE3
463 005102      PRINTB  #FMT4,#DIFMSG,DIFWD(R4),#LPS,LSTHDR(R4),#EPS,PRPOS(R4),#RPS,R1
(15) 005102 010146      MOV      R1,-(SP)
(14) 005104 012746 004066      MOV      #RPS,-(SP)
(13) 005110 016446 000124      MOV      PRPOS(R4),-(SP)
(12) 005114 012746 004054      MOV      #EPS,-(SP)
(11) 005120 016446 000050      MOV      LSTHDR(R4),-(SP)
(10) 005124 012746 004041      MOV      #LPS,-(SP)
(9) 005130 016446 000066      MOV      DIFWD(R4),-(SP)
(8) 005134 012746 002623      MOV      #DIFMSG,-(SP)
(7) 005140 012746 007104      MOV      #FMT4,-(SP)
(6) 005144 012746 000011      MOV      #11,-(SP)
(3) 005150 010600      MOV      SP,R0
(4) 005152 104414      TRAP    C$PNTB
(4) 005154 062706 000024      ADD     #24,SP
464 005160      ENDMSG
(3) 005160
(3) 005160 104423      L10001:  TRAP    C$MSG
465
466                                     ;SOFT ERROR RECOVERABLE ERROR REPORT
467 005162      BGNMSG  ERR3
468 005162 004737 006014      JSR      PC,LINE1
469 005166      PRINTB  #FMT2A,#CRLCS,SOFTCS(R4),#CRLBA,@BBA(R4),#CRLDA,LSTDA(R4)
(13) 005166 016446 000064      MOV      LSTDA(R4),-(SP)
(12) 005172 012746 002577      MOV      #CRLDA,-(SP)
(11) 005176 017446 000110      MOV      @BBA(R4),-(SP)
(10) 005202 012746 002565      MOV      #CRLBA,-(SP)
(9) 005206 016446 000116      MOV      SOFTCS(R4),-(SP)
(8) 005212 012746 002535      MOV      #CRLCS,-(SP)
(7) 005216 012746 006735      MOV      #FMT2A,-(SP)
(6) 005222 012746 000007      MOV      #7,-(SP)
(3) 005226 010600      MOV      SP,R0
(4) 005230 104414      TRAP    C$PNTB
(4) 005232 062706 000020      ADD     #20,SP
470 005236 016437 000064 002346      MOV      LSTDA(R4),TEMPO ;GET THE ADDRESS TO PRINT
471 005244 004537 006510      JSR      R5,TELCYL      ;CONVERT FOR PRINTING
```

472	005250			PRINTB	#FMT5,#MRT,RETRY(R4),#ERT,RTYPE(R4)
(11)	005250	016446	000052	MOV	RTYPE(R4),-(SP)
(10)	005254	012746	004161	MOV	#ERT,-(SP)
(9)	005260	016446	000036	MOV	RETRY(R4),-(SP)
(8)	005264	012746	004150	MOV	#MRT,-(SP)
(7)	005270	012746	007137	MOV	#FMT5,-(SP)
(6)	005274	012746	000005	MOV	#5,-(SP)
(3)	005300	010600		MOV	SP,R0
(4)	005302	104414		TRAP	C\$PNTB
(4)	005304	062706	000014	ADD	#14,SP
473	005310			ENDMSG	
(3)	005310			L10002:	
(3)	005310	104423		TRAP	C\$MSG
474					
475					
476					:GET STATUS ERROR REPORT
477	005312			BGNMSG	ERR4
478	005312	004737	006330	JSR	PC,LINE3
479	005316			PRINTB	#FMT6,#MST,E.MP,#MST1,ST1,ST2
(12)	005316	013746	002472	MOV	ST2,-(SP)
(11)	005322	013746	002470	MOV	ST1,-(SP)
(10)	005326	012746	004214	MOV	#MST1,-(SP)
(9)	005332	013746	002426	MOV	E.MP,-(SP)
(8)	005336	012746	004177	MOV	#MST,-(SP)
(7)	005342	012746	007153	MOV	#FMT6,-(SP)
(6)	005346	012746	000006	MOV	#6,-(SP)
(3)	005352	010600		MOV	SP,R0
(4)	005354	104414		TRAP	C\$PNTB
(4)	005356	062706	000016	ADD	#16,SP
480	005362			ENDMSG	
(3)	005362			L10003:	
(3)	005362	104423		TRAP	C\$MSG
481					
482					
483					:DATA ERROR SUMMARY
484					
485	005364			BGNMSG	ERR6
486	005364	004737	006220	JSR	PC,LINE2
487	005370	016400	000042	MOV	BMP(R4),R0
488	005374			PRINTB	#FMT9A,DECNT,R0
(9)	005374	010046		MOV	R0,-(SP)
(8)	005376	013746	002344	MOV	DECNT,-(SP)
(7)	005402	012746	007263	MOV	#FMT9A,-(SP)
(6)	005406	012746	000003	MOV	#3,-(SP)
(3)	005412	010600		MOV	SP,R0
(4)	005414	104414		TRAP	C\$PNTB
(4)	005416	062706	000010	ADD	#10,SP
489	005422			ENDMSG	
(3)	005422			L10004:	
(3)	005422	104423		TRAP	C\$MSG
490					
491					:NON-RECOVERABLE ERROR REPORT
492					
493	005424			BGNMSG	ERR7
494	005424			PRINTB	#FMT8,RETRY(R4),#RT1
(9)	005424	012746	004231	MOV	#RT1,-(SP)

J 4

(8) 005430 016446 000036
(7) 005434 012746 007215
(6) 005440 012746 000003
(3) 005444 010600
(4) 005446 104414
(4) 005450 062706 000010
495 005454 004737 006330
496 005460
(3) 005460
(3) 005460 104423
497
498
499
500 005462
501 005462 004737 006220
502 005466 016437 000040
503 005474 004537 006510
504 005500
(15) 005500 011246
(14) 005502 012746 004265
(13) 005506 013746 002402
(12) 005512 012746 004254
(11) 005516 016446 000040
(10) 005522 012746 002577
(9) 005526 017446 000110
(8) 005532 012746 002565
(7) 005536 012746 007373
(6) 005542 012746 000011
(3) 005546 010600
(4) 005550 104414
(4) 005552 062706 000024
505 005556
(8) 005556 010246
(7) 005560 012746 007444
(6) 005564 012746 000002
(3) 005570 010600
(4) 005572 104414
(4) 005574 062706 000006
506 005600
(3) 005600
(3) 005600 104423
507
508
509 005602
510
511 005602 004737 006330
512 005606
(11) 005606 016446 000050
(10) 005612 012746 004041
(9) 005616 010146
(8) 005620 012746 004177
(7) 005624 012746 007502
(6) 005630 012746 000005
(3) 005634 010600
(4) 005636 104414
(4) 005640 062706 000014

002346

MOV RETRY(R4),-(SP)
MOV #FMT8,-(SP)
MOV #3,-(SP)
MOV SP,R0
TRAP C\$PNTB
ADD #10,SP
JSR PC,LINE3
ENDMSG
L10005: TRAP C\$MSG
;BAD DATA COMPARE ERROR REPORT
BGNMSG ERR8
JSR PC,LINE2
MOV BDA(R4),TEMPO
JSR R5,TELCYL ;REPORT THE CYL # & SECTOR/HEAD
PRINTB #FMT10A,#CRLBA,@BBA(R4),#CRLDA,BDA(R4),#EXP,GDDAT,#RCD,(R2)
MOV (R2),-(SP)
MOV #RCD,-(SP)
MOV GDDAT,-(SP)
MOV #EXP,-(SP)
MOV BDA(R4),-(SP)
MOV #CRLDA,-(SP)
MOV @BBA(R4),-(SP)
MOV #CRLBA,-(SP)
MOV #FMT10A,-(SP)
MOV #11,-(SP)
MOV SP,R0
TRAP C\$PNTB
ADD #24,SP
PRINTB #FMT10B,R2
MOV R2,-(SP)
MOV #FMT10B,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C\$PNTB
ADD #6,SP
ENDMSG
L10006: TRAP C\$MSG
;DRIVE ERROR
BGNMSG ERR9
JSR PC,LINE3
PRINTB #FMT13,#MST,R1,#LPS,LSTHDR(R4)
MOV LSTHDR(R4),-(SP)
MOV #LPS,-(SP)
MOV R1,-(SP)
MOV #MST,-(SP)
MOV #FMT13,-(SP)
MOV #5,-(SP)
MOV SP,R0
TRAP C\$PNTB
ADD #14,SP

```
513 005644          ENDMSG
(3) 005644          L10007:
(3) 005644 104423   TRAP    C$MSG
514
515
516                                     ;INVALID ENTRY IN P-TABLE REPORT
517
518 005646          BGNMSG  ERR10
519 005646          PRINTB  #FMT11,#MPT,R1,#MRLCS,BCSR,#MVEC,BVEC
(13) 005646 013746 002332   MOV    BVEC,-(SP)
(12) 005652 012746 004004   MOV    #MVEC,-(SP)
(11) 005656 013746 002330   MOV    BCSR,-(SP)
(10) 005662 012746 002525   MOV    #MRLCS,-(SP)
(9) 005666 010146          MOV    R1,-(SP)
(8) 005670 012746 003756   MOV    #MPT,-(SP)
(7) 005674 012746 007452   MOV    #FMT11,-(SP)
(6) 005700 012746 000007   MOV    #7,-(SP)
(3) 005704 010600          MOV    SP,R0
(4) 005706 104414          TRAP   C$PNTB
(4) 005710 062706 000020   ADD    #20,SP
520 005714          ENDMSG
(3) 005714          L10010:
(3) 005714 104423   TRAP    C$MSG
521
522
523 005716          BGNMSG  ERR12
524
525 005716 004737 006330   JSR    PC,LINE3
526
527 005722          ENDMSG
(3) 005722          L10011:
(3) 005722 104423   TRAP    C$MSG
528
529 005724          BGNMSG  ERR13
530 005724 004737 006330   JSR    PC,LINE3
531 005730 016403 000104   MOV    DCS(R4),R3
532 005734 016337 000006 002426   MOV    MP(R3),E.MP ;GET HEADER
533 005742          PRINTB  #FMT14C ;CRLF
(7) 005742 012746 007562   MOV    #FMT14C,-(SP)
(6) 005746 012746 000001   MOV    #1,-(SP)
(3) 005752 010600          MOV    SP,R0
(4) 005754 104414          TRAP   C$PNTB
(4) 005756 062706 000004   ADD    #4,SP
534 005762          PRINTB  #FMT12,#ERRHDR,C.HDR ;PRINT THE HEADER MESSAGE
(9) 005762 013746 002434   MOV    C.HDR,-(SP)
(8) 005766 012746 004476   MOV    #ERRHDR,-(SP)
(7) 005772 012746 007472   MOV    #FMT12,-(SP)
(6) 005776 012746 000003   MOV    #3,-(SP)
(3) 006002 010600          MOV    SP,R0
(4) 006004 104414          TRAP   C$PNTB
(4) 006006 062706 000010   ADD    #10,SP
535 006012          ENDMSG
(3) 006012          L10012:
(3) 006012 104423   TRAP    C$MSG
536
537 006014 016437 000044 002460 LINE1: MOV    FUNC(R4),FASPNT ;GET FUNCTION
```



```

538 006022 012737 004777 002456      MOV      #MTCR,FASCII      ;FIRST FUNCTION ASCIZ
539 006030 042737 000100 002460      BIC      #INTEN,FASPNT    ;CLEAR INTERRUPT ENABLE
540 006036 006237 002460                ASR      FASPNT           ;ALIGN - NOW = 1 TO 7
541 006042 005337 002460                1$:     DEC      FASPNT    ;DOWN COUNT FUNCTION
542 006046 001404                BEQ      2$              ;FOUND?
543 006050 062737 000010 002456      ADD      #8.,FASCII      ;NO NEXT ONE
544 006056 000771                BR       1$              ;LOOP
545
546 006060                2$:     PRINTB   #FMT10,#TIME,HOUR,MINUTE,SECOND,#MRLCS,DCS(R4)#DRNM,<B,DRSEL+1(R4)>
(15) 006060 005046                CLR      -(SP)
(15) 006062 156416 000107      BISB    DRSEL+1(R4),(SP)
(14) 006066 012746 004030      MOV     #DRNM,-(SP)
(13) 006072 016446 000104      MOV     DCS(R4),-(SP)
(12) 006076 012746 002525      MOV     #MRLCS,-(SP)
(11) 006102 013746 002412      MOV     SECOND,-(SP)
(10) 006106 013746 002414      MOV     MINUTE,-(SP)
(9) 006112 013746 002416      MOV     HOUR,-(SP)
(8) 006116 012746 002516      MOV     #TIME,-(SP)
(7) 006122 012746 007337      MOV     #FMT10,-(SP)
(6) 006126 012746 000011      MOV     #11,-(SP)
(3) 006132 010600      MOV     SP,R0
(4) 006134 104414      TRAP    C$PNTB
(4) 006136 062706 000024      ADD     #24,SP
547 006142                PRINTB   #FMTDT,TDR(R4)
(8) 006142 016446 000120      MOV     TDR(R4),-(SP)
(7) 006146 012746 010164      MOV     #FMTDT,-(SP)
(6) 006152 012746 000002      MOV     #2,-(SP)
(3) 006156 010600      MOV     SP,R0
(4) 006160 104414      TRAP    C$PNTB
(4) 006162 062706 000006      ADD     #6,SP
548 006166                PRINTB   #FMT1A,#MFUNC,FASCII
(9) 006166 013746 002456      MOV     FASCII,-(SP)
(8) 006172 012746 002547      MOV     #MFUNC,-(SP)
(7) 006176 012746 006705      MOV     #FMT1A,-(SP)
(6) 006202 012746 000003      MOV     #3,-(SP)
(3) 006206 010600      MOV     SP,R0
(4) 006210 104414      TRAP    C$PNTB
(4) 006212 062706 000010      ADD     #10,SP
549 006216 000207      RTS     PC
550
551 006220                LINE2:  PRINTB   #FMT10,#TIME,HOUR,MINUTE,SECOND,#MRLCS,DCS(R4)#DRNM,<B,DRSEL+1(R4)>
(15) 006220 005046                CLR      -(SP)
(15) 006222 156416 000107      BISB    DRSEL+1(R4),(SP)
(14) 006226 012746 004030      MOV     #DRNM,-(SP)
(13) 006232 016446 000104      MOV     DCS(R4),-(SP)
(12) 006236 012746 002525      MOV     #MRLCS,-(SP)
(11) 006242 013746 002412      MOV     SECOND,-(SP)
(10) 006246 013746 002414      MOV     MINUTE,-(SP)
(9) 006252 013746 002416      MOV     HOUR,-(SP)
(8) 006256 012746 002516      MOV     #TIME,-(SP)
(7) 006262 012746 007337      MOV     #FMT10,-(SP)
(6) 006266 012746 000011      MOV     #11,-(SP)
(3) 006272 010600      MOV     SP,R0
(4) 006274 104414      TRAP    C$PNTB
(4) 006276 062706 000024      ADD     #24,SP
552 006302                PRINTB   #FMTDT,TDR(R4)

```

```

(8) 006302 016446 000120      MOV      TDR(R4),-(SP)
(7) 006306 012746 010164      MOV      #FMTDT, -(SP)
(6) 006312 012746 000002      MOV      #2, -(SP)
(3) 006316 010600                MOV      SP,R0
(4) 006320 104414                TRAP     C$PNTB
(4) 006322 062706 000006      ADD      #6,SP
553 006326 000207                RTS      PC
554
555 006330 004737 006014      LINE3:  JSR      PC,LINE1
556 006334                PRINTB  #FMT2,#CRLCS,BCSADR(R4),#CRLBA,@BBA(R4),#CRLDA,BDA(R4),#CRLMP,BMP(R4)
(15) 006334 016446 000042      MOV      BMP(R4),-(SP)
(14) 006340 012746 002611      MOV      #CRLMP, -(SP)
(13) 006344 016446 000040      MOV      BDA(R4),-(SP)
(12) 006350 012746 002577      MOV      #CRLDA, -(SP)
(11) 006354 017446 000110      MOV      @BBA(R4),-(SP)
(10) 006360 012746 002565      MOV      #CRLBA, -(SP)
(9) 006364 016446 000046      MOV      BCSADR(R4),-(SP)
(8) 006370 012746 002535      MOV      #CRLCS, -(SP)
(7) 006374 012746 006714      MOV      #FMT2, -(SP)
(6) 006400 012746 000011      MOV      #11, -(SP)
(3) 006404 010600                MOV      SP,R0
(4) 006406 104414                TRAP     C$PNTB
(4) 006410 062706 000024      ADD      #24,SP
557 006414                PRINTB  #FMT3,#CRLCS,E.CS,#CRLBA,E.BA,#CRLDA,E.DA,#CRLMP,E.MP
(15) 006414 013746 002426      MOV      E.MP, -(SP)
(14) 006420 012746 002611      MOV      #CRLMP, -(SP)
(13) 006424 013746 002424      MOV      E.DA, -(SP)
(12) 006430 012746 002577      MOV      #CRLDA, -(SP)
(11) 006434 013746 002422      MOV      E.BA, -(SP)
(10) 006440 012746 002565      MOV      #CRLBA, -(SP)
(9) 006444 013746 002420      MOV      E.CS, -(SP)
(8) 006450 012746 002535      MOV      #CRLCS, -(SP)
(7) 006454 012746 006757      MOV      #FMT3, -(SP)
(6) 006460 012746 000011      MOV      #11, -(SP)
(3) 006464 010600                MOV      SP,R0
(4) 006466 104414                TRAP     C$PNTB
(4) 006470 062706 000024      ADD      #24,SP
558 006474 013737 002424 002346      MOV      E.DA,TEMPO      ;GET ADDRESS TO PRINT
559 006502 004537 006510      JSR      R5,TELCYL      ;PRINT IT
560 006506 000207                RTS      PC              ;EXIT
561
562 006510 013737 002346 002300      TELCYL: MOV      TEMPO,CYL      ;GET THE ADDRESS
563 006516 042737 000177 002300      BIC      #177,CYL      ;SAVE ONLY CYLINDER BITS
564 006524 000337 002300      SWAB     CYL
565 006530 000241                CLC
566 006532 006137 002300      ROL     CYL
567 006536 103002                BCC     1$
568 006540 005237 002300      INC     CYL
569 006544 013737 002346 002304 1$:  MOV      TEMPO,SEC      ;GET SECTOR #
570 006552 042737 177700 002304      BIC      #177700,SEC    ;SAVE ONLY THE SECTOR BITS
571 006560 005037 002302                CLR     SUR              ;INIT TO HEAD 0
572 006564 032737 000100 002424      BIT     #100,E.DA      ;HEAD 1?
573 006572 001405                BEQ     2$              ;NO
574 006574 005237 002302                INC     SUR              ;YUP
575 006600 042737 177776 002302      BIC      #177776,SUR
576 006606                PRINTB  #FMT3A,#DRVER,CYL,SUR,SEC

```


(11)	006606	013746	002304	MOV	SEC,-(SP)
(10)	006612	013746	002302	MOV	SUR,-(SP)
(9)	006616	013746	002300	MOV	CYL,-(SP)
(8)	006622	012746	003061	MOV	#DRVER,-(SP)
(7)	006626	012746	007020	MOV	#FMT3A,-(SP)
(6)	006632	012746	000005	MOV	#5,-(SP)
(3)	006636	010600		MOV	SP,R0
(4)	006640	104414		TRAP	C\$PNTB
(4)	006642	062706	000014	ADD	#14,SP
577	006646	000205		RTS	R5

;FORMAT STATEMENTS

585	006650	052045	055045	022462	FMT1:	.ASCIZ	/%T%Z2%A:%Z2%A:%Z2/
586	006672	052045	047445	022466	FMT17:	.ASCIZ	/%T%06%T%01/
587	006705	045	022524	022524	FMT1A:	.ASCIZ	/%T%T%N/
588	006714	040445	042502	047506	FMT2:	.ASCIZ	/%ABEFORE ERR%T%06/
589	006735	045	022524	033117	FMT2A:	.ASCIZ	/%T%06%T%06%T%06%N/
590	006757	045	040501	020124	FMT3:	.ASCIZ	/%AAT ERR %T%06%T%06%T%06%T%06/
591	007020	047045	052045	040445	FMT3A:	.ASCIZ	/%N%T%A ADDR = CYL: %Z3%A. SUR: %01%A SECT: %Z2%A.%N/
592	007104	047045	052045	047445	FMT4:	.ASCIZ	/%N%T%06%T%06%N%T%06%T%06%N/
593	007137	045	022524	033117	FMT5:	.ASCIZ	/%T%06%T%T%N/
594	007153	045	022524	033117	FMT6:	.ASCIZ	/%T%06%T%06%A OR %06%N/
595	007201	045	022524	020101	FMT7:	.ASCIZ	/%T%A - %T%N/
596	007215	045	033104	052045	FMT8:	.ASCIZ	/%D6%T%N/
597	007225	045	022524	031132	FMT9:	.ASCIZ	/%T%Z2%A:%Z2%A:%Z2%T%06%T%01%N/
598	007263	045	033104	040445	FMT9A:	.ASCIZ	/%D6%A. WORDS BAD OUT OF %D6%A. WORDS READ%N/
599	007337	045	022524	031132	FMT10:	.ASCIZ	/%T%Z2%A:%Z2%A:%Z2%T%06%T%01/
600	007373	045	022524	033117	FMT10A:	.ASCIZ	/%T%06%T%06%N%T%06%T%06%A AT BUS ADDRESS /
601	007444	047445	022466	000116	FMT10B:	.ASCIZ	/%06%N/
602	007452	052045	047445	022462	FMT11:	.ASCIZ	/%T%02%T%06%T%03/
603	007472	052045	047445	022466	FMT12:	.ASCIZ	/%T%06%N/
604	007502	052045	047445	022466	FMT13:	.ASCIZ	/%T%06%T%06%N/
605	007517	045	022524	032132	FMT13D:	.ASCIZ	/%T%Z4%A NOW IS %Z4%N/
606	007544	047045	052045	047045	FMT14:	.ASCIZ	/%N%T%N/
607	007553	045	033117	040445	FMT14A:	.ASCIZ	/%06%A /
608	007562	047045	000		FMT14C:	.ASCIZ	/%N/
609	007565	045	053501	051117	FMT14B:	.ASCIZ	?%AWORD %D3%A. S/B %06%A WAS %06%N?
610	007627	045	042501	051122	FMT15:	.ASCIZ	/%AERROR(S) SET:%T%N%ZARECOVERY BEING ATTEMPTED/
611	007705	045	022516	047101	FRMT.16:	.ASCIZ	/%N%ANOT TESTING CS=%06%A DR=%01%N/
612	007747	045	022516	000124	FMT18:	.ASCIZ	/%N%T/
613	007754	047045	040445	043130	FMTXS:	.ASCIZ	/%N%AXFER SIZE = %Z6%A. WORDS%N/
614	010013	045	022516	022516	FMTS1:	.ASCIZ	/%N%N%S10%A*** RL01-RL02 PERFORMANCE REPORT ***%N%N/
615	010076	040445	025052	020052	FMTS1A:	.ASCIZ	/%A*** RUNNING%N/
616	010116	040445	025052	020052	FMTS1B:	.ASCIZ	/%A*** DROPPED %Z2%A:%Z2%N/
617	010151	045	022524	032517	FMTS2:	.ASCIZ	/%T%05%05%N/
618	010164	040445	042040	044522	FMTDT:	.ASCIZ	/%A DRIVE TYPE = RL0%01%N/
619	010215	045	052101	052117	FMTS2A:	.ASCIZ	/%ATOTAL SEEKS: %D6%Z3%N%AWORDS READ: %D6%Z4%Z4%N/
620	010304	040445	047527	042122	FMTS2B:	.ASCIZ	/%AWORDS WRITTEN: %D6%Z4%Z4%N/
621	010341	045	022516	042501	FMTS3:	.ASCIZ	/%N%AEERRORS%N%ADR-ER:%D6%A SEEK: %D6%A TRACK: %D6%A DATA: %D6%N/
622	010450	040445	040510	042122	FMTS3A:	.ASCIZ	/%AHARD: %D6%A SOFT: %D6%N/
623	010505	045	042101	045503	FMTS4:	.ASCIZ	/%ADCK: %D6%A HCRC: %D6%A NXM: %D6%A HNF: %D6%N/
624	010600	040445	046104	035124	FMTS5:	.ASCIZ	/%ADLT: %D6%A OPI: %D6%N%N/
625							
629							


```
630          010640          .EVEN
631
632 010640          ENDMOD
633
634
635          .SBTTL  DEFAULT HARDWARE P-TABLE PARAMETERS
636
637 010640          BGNMOD  HPTCODE
638
639 010640          BGNHW
(3) 010640 000006          .WORD  L10013-L$HW/2
640
641 010642 174400          .WORD  174400          ;DRIVE CSR
642 010644 000160          .WORD  160           ;DRIVE VECTOR
643 010646 000240          .WORD  240           ;DRIVE PRIORITY
644 010650 000001          .WORD  1             ;DRIVE TYPE
645 010652 000000          .WORD  0             ;DRIVE NUMBER
646 010654 000001          .WORD  1             ;CONTROLLER TYPE
647
648 010656          ENDMOD
(3) 010656          L10013:
649
650 010656          ENDMOD
651
652          .SBTTL  DEFAULT SOFTWARE P-TABLE PARAMETERS
653
654 010656          BGNMOD  SPTCODE
655
656 010656          BGNMW
(3) 010656 000037          .WORD  L10014-L$SW/2
657
658 010660 000001          LIMIT: .WORD  1           ;RETRY LIMIT
659 010662 000003          ERLMT: .WORD  3           ;ERROR LIMIT
660 010664 000003          SELMT: .WORD  3           ;SEEK ERROR LIMIT
661 010666 060650          DALMT: .WORD  25000.        ;DATA XFER LIMIT (*(10*3)) (BITS)
662 010670 023420          SKLMT: .WORD  10000.       ;SEEK LIMIT
663 010672 000360          TYINT: .WORD  240.       ;TIME INTERVAL IN MINS. BETWEEN STATISTICAL
664                                     ;/REPORTS (4 HRS. TOTAL)
665 010674 000020          CMRD:  .WORD  16.         ;WORDS TO COMPARE ON READ
666 010676 000003          DELMT: .WORD  3           ;ERRORS TO REPORT ON DATA COMPARE
667 010700 000000          XCHFLG: .WORD  0           ;CHANGE OTHER PARAMETERS
668 010702 002400          T.MXB:  .WORD  1280.       ;MAXIMUM R/W TRANSFER BUFFER
669 010704 000100          T.MXH:  .WORD  100         ;MAXIMUM HEAD SELECT
670 010706 000000          T.MNH:  .WORD  0           ;MINIMUM HEAD SELECT
671 010710 177600          T.MXC:  .WORD  177600     ;MAXIMUM CYLINDER
672 010712 000000          T.MNC:  .WORD  0           ;MINIMUM CYLINDER
673 010714 000000          T.MXS:  .WORD  0           ;MAXIMUM START SECTOR
674 010716 000000          T.MNS:  .WORD  0           ;MINIMUM START SECTOR
675 010720 000001          T.DCK:  .WORD  1           ;DATA DUMP ON DATA CHECK ERROR
676 010722 000001          T.DRP:  .WORD  1           ;DROP ON LIMIT REACHED
677 010724 000003          T.MNB:  .WORD  3           ;MINIMUM BUFFER TRANSFER SIZE
678 010726 000012          SFLMT:  .WORD  10.        ;SOFT ERROR LIMIT
679 010730 000000          T.STA:  .WORD  0           ;DROP DRIVE ON PERFORMANCE REACHED
680 010732 000003          DRLMT:  .WORD  3           ;DRIVE ERROR LIMIT
681 010734 000000          T.ROF:  .WORD  0           ;READ ONLY FLAG
682 010736 000001          T.RAN:  .WORD  1           ;RANDOM SELECT OF PATTERNS
```


683 010740 000004
684 010742 000001
685 010744 000200
686 010746 000000
687 010750 000001
688 010752 000012
689 010754 000001
690
691 010756
(3) 010756
692
693 010756
694
695 010756
696
697 010756
(4) 010756 000001
(6) 010760 014534
698
699 010762
700
701
702
703 010762
704
705 010762
706 010762
(7) 010762 012746 010013
(6) 010766 012746 000001
(3) 010772 010600
(4) 010774 104416
(4) 010776 062706 000004
707
708
709 011002 010446
710 011004 012704 030432
711 011010 005764 000104
712 011014 001402
713 011016 004737 014024
714 011022 062704 000126
715 011026 020427 031712
716 011032 001366
717 011034 012604
718 011036
(3) 011036
(3) 011036 104425
719
720 011040
721
722

T.PAT: .WORD 4
T.SLT: .WORD 1
T.CLT: .WORD 128.
T.STIP: .WORD 0
T.WCK: .WORD 1
T.DCD: .WORD 10.
T.ANS: .WORD 1

: ONLY ONE PATTERN 4 = WORST CASE
: SEEK RETRY LIMIT
: NUMBER OF ERRORS ON DCK DUMP
: RESTRICT BUFFER SIZE
: DO WRITE CHECK

ENDSW
L10014:

ENDMOD

BGNMOD DSPCODE

DISPATCH 1
.WORD 1
.WORD T1

ENDMOD

.SBTTL STATISTICAL CODE

BGNMOD RPTCODE

BGNRPT

PRINTS #FMST1
MOV #FMST1, -(SP)
MOV #1, -(SP)
MOV SP, R0
TRAP C\$PNTS
ADD #4, SP

: PRINT STATISTICAL HEADER

: 'RL01-RL02 PERFORMANCE REPORT'

MOV R4, -(SP)
MOV #DRBUF, R4
1\$: TST DCS(R4)
BEQ 2\$
JSR PC, REPORT
2\$: ADD #PRPOS+2, R4
CMP R4, #ENDBUF
BNE 1\$
MOV (SP)+, R4

: SAVE PRESENT VALUE OF R4
: START OF DRIVE BUFFER
: IS THERE A DRIVE?
: NO, GET NEXT ONE
: TYPE OUT SUMMARY
: NEXT DRIVE
: AT THE END?
: NO, TRY NEXT
: RESTORE R4

ENDRPT
L10015:

TRAP C\$RPT

ENDMOD

```

724
725      .SBTTL  LOAD PROTECTION TABLE
726 011040 BGNPROT
727 011040 000000      .WORD  0      ;P-TABLE OFFSET OF CSR
728 011042 177777      .WORD -1      ;NOT A MASS-BUS DRIVE
729 011044 000010      .WORD 10     ;P-TABLE OFFSET OF DRIVE
730 011046 ENDPROT
731
732
733      .SBTTL  INITIALIZATION CODE
734
735 011046 BGNMOD  INITCODE      ;START OF INITIALIZE CODE
736
737 011046      BGNINIT
738
739 011046      SETPRI #340      ;PRIORITY TO 7 TO INHIBIT INTERRUPTS
(3) 011046 012700 000340      MOV #340,R0
(3) 011052 104441      TRAP C$SPRI
740
741 011054      BRESET      ;FOR LSI-11 CPU'S
(3) 011054 104433      TRAP C$RESET
742      ;CLEAR OPERATION FLAGS
743 011056 005037 000050      CLR OPFLG
744 011062 005037 002476      CLR INCALL
745 011066 005037 002452      CLR STFLG
746 011072 005037 002454      CLR CNTFLG      ;CLEAR CONT
747      ;CHECK FOR PRESENCE OF A SYSTEM CLOCK
748 011076 005037 002502      CLR SYSCLK      ;CLEAR SYSTEM CLOCK FLAG
749 011102      CLOCK P,CLKADR      ;P-CLOCK?
(3) 011102 012700 000120      MOV #P,R0
(3) 011106 104462      TRAP C$CLCK
(3) 011110 010037 002316      MOV R0,CLKADR
750 011114      BNCOMPLETE LCLKCH ;BRANCH IF NO P-CLOCK
(2) 011114 103006      BCC LCLKCH
751 011116 012737 000001 002314      MOV #1,CLKTYP      ;IDENTIFY P-CLOCK TYPE
752 011124 005237 002502      INC SYSCLK      ;INDICATE PRESENCE OF A SYSTEM CLOCK
753 011130 000522      BR PWRCH      ;BRANCH TO CHECK POWER
754 011132      LCLKCH: CLOCK L,CLKADR      ;L-CLOCK?
(3) 011132 012700 000114      MOV #L,R0
(3) 011136 104462      TRAP C$CLCK
(3) 011140 010037 002316      MOV R0,CLKADR
755 011144      BCOMPLETE 1$      ;BRANCH IF L-CLOCK
(2) 011144 103401      BCS 1$
756 011146 000467      BR NILCLK      ;ELSE, INDICATE CLOCK IS NOT PRESENT
757 011150      1$: READBUS      ;CHECK TYPE OF BUS
(3) 011150 104407      TRAP C$RDBU
758 011152      BNCOMPLETE 2$      ;BRANCH IF NOT Q-BUS
(2) 011152 103057      BCC 2$
759 011154 005037 002514      CLR CLKFLD      ;CLEAR CLOCK FIELD FOR STORING 'TICKS'
760 011160      SETVEC #100,#CLKTIK,#340 ;SET UP L-CLOCK INTERRUPT VECTOR TO CHECK
(7) 011160 012746 000340      MOV #340,-(SP)
(6) 011164 012746 017110      MOV #CLKTIK,-(SP)
(5) 011170 012746 000100      MOV #100,-(SP)
(4) 011174 012746 000003      MOV #3,-(SP)
(3) 011200 104437      TRAP C$SVEC
(2) 011202 062706 000010      ADD #10,SP
  
```



```

761
762 011206          SETPRI #240          ;/IF CLOCK IS 'TICKING'
(3) 011206 012700 000240 MOV #240,R0          ;SET PRIORITY TO 5 TO ALLOW CLOCK INTERRUPTS
(3) 011212 104441 TRAP C$SPRI
763 011214          WAITMS #5          ;PAUSE TO ALLOW CLOCK INTERRUPTS
(3) 011232 012727 000372 MOV ##250.,(PC)+
(3) 011236 000000 .WORD 0
(3) 011240 013727 002116 MOV L$DLY,(PC)+
(3) 011244 000000 .WORD 0
(3) 011246 005367 177772 DEC -6(PC)
(3) 011252 001375 BNE -4
(3) 011254 005367 177756 DEC -22(PC)
(3) 011260 001367 BNE -20
764 011270          SETPRI #340          ;RESTORE PRIORITY TO 7 TO INHIBIT INTERRUPTS
(3) 011270 012700 000340 MOV #340,R0
(3) 011274 104441 TRAP C$SPRI
765 011276          CLRVEC #100          ;CLEAR L-CLOCK INTERRUPT VECTOR
(3) 011276 012700 000100 MOV #100,R0
(3) 011302 104436 TRAP C$CVEC
766 011304 005737 002514 TST CLKFLD          ;L-CLOCK 'TICKS'?
767 011310 001406 BEQ NILCLK          ;BRANCH IF NO 'TICKS'
768 011312 012737 000002 002314 2$: MOV #2,CLKTYP          ;IDENTIFY L-CLOCK TYPE
769 011320 005237 002502 INC SYSCLK          ;INDICATE PRESENCE OF A SYSTEM CLOCK
770 011324 000424 BR PWRCH          ;BRANCH TO CHECK POWER
771 011326          NILCLK: PRINTF #FMT14,#NOCLK ;REPORT 'SYSTEM CLOCK IS NOT AVAILABLE'
(8) 011326 012746 004530 MOV #NOCLK,-(SP)
(7) 011332 012746 007544 MOV #FMT14,-(SP)
(6) 011336 012746 000002 MOV #2,-(SP)
(3) 011342 010600 MOV SP,R0
(4) 011344 104417 TRAP C$PNTF
(4) 011346 062706 000006 ADD #6,SP
772 011352          PRINTF #FMT14,#NOREPT ;PRINT 'PERFORMANCE REPORTS WILL NOT BE PRINTED'
(8) 011352 012746 004566 MOV #NOREPT,-(SP)
(7) 011356 012746 007544 MOV #FMT14,-(SP)
(6) 011362 012746 000002 MOV #2,-(SP)
(3) 011366 010600 MOV SP,R0
(4) 011370 104417 TRAP C$PNTF
(4) 011372 062706 000006 ADD #6,SP
773          ;POWER FAIL SEQUENCE
774 011376          PWRCH: REAF #EF.PWR          ;POWER FAILURE?
(3) 011376 012700 000034 MOV #EF.PWR,R0
(3) 011402 104447 TRAP C$REFG
775 011404          BNCOMPLETE 3$          ;BRANCH IF NO POWER FAILURE
(2) 011404 103121 BCC 3$
776 011406 005237 002446 INC PWRFLG          ;INDICATE POWER FAIL
777 011412 012704 030432 MOV #DRBUF,R4          ;INITIALIZE POINTER TO DRIVE PARAMETER BUFFERS
778 011416 012702 000001 MOV #1,R2
779 011422 130237 002252 11$: BITB R2,DRUT
780 011426 001471 BEQ 13$
781 011430 016400 000106 MOV DRSEL(R4),R0
782 011434 052700 000200 BIS #200,R0
783 011440 010074 000104 MOV R0,@DCS(R4)
784 011444 012701 000170 MOV #120.,R1
785 011450 032774 000001 000104 12$: BIT #1,@DCS(R4)
786 011456 001037 BNE 15$
787 011460          WAITMS #10.          ;IMPLEMENT 1 SECOND TIME DELAY

```

```

(3) 011476 012727 000372      MOV    ##250.,(PC)+
(3) 011502 000000      .WORD 0
(3) 011504 013727 002116      MOV    L$DLY,(PC)+
(3) 011510 000000      .WORD 0
(3) 011512 005367 177772      DEC    -6(PC)
(3) 011516 001375      BNE    -4
(3) 011520 005367 177756      DEC    -22(PC)
(3) 011524 001367      BNE    -20
788 011534 005301      DEC    R1
789 011536 001344      BNE    12$
790 011540 012737 004101 002246  MOV    #NOPWR,WHY      ;MSG. 'DR DID REC'R FROM PWR UP'
791 011546 004537 023520      JSR    R5,DRDRV
792 011552 000137 011612      JMP    13$
793
794 011556 004537 024446      15$: JSR    R5,ISDRST
795 011562 004537 025670      JSR    R5,HDHOME
796 011566 005064 000056      CLR    PRFLGS(R4)
797 011572 005064 000036      CLR    RETRY(R4)
798 011576 005064 000076      CLR    DOWCK(R4)
799 011602 005064 000052      CLR    RTYPE(R4)
800 011606 005064 000114      CLR    RSEEK(R4)
801 011612 062704 000126      13$: ADD    #PRPOS+2,R4
802 011616 106302      ASLB   R2
803 011620 103300      BCC    11$
804 011622 005737 002502      TST    SYSCLK      ;SYSTEM CLOCK AVAILABLE?
805 011626 001406      BEQ    4$
806 011630      CLKON
807 011640      REQTIM R0      ;ACTIVATE CLOCK WITH 1-SECOND INCREMENTS
808 011644 000137 012674      JMP    INIEND      ;REQUEST ELAPSED SUPERVISOR TIME
809
810 011650      .CONTINUE'' COMMAND SEQUENCE
811 (3) 011650 012700 000036      3$: READEF #EF.CONTINUE      ;CONTINUE FROM CONSOLE?
812 (3) 011654 104447      MOV    #EF.CONTINUE,R0
813 011656 103004      TRAP   C$REFG
814 (2) 011656 103004      BNCOMPLETE 1$      ;NO, CONTINUE W/ INIT CODE
815      BCC    1$
816 011660 005237 002454      INC    CNTFLG      ;YES SET CONT FLAG, GO TO END OF INIT
817 011664 000137 012222      JMP    END
818
819 011670 004537 027174      1$: JSR    R5,CLEAR      ;CLEAR ALL DRIVE BUFFERS
820 011674 012737 176543 002260  MOV    #176543,HINUM      ;PRIME RANDOM GENERATOR
821 011702 012737 123456 002262  MOV    #123456,LONUM
822 011710 012700 002320      2$: MOV    #CNTLR1,R0      ;INITIALIZE POINTER TO GLOBAL DATA AREA
823 011714 005020      CLRDAT: CLR    (R0)+      ;MASS CLEAR OF GLOBAL DATA AREA
824 011716 020027 002454      CMP    R0,#STFLG+2      ;AT END OF GLOBAL DATA AREA?
825 011722 001374      BNE
826
827 011724 012704 030432      MOV    #DRBUF,R4      ;SET UP DRIVE INFORMATION BUFFER POINTER
828 011730 012702 027364      MOV    #BSECO,R2      ;SET UP BAD SECTOR POINTER
829 011734 013703 002012      MOV    L$UNIT,R3      ;GET NUMBER OF UNITS
830 011740 010337 002444      MOV    R3,UUT      ;SAVE L$UNIT
831 011744 005001      CLR    R1      ;INITIALIZE P-TABLE FOR LOGICAL UNIT
832 011746 005703      1$: TST    R3      ;ANY P-TABLES LEFT?
833 011750 001524      BEQ    END      ;NO,GO TO END
834 (3) 011752 010100      GPHARD R1,R0      ;REQUEST A P-TABLE FOR DRIVE
      MOV    R1,R0

```



```
(3) 011754 104442          TRAP      C$GPHRD
832 011756          BNCOMPLET 12$
(2) 011756 103112          BCC      12$
833          ;MOVE P-TABLE CONTENTS TO LOCAL STORAGE
834 011760 012037 002330      MOV      (R0)+,BCSR          ;GET CSR
835 011764 012037 002332      MOV      (R0)+,BVEC         ;GET VECTOR
836 011770 012037 002334      MOV      (R0)+,BPRIOR       ;GET PRIORITY
837 011774 012037 002254      MOV      (R0)+,T.DRIVE      ;GET DRIVE TYPE
838 012000 011037 002336      MOV      (R0),BDRSEL        ;GET DRIVE NUMBER
839 012004 005737 002320      TST      CNTRL1            ;DO WE HAVE CSR 1 YET?
840 012010 001011          BNE      2$                ;YES, THEN SEE IF IT THIS DRIVE IS
841          ;/ASSOCIATED WITH CNTRL1
842 012012 013737 002334 002376      MOV      BPRIOR,PRIOR1
843 012020 013737 002330 002320      MOV      BCSR,CNTRL1        ;NO, MAKE THIS ONE CSR 1
844 012026 013737 002332 002372      MOV      BVEC,VECT1         ;MAKE THIS VECTOR VECT1
845 012034 023737 002330 002320 2$:   CMP      BCSR,CNTRL1        ;IS THIS CSR CNTRL1?
846 012042 001012          BNE      5$                ;NO, GO CHECK AGAINST #2
847 012044 023737 002332 002372      CMP      BVEC,VECT1         ;IS VECTOR PROPER?
848 012052 001050          BNE      10$               ;NO, REPORT ERROR
849 012054 012737 002436 002350      MOV      #BUF1,TEMP1        ;FIRST CONTROLLER/FIRST BUFFER
850 012062 004537 013474          JSR      R5,FILINF          ;FILL BUFFER
851 012066 000450          BR       11$               ;GO GET NEXT P-TABLE
852 012070 005737 002322          5$:   TST      CNTRL2            ;HAVE WE GOT CSR #2 YET?
853 012074 001015          BNE      6$                ;YES, CHECK THIS ONE AGAINST IT
854 012076 023737 002372 002330      CMP      VECT1,BCSR         ;IS THIS VECTOR SAME AS CNTRL1
855 012104 001433          BEQ     10$                ;IF SO, DON'T ALLOW IT
856 012106 013737 002330 002322      MOV      BCSR,CNTRL2        ;MAKE THIS ONE CSR 2
857 012114 013737 002332 002374      MOV      BVEC,VECT2         ;SETUP SECOND VECTOR
858 012122 013737 002334 002400      MOV      BPRIOR,PRIOR2
859 012130 023737 002330 002322 6$:   CMP      BCSR,CNTRL2        ;IS THIS CSR # 2?
860 012136 001016          BNE      10$                ;NO, WELL WE DON'T ALLOW 3
861 012140 023737 002332 002374      CMP      BVEC,VECT2         ;DOES IT HAVE PROPER VECTOR
862 012146 001012          BNE      10$                ;NO, GO REPORT ERROR
863 012150 023737 002374 002372      CMP      VECT2,VECT1        ;IS VECTOR OF FIRST EQUAL TO
864 012156 001406          BEQ     10$                ;VECTOR OF SECOND, YES REPORT ERROR
865 012160 012737 002440 002350      MOV      #BUF2,TEMP1        ;OTHER CNTRLR/OTHER BUFFER
866 012166 004537 013474          JSR      R5,FILINF          ;LOAD BUFFER
867 012172 000406          BR       11$                ;NEXT
868 012174          10$:  ERRDF 160.,ILLEG,ERR10 ;BAD P-TABLE
(4) 012174 104455          TRAP      C$ERDF
(5) 012176 000240          .WORD    160
(5) 012200 003770          .WORD    ILLEG
(5) 012202 005646          .WORD    ERR10
869 012204 005064 000104          12$:  CLR      DCS(R4)
870 012210 005201          11$:  INC      R1                ;POINT TO NEXT
871 012212 005303          DEC      R3                ;DOWN COUNT
872 012214 062702 000042          ADD     #34.,R2            ;NEXT BAD SECTOR FILE
873 012220 000652          BR       1$                ;DO WHILE
874
875
876 012222          END:
877
878 012222 012737 177770 002256      MOV     #177770,SYSMSK      ;SETUP FOR EIGHT DRIVES
879 012230 023727 002444 000004      CMP     UUT,#4              ;MORE THAN FOUR
880 012236 003012          BGT     2$                ;YES, THEN MASK IS OKAY
881 012240 052737 000004 002256      BIS     #4,SYSMSK          ;SETUP FOR FOUR DRIVES
```

```

882 012246 023727 002444 000002      CMP      UUT,#2      ;MORE THAN TWO
883 012254 003003                BGT      2$          ;YES, IT'S OKAY
884 012256 052737 000002 002256      BIS      #2,SYSMSK  ;SET FOR ONE OR TWO
885
886                                     ;'START' COMMAND SEQUENCE
887 012264                2$:      READEF #EF.START      ;START COMMAND
(3) 012264 012700 000040      MOV      #EF.START,RO
(3) 012270 104447                TRAP     C$REFG
888 012272                BNCOMplete RESTART      ;NO, CHK RESTART
(2) 012272 103006                BCC      RESTART
889 012274 005237 002452      INC      STFLG      ;SET START INDICATOR
890 012300 005037 002274      CLR      WRINIT     ;CLEAR THE WRITE INIT FLAG ON START
891 012304 005037 002310      CLR      KILLDC     ;CLEAR DATA COMP FLAG ON START ONLY
892
893 RESTART:
894 012310 005737 002454      TST      CNTFLG     ;CONTINUING
895 012314 001047                BNE      3$          ;YES GO TO 3$
896 012316 005737 002274      TST      WRINIT     ;IN PROCESS OF INITTING THE PACK?
897 012322 001420                BEQ      11$         ;NO
898 012324 005037 002274      CLR      WRINIT     ;YES - CLEAR THE FLAG
899 012330 005237 002310      INC      KILLDC     ;INHIBIT DATA COMPARES!
900 012334 005037 010674      CLR      CMRD       ;AND SET DAT COMPARE TO 0 WORDS
901 012340                PRINTF #FMT18,#NORDDC ;TELL OPR PACK NOT INITTED YET
(8) 012340 012746 004416      MOV      #NORDDC,-(SP)
(7) 012344 012746 007747      MOV      #FMT18,-(SP)
(6) 012350 012746 000002      MOV      #2,-(SP)
(3) 012354 010600                MOV      SP,RO
(4) 012356 104417                TRAP     C$PNTF
(4) 012360 062706 000006      ADD      #6,SP
902
903                                     ;LET'S CREATE INTERNAL BITMAP
904
905 012364 012701 000001      11$:     MOV      #1,R1      ;BIT MASK
906 012370 105037 002253      CLRB     DRPRS      ;CLEAR OUT DRIVES PRESENT
907 012374 012704 030432      MOV      #DRBUF,R4  ;START OF DRIVE BUFFERS
908 012400 005764 000104      1$:     TST      DCS(R4)    ;ANY CSR?
909 012404 001402                BEQ      2$          ;NO, NO DRIVE THEN
910 012406 150137 002253      BISB     R1,DRPRS   ;INDICATE DRIVE IN BITMAP
911 012412 006301                ASL      R1          ;NEXT POSITION
912 012414 062704 000126      ADD      #PRPOS+2,R4 ;NEXT DRIVE BUFFER
913 012420 022704 031712      CMP      #ENDBUF,R4 ;DONE
914 012424 001365                BNE      1$         ;NO
915
916 012426 113737 002253 002252      MOVB     DRPRS,DRUT ;SET UP DRIVES UNDER TEST
917
918 012434                3$:
919
920 012434                SETVEC  VECT1,#INTR1,PRIOR1 ;SET CONTROLLER 1'S VECTOR
(7) 012434 013746 002376      MOV      PRIOR1,-(SP)
(6) 012440 012746 017116      MOV      #INTR1,-(SP)
(5) 012444 013746 002372      MOV      VECT1,-(SP)
(4) 012450 012746 000003      MOV      #3,-(SP)
(3) 012454 104437                TRAP     C$SVEC
(2) 012456 062706 000010      ADD      #10,SP
921
922 012462 005737 002322      TST      CNTLR2     ;RUNNING TWO CONTROLLERS?

```



```

923 012466 001413          BEQ    4$          ;NO
924
925 012470          SETVEC  VECT2,#INTR2,PRIOR2 ;YES SET CONTROLLER 2'S VECTOR
(7) 012470 013746 002400  MOV    PRIOR2,-(SP)
(6) 012474 012746 017126  MOV    #INTR2,-(SP)
(5) 012500 013746 002374  MOV    VECT2,-(SP)
(4) 012504 012746 000003  MOV    #3,-(SP)
(3) 012510 104437          TRAP   C$$SVEC
(2) 012512 062706 000010  ADD    #10,SP
926
927 012516 005737 002454  4$:   TST    CNTFLG          ;CONTINUE?
928 012522 001412          BEQ    FINDBF          ;NO, GO PAST RESTART OF CLOCK
929
930 012524 005737 002502          TST    SYSCLK          ;DO WE HAVE SYSTEM CLOCK?
931 012530 001461          BEQ    INIEND          ;NO
932
933 012532          CLKON          ;ACTIVATE SYSTEM CLOCK
934 012542          REQTIM R0          ;REQUEST ELAPSED SUPERVISOR TIME
935 012546 000452          BR     INIEND          ;GO TO END
936
937          ;REQUEST MEMORY BUFFER SPACE TO PERFORM READ/WRITE OPERATIONS
938 012550          FINDBF: MEMORY R2          ;REQUEST MEMORY BUFFER SPACE
(3) 012550 104431          TRAP   C$MEM
(3) 012552 010002          MOV    R0,R2
939 012554 022712 002400          CMP    #1280.,(R2)          ;DO WE HAVE A MINIMUM OF 1280 WORDS?
940 012560 003413          BLE   1$          ;YES - BRANCH
941 012562          PRINTF #FMT14,#INSMEM          ;NO - PRINT MSG. 'SYSTEM FATAL ERROR -
(8) 012562 012746 004711          MOV    #INSMEM,-(SP)
(7) 012566 012746 007544          MOV    #FMT14,-(SP)
(6) 012572 012746 000002          MOV    #2,-(SP)
(3) 012576 010600          MOV    SP,R0
(4) 012600 104417          TRAP   C$PNTF
(4) 012602 062706 000006          ADD    #6,SP
942
943 012606 000000          ;/INSUFFICIENT MEMORY BUFFER SPACE''
944 012610 010237 002436  1$:   MOV    R2,BUF1          ;GET ADDRESS OF FREE MEMORY
945 012614 005737 002322          TST    CNTLR2          ;TWO CONTROLLERS?
946 012620 001410          BEQ    2$          ;NO - ASSIGN ALL BUFFER TO SINGLE CONTROLLER
947 012622 042712 000001          BIC    #1,(R2)          ;MAKE LENGTH OF FREE MEMORY EVEN
948 012626 013737 002436 002440          MOV    BUF1,BUF2          ;SET UP FOR BUFFER 2
949 012634 061237 002440          ADD    (R2),BUF2          ;ADD HALF OF BUFFER
950 012640 006212          ASR    (R2)          ;DIVIDE BUFFER SPACE BY 2
951 012642 011237 002442          MOV    (R2),MAXWC          ;INITIALIZE MAXIMUM WORD COUNT
952 012646 023727 002442 012000          CMP    MAXWC,#5120.          ;IS WORD COUNT LESS THAN OR EQUAL TO 5120?
953 012654 003403          BLE   3$          ;BRANCH IF TRUE
954 012656 012737 012000 002442          MOV    #5120.,MAXWC          ;NO - INITIALIZE VALUE TO 5120 WORDS
955
956 012664          3$:   CLKON          ;ACTIVATE SYSTEM CLOCK TO INITIATE GENERATION
957          ;/OF TIMING INTERVALS
958 012674          INIEND:
959 012674          ENDINIT
(3) 012674          L10017:
(3) 012674 104411          TRAP   C$INIT
960 012676          ENDMOD
961

```

```

963
964
965
966
967
968
969
970
971
972 012676
973 012676 010346
974 012700 010446
975 012702 013703 002012
976 012706 012704 030432
977 012712 005037 002450
978 012716
(7) 012716 012746 000340
(6) 012722 012746 014016
(5) 012726 013746 002466
(4) 012732 012746 000003
(3) 012736 104437
(2) 012740 062706 000010
979
980 012744 005774 000104
981 012750 005737 002450
982 012754 001425
983 012756
(9) 012756 005046
(9) 012760 156416 000107
(8) 012764 016446 000104
(7) 012770 012746 007705
(6) 012774 012746 000003
(3) 013000 010600
(4) 013002 104417
(4) 013004 062706 000010
984
985 013010 012737 004673 002246
986
987 013016 004537 023520
988 013022 005064 000104
989 013026 000436
990 013030 056474 000106 000104 2$:
991 013036 052774 000200 000104
992 013044 032774 000001 000104
993 013052 001024
994 013054
(9) 013054 005046
(9) 013056 156416 000107
(8) 013062 016446 000104
(7) 013066 012746 007705
(6) 013072 012746 000003
(3) 013076 010600
(4) 013100 104417
(4) 013102 062706 000010
995
996 013106 012737 004636 002246
  
```

.SBTTL AUTO DROP SECTION

:THE AUTO DROP SECTION IS CONDITIONALLY EXECUTED AFTER THE INITIALIZATION CODE
 :WHEN THE OPERATOR 'ADR' FLAG IS SET. EACH DRIVE IS CHECKED TO DETERMINE IF IT
 :IS READY TO TRANSFER DATA. IF THE DRIVE DOES NOT RESPOND WITH 'READY' IT IS
 :DROPPED FROM THE TEST CYCLE. THE HARDWARE TESTS ARE PERFORMED IMMEDIATELY
 :AFTER THE READY STATUS OF ALL DRIVES HAVE BEEN CHECKED.

BGNAUTO

```

MOV R3,-(SP) ;SAVE REGISTERS
MOV R4,-(SP)
MOV L$UNIT,R3 ;INITIALIZE NUMBER OF DRIVES UNDER TEST
MOV #DRBUF,R4 ;INITIALIZE START OF DRIVE BUFFERS
1$: CLR TRPFLG ;CLEAR TRAP FLAG
SETVEC ERRVEC,#TRPHAN,#340 ;SET UP TIME-OUT VECTOR TO DETECT
MOV #340,-(SP)
MOV #TRPHAN,-(SP)
MOV ERRVEC,-(SP)
MOV #3,-(SP)
TRAP C$SVEC
ADD #10,SP

; /NON-EXISTENT CONTROLLER
TST @DCS(R4) ;ACCESS CONTROLLER
TST TRPFLG ;DID TRAP OCCUR?
BEQ 2$ ;BRANCH TO CHECK DRIVE IF TRAP DID NOT OCCUR
PRINTF #FRMT16,DCS(R4),<B,DRSEL+1(R4)> ;PRINT CONTROL STATUS AND DRIVE
CLR -(SP)
BISB DRSEL+1(R4),(SP)
MOV DCS(R4),-(SP)
MOV #FRMT16,-(SP)
MOV #3,-(SP)
MOV SP,R0
TRAP C$PNTF
ADD #10,SP

; /NUMBER INFORMATION
MOV #NOCTLR,WHY ;PROVIDE REASON FOR DROPPING DRIVE -
; /'NO CONTROLLER'
JSR R5,DRDRV ;DO DROP UNIT ON DRIVE FROM TEST CYCLE
CLR DCS(R4) ;TAKE DRIVE OUT OF BUFFER
BR 3$ ;BRANCH TO GET NEXT DRIVE
2$: BIS DRSEL(R4),@DCS(R4) ;GET SELECTED DRIVE NUMBER
BIS #200,@DCS(R4) ;SET CONTROLLER READY
BIT #1,@DCS(R4) ;IS DRIVE READY?
BNE 3$ ;BRANCH TO CHECK NEXT DRIVE IF READY
PRINTF #FRMT16,DCS(R4),<B,DRSEL+1(R4)> ;PRINT CONTROL STATUS AND DRIVE
CLR -(SP)
BISB DRSEL+1(R4),(SP)
MOV DCS(R4),-(SP)
MOV #FRMT16,-(SP)
MOV #3,-(SP)
MOV SP,R0
TRAP C$PNTF
ADD #10,SP

; /NUMBER INFORMATION
MOV #NOTRDY,WHY ;PROVIDE REASON FOR DROPPING DRIVE -
  
```



```
997  
998 013114 004537 023520  
999 013120 005064 000104  
1000 013124 013700 002466  
   (3) 013124 013700 002466  
   (3) 013130 104436  
1001 013132 062704 000126  
1002  
1003 013136 005303  
1004 013140 001264  
1005 013142 012604  
1006 013144 012603  
1007 013146  
   (3) 013146  
   (3) 013146 104461  
1008  
1009  
1010
```

```
3$: JSR R5,DRDRV ;/'DID NOT RESPOND WITH 'READY''  
    CLR DCS(R4) ;DO DROP UNIT ON DRIVE FROM TEST CYCLE  
    CLRVEC ERRVEC ;TAKE DRIVE OUT OF BUFFER  
    MOV ERRVEC,R0 ;RELEASE THE ERROR VECTOR  
    TRAP C$CVEC  
    ADD #PRPOS+2,R4 ;UPDATE POINTER TO ACCESS DRIVE BUFFER  
    DEC R3 ;/FOR NEXT DRIVE  
    BNE 1$ ;DECREMENT DRIVE COUNT  
    MOV (SP)+,R4 ;BRANCH TO GET NEXT DRIVE IF MORE  
    MOV (SP)+,R3 ;RESTORE REGISTERS  
ENDAUTO  
L10020: TRAP C$AUTO
```

```

1012
1013 013150          BGNMOD  CLNCODE
1014
1015
1016 013150          BGNCLN
1017
1018 013150          SETVEC  ERRVEC,#TRPHAN,#340
(7) 013150 012746 000340  MOV      #340,-(SP)
(6) 013154 012746 014016  MOV      #TRPHAN,-(SP)
(5) 013160 013746 002466  MOV      ERRVEC,-(SP)
(4) 013164 012746 000003  MOV      #3,-(SP)
(3) 013170 104437          TRAP     C$$SVEC
(2) 013172 062706 000010  ADD      #10,SP
1019 013176          SETPRI  #PRI00          ;PRIORITY TO ZERO
(3) 013176 012700 000000  MOV      #PRI00,R0
(3) 013202 104441          TRAP     C$$SPRI
1020
1021 013204 032777 000200 167106 1$:  BIT      #CRDY,@CNTLR1          ;WAIT FOR CONTROLLER TO FINISH
1022 013212 001774          BEQ      1$
1023 013214 042777 000100 167076          BIC      #INTEN,@CNTLR1          ;CLEAR INTERRUPT IF PENDING
1024 013222          CLRVEC  VECT1          ;RELEASE VECTOR OF FIRST CONTROLLER
(3) 013222 013700 002372  MOV      VECT1,R0
(3) 013226 104436          TRAP     C$$CVEC
1025
1026 013230 005737 002322          TST     CNTLR2          ;TWO CONTROLLERS
1027 013234 001412          BEQ      3$          ;NO
1028
1029 013236 032777 000200 167056 2$:  BIT      #CRDY,@CNTLR2          ;WAIT FOR OTHER CONTROLLER TO FINISH
1030 013244 001774          BEQ      2$
1031 013246 042777 000100 167046          BIC      #INTEN,@CNTLR2          ;CLEAR OUT INTERRUPT ENABLE
1032 013254          CLRVEC  VECT2          ;YES, WELL RELEASE ITS VECTOR
(3) 013254 013700 002374  MOV      VECT2,R0
(3) 013260 104436          TRAP     C$$CVEC
1033
1034 013262 005037 002476          CLR     INCALL          ;
1035 013266 005037 002474          CLR     OPCALL          ;
1036 013272          CLRVEC  ERRVEC          ;
(3) 013272 013700 002466  MOV      ERRVEC,R0
(3) 013276 104436          TRAP     C$$CVEC
1037 013300 005737 002502          TST     SYSCLK
1038 013304 001416          BEQ      4$
1039 013306          CLKOFF          ;DEACTIVATE SYSTEM CLOCK
1040 013342          BRESET          ;TAKE CARE OF LSI-11
(3) 013342 104433          TRAP     C$$RESET
1041 013344          ENDCLN
(3) 013344          L10021: TRAP     C$$CLEAN
(3) 013344 104412
1042
1043 013346          ENDMOD
1044
1045
1046 013346          BGNMOD  ADDCODE
1047
1048 013346          BGNAU
1049
1050 013346 012704 030432          MOV      #DRBUF,R4          ;START OF DRIVE BUFFERS
  
```



```
1051 013352 012701 000001      MOV      #1,R1      ;MASK TO FIND DRIVE
1052 013356 010002      MOV      R0,R2      ;SAVE WHICH TO FIND
1053 013360 005700      1$:    TST      R0      ;THIS ONE
1054 013362 001405      BEQ      2$          ;YES
1055 013364 062704 000126      ADD      #PRPOS+2,R4 ;NEXT
1056 013370 006301      ASL      R1          ;NEXT MASK
1057 013372 005300      DEC      R0
1058 013374 000771      BR       1$
1059 013376 150137 002252      2$:    BISB    R1,DRUT  ;INSERT IN DRIVE UNDER TEST
1060 013402      GPHARD  R2,R1
      (3) 013402 010200      MOV      R2,R0
      (3) 013404 104442      TRAP    C$GPHRD
      (3) 013406 010001      MOV      R0,R1
1061 013410 011164 000104      MOV      (R1),DCS(R4)
1062 013414 012700 000100      MOV      #SERNM1,R0 ;SETUP TO CLEAR STATUS
1063 013420 006200      ASR      R0
1064 013422 005024      4$:    CLR      (R4)+
1065 013424 005300      DEC      R0
1066 013426 001375      BNE     4$
1067 013430      5$:
1068
1069 013430      ENDAU
      (3) 013430      L10022:
      (3) 013430 104452      TRAP    C$AU
1070
1071 013432      ENDMOD
1072
1073 013432      BGNMOD  DROPCODE
1074
1075 013432      BGNDU
1076
1077 013432 005737 002476      TST      INCALL
1078 013436 001015      BNE     3$
1079 013440 012704 030432      MOV      #DRBUF,R4
1080 013444 005700      2$:    TST      R0
1081 013446 001404      BEQ     1$
1082 013450 005300      DEC     R0
1083 013452 062704 000126      ADD     #PRPOS+2,R4
1084 013456 000772      BR      2$
1085
1086 013460 012737 003510 002246 1$:    MOV     #REQ,WHY
1087 013466 004537 023514      JSR     R5,ODRDRV
1088 013472      3$:
1089
1090
1091 013472      ENDDU
      (3) 013472      L10023:
      (3) 013472 104453      TRAP    C$DU
1092
1093 013474      ENDMOD
1094
```

```
1096  
1097  
1098  
1099 013474  
1100  
1101  
1102  
1103 013474 013764 002336 000106  
1104 013502 022737 000001 002254  
1105 013510 001403  
1106 013512 012737 000002 002254  
1107 013520 013764 002254 000120  
1108 013526 013764 002330 000104  
1109 013534 013764 002350 000110  
1110 013542 010264 000112  
1111 013546 062704 000126  
1112 013552 000205  
1113
```

.SBTTL GLOBAL SUBROUTINES
BGNMOD GLBSUB
;ROUTINE TO FILL DRIVE PARAMETER BUFFERS WITH INFORMATION

FILINF:	MOV	BDRSEL,DRSEL(R4)	;SET DRIVE SELECT BITS
	CMP	#1,T.DRIVE	;DRIVE = RL01?
	BEQ	FILTD	;YES
	MOV	#2,T.DRIVE	;DRIVE IS AN RL02
FILTD:	MOV	T.DRIVE,TDR(R4)	
	MOV	BCSR,DCS(R4)	;SET CSR
	MOV	TEMP1,BBA(R4)	;SET R/W BUFFER
	MOV	R2,BSECPT(R4)	;SETUP BAD SECTOR POINTER
	ADD	#PRPOS+2,R4	;UPDATE POINTER
	RTS	R5	


```

1115 ;SETS UP CLOCK INTERRUPT VECTOR, CLOCK COUNT, AND IDENTIFIES CLOCK FREQUENCY
1116
1117 013554 010346 CLKINI: MOV R3,-(SP) ;SAVE R3
1118 013556 022737 000001 002314 CMP #1,CLKTYP ;P-CLOCK?
1119 013564 001014 BNE LCLK ;BRANCH IF NOT P-CLOCK
1120 013566 SETVEC #104,#UPDATE,#340 ;SET P-CLOCK INTERRUPT VECTOR
(7) 013566 012746 000340 MOV #340,-(SP)
(6) 013572 012746 016706 MOV #UPDATE,-(SP)
(5) 013576 012746 000104 MOV #104,-(SP)
(4) 013602 012746 000003 MOV #3,-(SP)
(3) 013606 104437 TRAP CSSVEC
(2) 013610 062706 000010 ADD #10,SP
1121 013614 000417 BR FRQCHK ;BRANCH FOR SYSTEM FREQUENCY CHECK
1122 013616 022737 000002 002314 LCLK: CMP #2,CLKTYP ;L-CLOCK?
1123 013624 001036 BNE ENDINI ;BRANCH IF NO CLOCLK
1124 013626 SETVEC #100,#UPDATE,#340 ;SET L-CLOCK INTERRUPT VECTOR
(7) 013626 012746 000340 MOV #340,-(SP)
(6) 013632 012746 016706 MOV #UPDATE,-(SP)
(5) 013636 012746 000100 MOV #100,-(SP)
(4) 013642 012746 000003 MOV #3,-(SP)
(3) 013646 104437 TRAP CSSVEC
(2) 013650 062706 000010 ADD #10,SP
1125 013654 013703 002316 FRQCHK: MOV CLKADR,R3 ;GET BASE ADDRESS OF THE SUPERVISOR CLOCK TABLE
1126 013660 022763 000074 000006 CMP #60,6(R3) ;60 HZ?
1127 013666 001007 BNE FRQ50 ;BRANCH FOR 50 HZ
1128 013670 012737 000074 002506 MOV #60.,CLKCNT ;INITIALIZE CLOCK COUNT FOR 60 TICKS
1129 ;/PER SECOND
1130 013676 012737 000001 002312 MOV #1,CLKFRQ ;IDENTIFY CLOCK FREQUENCY IS 60 HZ
1131 013704 000406 BR ENDINI ;RETURN
1132 013706 012737 000062 002506 FRQ50: MOV #50.,CLKCNT ;INITIALIZE CLOCK COUNT FOR 50 TICKS
1133 ;/PER SECOND
1134 013714 012737 000002 002312 MOV #2,CLKFRQ ;IDENTIFY CLOCK FREQUENCY IS 50 HZ
1135 013722 012603 ENDINI: MOV (SP)+,R3 ;RESTORE R3
1136 013724 000207 RTS PC
1137
1138
1139 ;DETERMINES CLOCK TYPE AND INITIALIZES THE CLOCK FOR OPERATION IN REPEAT
1140 ;INTERRUPT MODE AT LINE FREQUENCY
1141
1142 013726 005037 002512 CLKST: CLR CLKACC ;CLEAR CLOCK ELAPSED TIME INDICATOR
1143 013732 022737 000002 002314 CMP #2,CLKTYP ;L-CLOCK?
1144 013740 001006 BNE 1$ ;BRANCH FOR P-CLOCK
1145 013742 012737 000100 177546 MOV #100,@#177546 ;SET INTERRUPT ENABLE BIT TO 1
1146 013750 005237 002504 INC CLKSON ;INDICATE 'CLOCK ON'
1147 013754 000414 BR 2$ ;BRANCH TO SET UP TIME INCREMENTS
1148 013756 022737 000001 002314 1$: CMP #1,CLKTYP ;P-CLOCK?
1149 013764 001013 BNE 3$ ;BRANCH IF NO CLOCK
1150 013766 012737 000001 172542 MOV #1,@#172542 ;SET UP P-CLOCK FOR 1 INTERRUPT PER TICK
1151 013774 012737 000115 172540 MOV #115,@#172540 ;SET INTERRUPT ENABLE,REPEAT INTERRUPT MODE,
1152 ;/LINE FREQUENCY RATE,START CLOCK
1153 014002 005237 002504 INC CLKSON ;INDICATE 'CLOCK ON'
1154 014006 013737 002506 002510 2$: MOV CLKCNT,CLKBFR ;SET UP TIME INCREMENTS
1155 014014 000207 3$: RTS PC ;RETURN
1156
1157
1158 014016 005237 002450 TRPHAN: INC TRPFLG

```

CZRLKBO RL01/02 PERF EXER
CZRLKB.MAC 07-DEC-79 09:46

MACY11 30A(1052) 17-DEC-79
GLOBAL SUBROUTINES

11:31 C 6 PAGE 1-32

SEQ 0067

1159 014022 000002

RTI

1161
1162
1163
1164 014024
(15) 014024 005046
(15) 014026 156416 000107
(14) 014032 012746 004030
(13) 014036 016446 000104
(12) 014042 012746 002525
(11) 014046 013746 002412
(10) 014052 013746 002414
(9) 014056 013746 002416
(8) 014062 012746 002516
(7) 014066 012746 007337
(6) 014072 012746 000011
(3) 014076 010600
(4) 014100 104416
(4) 014102 062706 000024
1165 014106
(8) 014106 016446 000120
(7) 014112 012746 010164
(6) 014116 012746 000002
(3) 014122 010600
(4) 014124 104416
(4) 014126 062706 000006
1166 014132 005764 000070
1167 014136 001417
1168
1169
1170
1171 014140
(9) 014140 005046
(9) 014142 156416 000071
(8) 014146 005046
(8) 014150 156416 000070
(7) 014154 012746 010116
(6) 014160 012746 000003
(3) 014164 010600
(4) 014166 104416
(4) 014170 062706 000010
1172 014174 000410
1173
1174 014176
(7) 014176 012746 010076
(6) 014202 012746 000001
(3) 014206 010600
(4) 014210 104416
(4) 014212 062706 000004
1175
1176 014216
(10) 014216 016446 000100
(9) 014222 016446 000102
(8) 014226 012746 002634
(7) 014232 012746 010151
(6) 014236 012746 000004
(3) 014242 010600

:SBTTL REPORT ROUTINE
:ROUTINE TO PRINT STATISTICAL REPORT OF DRIVE(S)
REPORT: PRINTS #FMT10,#TIME,HOUR,MINUTE,SECOND,#MRLCS,DCS(R4),#DRNM,<B,DRSEL+1(R4)>
CLR -(SP)
BISB DRSEL+1(R4),(SP)
MOV #DRNM,-(SP)
MOV DCS(R4),-(SP)
MOV #MRLCS,-(SP)
MOV SECOND,-(SP)
MOV MINUTE,-(SP)
MOV HOUR,-(SP)
MOV #TIME,-(SP)
MOV #FMT10,-(SP)
MOV #11,-(SP)
MOV SP,R0
TRAP C\$PNTS
ADD #24,SP
PRINTS #FMTDT,TDR(R4)
MOV TDR(R4),-(SP)
MOV #FMTDT,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C\$PNTS
ADD #6,SP
TST DPHOUR(R4) ;DO WE HAVE ANY DROPPED TIME
BEQ 1\$;NO, THEN PRINT 'RUNNING'

:PRINT THE TIME THE DRIVE WAS DROPPED FROM TESTING

PRINTS #FMST1B,<B,DPHOUR(R4)>,<B,DPMIN(R4)>
CLR -(SP)
BISB DPMIN(R4),(SP)
CLR -(SP)
BISB DPHOUR(R4),(SP)
MOV #FMST1B,-(SP)
MOV #3,-(SP)
MOV SP,R0
TRAP C\$PNTS
ADD #10,SP
BR 2\$

1\$: PRINTS #FMST1A ;PRINT '*** RUNNING'
MOV #FMST1A,-(SP)
MOV #1,-(SP)
MOV SP,R0
TRAP C\$PNTS
ADD #4,SP

2\$: PRINTS #FMST2,#CART,SERNM2(R4),SERNM1(R4)
MOV SERNM1(R4),-(SP)
MOV SERNM2(R4),-(SP)
MOV #CART,-(SP)
MOV #FMST2,-(SP)
MOV #4,-(SP)
MOV SP,R0

(4)	014244	104416		TRAP	C\$PNTS
(4)	014246	062706	000012	ADD	#12,SP
1177	014252			PRINTS	#FMTS2A,SKCNT(R4),SKCNT1(R4),RXFR3(R4),RXFR2(R4),RXFR1(R4)
(12)	014252	016446	000002	MOV	RXFR1(R4),-(SP)
(11)	014256	016446	000004	MOV	RXFR2(R4),-(SP)
(10)	014262	016446	000060	MOV	RXFR3(R4),-(SP)
(9)	014266	016446	000054	MOV	SKCNT1(R4),-(SP)
(8)	014272	016446	000000	MOV	SKCNT(R4),-(SP)
(7)	014276	012746	010215	MOV	#FMTS2A,-(SP)
(6)	014302	012746	000006	MOV	#6,-(SP)
(3)	014306	010600		MOV	SP,R0
(4)	014310	104416		TRAP	C\$PNTS
(4)	014312	062706	000016	ADD	#16,SP
1178	014316			PRINTS	#FMTS2B,WXFR3(R4),WXFR2(R4),WXFR1(R4)
(10)	014316	016446	000006	MOV	WXFR1(R4),-(SP)
(9)	014322	016446	000010	MOV	WXFR2(R4),-(SP)
(8)	014326	016446	000062	MOV	WXFR3(R4),-(SP)
(7)	014332	012746	010304	MOV	#FMTS2B,-(SP)
(6)	014336	012746	000004	MOV	#4,-(SP)
(3)	014342	010600		MOV	SP,R0
(4)	014344	104416		TRAP	C\$PNTS
(4)	014346	062706	000012	ADD	#12,SP
1179	014352			PRINTS	#FMTS3,DERCNT(R4),SKECNT(R4),TRERR(R4),DATCER(R4)
(11)	014352	016446	000074	MOV	DATCER(R4),-(SP)
(10)	014356	016446	000072	MOV	TRERR(R4),-(SP)
(9)	014362	016446	000016	MOV	SKECNT(R4),-(SP)
(8)	014366	016446	000020	MOV	DERCNT(R4),-(SP)
(7)	014372	012746	010341	MOV	#FMTS3,-(SP)
(6)	014376	012746	000005	MOV	#5,-(SP)
(3)	014402	010600		MOV	SP,R0
(4)	014404	104416		TRAP	C\$PNTS
(4)	014406	062706	000014	ADD	#14,SP
1180	014412			PRINTS	#FMTS3A,ERRCNT(R4),SFTCNT(R4)
(9)	014412	016446	000014	MOV	SFTCNT(R4),-(SP)
(8)	014416	016446	000012	MOV	ERRCNT(R4),-(SP)
(7)	014422	012746	010450	MOV	#FMTS3A,-(SP)
(6)	014426	012746	000003	MOV	#3,-(SP)
(3)	014432	010600		MOV	SP,R0
(4)	014434	104416		TRAP	C\$PNTS
(4)	014436	062706	000010	ADD	#10,SP
1181	014442			PRINTS	#FMTS4,DCRCER(R4),HRCRCER(R4),NXMCNT(R4),HNFERR(R4)
(11)	014442	016446	000032	MOV	HNFERR(R4),-(SP)
(10)	014446	016446	000034	MOV	NXMCNT(R4),-(SP)
(9)	014452	016446	000024	MOV	HRCRCER(R4),-(SP)
(8)	014456	016446	000022	MOV	DCRCER(R4),-(SP)
(7)	014462	012746	010505	MOV	#FMTS4,-(SP)
(6)	014466	012746	000005	MOV	#5,-(SP)
(3)	014472	010600		MOV	SP,R0
(4)	014474	104416		TRAP	C\$PNTS
(4)	014476	062706	000014	ADD	#14,SP
1182	014502			PRINTS	#FMTS5,DLTCNT(R4),OPICNT(R4)
(9)	014502	016446	000030	MOV	OPICNT(R4),-(SP)
(8)	014506	016446	000026	MOV	DLTCNT(R4),-(SP)
(7)	014512	012746	010600	MOV	#FMTS5,-(SP)
(6)	014516	012746	000003	MOV	#3,-(SP)
(3)	014522	010600		MOV	SP,R0

CZRLKBO RL01/02 PERF EXER
CZRLKB.MAC 07-DEC-79 09:46

MACY11 30A(1052) 17-DEC-79 11:31 F 6 PAGE 1-35
REPORT ROUTINE

SEQ 0070

(4) 014524 104416
(4) 014526 062706 000010
1183 014532 000207
1184
1185
1186 014534
1187

TRAP C\$PNTS
ADD #10,SP
RTS PC

ENDMOD

```
1189 .SBTTL PROGRAM MAIN LOOP
1190 014534 BGNTST
1191 014534 STARS
(2) ;*****
1192 ;PROGRAM WILL RANDOMLY PICK ONE OF THE DRIVES TO
1193 ;PERFORM AN OPERATION. WE WILL ALWAYS PICK ONE OF FOUR
1194 ;OR EIGHT DRIVES (ONE OR TWO CONTROLLERS) 'DRUT' WILL BE
1195 ;CHECKED TO SEE IF DRIVE IS ON SYSTEM. ONCE DRIVE IS PICKED
1196 ;THEN A FUNCTION WILL BE SELECTED RANDOMLY FOR THAT
1197 ;DRIVE. FUNCTIONS OF CONTROLLER RESET, GET STATUS, SEEK, READ, WRITE
1198 ;WILL BE SELECTED, EACH FUNCTION WILL HAVE ITS OWN ROUTINE
1199 ;TO GET PARAMETERS FOR THE DRIVE.
1200 014534 STARS
(2) ;*****
1201
1202 014534 MTEST: SETPRI #240 ;PRIORITY TO 5 TO ALLOW CLOCK INTERRUPTS
(3) 014534 012700 000240 MOV #240,R0
(3) 014540 104441 TRAP C$SPRI
1203 ;/AND TO INHIBIT DRIVE INTERRUPTS
1204
1205 014542 005737 002274 TST WRINIT ;HERE AFTER PWR FAIL DURING WRITE
1206 014546 001407 BEQ 161$ ;NO
1207 014550 013704 002274 MOV WRINIT,R4 ;YES - RESET R4
1208 014554 013701 002276 MOV WRPOS,R1 ;AND R1 POINTERS
1209 014560 005237 002452 INC STFLG ;FAKE OUT THE START FLAG
1210 014564 000410 BR 16$ ;AND CONTINUE WRITE INIT CODE
1211 014566 012704 030432 161$: MOV #DRBUF,R4 ;GET DRIVE BUFFERS
1212 014572 012701 000001 MOV #1,R1 ;MASK
1213 014576 010437 002274 MOV R4,WRINIT ;COPY THE R4 AND
1214 014602 010137 002276 MOV R1,WRPOS ;POINTERS
1215
1216 014606 130137 002252 16$: BITB R1,DRUT ;DRIVE UNDER TEST
1217 014612 001442 BEQ 15$ ;NO
1218
1219 014614 012774 000200 000104 MOV #200,@DCS(R4) ;CHECK IF DRIVE THERE
1220 014622 056474 000106 000104 BIS DRSEL(R4),@DCS(R4)
1221 014630 012700 000000 MOV #0.,R0 ;STALL
1222 014634 005300 13$: DEC R0
1223 014636 001376 BNE 13$
1224 014640 032774 000001 000104 BIT #DRDY,@DCS(R4) ;WAIT FOR DRIVE TO BECOME 'READY'
1225 014646 001006 BNE 14$ ;AFTER THE HEADS HOME COMMAND
1226
1227 014650 012737 002664 002246 MOV #DNRDY,WHY ;MSG. 'DRIVE NOT READY'
1228 014656 004537 023520 JSR R5,DRDRV
1229 014662 000416 BR 15$
1230
1231 014664 004537 022654 14$: JSR R5,RDBDSC ;GO GET BAD SECTORS
1232 014670 005064 000056 CLR PRFLGS(R4)
1233 014674 005064 000114 CLR RSEEK(R4)
1234 014700 005764 000122 TST WRIPG(R4) ;SEE IF WRITE IN PROGRESS FLAG SET
1235 014704 001003 BNE 99$ ;JUMP IF SET
1236 014706 005737 002452 TST STFLG
1237 014712 001402 BEQ 15$
1238
1239 014714 004537 024622 99$: JSR R5,WRPACK
1240
```



```

1241 014720 062704 000126      15$:  ADD      #PRPOS+2,R4      ;NEXT DRIVE
1242 014724 010437 002274      MOV      R4,WRINIT          ;SAVE CURRENT R4 POINTER
1243 014730 006337 002276      ASL      WRPOS              ;AND SHIFT COPY OF R1 POINTER
1244 014734 106301              ASLB     R1                  ;DONE?
1245 014736 103323              BCC     16$                 ;NO GO FOR NEXT ONE
1246
1247
1248
1249 014740 005037 002274      ;HERE WHEN ALL FINISHED WITH THE WRITE INIT CODE
1250 014744              12$:  CLR      WRINIT              ;CLEAR THE WRITE INIT FLAG
      PRINTF #FMT14,#MSTART    ;MSG. 'TESTING STARTED'
      MOV      -#MSTART,-(SP)
      MOV      #FMT14,-(SP)
      MOV      #2,-(SP)
      MOV      SP,R0
      TRAP    C$PNTF
      ADD     #6,SP
      SETPRI  #0              ;PRIORITY TO 0 TO ALLOW BOTH
      MOV     #0,R0
      TRAP    C$SPRI
1251 014770              ;/CLOCK AND DRIVE INTERRUPTS
      (3) 014770 012700 000000
      (3) 014774 104441
1252
1253
1254 014776 004537 024524      MAIN:  JSR      R5,RAND        ;GET A DRIVE?(LUN)
1255 015002 013702 002262      MOV     LONUM,R2           ;GET THE SELECTED DRIVE (LUN)
1256 015006 043702 002256      PEROTH: BIC     SYMSK,R2     ;MASK TO DRIVES ON SYSTEM
1257 015012 012701 000001      MOV     #1,R1             ;LET'S SEE IF DRIVE IS THERE
1258 015016 005702              1$:   TST     R2            ;HAVE WE GOT PROPER MASK YET
1259 015020 001403              BEQ     2$                ;YES, GO TO 2$
1260 015022 006301              ASL     R1                 ;NO, SHIFT FOR NEXT DRIVE
1261 015024 005302              DEC     R2                 ;DECREMENT DRIVE NUMBER
1262 015026 000773              BR      1$                 ;GO CHECK NEW DRIVE NUMBER
1263 015030 105737 002252      2$:   TSTB   DRUT           ;ANY DRIVES ON LINE
1264 015034 001006              BNE     5$                 ;YES, CHECK
1265
1266 015036              ERRSF  170,,NODRIV        ;NO DRIVES
      (4) 015036 104454      TRAP   C$ERSF
      (5) 015040 000252      .WORD 170
      (5) 015042 004016      .WORD NODRIV
      (5) 015044 000000      .WORD 0
1267
1268 015046 000137 030424      JMP     ENDOFPROGRAM
1269
1270 015052 130137 002252      5$:   BITB   R1,DRUT        ;IS THIS DRIVE PRESENT?
1271 015056 001747              BEQ     MAIN              ;NO, GO BACK TRY AGAIN
1272 015060 010137 002250      MOV     R1,TSTDRV        ;COPY UNIT UNDER TEST FOR LATER CHECK
1273
1274
1275
1276
1277 015064 023737 002406 010672      ;WE NOW HAVE A DRIVE, CHECK TO SEE IF ITS CONTROLLER
1278 015072 002403              ;IS FREE BEFORE WE GO ANY FURTHER
1279 015074 005037 002406      CMP     INTERVAL,TYINT    ;TIME FOR STATISTICAL REPORT?
1280
1281
1282 015100              BLT     6$                ;NO, PERFORM FUNCTION
      (3) 015100 104424      CLR     INTERVAL         ;CLEAR INTERVAL TO INITIALIZE TIME INTERVAL
1283
                                ;/BEFORE THE NEXT STATISTICAL REPORT
                                DORPT  ;PRINT STATISTICAL REPORT
                                TRAP   C$DRPT

```

```

1284 015102 012704 030432      6$:  MOV    #DRBUF,R4      ;GET START OF DRIVE BUFFERS
1285 015106 013702 002262      MOV    LONUM,R2      ;GET RANDOM DRIVE BACK (LUN)
1286 015112 043702 002256      BIC    SYMSK,R2      ;MASK TO SYSTEM SYS
1287 015116 005702      3$:  TST    R2          ;DO WE HAVE BUFFER FOR THAT DRIVE
1288 015120 001404      BEQ    4$          ;YES, GO CHECK ITS CONTROLLER
1289 015122 062704 000126      ADD    #PRPOS+2,R4   ;NO, UPDATE FOR NEXT BUFFER
1290 015126 005302      DEC    R2          ;DOWN COUNT DRIVE NUMBER (LUN)
1291 015130 000772      BR     3$          ;GO BACK AND CHECK FOR FOUND
1292 015132 032774 000200 000104 4$:  BIT    #BIT7,@DCS(R4) ;CONTROLLER ASSOCIATED WITH DRIVE
1293 015140 001716      BEQ    MAIN        ;BUSY
1294 015142 032774 000100 000104  BIT    #BIT6,@DCS(R4) ;INTERRUPT BEEN SERVICED?
1295 015150 001312      BNE    MAIN        ;NO - WAIT FOR THE INTERRUPT
1296
1297
1298      ;WE CAN NOW PROCEED IN GETTING A FUNCTION AND RELATED DATA
1299      ;FOR THE DRIVE RANDOMLY. R4 HAS DRIVE BUFFER POINTER
1300 015152      TAGX:
1301 015152 005737 010722      TST    T.DRP        ;DROP ON ERROR LIMITS REACHED?
1302 015156 001456      BEQ    GETFNC       ;NO
1303 015160 026437 000012 010662  CMP    ERRCNT(R4),ERLMT ;HARD REACHED?
1304 015166 103404      BLO   9$
1305 015170 012737 003322 002246  MOV    #ERLMTM,WHY
1306 015176 000442      BR     11$
1307 015200 026437 000014 010726  9$:  CMP    SFTCNT(R4),SFLMT ;SOFT REACHED?
1308 015206 103404      BLO   10$
1309 015210 012737 003365 002246  MOV    #SFEMSG,WHY
1310 015216 000432      BR     11$
1311 015220 026437 000074 010752 10$:  CMP    DATCER(R4),T.DCD
1312 015226 103404      BLO   110$
1313 015230 012737 003407 002246  MOV    #DCDMSG,WHY
1314 015236 000422      BR     11$
1315 015240 016401 000016      110$: MOV    SKECNT(R4),R1
1316 015244 066401 000072      ADD    TRERR(R4),R1
1317 015250 020137 010664      CMP    R1,SELMT
1318 015254 103404      BLO   7$
1319 015256 012737 003344 002246  MOV    #SERLMT,WHY
1320 015264 000407      BR     11$
1321 015266 026437 000020 010732 7$:  CMP    DERCNT(R4),DRLMT ;DRIVE ERROR REACHED?
1322 015274 103407      BLO   GETFNC       ;NO - TIME TO DO SOMETHING
1323 015276 012737 003432 002246  MOV    #DERMSG,WHY
1324
1325 015304 004537 023520      11$: JSR    R5,DRDRV    ;DROP THIS DRIVE!!!
1326 015310 000137 014776      JMP    MAIN        ;GO GET ANOTHER

```



```
1328 ;HERE TO GET A 'STRING' FUNCTION - LIST OF COMMANDS TO ISSUE
1329
1330 015314 GETFNC:
1331 015314 005737 010730 8$: TST T.STA ;DO WE WISH TO DROP ON OPR LIMITS
1332 015320 001422 BEQ 98$ ;NO
1333
1334 015322 026437 000000 010670 CMP SKCNT(R4),SKLMT ;PAST THE SEEK LIMIT??
1335 015330 103416 BLO 98$ ;NO, THEN GO TEST
1336 015332 016400 000060 MOV RXFR3(R4),R0 ;GET READ COUNT
1337 015336 066400 000062 ADD WXFR3(R4),R0 ;ADD IN WRITE COUNT
1338 015342 020037 010666 CMP R0,DALMT ;LIMIT REACHED??
1339 015346 103407 BLO 98$ ;NO, THEN GO TEST
1340 015350 012737 003611 002246 MOV #SOPLMT,WHY
1341 015356 004537 023520 JSR R5,DRDRV ;DROP THE DRIVE
1342 015362 000137 014776 JMP MAIN ;GO FOR ANOTHER DRIVE
1343
1344 015366 004537 024524 98$: JSR R5,RAND ;GET A RANDOM FUNCTION INDEX NUMBER
1345 ;0 & 7 ARE NOT LEGIT
1346 015372 013702 002262 MOV LONUM,R2 ;GET IT
1347 015376 042702 177770 BIC #177770,R2 ;MASK TO 0-7
1348 015402 001001 BNE 6$ ;IF 0, MAKE 1
1349 015404 005202 INC R2
1350 015406 022702 000007 6$: CMP #7,R2 ;IS IT 7?
1351 015412 001001 BNE 5$ ;IF 7, MAKE 6
1352 015414 005302 DEC R2
1353 015416 006302 5$: ASL R2 ;SHIFT LEFT (X2)
1354 015420 000172 022636 JMP @LIST(R2) ;GO TO FUNCTION ROUTINE
1355
1356 015424 STARS
(2) ;:*****
1357 ;SKWRT -- ISSUE:
1358 ;SEEK TO A CYLINDER
1359 ;WRITE DATA
1360 ;WRITE CHECK
1361 015424 STARS
(2) ;:*****
1362
1363 015424 004537 015766 SKWRT: JSR R5,SKFNC ;RANDOM SEEK LOAD
1364 015430 004537 015540 JSR R5,OPROK ;WAIT TILL DONE
1365 015434 004537 016430 JSR R5,WRTFNC ;WRITE DATA LOAD
1366 015440 004537 015540 JSR R5,OPROK
1367 015444 004537 015724 JSR R5,WRTCKF ;WRITE CHECK LOAD
1368 015450 004537 015540 JSR R5,OPROK
1369 015454 000137 014776 JMP MAIN ;GET NEXT COMMAND
```

1371 015460
(2)
1372
1373
1374
1375 015460
(2)
1376
1377 015460 004537 015766
1378 015464 004537 015540
1379 015470 004537 016500
1380 015474 004537 015540
1381 015500 000137 014776
1382
1383 015504
(2)
1384
1385
1386
1387
1388 015504
(2)
1389
1390 015504 004537 015766
1391 015510 004537 015540
1392 015514 004537 016500
1393 015520 004537 015540
1394 015524 004537 016500
1395 015530 004537 015540
1396 015534 000137 014776

```
STARS
:*****
:SKRD -- ISSUE:
:      RANDOM SEEK TO A CYLINDER
:      READ DATA
STARS
:*****
SKRD:  JSR      R5,SKFNC      ;LOAD SEEK
        JSR      R5,OPROK
        JSR      R5,RDDFNC    ;LOAD READ DATA CMD
        JSR      R5,OPROK
        JMP      MAIN        ;GET THE NEXT COMMAND

STARS
:*****
:SKRDRD -- ROUTINE TO DO:
:      SEEK TO A CYLINDER
:      READ (AND COMPARE DATA)
:      READ (AGAIN)
STARS
:*****
SKRDRD: JSR      R5,SKFNC      ;LOAD SEEK
        JSR      R5,OPROK
        JSR      R5,RDDFNC    ;LOAD READ
        JSR      R5,OPROK
        JSR      R5,RDDFNC    ;LOAD READ
        JSR      R5,OPROK
        JMP      MAIN        ;EXIT
```



```
1398 015540 STARS  
(2) :*****  
1399 :OPROK -- ROUTINE TO ISSUE THE FUNCTION AND WAIT FOR 'READY'...IF AN  
1400 : ERROR RETRY IS NEEDED - THEN ISSUE THE FUNCTION AGAIN.  
1401 015540 STARS  
(2) :*****  
1402  
1403 015540 004537 016574 OPROK: JSR R5, LDFUNC ;ISSUE THE FUNCTION  
1404 015544 004537 024340 JSR R5, WTRDY ;WAIT TILL READY  
1405 015550 133737 002250 002252 BITB TSTDV, DRUT ;DRIVE STILL AVAILABLE?  
1406 015556 001003 BNE 1$ ;YUP - CONTINUE  
1407 015560 005726 TST (SP)+ ;NO - FIX THE STACK  
1408 015562 000137 014776 JMP MAIN ;BACK TO THE MAIN LOOP - FORCED EXIT FROM  
1409 ;THE STRING FUNCTION  
1410 015566 005764 000036 1$: TST RETRY(R4) ;NEED TO RETRY FUNCTION?  
1411 015572 001403 BEQ 3$ ;NO  
1412 015574 004537 016542 2$: JSR R5, ISSUE ;YES - ISSUE THE FUNCTION AGAIN  
1413 015600 000757 BR OPROK ;AND DO IT  
1414 015602 005764 000114 3$: TST RSEEK(R4) ;SEEK RETRY?  
1415 015606 001403 BEQ 4$ ;NO - EXIT NOW  
1416 015610 004537 015766 JSR R5, SKFNC ;DO A SEEK AGAIN  
1417 015614 000751 BR OPROK ;ISSUE & EXECUTE THE SEEK  
1418 015616 000205 4$: RTS R5 ;EXIT
```

```
1420 015620 STARS
(2) :*****
1421 :SKRH -- ISSUE:
1422 :RANDOM SEEK
1423 :READ HEADERS
1424 :READ DATA W/NO HDR CMP
1425 :GET STATUS
1426 015620 STARS
(2) :*****
1427
1428 015620 004537 015766 SKRH: JSR R5,SKFNC ;LOAD SEEK
1429 015624 004537 016574 JSR R5,LDFUNC ;ISSUE
1430 015630 004537 024340 JSR R5,WTRDY
1431 015634 004537 016416 JSR R5,RDHFNC ;LOAD READ HDRS
1432 015640 004537 016574 JSR R5,LDFUNC ;ISSUE
1433 015644 004537 024340 JSR R5,WTRDY
1434 015650 004537 015704 JSR R5,RDNHC ;LOAD READ W/NO HDRS
1435 015654 004537 016574 JSR R5,LDFUNC ;ISSUE
1436 015660 004537 024340 JSR R5,WTRDY
1437 015664 004537 015746 JSR R5,GSTFNC ;LOAD GET STATUS
1438 015670 004537 016574 JSR R5,LDFUNC ;ISSUE
1439 015674 004537 024340 JSR R5,WTRDY
1440 015700 000137 014776 JMP MAIN ;GET THE NEXT COMMAND
1441
1442 015704 STARS
(2) :*****
1443 :READ DATA W/NO HDR COMPARE
1444 015704 STARS
(2) :*****
1445
1446 015704 012764 177600 000042 RDNHC: MOV #-128.,BMP(R4) ;SET FOR A 1 SECTOR READ
1447 015712 012764 000016 000044 MOV #16,FUNC(R4) ;LOAD THE COMMAND
1448 015720 000137 016542 JMP ISSUE ;PROCESS IT
1449
1450 015724 STARS
(2) :*****
1451 :WRTCKF - WRITE CHECK FUNCTION
1452 015724 STARS
(2) :*****
1453
1454 015724 005737 010734 WRTCKF: TST T,ROF ;READ ONLY SET?
1455 015730 001401 BEQ 1$ ;NO - DO THE WRITE-CHECK FUNCTION
1456 015732 000205 RTS R5 ;YES - EXIT NOW
1457
1458 015734 012764 000002 000044 1$: MOV #WRCHK,FUNC(R4) ;SAVE CMD
1459 015742 000137 016542 JMP ISSUE ;PROCESS IT
```



```
1461 .SBTTL ROUTINES TO SETUP AND ISSUE GET STATUS & SEEK
1462 015746 STARS
(2) :*****
1463 :GET STATUS FUNCTION
1464 015746 STARS
(2) :*****
1465
1466 015746 012764 000004 000044 GSTFNC: MOV #GSTAT, FUNC(R4) ;LOAD GET STATUS
1467 015754 012764 000003 000040 MOV #GSBIT, BDA(R4) ;SET GSBIT IN COMMAND WORD
1468 015762 000137 016542 JMP ISSUE ;GO ISSUE FUNCTION
1469
1470 015766 STARS
(2) :*****
1471 :SEEK FUNCTION
1472 015766 STARS
(2) :*****
1473
1474 ;WE WILL CALL 'RAND' FOR A NEW DISK ADDRESS TO SEEK
1475 ;TO. ANY TRACK BUT LAST IS LEGAL. WE WILL ALSO INCREMENT
1476 ;ITS SEEK COUNT
1477
1478 015766 005764 000114 SKFNC: TST RSEEK(R4) ;TRYING TO RECOVER
1479 015772 001003 BNE 10$ ;YES - DO IT
1480 015774 005764 000036 TST RETRY(R4) ;RECOVERY FROM A 'DRIVE' ERROR?
1481 016000 001411 BEQ 98$ ;NO - NORMAL SEEK REQUIRED
1482 016002 016401 000050 10$: MOV LSTHDR(R4), R1 ;YES SET UP FOR RESEEK
1483 016006 016402 000124 MOV PRPOS(R4), R2 ;TO CYLINDER
1484 016012 042701 000100 BIC #100, R1 ;HEAD SET IN LATER
1485 016016 042702 000100 BIC #100, R2
1486 016022 000546 BR 4$ ;SKIP RANDOM PART
1487 016024 004537 024524 98$: JSR R5, RAND ;GET A RANDOM NUMBER
1488 016030 013702 002262 MOV LONUM, R2 ;GET THE RANDOM NUMBER
1489 016034 043702 002272 BIC SMSK, R2 ;LEAVE CYL AND HEAD
1490 016040 020264 000124 CMP R2, PRPOS(R4) ;ON THAT TRACK ALREADY
1491 016044 001767 BEQ 98$ ;YES - RESELECT
1492
1493 016046 022764 000001 000120 980$: CMP #1, TDR(R4) ;THIS DRIVE AN RL01?
1494 016054 001006 BNE 981$ ;NO - MUST BE AN RL02
1495 016056 042702 100000 BIC #BIT15, R2 ;KILL UPPER BIT OF CYL ADDRESS
1496 016062 022702 077700 CMP #077700, R2 ;POINTING TO THE BAD SEC FILE?
1497 016066 001007 BNE 96$ ;NO - PROCEED
1498 016070 000403 BR 982$ ;YUP - CORRECT THE POSITION
1499 016072 022702 177700 981$: CMP #177700, R2 ;RL02 BAD SECTOR FILE?
1500 016076 001003 BNE 96$ ;NO - PROCEED
1501 016100 000240 982$: NOP ;TRAP
1502 016102 042702 000100 BIC #HEAD, R2 ;POINT TO HEAD 0 LAST TRACK
1503
1504
1505 016106 010237 002342 96$: MOV R2, CHKSEC ;SAVE THE ADDRESS FOR THE BAD SEC FILE CHECK
1506 016112 004537 027274 JSR R5, CKBDTK ;SEE IF THIS ADDR IN BAD SECTOR FILE
1507 016116 005737 002340 TST HDRFND ;WAS IT?
1508 016122 001340 BNE 98$ ;YES - RESELECT THE ADDRESS
```


1511	016124	005003			90\$:	CLR	R3	
1512	016126	010200				MOV	R2,R0	:COPY ADDRESS - NO SECTOR YET
1513	016130	042700	177677			BIC	#177677,R0	:LEAVE ONLY HEAD
1514	016134	023737	010710	010712		CMP	T.MXC,T.MNC	:MIN AND MAX CYLINDERS THE SAME
1515	016142	001011				BNE	95\$:NO, BRANCH AND STAY IN LIMITS
1516	016144	013702	010710			MOV	T.MXC,R2	:MAKE CYLINDER MAX/MIN
1517	016150	022764	000001	000120		CMP	#1,TDR(R4)	:DRIVE = RL01?
1518	016156	001031				BNE	92\$:NO
1519	016160	042702	100000			BIC	#BIT15,R2	:FORCE CYL TO PROPER LIMIT
1520	016164	000426				BR	92\$:GO CALCULATE DIFF AND SEEK
1521	016166	042702	000100		95\$:	BIC	#HEAD,R2	:STRIP OUT H.S. BIT
1522	016172	023702	010710		94\$:	CMP	T.MXC,R2	:IS ADDRESS LESS/EQUAL THAN MAX
1523	016176	10301C				BHIS	93\$:YES, CHECK LOW END
1524	016200	005203				INC	R3	:BUMP A TALLY COUNTER
1525	016202	020327	000012			CMP	R3,#10.	:IF CAN'T FIND ADDRESS IN 10 TIMES THEN RESELECT
1526	016206	001706				BEQ	98\$:RESELECT
1527	016210	006202				ASR	R2	:HALF IT AND CHECK AGAIN
1528	016212	062702	000200		91\$:	ADD	#BIT7,R2	:JUST TO MAKE NON ZERO
1529	016216	000763				BR	95\$:GO BACK AND CHECK AGAIN
1530	016220	023702	010712		93\$:	CMP	T.MNC,R2	:IS MIN GREATER/EQUAL THAN ADDRESS
1531	016224	101406				BLOS	92\$:YES, CALCULATE DIFF AND SEEK
1532	016226	005203				INC	R3	
1533	016230	020327	000012			CMP	R3,#10.	:TIME TO RESELECT?
1534	016234	001673				BEQ	98\$:YUP - DO IT NOW
1535	016236	006302				ASL	R2	:NO, DOUBLE IT
1536	016240	000764				BR	91\$:GO CHECK MAX/MIN AGAIN
1537	016242	016401	000124		92\$:	MOV	PRPOS(R4),R1	:GET PRESENT DISK POSITION
1538	016246	042701	000177			BIC	#177,R1	
1539	016252	022764	000001	000120		CMP	#1,TDR(R4)	:RL01=1
1540	016260	001002				BNE	25\$:BRANCH...MUST BE RL02
1541	016262	042702	100000			BIC	#BIT15,R2	:CLEAR THE HIGH BIT FOR RL02 CYL #
1542	016266	016464	000124	000050	25\$:	MOV	PRPOS(R4),LSTHDR(R4)	
1543	016274	010264	000124			MOV	R2,PRPOS(R4)	:NEW HEADER AFTER SEEK
1544	016300	050064	000124			BIS	R0,PRPOS(R4)	:SET IN RANDOM HEAD GOTTEN
1545	016304	023737	010704	010706		CMP	T.MXH,T.MNH	:MIN AND MAX HEAD SELECT THE SAME
1546	016312	001012				BNE	4\$:NO, THEN WE CAN USE BOTH SURFACES
1547	016314	005737	010704			TST	T.MXH	:WHICH IS OUR SURFACE FOR USE
1548	016320	001004				BNE	97\$:TOP SURFACE BRANCH
1549	016322	042764	000100	000124		BIC	#HEAD,PRPOS(R4)	:LOWER SURFACE ONLY
1550	016330	000403				BR	4\$	
1551	016332	052764	000100	000124	97\$:	BIS	#HEAD,PRPOS(R4)	:TOP SURFACE ONLY

1553 016340
(2)
1554
1555 016340
(2)
1556
1557 016340 160102
1558 016342 103002
1559 016344 005402
1560 016346 000402
1561 016350 052702 000004
1562 016354 052702 000001
1563 016360 032764 000100 000124
1564 016366 001402
1565 016370 052702 000020
1566 016374 010264 000040
1567 016400 010264 000066
1568 016404 012764 000006 000044
1569 016412 000137 016542

```
STARS
:*****
:CALCULATE THE DIFFERENCE WORD AND STORE IT IN BDA
STARS
:*****
4$: SUB R1,R2 ;SUBTRACT PRESENT FROM NEXT
    BCC 1$ ;IF POSITIVE RESULT GO TO 1$
    NEG R2 ;NEG RESULT, NEGATE IT
    BR 2$ ;GO SET DIRECTION OUT
1$: BIS #SIGN,R2 ;DIRECTION OUT, MARKER
2$: BIS #MK,R2 ;MARKER BIT
    BIT #HEAD,PRPOS(R4) ;WHICH SURFACE SELECTED?
    BEQ 3$ ;TOP, THEN 3$
3$: BIS #SKHS,R2 ;BOTTOM SET HEAD BIT
    MOV R2,BDA(R4) ;MOVE DIFFERENCE WORD TO DA
    MOV R2,DIFWD(R4) ;LOAD DIFFERENCE WORD
    MOV #SEEK,FUNC(R4) ;LOAD SEEK
    JMP ISSUE
```

```

1571      .SBTTL ROUTINE TO LOAD READ HEADER AND ISSUE IT
1572
1573 016416 012764 000010 000044 RDHFNC: MOV #RDHDR, FUNC(R4) ;LOAD READ HEADER
1574 016424 000137 016542      JMP      ISSUE
1575
1576      .SBTTL ROUTINE TO LOAD WRITE DATA COMMAND
1577
1578 016430 005737 010734 WRTFNC: TST      T,ROF          ;READ ONLY
1579 016434 001021      BNE      RDDDFNC          ;YES
1580 016436 004537 025764      JSR      R5,GWCDA         ;GET WORD COUNT,DA
1581 016442 005737 010674      TST      CMRD            ;COMPARE DATA ON A READ?
1582 016446 001404      BEQ      1$              ;NO - SO DON'T GEN A WRITE BUFFER
1583 016450 005237 002306      INC      REGEN           ;YES - SET THE GENERATE DATA FLAG
1584
1585      ;WE NOW HAVE SECTOR AND WORD COUNT, LET'S WRITE BUFFER IN MEMORY
1586      ;TO WRITE OUT TO DISK
1587      ;FORMAT:          WORD 1 - # OF WORDS IN SECTOR
1588      ;                  :          WORD 2 - ADDRESS OF PATTERN WRITTEN ON SECTOR
1589      ;                  :          WORD 3 - 127 DATA PATTERN
1590      ;
1591
1592 016454 004537 022370      JSR      R5,WRBUF        ;WRITE BUFFER INTO MEMORY
1593 016460 012764 000012 000044 1$: MOV #WRITE, FUNC(R4) ;LOAD WRITE
1594 016466 012764 000001 000122      MOV #1,WRIPG(R4) ;SET THE WRITE IN PROGRESS FLAG
1595 016474 000137 016542      JMP      ISSUE          ;GO ISSUE FUNCTION
1596
1597      .SBTTL ROUTINE TO LOAD READ DATA COMMAND
1598
1599      ;THIS ROUTINE WILL FIRST CLEAR OUT THE BUFFER AREA,
1600      ;SELECT A RANDOM NUMBER OF WORDS TO READ AND A
1601      ;RANDOM SECTOR ON THE PRESENT CYLINDER TO READ FROM
1602
1603 016500 004537 025764 RDDDFNC: JSR      R5,GWCDA         ;GET WORD COUNT, DA
1604 016504 005737 010674      TST      CMRD            ;GOING TO COMPARE DATA AFTER READING?
1605 016510 001407      BEQ      2$              ;NO - SO SKIP THE CLEAR BUFFER CODE
1606 016512 016402 000042      MOV      BMP(R4),R2      ;CLEAR OUT BUFFER AREA
1607 016516 017401 000110      MOV      @BBA(R4),R1    ;SO WE KNOW READ
1608 016522 005021      1$: CLR      (R1)+        ;WORKED!!
1609 016524 005202      INC      R2
1610 016526 001375      BNE      1$
1611 016530 012764 000014 000044 2$: MOV #READ, FUNC(R4) ;LOAD READ
1612 016536 000137 016542      JMP      ISSUE

```



```
1614 .SBTTL SETUP CONTROLLER AND DRIVE INFO FOR INTERRUPT PROCESSING
1615
1616 ;WE COME HERE BEFORE ISSUING ANY FUNCTION SO THAT ON INTERRUPT
1617 ;WE CAN PROPERLY PROCESS THE INTERRUPT. WE WILL CHECK WHICH
1618 ;CONTROLLER WE ARE WORKING WITH AND STORE OFF THE DRIVE BUFFER
1619 ;POINTER IN ITS 'LSTDR'
1620 ;
1621
1622 016542 026437 000104 002320 ISSUE: CMP DCS(R4),CNTLR1 ;DRIVE ON CONTROLLER 1?
1623 016550 001003 BNE 1$ ;NO, ASSUME ON CONTROLLER 2
1624 016552 010437 002324 MOV R4,LSTDR1 ;PUT BUFFER POINTER IN 1
1625 016556 000402 BR 2$ ;SKIP OVER NEXT INSTRUCTION
1626 016560 010437 002326 1$: MOV R4,LSTDR2 ;PUT BUFFER POINTER IN 2
1627 016564 052764 000100 000044 2$: BIS #INTEN,FUNC(R4) ;ALLOW INTERRUPTS
1628 016572 000205 RTS R5 ;EXIT
1629
1630 .SBTTL ROUTINE TO LOAD FUNCTION
1631 016574 STARS
1632 (2) ;*****
1633 ;CALL JSR R5,LDFUNC
1634 ;ALL INFORMATION MUST BE SET UP IN DRIVE BUFFER
1635 016574 ;R4 HAS POINTER TO BUFFER
1636 (2) STARS
1637 ;*****
1637 016574 016403 000104 LDFUNC: MOV DCS(R4),R3 ;GET CSR FOR DRIVE
1638 016600 032713 000200 BIT #BIT7,(R3) ;CAN WE ISSUE COMMAND?
1639 016604 001004 BNE 1$ ;YES, GO ISSUE COMMAND
1640
1641 016606 ERRSF 200,PRGER ;THIS ERROR SHOULD NEVER PRINT
1642 (4) 016606 104454 TRAP C$ERSF
1643 (5) 016610 000310 .WORD 200
1644 (5) 016612 002732 .WORD PRGER
1645 (5) 016614 000000 .WORD 0
1646
1643 016616 017463 000110 000002 1$: MOV @BBA(R4),BA(R3) ;LOAD BUS ADDRESS REGISTER
1644 016624 016463 000040 000004 MOV BDA(R4),DA(R3) ;LOAD DISK ADDRESS REGISTER
1645 016632 016463 000042 000006 MOV BMP(R4),MP(R3) ;LOAD MULTI-PURPOSE REGISTER
1646 016640 016464 000044 000046 MOV FUNC(R4),BCSADR(R4) ;GET FUNCTION
1647 016646 056464 000106 000046 BIS DRSEL(R4),BCSADR(R4) ;SET DRIVE SELECT BITS
1648 016654 052764 000201 000046 BIS #CRDY!DRDY,BCSADR(R4) ;SET CRDY:DRDY IN IMAGE
1649 016662 042764 002000 000046 BIC #OPI,BCSADR(R4) ;WE'RE CLEAR BIT 10 FOR DRIVE 7-4 (OKAY?)
1650 016670 016463 000046 000000 MOV BCSADR(R4),CS(R3) ;LOAD CSR
1651 016676 042763 000200 000000 BIC #CRDY,CS(R3) ;ISSUE FUNCTION
1652 016704 000205 RTS R5 ;EXIT
1653
```

```
1655 .SBTTL INTERRUPT SERVICE ROUTINES
1656
1657 ;CLOCK INTERRUPT HANDLER
1658 ;UPDATES TIME EVERY 1/60 SECOND (60 HZ) OR EVERY 1/50 SECOND (50 HZ)
1659 016706 BGNSRV UPDATE
1660 016706 010446 MOV R4,-(SP) ;SAVE R4
1661 ;CLEAR CLOCK INTERRUPT ENABLE TO INHIBIT CLOCK INTERRUPTS DURING UPDATING
1662 ;OF TIME FIELDS
1663 016710 022737 000001 002314 CMP #1,CLKTYP ;P-CLOCK?
1664 016716 001004 BNE 1$ ;BRANCH IF NOT P-CLOCK
1665 016720 042737 000100 172540 BIC #100,@#172540 ;DISABLE P-CLOCK INTERRUPT FACILITY
1666 ;UPDATE TIME FIELDS
1667 016726 000403 BR 2$
1668 016730 042737 000100 177546 1$: BIC #100,@#177546 ;DISABLE L-CLOCK INTERRUPT FACILITY
1669 016736 012704 002410 2$: MOV #TICK,R4 ;INITIALIZE TICK ADDRESS
1670 016742 005214 INC (R4) ;INCREMENT TICK TIME FIELD
1671 016744 023727 002312 000002 CMP CLKFRQ,#2 ;50 HZ CLOCK?
1672 016752 001005 BNE 3$ ;NO--BRANCH FOR SERVICING 60 HZ CLOCK
1673 016754 021427 000062 CMP (R4),#50. ;((R4))=50?
1674 016760 001026 BNE EXIT2 ;IF NOT,UPDATING IS COMPLETE
1675 016762 005014 CLR (R4) ;ELSE,((R4))=0 (RESET COUNT)
1676 016764 000404 BR 4$ ;BRANCH TO UPDATE 'SECOND' TIME FIELD
1677 016766 021427 000074 3$: CMP (R4),#60. ;((R4))=60?
1678 016772 001021 BNE EXIT2 ;IF NOT,UPDATING IS COMPLETE
1679 016774 005014 CLR (R4) ;ELSE,((R4))=0 (RESET COUNT)
1680 016776 005724 4$: TST (R4)+ ;(R4)=(R4)+2 (GO TO NEXT TIME FIELD)
1681 017000 005214 INC (R4) ;INCREMENT 'SECOND' TIME FIELD
1682 017002 021427 000074 CMP (R4),#60. ;((R4))=60?
1683 017006 001013 BNE EXIT2 ;IF NOT,UPDATING IS COMPLETE
1684 017010 005237 002406 INC INTERVAL ;INCREMENT INTERVAL TIME FIELD (STORES
1685 ;/RUNNING TIME BETWEEN STATISTICAL REPORTS)
1686 017014 005014 CLR (R4) ;ELSE,((R4))=0 (RESET COUNT)
1687 017016 005724 TST (R4)+ ;ACCESS 'MINUTE' TIME FIELD
1688 017020 005214 INC (R4) ;INCREMENT 'MINUTE' TIME FIELD
1689 017022 021427 000074 CMP (R4),#60. ;((R4))=60?
1690 017026 001003 BNE EXIT2 ;IF NOT,UPDATING IS COMPLETE
1691 017030 005014 CLR (R4) ;ELSE,((R4))=0 (RESET COUNT)
1692 017032 005724 TST (R4)+ ;ACCESS 'HOUR' TIME FIELD
1693 017034 005214 INC (R4) ;INCREMENT 'HOUR' TIME FIELD
1694 017036 005337 002510 EXIT2: DEC CLKBFR ;COUNT CLOCK TICKS
1695 017042 003005 BGT 5$ ;TIME NOT EXPIRED
1696 017044 005237 002512 INC CLKACC ;BUMP ELAPSED TIME
1697 017050 013737 002506 002510 MOV CLKCNT,CLKBFR ;RE-INITIALIZE TIME INCREMENT
1698 ;RE-ENABLE CLOCK INTERRUPT FACILITY
1699 017056 022737 000001 002314 5$: CMP #1,CLKTYP ;P-CLOCK?
1700 017064 001004 BNE 6$ ;BRANCH IF NOT P-CLOCK
1701 017066 052737 000100 172540 BIS #100,@#172540 ;SET P-CLOCK INTERRUPT ENABLE BIT
1702 017074 000403 BR 7$ ;EXIT
1703 017076 052737 000100 177546 6$: BIS #100,@#177546 ;SET L-CLOCK INTERRUPT ENABLE BIT
1704 017104 012604 7$: MOV (SP)+,R4 ;RESTORE R4
1705 017106 ENDSRV
(3) 017106 L10025:
(2) 017106 000002 RTI
1706
```



```
1708 ;L-CLOCK 'TICK' CHECK ROUTINE FOR LSI-11
1709 017110 BGNSRV CLKTIK
1710
1711 017110 005237 002514 INC CLKFLD ;INCREMENT CLOCK FIELD TO INDICATE THAT
1712 ;/CLOCK IS 'TICKING'
1713
1714 017114 ENDSRV
(3) 017114 L10026:
(2) 017114 000002 RTI
1715
1716
1717
1718 017116 BGNSRV INTR1
1719
1720
1721
1722 ;ON INTERRUPT WE CHECK FOR ERRORS FIRST, IF NO ERRORS WE
1723 ;CHECK FUNCTION PERFORMED. WE ACT ACCORDING IF FUNCTION IS:
1724 ; 1- WRITE CHECK - NOTHING IF NO ERROR
1725 ; 2- GET STATUS - READ AND CHECK DRIVE STATUS
1726 ; 3- SEEK - NOTHING RTI; SET RD HDR AS NEXT COMMAND
1727 ; 4- RDHDR - COMPARE HEADER TO PRESENT POSITION
1728 ; 5- WRITE - UPDATE XFER COUNT, EXIT
1729 ; 6- READ - COMPARE DATA IF REQUESTED, UPDATE XFER COUNT, EXIT
1730 ; 7- READ W/NO HDR COMPARE - UPDATE XFER COUNT, EXIT
1731
1732 ;ALL SUCCESSFUL EXITS FROM INTERRUPT ROUTINE TEST RETRY
1733 ;LIMIT IF RETRY IS LESS THEN LIMIT THEN LOG SOFT ERROR, CLEAR RETRY
1734 ;IF RETRY = 0, THEN NOTHING
1735
1736 ;ON ERRORS - IF DRIVE ERROR - UNDER NON-INTERRUPT
1737 DO: GET STATUS - INVESTIGATE ERROR TYPE
1738
1739 DO: DRIVE RESET - IF ERROR OCCURS AGAIN - FATAL ERROR
1740 ; IF NO ERROR, EXIT
1741 DRIVE ERROR IS LOGGED UNDER ALL CIRCUMSTANCES
1742
1743 IF DCRC, HCRC, HNF CHECK BAD SECTOR LIST, IF IN LIST
1744 IGNORE ERROR EXIT AS NORMAL, IF NOT IN LIST
1745 INCREMENT RETRY; IF RETRY LIMIT EXCEEDED
1746 LOG HARD ERROR, ELSE RETRY FUNCTION
1747
1748 IF OPI,NXM INCREMENT RETRY CHECK RETRY LIMIT
1749 IF RETRY EXCEEDED LOG HARD ERROR EXIT
1750 IF RETRY NOT EXCEEDED RETRY FUNCTION
1751
1752
```

```
1754 017116 010446          INTR1: MOV R4,-(SP)      ;SAVE PRESENT R4 VALUE
1755 017120 013704 002324   MOV LSTDR1,R4      ;GET THE DRIVE BUFFER OF INTERRUPTING DRIVE
1756 017124 000403          BR SAVE           ;GO SAVE R0-R3
1757 017126 010446          INTR2: MOV R4,-(SP)      ;SAVE PRESENT R4 VALUE
1758 017130 013704 002326   MOV LSTDR2,R4      ;GET THE DRIVE BUFFER OF INTERRUPTING DRIVE
1759 017134 013746 002420   SAVE:  MOV E.CS,-(SP)
1760 017140 013746 002422   MOV E.BA,-(SP)
1761 017144 013746 002424   MOV E.DA,-(SP)
1762 017150 013746 002426   MOV E.MP,-(SP)
1763 017154 013746 002430   MOV E.MP1,-(SP)
1764 017160 013746 002432   MOV E.MP2,-(SP)
1765 017164 013746 002342   MOV CHKSEC,-(SP)
1766 017170 013746 002340   MOV HDRFND,-(SP)
1767 017174 013746 002350   MOV TEMP1,-(SP)
1768 017200 013746 002246   MOV WHY,-(SP)
1769 017204 013746 002474   MOV OPCALL,-(SP)
1770 017210 013746 002476   MOV INCALL,-(SP)
1771 017214 010346          MOV R3,-(SP)      ;SAVE R3
1772 017216 010246          MOV R2,-(SP)      ;R2
1773 017220 010146          MOV R1,-(SP)      ;R1
1774 017222 010046          MOV R0,-(SP)      ;R0
1775 017224 005064 000122   CLR WRIPG(R4)     ;CLEAR THE WRITE IN PROGRESS FLAG
1776 017230 016403 000104   MOV DCS(R4),R3    ;GET CSR FOR INTERRUPT
1777 017234 016337 000000 002420  MOV CS(R3),E.CS   ;SAVE ALL REGISTERS NOW!!
1778 017242 016337 000002 002422  MOV BA(R3),E.BA
1779 017250 016337 000004 002424  MOV DA(R3),E.DA
1780 017256 016337 000006 002426  MOV MP(R3),E.MP
1781 017264 016337 000006 002430  MOV MP(R3),E.MP1
1782 017272 016337 000006 002432  MOV MP(R3),E.MP2
1783 017300 005737 002420   TST E.CS          ;ANY ERRORS?
1784 017304 100402          BMI 1$           ;YES, GO SOLVE ERROR MYSTERY
1785 017306 000137 020432   JMP CHKFNC       ;NO, GO SEE IF WE HAVE TO DO ANYTHING
```



```

1787          .SBTTL  CONTROLLER ERROR CHECK ROUTINE
1788
1789          ;WE HAVE SOME SORT OF ERROR LET'S FIND OUT WHICH ONE
1790          ;IT IS.
1791
1792 017312 013764 002424 000064 1$:  MOV  E.DA,LSTDA(R4)  ;SAVE DA FOR SOFT ERROR PRINT
1793 017320 032737 040000 002420      BIT  #DERR,E.CS      ;DRIVE ERROR?
1794 017326 001402                BEQ  2$          ;NO, CONTINUE
1795 017330 000137 021420                JMP  CKDERR      ;YES, GO CHECK DRIVE ERROR
1796 017334 032737 000001 002420 2$:  BIT  #DRDY,E.CS      ;DRIVE READY THERE
1797 017342 001017                BNE  23$        ;YES, CONTINUE CHECKING
1798 017344 004537 024432                JSR  R5,GETDST   ;NO,GET DRIVE STATUS
1799 017350 042701 000100                BIC  #100,R1     ;GET RID OF HEAD
1800 017354 020127 000034                CMP  R1,#34     ;ALLOW ONLY SEEK TRACKING STATE
1801 017360 001410                BEQ  23$        ;WAS 34 SKIP ERROR
1802
1803 017362 005264 000012                INC  ERRCNT(R4)  ;INDICATE HARD ERROR
1804 017366                ERRDF 1000.,NORDY,ERR9
      (4) 017366 104455                TRAP C$ERDF
      (5) 017370 001750                .WORD 1000
      (5) 017372 002704                .WORD NORDY
      (5) 017374 005602                .WORD ERR9
1805
1806 017376 000137 021254                JMP  EXIT1
1807
1808 017402 032737 020000 002420 23$:  BIT  #NXM,E.CS      ;NON-EXISTENT MEMORY?
1809 017410 001407                BEQ  3$          ;NO, KEEP CHECKING
1810 017412 012764 004346 000052                MOV  #MTNXM,RTYPE(R4) ;ERROR MESSAGE
1811 017420 005264 000034                INC  NXMCNT(R4)    ;LOG ERROR
1812 017424 000137 020036                JMP  111$        ;CHECK RETRY, EXIT BACK
1813
1814 017430 032737 014000 002420 3$:  BIT  #BIT12!BIT11,E.CS ;QUALIFING BITS SET?
1815 017436 001020                BNE  5$          ;YES, CAN'T BE OPI ALONE
1816
1817 017440 032737 002000 002420                BIT  #OPI,E.CS    ;OPI SET?
1818 017446 001006                BNE  4$          ;YES, CONTINUE
1819
1820 017450                ERRSF 10.,UDERR,ERR1 ;WE HAVE AN UNDIAGNOSABLE CONDITION, ONLY COMPOSITE SET
      (4) 017450 104454                TRAP C$ERSF
      (5) 017452 000012                .WORD 10
      (5) 017454 003007                .WORD UDERR
      (5) 017456 005070                .WORD ERR1
1821 017460                33$:  BREAK
      (3) 017460 104422                TRAP C$BRK
1822 017462 000776                BR   33$
1823
1824
1825 017464 012764 004341 000052 4$:  MOV  #MTOPI,RTYPE(R4);SET UP FOR 'OPI' PRINT
1826 017472 005264 000030                INC  OPICNT(R4)   ;LOG ERROR
1827 017476 000557                BR   111$        ;CHECK RETRY EXIT BACK
1828
1829          ;WE KNOW IT'S NOW EITHER DLT, DCRC,HNF, OR HCRC
1830          ;CHECK FOR EACH
1831
1832 017500 032737 002000 002420 5$:  BIT  #OPI,E.CS    ;OPI QUALIFIER SET?
1833 017506 001060                BNE  7$          ;YES, THEN IT'S HCRC OR HNF
  
```

```

1834
1835 ;IT'S NOW DOWN TO DLT OR DCRC
1836
1837 017510 032737 010000 002420 BIT #DLT,E.CS ;DATA LATE?
1838 017516 001406 BEQ 6$ ;NO, MUST BE DATA CRC
1839 017520 012764 004334 000052 MOV #MTDLT,RTYPE(R4) ;SET UP FOR 'DLT' PRINT
1840 017526 005264 000026 INC DLT CNT(R4) ;LOG ERROR
1841 017532 000541 BR 111$ ;CHECK RETRY, EXIT
1842
1843 017534 013737 002424 002342 6$: MOV E.DA,CHKSEC ;SET UP SECTOR TO LOOK FOR
1844 017542 005364 000064 DEC LSTDA(R4) ;DOWN COUNT FOR PRINT OUT
1845 017546 005337 002342 DEC CHKSEC ;DOWN COUNT FOR LOOP UP
1846 017552 004537 027216 JSR R5,CKBDSC ;CHECK BAD SECTOR LIST
1847 017556 005737 002340 TST HDRFND ;WAS HEADER THERE?
1848 017562 001117 BNE 110$ ;IGNORE ERROR, RETURN
1849 017564 005264 000022 117$: INC DCR CER(R4) ;ACCOUNT FOR ERROR
1850 017570 012764 004327 000052 MOV #MTDCRC,RTYPE(R4) ;SET UP FOR 'DCRC' PRINT
1851 017576 022764 000102 000044 CMP #INTEN!WRCHK,FUNC(R4)
1852 017604 001001 BNE 118$
1853 017606 000513 BR 111$
1854
1855 017610 005737 010720 118$: TST T.DCK ;DUMP BUFFER?
1856 017614 001510 BEQ 111$ ;NO, EXIT
1857 017616 PRINTF #FMT14,#DMPDCK
(8) 017616 012746 003265 MOV #DMPDCK,-(SP)
(7) 017622 012746 007544 MOV #FMT14,-(SP)
(6) 017626 012746 000002 MOV #2,-(SP)
(3) 017632 010600 MOV SP,R0
(4) 017634 104417 TRAP C$PNTF
(4) 017636 062706 000006 ADD #6,SP
1858 017642 004537 026272 JSR R5,DMPBUF ;DUMP BUFFER
1859
1860 017646 000473 BR 111$ ;EXIT
1861
1862 ;IT'S NOW EITHER HNF OR HCRC.
1863 ;IF HCRC AND RDHDR, DETERMINE IF BAD SECTOR BY DOING 40 RDHDRS
1864 ;IF HCRC AND R/W, CHECK IF DA IS IN BAD SECTOR FILE
1865 ;IF HNF READ HEADER TO VERIFY IF ON CORRECT CYLINDER
1866 ;THEN IF ON CORRECT CYLINDER SEE IF DA IS A BAD SECTOR
1867 ;IF NOT ON CORRECT CYLINDER REPORT MISSEK, LOG MISEEK
1868 ;AND PRESENT POSITION UPDATE.
1869
1870 017650 032737 010000 002420 7$: BIT #HNF,E.CS ;HEADER NOT FOUND SET?
1871 017656 001470 BEQ 112$ ;NO IT MUST BE HCRC
1872 017660 012701 000051 MOV #41,R1 ;ALLOW FORTY READ HEADERS TO,
1873 017664 004537 024446 8$: JSR R5,ISDRST
1874 017670 016402 000106 MOV DRSEL(R4),R2 ;FIND CYLINDER
1875 017674 052702 000010 BIS #RDHDR,R2 ;READ HEADER
1876 017700 016403 000104 MOV DCS(R4),R3
1877 017704 010263 000000 MOV R2,CS(R3) ;ISSUE READ HEADER
1878 017710 004537 024340 JSR R5,WTRDY ;WAIT
1879 017714 005301 DEC R1 ;DONE 40 OF THESE?
1880 017716 001424 BEQ 9$ ;YES, GIVE UP WE DON'T HAVE ALL DAY!
1881 017720 005763 000000 TST CS(R3) ;IS ERROR SET?
1882 017724 100757 BMI 8$ ;YES, GO DO IT AGAIN
1883

```



```

1884 017726 016301 000006      MOV      MP(R3),R1      ;GET HEADER
1885 017732 010137 002434      MOV      R1,C.HDR     ;SAVE FOR ERROR REPORTING
1886 017736 043701 002272      BIC      SMSK,R1      ;MASK OUT SECTOR BITS
1887 017742 020164 000124      CMP      R1,PRPOS(R4) ;IS CYLINDER HEAD CORRECT?
1888 017746 001415      BEQ      10$          ;YES, GO CHECK BAD SECTOR LIST
1889
1890
1891 017750 005264 000072      INC      TRERR(R4)
1892 017754      ERRHRD  20.,TRACK,ERR2 ;TRACKING DRIFT ERROR
   (4) 017754 104456      TRAP    C$ERRHRD
   (5) 017756 000024      .WORD  20
   (5) 017760 003305      .WORD  TRACK
   (5) 017762 005076      .WORD  ERR2
1893
1894
1895 017764 000137 020750      JMP      SKRETRY      ;FIX TRACKING ERROR
1896
1897
1898 017770      9$:      ERRHRD  30.,EXHAUS,ERR1 ;WE CAN'T FIND GOOD HEADER ON THIS TRACK
   (4) 017770 104456      TRAP    C$ERRHRD
   (5) 017772 000036      .WORD  30
   (5) 017774 002773      .WORD  EXHAUS
   (5) 017776 005070      .WORD  ERR1
1899
1900 020000 000410      BR      110$
1901
1902 020002 013737 002424 002342 10$:      MOV      E.DA,CHKSEC
1903 020010 004537 027274      JSR      R5,CKBDTK    ;GO CHECK BAD SECTOR FILE
1904 020014 005737 002340      TST      HDRFND      ;WAS IT THERE
1905 020020 001401      BEQ      11$          ;NO, LOG IT EXIT
1906 020022 000577      110$:    BR      GOERRX    ;YES IGNORE ERROR
1907
1908 020024 005264 000032      11$:      INC      HNFERR(R4)   ;LOG IT
1909 020030 012764 004314 000052      MOV      #MTHNF,RTYPE(R4);SET UP FOR 'HNF' PRINT
1910 020036 000573      111$:    BR      GOFIN      ;EXIT
1911
1912
1913      ;
1914      ;IT WAS A HEADER CRC ERROR, FIGURE OUT IF IT WAS
1915      ;ON A READ HEADER OR READ/WRITE
1916      ;
1917
1918 020040 022764 000110 000044 112$:    CMP      #INTEN!RDHDR,FUNC(R4) ;READ HEADER?
1919 020046 001417      BEQ      13$          ;YES, GO FIND OUT MORE ABOUT IT
1920      ;NO, IT MUST BE R/W
1921 020050 013737 002424 002342      -MOV     E.DA,CHKSEC
1922 020056 004537 027216      JSR      R5,CKBDSC    ;BAD SECTOR SEARCH
1923 020062 005737 002340      TST      HDRFND      ;WAS OUR DA THERE?
1924 020066 001401      BEQ      12$          ;NO, MUST BE LEGIT ERROR
1925 020070 000554      BR      GOERRX      ;YES, IGNORE ERROR
1926
1927 020072 005264 000024      12$:      INC      HRCRCR(R4)   ;LOG ERROR
1928 020076 012764 004321 000052      MOV      #MTHCRC,RTYPE(R4)
1929 020104 000550      BR      GOFIN
1930
1931 020106 017401 000110      13$:      MOV      @BBA(R4),R1   ;USE IT'S BUFFER TO STORE HDRS
  
```

```

1932 020112 012737 000050 002350      MOV      #40.,TEMP1      ;40 CONSECUTIVE HEADERS
1933 020120 012702 000010      14$:    MOV      #RDHDR,R2      ;READ HEADER
1934 020124 056402 000106      BIS      DRSEL(R4),R2      ;
1935 020130 016403 000104      MOV      DCS(R4),R3      ;
1936 020134 010263 000000      MOV      R2,CS(R3)      ;
1937 020140 004537 024340      JSR      R5,WTRDY      ;WAIT FOR READY
1938 020144 016321 000000      MOV      CS(R3),(R1)+      ;READ ALL REGISTERS
1939 020150 016321 000006      MOV      MP(R3),(R1)+      ;
1940 020154 016321 000006      MOV      MP(R3),(R1)+      ;
1941 020160 016321 000006      MOV      MP(R3),(R1)+      ;
1942 020164 005337 002350      DEC      TEMP1      ;DONE 40 YET?
1943 020170 001353      BNE      14$      ;NO, GO BACK
1944
1945      ;WE HAVE 40 HEADERS NOW LETS SEE IF WE CAN VERIFY WHETHER
1946      ;OR NOT A BAD SECTOR CAUSED THE ERROR. CHECK FIRST TO SEE
1947      ;IF WE HAVE ANY BAD SECTORS ON THIS TRACK.
1948
1949 020172 017402 000110      99$:    MOV      @BBA(R4),R2      ;GET BUFFER START
1950 020176 012701 000050      MOV      #40.,R1      ;FORTY HEADERS
1951 020202 032712 002000      15$:    BIT      #OPI,(R2)      ;IS OPI SET IN CS
1952 020206 001403      BEQ      16$      ;NO, WELL CAN'T BE HCRC
1953 020210 032712 004000      BIT      #HCRC,(R2)      ;INSURE HCRC W/OPI
1954 020214 001005      BNE      17$      ;FOUND GO SEE IF IT COMPARES
1955 020216 062702 000010      16$:    ADD      #10,R2      ;NEXT CS IMAGE
1956 020222 005301      DEC      R1      ;DONE 40
1957 020224 001366      BNE      15$
1958 020226 000721      BR       12$
1959
1960 020230 020274 000110      17$:    CMP      R2,@BBA(R4)      ;IS HEADER FIRST ONE?
1961 020234 001046      BNE      21$      ;NO, READ PREVIOUS HEADER
1962      ;YES, WE'LL HAVE TO GO THRU
1963      ;AND CHECK OTHERS BEFORE WE
1964      ;CAN SAFELY CALCULATE
1965      ;'SUPPOSED' BAD SECTOR
1966 020236 017401 000110      MOV      @BBA(R4),R1
1967 020242 012703 000001      MOV      #1,R3
1968 020246 062701 000010      18$:    ADD      #10,R1
1969 020252 032711 002000      BIT      #OPI,(R1)
1970 020256 001416      BEQ      19$
1971 020260 032711 004000      BIT      #HCRC,(R1)
1972 020264 001413      BEQ      19$
1973 020266 005203      INC      R3
1974 020270 022703 000017      CMP      #15.,R3
1975 020274 001364      BNE      18$
1976
1977
1978 020276 012737 003667 002246      MOV      #MBDMSC,WHY      ;DROP DRIVE DUE TO
1979 020304 004537 023520      JSR      R5,DRDRV      ;MORE THAN 16 BAD SECTORS
1980 020310 000137 021254      JMP      EXIT1
1981
1982
1983 020314 005012      19$:    CLR      (R2)      ;CLEAR THIS CS
1984 020316 062701 000002      ADD      #2,R1      ;GET IT'S HEADER ADDRESS
1985 020322 011102      MOV      (R1),R2      ;GET HEADER
1986 020324 010201      MOV      R2,R1      ;SAVE HEADER
1987 020326 042702 177700      BIC      #177700,R2      ;MASK ONLY SECTOR

```


1988	020332	160301		SUB	R3,R1		:BACK UP TO SECTOR WHICH IS BAD
1989	020334	100402		BMI	20\$:IF MINUS DO MAGIC
1990	020336	160302		SUB	R3,R2		:NO THEN SUBTRACT IS LEGAL
1991	020340	000421		BR	22\$:BRANCH TO CHECK FILE
1992	020342	160302		20\$: SUB	R3,R2		:THIS SUB PRODUCES WRONG ANSWER
1993	020344	062702	000050	ADD	#50,R2		:FIX IT UP
1994	020350	000415		BR	22\$:GO CHECK FILE
1995							
1996	020352	005012		21\$: CLR	(R2)		:CLEAR THIS CS OUT
1997	020354	162702	000006	SUB	#6,R2		:GET PREVIOUS HEADER
1998	020360	011201		MOV	(R2), R1		
1999	020362	005201		INC	R1		
2000	020364	010102		MOV	R1,R2		
2001	020366	042701	177700	BIC	#177700,R1		
2002	020372	022701	000050	CMP	#40.,R1		
2003	020376	002402		BLT	22\$		
2004	020400	162702	000050	SUB	#40.,R2		
2005	020404	010237	002342	22\$: MOV	R2,CHKSEC		
2006	020410	004537	027216	JSR	R5,CKBDSC		
2007	020414	005737	002340	TST	HDRFND		
2008	020420	001664		BEQ	99\$		
2009	020422	000137	021260	GOERRX: JMP	ERREX		
2010							
2011							
2012	020426	000137	021362	GOF IN: JMP	FINERR		

```
2014 .SBTTL COMMAND SERVICE ROUTINES
2015
2016 ;THERE WAS NO ERROR SO.....
2017 ;NOW WE WILL FIND OUT WHICH FUNCTION WE DID TO CAUSE
2018 ;INTERRUPT AND ACT ACCORDINGLY.
2019 ;
2020
2021 020432 016401 000044 CHKFUNC: MOV FUNC(R4),R1 ;GET FUNCTION OF DRIVE
2022 020436 006201 ASR R1 ;ALIGN THE FUNCTION CODE
2023 020440 042701 000040 BIC #40,R1 ;WIPE OUT INT. ENAB (SHIFTED)
2024 020444 005301 DEC R1 ;WRITE CHECK??
2025 020446 001002 BNE 2$ ;NO, BRANCH
2026 020450 000137 020610 JMP AFWRCK ;FUNCTION #1
2027
2028 020454 005301 2$: DEC R1 ;GET STATUS?
2029 020456 001565 BEQ AGSTAT ;BRANCH IF SO...FUNCTION #2
2030 020460 005301 DEC R1 ;SEEK?
2031 020462 001421 BEQ ASEEK ;BRANCH IF SO...FUNCTION #3
2032 020464 005301 DEC R1 ;RDHDR?
2033 020466 001500 BEQ ARDHDR ;BRANCH IF SO...FUNCTION #4
2034 020470 005301 DEC R1 ;WRITE?
2035 020472 001002 BNE 1$ ;NO, BRANCH
2036 020474 000137 021136 JMP AWRITE ;FUNCTION #5
2037 020500 005301 1$: DEC R1 ;READ?
2038 020502 001432 BEQ AFREAD ;BRANCH IF SO...FUNCTION #6
2039 020504 005301 DEC R1 ;READ W/NO HDR COMPARE?
2040 020506 001440 BEQ AFWRCK ;YES - TREAT AS IF WRITE CHECK
2041
2042 020510 ERRSF 210,PRGER ;SHOULD NEVER GET HERE!!!
(4) 020510 104454 TRAP CSERSF
(5) 020512 000322 .WORD 210
(5) 020514 002732 .WORD PRGER
(5) 020516 000000 .WORD 0
2043 020520 000000 HALT
2044 020522 000137 021222 XEXIT: JMP EXIT
```



```

2046          .SBTTL  SEEK INTERRUPT SERVICE
2047
2048 02052f 052764 000001 000056 ASEEK: BIS      #SKDON,PRFLGS(R4)  ;SET SEEK VERIFY NEEDED
2049 02053  005064 000114          CLR      RSEEK(R4)      ;CLEAR THE RETRY FLAG
2050 020540 005264 000054          INC      SKCNT1(R4)     ;INCREMENT COUNT
2051 020544 026427 000054 001750 .    CMP      SKCNT1(R4),#1000. ;10(3) REACHED
2052 020552 002404          BLT     99$            ;NO, EXIT
2053 020554 005264 000000          INC      SKCNT(R4)      ;YES, BUMP THOUSANDS
2054 020560 005064 000054          CLR      SKCNT1(R4)
2055 020564 000137 021222 99$:   JMP      EXIT
2056
2057          .SBTTL  READ INTERRUPT SERVICE
2058
2059 020570 042764 000001 000056 AFREAD: BIC      #SKDON,PRFLGS(R4)  ;CLEAR THE SEEK VERIFY FLAG
2060 020576          SETPRI  #340
2061 (3) 020576 012700 000340          MOV      #340,R0
2062 (3) 020602 104441          TRAP    C$SPRI
2063 020604 004537 023742          JSR     R5,CKDATA      ;CHECK DATA
2064
2063 020610 016401 000042          AFWRCK: MOV      BMP(R4),R1      ;BUMP UP XFER COUNT
2064 020614 005401          NEG     R1             ;MAKE POSITIVE
2065 020616 060164 000002          ADD     R1,RXFR1(R4)   ;ADD THE BITS
2066 020622 022764 023420 000002          CMP     #10000.,RXFR1(R4) ;10(8) REACHED YET
2067 020630 101016          BHI    2$             ;NO, EXIT
2068 020632 005264 000004          INC     RXFR2(R4)      ;BUMP 10(10)
2069 020636 162764 023420 000002          SUB     #10000.,RXFR1(R4) ;START 10(8) AT 0
2070 020644 022764 023420 000004          CMP     #10000.,RXFR2(R4) ;10(10) REACHED YET
2071 020652 101005          BHI    2$             ;NO, EXIT
2072 020654 005264 000060          INC     RXFR3(R4)      ;YES BUMP 65K 10(10)
2073 020660 162764 023420 000004          SUB     #10000.,RXFR2(R4) ;MAKE 10(10) 0
2074 020666 000555          2$:   BR      EXIT      ;EXIT
2075
2076          .SBTTL  READ HEADER INTERRUPT SERVICE
2077
2078 020670 013701 002426          ARDHDR: MOV      E.MP,R1      ;GET HEADER
2079 020674 043701 002272          BIC     SMSK,R1        ;MASK OUT SECTOR BITS
2080 020700 026401 000124          CMP     PRPOS(R4),R1   ;IS HEADER CORRECT?
2081 020704 001442          BEQ    1$             ;YES, CONTINUE
2082
2083 020706 032764 000001 000056          BIT     #SKDON,PRFLGS(R4) ;IS THIS MIS-SEEK OR TRACKING ERROR
2084 020714 001407          BEQ    2$             ;BRANCH IF TRACKING
2085
2086 020716 005264 000016          INC     SKECNT(R4)     ;ACCOUNT FOR SEEK ERROR
2087 020722          ERRHRD 50.,MSKER,ERR2
2088 (4) 020722 104456          TRAP   C$ERHRD
2089 (5) 020724 000062          .WORD  50
2090 (5) 020726 003031          .WORD  MSKER
2091 (5) 020730 005076          .WORD  ERR2
2092 020732 000406          BR     3$             ;BRANCH AROUND TRACKING ERROR REPORT
2093
2094 020734 005264 000072          2$:   INC     TRERR(R4)     ;ACCOUNT FOR TRACKING ERROR
2095 020740          ERRHRD 55.,TRACK,ERR2 ;TRACKING ERROR
2096 (4) 020740 104456          TRAP   C$ERHRD
2097 (5) 020742 000067          .WORD  55
2098 (5) 020744 003305          .WORD  TRACK
2099 (5) 020746 005076          .WORD  ERR2
  
```

```
2093          020750          SKRETRY=.
2094
2095 020750 005264 000114          3$:  INC      RSEEK(R4) ;SET RETRY IN PROGRESS
2096 020754 026437 000114 010742  CMP      RSEEK(R4),T.SLT ;RETRY EXHAUSTED?????
2097 020762 101405          BLOS     4$          ;NO, THEN RETRY
2098
2099 020764          ERRHRD  333.,SEXHAU,ERR2
   (4) 020764 104456          TRAP     C$ERRHRD
   (5) 020766 000515          .WORD   333
   (5) 020770 003523          .WORD   SEXHAU
   (5) 020772 005076          .WORD   ERR2
2100 020774 000406          BR       1$
2101
2102 020776 010164 000050          4$:  MOV      R1,LSTHDR(R4) ;SET UP RETRY
2103 021002 042764 000001 000056  BIC      #SKDON,PRFLGS(R4) ;ALLOW SEEK
2104 021010 000504          BR       EXIT          ;EXIT
2105 021012 042764 000001 000056  1$:  BIC      #SKDON,PRFLGS(R4) ;SET VERIFICATION DONE
2106 021020 005064 000114          CLR      RSEEK(R4)
2107 021024 010164 000124          MOV      R1,PRPOS(R4) ;MAKE THIS HEADER PRESENT POSITION
2108 021030 000474          BR       EXIT          ;EXIT
2109
2110          .SBTTL  GET STATUS INTERRUPT SERVICE
2111
2112 021032 013701 002426          AGSTAT: MOV     E.MP,R1          ;GET STATUS
2113 021036 042701 000100          BIC      #100,R1          ;CLEAR OUT HEAD SELECT
2114 021042 005737 010734          TST     T.ROF          ;READ ONLY
2115 021046 001402          BEQ     2$
2116 021050 042701 020000          BIC      #WL,R1
2117 021054 032701 177400          2$:  BIT     #177400,R1          ;ANY BITS WRONG
2118 021060 001406          BEQ     1$          ;NO, CONTINUE
2119
2120 021062 005264 000012          INC      ERRCNT(R4)          ;STATUS BITS WRONG
2121 021066          ERRHRD  60.,MDSER,ERR4
   (4) 021066 104456          TRAP     C$ERRHRD
   (5) 021070 000074          .WORD   60
   (5) 021072 003116          .WORD   MDSER
   (5) 021074 005312          .WORD   ERR4
2122
2123 021076 010102          1$:  MOV      R1,R2          ;COPY STATUS WORD
2124 021100 042702 177700          BIC      #177700,R2          ;GET STATE BITS
2125 021104 022702 000034          CMP     #34,R2          ;COVER CLSD, HEADS OUT, BRUSHES HOME, SEEK TRACK COUNTIN
2126 021110 001444          BEQ     EXIT          ;YES, EXIT
2127 021112 022702 000035          CMP     #35,R2          ;COVER CLSD, HEADS OUT, BRUSHES HOME, SEEK LINEAR MODE
2128 021116 001441          BEQ     EXIT          ;YES, EXIT
2129
2130 021120 005264 000012          INC      ERRCNT(R4)
2131 021124          ERRHRD  70.,MDSER,ERR4
   (4) 021124 104456          TRAP     C$ERRHRD
   (5) 021126 000106          .WORD   70
   (5) 021130 003116          .WORD   MDSER
   (5) 021132 005312          .WORD   ERR4
2132
2133 021134 000432          BR       EXIT
```



```
2135          .SBTTL WRITE INTERRUPT SERVICE
2136
2137 021136 042764 000001 000056 AWRITE: BIC    #SKDON,PRFLGS(R4) ;CLEAR SEEK VERIFY FLAG
2138 021144 016401 000042          MOV    BMP(R4),R1      ;GET WORD COUNT
2139 021150 005401          NEG    R1              ;MAKE POSITIVE
2140 021152 060164 000006          ADD    R1,WXFR1(R4)   ;ADD THE BITS
2141 021156 022764 023420 000006    CMP    #10000.,WXFR1(R4) ;10(5) YET?
2142 021164 101016          BHI   EXIT           ;NO - EXIT
2143 021166 005264 000010          INC    WXFR2(R4)     ;YES BUMP 10(10)
2144 021172 162764 023420 000006    SUB    #10000.,WXFR1(R4) ;10(5) GOES TO ZERO
2145 021200 022764 023420 000010    CMP    #10000.,WXFR2(R4) ;10(10) YET?
2146 021206 101005          BHI   EXIT           ;NO - EXIT
2147 021210 005264 000062          INC    WXFR3(R4)     ;INC 65K (10)(10)
2148 021214 162764 023420 000010    SUB    #10000.,WXFR2(R4) ;MAKE 10(10)
2149
2150 021222 005764 000036          EXIT:  TST   RETRY(R4) ;IN PROCESS OF RETRYING?
2151 021226 001414          BEQ   ERREX          ;NO
2152 021230 026427 000052 004353    CMP    RTYPE(R4),#MTDRV
2153 021236 001406          BEQ   EXIT1
2154 021240 005264 000014          INC    SFTCNT(R4)   ;YES, LOG SOFT ERROR
2155
2156 021244          ERRSOFT 80.,MSFER,ERR3 ;REPORT SOFT ERROR
(4) 021244 104457          TRAP  C$ERSOFT
(5) 021246 000120          .WORD 80
(5) 021250 003042          .WORD MSFER
(5) 021252 005162          .WORD ERR3
2157
2158 021254 005064 000036          EXIT1: CLR   RETRY(R4) ;CLEAR RETRY
```

```
2160 .SBTTL EXIT FOR INTERRUPT SERVICE
2161
2162 021260 042774 000100 000104 ERREX: BIC #INTEN,@DCS(R4)
2163 021266 012600 MOV (SP)+,R0
2164 021270 012601 MOV (SP)+,R1
2165 021272 012602 MOV (SP)+,R2
2166 021274 012603 MOV (SP)+,R3
2167 021276 012637 002476 MOV (SP)+,INCALL
2168 021302 012637 002474 MOV (SP)+,OPCALL
2169 021306 012637 002246 MOV (SP)+,WHY
2170 021312 012637 002350 MOV (SP)+,TEMP1
2171 021316 012637 002340 MOV (SP)+,HDRFND
2172 021322 012637 002342 MOV (SP)+,CHKSEC
2173 021326 012637 002432 MOV (SP)+,E.MP2
2174 021332 012637 002430 MOV (SP)+,E.MP1
2175 021336 012637 002426 MOV (SP)+,E.MP
2176 021342 012637 002424 MOV (SP)+,E.DA
2177 021346 012637 002422 MOV (SP)+,E.BA
2178 021352 012637 002420 MOV (SP)+,E.CS
2179 021356 012604 MOV (SP)+,R4
2180 021360 ENDSRV
(3) 021360 L10027:
(2) 021360 000002 RTI
2181
2182 021362 004537 022616 FINERR: JSR R5,RCNT ;CHECK TO SEE IF WE HAVE EXCEEDED
2183 021366 000405 BR 1$ ;RETRY LIMIT, IF SO 1$ AND REPORT HARD
2184 021370 013764 002420 000116 MOV E.CS,SOFTCS(R4)
2185 021376 000137 021260 JMP ERREX ;NOT EXCEEDED EXIT
2186 021402 005264 000012 1$: INC ERRCNT(R4) ;INDICATE ERROR
2187
2188 021406 ERRHRD 90.,MHDER,ERR13 ;NON-RECOVERABLE ERROR
(4) 021406 104456 TRAP C$ERHRD
(5) 021410 000132 .WORD 90
(5) 021412 003252 .WORD MHDER
(5) 021414 005724 .WORD ERR13
2189 021416 000716 BR EXIT1
```



```

2191 .SBTTL DRIVE ERROR INTERRUPT SERVICE
2192
2193 ;WE HAVE A DRIVE ERROR, LET'S GET THE STATUS
2194
2195 021420 005264 000020 CKDERR: INC DERCNT(R4) ;ACCOUNT FOR ERROR
2196 021424 004537 024432 JSR R5,GETDST ;GET DRIVE STATUS
2197 ;REPORT DRIVE ERROR
2198 021430 ERRHRD 224.,DRVER,ERR9 ;DRIVE ERROR
(4) 021430 104456 TRAP C$ERHRD
(5) 021432 000340 .WORD 224
(5) 021434 003061 .WORD DRVER
(5) 021436 005602 .WORD ERR9
2199
2200 ;ACT ACCORDINGLY TO DRIVE ERROR
2201
2202 021440 032701 001000 BIT #VC,R1 ;VOLUME CHECK?
2203 021444 001027 BNE 9$ ;YES, GO ISSUE RESET
2204 021446 032701 010000 BIT #SKTO,R1 ;SEEK TIME OUT?
2205 021452 001070 BNE 12$ ;YES, ISSUE RESET
2206 021454 032701 144000 BIT #WDE!HCE!SPE,R1 ;WRITE DATA, CURRENT HEAD, SPINDLE?
2207 021460 001153 BNE 15$ ;GO WAIT FOR HEADS TO UNLOAD
2208 021462 032701 002000 BIT #WGE,R1 ;WRITE GATE ERROR
2209 021466 001003 BNE 20$ ;YES, ISSUE RESET
2210 021470 004537 024446 JSR R5,ISDRST ;ISSUE RESET
2211 021474 000431 BR 10$ ;GO CHECK DRIVE READY
2212 021476 004537 024446 20$: JSR R5,ISDRST ;ISSUE RESET
2213 021502 004537 024432 JSR R5,GETDST ;RESET WORK?
2214 021506 032701 002000 BIT #WGE,R1 ;WGE CLEAR
2215 021512 001422 BEQ 10$ ;YES GO CHECK DRIVE READY
2216 021514 012737 003157 002246 MOV #WGEST,WHY ;REPORT WGE DIDN'T CLR
2217 021522 000412 BR 91$ ;DROP DRIVE
2218
2219 021524 004537 024446 9$: JSR R5,ISDRST ;ISSUE RESET
2220 021530 004537 024432 JSR R5,GETDST ;RESET WORK
2221 021534 032701 001000 BIT #VC,R1 ;VOL CHK CLEAR
2222 021540 001407 BEQ 10$ ;YES, CHECK DRIVE READY
2223 021542 012737 003132 002246 MOV #MVCER,WHY ;DROP THE DRIVE
2224
2225 021550 004537 023520 91$: JSR R5,DRDRV
2226 021554 000137 021254 JMP EXIT1
2227 021560 032763 000001 000000 10$: BIT #DRDY,CS(R3) ;DRIVE READY POSTED?
2228 021566 001004 BNE 101$ ;YES, PRINT RECOVERED
2229
2230 021570 012737 002664 002246 MOV #DNRDY,WHY
2231 021576 000764 BR 91$ ;NO, DROP DRIVE
2232
2233 021600 101$: PRINTB #FMT14,#MRDER ;PRINT DRIVE RECOVERED
(8) 021600 012746 003212 MOV #MRDER,-(SP)
(7) 021604 012746 007544 MOV #FMT14,-(SP)
(6) 021610 012746 000002 MOV #2,-(SP)
(3) 021614 010600 MOV SP,R0
(4) 021616 104414 TRAP C$PNTB
(4) 021620 062706 000006 ADD #6,SP
2234 021624 004537 022316 JSR R5,GHDR ;GET THE CURRENT DISK POSITION - HEADER
2235 021630 000137 021362 JMP FINERR
2236 021634 012702 000004 12$: MOV #4,R2 ;SEEK TIME OUT

```

```

2237 021640 004537 024446      13$: JSR      R5,ISDRST      ;ISSUE DRIVE RESET
2238                                     ;FOUR TIMES BEFORE
2239 021644                                     ;DROPPING DRIVE
      (3) 021662 012727 000372      WAITMS   #15.
      (3) 021666 000000      MOV      ##250.,(PC)+
      (3) 021670 013727 002116      .WORD   0
      (3) 021674 000000      MOV      L$DLY,(PC)+
      (3) 021676 005367 177772      .WORD   0
      (3) 021702 001375      DEC      -6(PC)
      (3) 021704 005367 177756      BNE     -.4
      (3) 021710 001367      DEC      -22(PC)
      BNE     -.20
2240
2241 021720 032763 000001 000000      BIT     #DRDY,CS(R3)      ;DRIVE READY YET?
2242 021726 001006      BNE     14$              ;YES, CHECK IF ERROR CLEARED
2243 021730 005302      DEC     R2                ;NO, HAVE WE DONE IT FOUR TIMES
2244 021732 001342      BNE     13$              ;YET
2245
2246 021734 012737 003070 002246 141$: MOV     #MDERS,WHY        ;YES, DROP DRIVE
2247 021742 000702      BR      91$
2248
2249 021744 032763 040000 000000 14$: BIT     #DERR,CS(R3)      ;DRIVE ERROR SET STILL
2250 021752 001370      BNE     141$            ;YES, DROP DRIVE
2251 021754      PRINTB  #FMT14,#MRDER
      (8) 021754 012746 003212      MOV     #MRDER,-(SP)
      (7) 021760 012746 007544      MOV     #FMT14,-(SP)
      (6) 021764 012746 000002      MOV     #2,-(SP)
      (3) 021770 010600      MOV     SP,R0
      (4) 021772 104414      TRAP   C$PNTB
      (4) 021774 062706 000006      ADD     #6,SP
2252 022000 004537 022316      JSR     R5,GHDR
2253 022004 000137 021222      JMP     EXIT
2254
2255 022010 012702 000004      15$: MOV     #4,R2
2256 022014 004537 024432      16$: JSR     R5,GETDST      ;WAIT FOR HEADS TO UNLOAD
2257 022020 032701 000020      BIT     #BIT4,R1        ;GET STATUS
2258 022024 001434      BEQ     17$              ;UNLOAD STATE
2259 022026      WAITMS  #15.            ;YES, CONTINUE W/ RECOVERY
      (3) 022044 012727 000372      MOV     ##250.,(PC)+    ;WAIT A WHILE
      (3) 022050 000000      .WORD   0
      (3) 022052 013727 002116      MOV     L$DLY,(PC)+
      (3) 022056 000000      .WORD   0
      (3) 022060 005367 177772      DEC     -6(PC)
      (3) 022064 001375      BNE     -.4
      (3) 022066 005367 177756      DEC     -22(PC)
      (3) 022072 001367      BNE     -.20
2260 022102 005302      DEC     R2                ;WAIT LONG ENOUGH
2261 022104 001343      BNE     16$              ;NO, GO BACK
2262 022106 012737 003547 002246      MOV     #UNLOAD,WHY     ;DROP DRIVE
2263 022114 000615      BR      91$
2264
2265 022116 004537 024446      17$: JSR     R5,ISDRST      ;ISSUE RESET
2266 022122      WAITMS  #1.
      (3) 022140 012727 000372      MOV     ##250.,(PC)+
      (3) 022144 000000      .WORD   0
      (3) 022146 013727 002116      MOV     L$DLY,(PC)+
      (3) 022152 000000      .WORD   0

```



```
(3) 022154 005367 177772      DEC      -6(PC)
(3) 022160 001375              BNE      .-4
(3) 022162 005367 177756      DEC     -22(PC)
(3) 022166 001367              BNE     .-20
2267 022176 032763 040000 000000  BIT     #DERR,CS(R3)      ;DRIVE ERROR CLEAR?
2268 022204 001253              BNE     141$              ;NO, DROP DRIVE
2269 022206 012702 000075      MOV     #61.,R2          ;YES, WAIT 60 SECONDS
2270 022212              18$: WAITMS #10.         ;FOR DRIVE READY TO
(3) 022230 012727 000372      MOV     ##250.,(PC)+
(3) 022234 000000              .WORD   0
(3) 022236 013727 002116      MOV     L$DLY,(PC)+
(3) 022242 000000              .WORD   0
(3) 022244 005367 177772      DEC     -6(PC)
(3) 022250 001375              BNE     .-4
(3) 022252 005367 177756      DEC     -22(PC)
(3) 022256 001367              BNE     .-20
2271 022266 032763 000001 000000  BIT     #DRDY,CS(R3)      ;COME BACK
2272 022274 001223              BNE     14$              ;
2273 022276 005302              DEC     R2
2274 022300              BREAK   ;INITIATE PROGRAM CALL TO SUPERVISOR
(3) 022300 104422              TRAP   C$BRK
2275 022302 001343              BNE     18$
2276 022304 012737 003573 002246  MOV     #NOLOAD,WHY      ;NO READY DROP DRIVE
2277 022312 000137 021550      JMP     91$
2278
2279
2280 022316 012763 000210 000000  GHDR:  MOV     #CRDY!RDHDR,CS(R3)
2281 022324 056463 000106 000000      BIS     DRSEL(R4),CS(R3)
2282 022332 042763 000200 000000      BIC     #200,CS(R3)
2283 022340 004537 024340              JSR     R5,WTRDY
2284 022344 016301 000006      MOV     MP(R3),R1
2285 022350 043701 002272      BIC     SMSK,R1
2286 022354 010164 000124      MOV     R1,PRPOS(R4)
2287 022360 012764 004353 000052  MOV     #MTDRV,RTYPE(R4) ;SETUP DRIVE ERROR
2288 022366 000205              RTS     R5
2289
2290
```

```
2292 .SBTTL BUFFER GENERATION ROUTINE FOR THE 'WRITE' FUNCTION
2293 STARS
(2) :*****
2294 :WRBUF -- ROUTINE TO WRITE A BUFFER INTO MEMORY. USES WORD COUNT AND BUS
2295 : ADDRESS FROM DRIVE BUFFER (R4). WILL WRITE RANDOM FROM ONE OF
2296 : 8 PATTERNS. USED BY WRITE FUNCTION AND WRPACK ROUTINE.
2297 STARS
(2) :*****
2298
2299 022370 005737 002306 WRBUF: TST REGEN ;REBUILD THE DATA BUFFER?
2300 022374 001507 BEQ 9$ ;NO --EXIT
2301 022376 010346 MOV R3,-(SP) ;SAVE REGISTERS
2302 022400 010246 MOV R2,-(SP)
2303 022402 010146 MOV R1,-(SP)
2304 022404 010046 MOV R0,-(SP)
2305 022406 016402 000042 MOV BMP(R4),R2 ;R2 HAS TOTAL WORDS TO SET UP FOR
2306 022412 005402 NEG R2 ;POSITIVE NUMBER
2307 022414 017401 000110 MOV @BBA(R4),R1 ;WHERE BUFFER IS
2308 022420 020227 000200 2$: CMP R2,#128. ;MORE THAN 128 WORDS
2309 022424 002015 BGE 4$ ;YES, BRANCH
2310 022426 020227 000003 CMP R2,#3 ;GREATER THAN THREE WORDS
2311 022432 002005 BGE 3$ ;YES, BRANCH
2312 022434 062702 000003 ADD #3,R2 ;ADD 3
2313 022440 162764 000003 000042 SUB #3,BMP(R4) ;WC UP BY 3
2314 022446 010221 3$: MOV R2,(R1)+ ;STORE WC
2315 022450 005302 DEC R2 ;ACCOUNT FOR WC
2316 022452 010237 002362 MOV R2,TEMP6 ;LOAD DOWN COUNTER
2317 022456 000405 BR 5$
2318 022460 012737 000177 002362 4$: MOV #127.,TEMP6 ;LOAD DOWN COUNTER
2319 022466 012721 000200 MOV #128.,(R1)+
2320 022472 005737 010736 5$: TST T.RAN ;RANDOM SELECT OF PATTERNS
2321 022476 001003 BNE 55$ ;YEA
2322 022500 013703 010740 MOV T.PAT,R3 ;NO GET.PATTERN OPERATOR
2323 022504 000406 BR 56$ ;WANTS TO USE
2324 022506 004537 024524 55$: JSR R5,RAND ;GET RANDOM # FOR PATTERN
2325 022512 013703 002262 MOV LONUM,R3 ;GET RANDOM PATTERN
2326 022516 042703 177770 BIC #177770,R3 ;0,7
2327 022522 006303 56$: ASL R3 ;WORD OFFSET
2328 022524 062703 030004 ADD #PATLST,R3 ;GET PATTERN LIST
2329 022530 011303 MOV (R3),R3 ;GET LIST ADDRESS
2330 022532 010337 002364 MOV R3,TEMP7 ;STOR FOR RECALL
2331 022536 010321 MOV R3,(R1)+ ;LOAD IT
2332 022540 005337 002362 DEC TEMP6 ;ACCOUNT FOR IT
2333 022544 013703 002364 6$: MOV TEMP7,R3 ;PATTERN START
2334 022550 012737 000020 002366 MOV #16.,TEMP8 ;16 ENTRIES
2335 022556 012321 7$: MOV (R3)+,(R1)+ ;STORE PATTERN
2336 022560 005337 002362 DEC TEMP6 ;DOWN COUNT
2337 022564 001404 BEQ 8$ ;DONE?
2338 022566 005337 002366 DEC TEMP8 ;DONE WITH PATTERN
2339 022572 001371 BNE 7$ ;NO, GO BACK
2340 022574 000763 BR 6$ ;RESTART PATTERN
2341 022576 162702 000200 8$: SUB #128.,R2 ;ANOTHER SECTOR TO USE
2342 022602 003306 BGT 2$ ;YES GO BACK
2343 022604 012600 MOV (SP)+,R0 ;RESTORE REGISTERS
2344 022606 012601 MOV (SP)+,R1
2345 022610 012602 MOV (SP)+,R2
```



```
2346 022612 012603
2347 022614 000205
2348
2349
2350
2351
2352
2353
2354 022616 026437 000036 010660 RCNT:  CMP   RETRY(R4),LIMIT ;LIMIT REACHED?
2355 022624 001403          BEQ   1$           ;YES TAKE FIRST RETURN
2356 022626 005264 000036          INC   RETRY(R4)    ;ACCOUNT FOR RETRY
2357 022632 005725          TST   (R5)+       ;NEXT RETURN
2358 022634 000205          1$:  RTS   R5           ;RETURN
2359
2360
2361
2362
2363
2364
2365 022636 000000
2366 022640 015424
2367 022642 015460
2368 022644 015620
2369 022646 015424
2370 022650 015460
2371 022652 015504

.SBTTL  RETRY LIMIT ROUTINE
;RETRY BUMP, TWO RETURNS - CALL +2 - RETRY EXCEEDED
;CALL +4 - CONTINUE RETRY

.SBTTL  LIST OF FUNCTION ROUTINES
;WE GO THRU THIS LIST WHEN CALLED IN 'GETFNC'
;LIST IS IN NUMERICAL ORDER 1-6

LIST:   .WORD   0
        SKWRT   ;SEEK - WRITE DATA - WRITE CHECK
        SKRD    ;SEEK - READ DATA
        SKRH    ;SEEK - READ HDR - READ W/NO HDR CMP - GET STATUS
        SKWRT   ;SEEK - WRITE DATA - WRITE CHECK
        SKRD    ;SEEK - READ DATA
        SKRDRD  ;SEEK - READ DATA - READ DATA
```

```
2373 .SBTTL BAD SECTOR FILE ROUTINE
2374 022654 STARS
(2) :*****
2375 :RDBDSC -- ROUTINE TO RECOVER BAD SECTOR FILE AND SAVE IT FOR
2376 :COMPARISON UPON ERROR ON READS/Writes & FOR THE SEEK FUNCTION. WE
2377 :WILL ONLY RESERVE SPACE FOR 16 BAD SECTORS PER DRIVE AND 1 ENTRY FOR
2378 :THE BAD SECTOR FILE AREA POINTER - LAST TRACK ON THE CARTRIDGE.
2379 :WE WILL ISSUE A DRIVE RESET FIRST, READ HEADER, POSITION TO THE LAST
2380 :TRACK (CYLINDER 255. OR 511., SURFACE 1) AND READ IN THE FIRST SECTOR
2381 :FOR FACTORY BAD, AND THE 20TH FOR FIELD BAD SECTORS. R4 WILL CONTAIN
2382 :THE BUFFER POINTER TO THE DRIVE WE WANT TO READ.
2383
2384 :CALL JSR R5,RDBDSC ;GET THE BAD SECT FILE ENTRYS
2385
2386 :THE BAD SECTOR FILE (BOTH FACTORY AND FIELD) LOOKS LIKE THIS:
2387
2388 SERIAL NUMBER LOW 5 DIGITS (OCTAL SERIAL NUMBER)
2389 SERIAL NUMBER HIGH 5 DIGITS
2390
2391 0'S
2392 0'S
2393
2394 ENTRY - CYLINDER # FROM 0 TO 1777 MAX (RL02) OR 777 (RL01)
2395 ENTRY - HEAD & SECTOR NUMBER
2396
2397 ENTRY - CYL
2398 ENTRY - HEAD & SECTOR
2399
2400 -1 ...END OF ENTRYS
2401 -1 ...TO WORD 256. (END OF SECOND SECTOR IN PAIR)
2402 022654 STARS
(2) :*****
2403
2404 022654 010046 RDBDSC: MOV R0,-(SP) ;SAVE REGISTERS
2405 022656 010146 MOV R1,-(SP) ;
2406 022660 010246 MOV R2,-(SP) ;
2407 022662 010346 MOV R3,-(SP) ;
2408 022664 004537 024446 21$: JSR R5,ISDRST ;ISSUE A DRIVE RESET
2409 022670 012764 000010 000044 MOV #RDHDR,FUNC (R4);READ HEADER TO FIND POSITION
2410 022676 004537 016574 JSR R5,LDFUNC ;ON DISK
2411 022702 004537 024340 JSR R5,WTRDY
2412
2413 022706 016300 000006 MOV MP(R3),R0 ;GET HEADER AND CALCULATE
2414 022712 022764 000001 000120 CMP #1,TDR(R4) ;RL02 TYPE DRIVE?
2415 022720 001005 BNE 23$ ;JUMP IF RL02
2416 022722 043700 002264 BIC CYLSK,R0 ;HERE FOR RL01
2417 022726 012701 077600 MOV #77600,R1
2418 022732 000404 BR 25$
2419 022734 043700 002270 23$: BIC CMSK,R0 ;HERE FOR RL02
2420 022740 012701 177600 MOV #177600,R1
2421 022744 160001 25$: SUB R0,R1
2422 022746 010164 000040 MOV R1,BDA(R4)
2423 022752 052764 000025 000040 BIS #SKHS!SIGN!MK,BDA(R4)
2424 022760 012764 000006 000044 MOV #SEEK,FUNC(R4)
2425 022766 004537 016574 JSR R5,LDFUNC ;SEEK TO THE BAD SECTOR FILE AREA
2426 022772 004537 024340 JSR R5,WTRDY ;WAIT FOR DRIVE READY
```


2427	022776	012764	000010	000044		MOV	#RDHDR, FUNC(R4)	
2428	023004	004537	016574			JSR	R5, LDFUNC	:READ A HEADER ON THE BSF
2429	023010	004537	024340			JSR	R5, WTRDY	:WAIT FOR DRIVE READY
2430	023014	016300	000006			MOV	MP(R3), R0	:GET THE HEADER WORD READ
2431	023020	042700	000077			BIC	#77, R0	:CLEAR SECTOR NUMBER READ
2432	023024	022764	000001	000120		CMP	#1, TDR(R4)	:DRIVE = RL01?
2433	023032	001007				BNE	300\$:NO - MUST BE AN RL02
2434	023034	022700	077700			CMP	#77700, R0	:YES - ON BSF AREA?
2435	023040	001311				BNE	21\$:NO - SEEK AGAIN
2436	023042	012764	077700	000040		MOV	#77700, BDA(R4)	:SAVE THIS HEADER FOR READ COMMAND
2437	023050	000406				BR	555\$	
2438	023052	022700	177700		300\$:	CMP	#177700, R0	:RL02 BSF AREA?
2439	023056	001302				BNE	21\$:NO - SEEK AGAIN
2440	023060	012764	177700	000040		MOV	#177700, BDA(R4)	:YES - SAVE FOR THE READ COMMAND
2441	023066	012764	177400	000042	555\$:	MOV	#-256., BMP(R4)	:SETUP FOR A 2 SECTOR READ IN BSF
2442	023074	012764	000014	000044		MOV	#READ, FUNC(R4)	:GET THE READ FUNCTION #
2443								
2444	023102	005037	002354			CLR	TEMP3	:MANUFACTURING/FIELD FILE SWITCH
2445	023106	012737	003720	002246		MOV	#HWSEC, WHY	:START WITH MANUFACTURING BAD
2446	023114	016402	000112			MOV	BSECTP(R4), R2	:INITIALIZE LIST TO ALL 1'S
2447	023120	012700	000021			MOV	#17., R0	:SIXTEEN ENTRIES + 1 FOR BSF POINTER
2448	023124	012722	177777		11\$:	MOV	#-1, (R2)+	:INIT STORAGE TO -1'S
2449	023130	005300				DEC	R0	:DONE?
2450	023132	001374				BNE	11\$:NO - DO THE NEXT ONE
2451								
2452	023134	016402	000112			MOV	BSECTP(R4), R2	:GET POINTER TO LIST TO STORE BSF ENTRIES
2453	023140	016422	000040			MOV	BDA(R4), (R2)+	:SAVE 1ST ENTRY AS BSF POINTER
2454	023144	012700	000020			MOV	#16., R0	:SIXTEEN ENTRIES
2455	023150	004537	016574		4\$:	JSR	R5, LDFUNC	:READ THE BSF SECTOR PAIR
2456	023154	004537	024340			JSR	R5, WTRDY	:WAIT FOR DRIVE READY
2457								
2458	023160	005774	000104			TST	@DCS(R4)	:WAS THE READ GOOD?
2459	023164	100042				BPL	3\$:YES
2460								
2461	023166	004537	024446			JSR	R5, ISDRST	:NO - ISSUE A DRIVE RESET
2462	023172	062764	000004	000040		ADD	#4, BDA(R4)	:POINT TO NEXT SECTOR
2463	023200	005737	002354			TST	TEMP3	:MANUFACTURING OR FIELD BAD
2464	023204	001414				BEQ	5\$:MANUFACTURING = 0
2465	023206	012737	003740	002246		MOV	#SWSEC, WHY	:FIELD BAD
2466	023214	022764	000001	000120		CMP	#1, TDR(R4)	:DRIVE = RL01?
2467	023222	001011				BNE	400\$:NO - MUST BE RL02
2468	023224	022764	077750	000040		CMP	#77750, BDA(R4)	:YES - AT END OF FIELD FILE?
2469	023232	001346				BNE	4\$:NO - CONTINUE
2470	023234	000516				BR	6\$:DROP DRIVE AND EXIT
2471								
2472	023236	026427	000040	077724	5\$:	CMP	BDA(R4), #77724	:AT END OF MANUFACTURING BAD
2473	023244	000410				BR	55\$:SEE IF DONE
2474	023246	022764	177750	000040	400\$:	CMP	#177750, BDA(R4)	:AT END OF FIELD BAD FOR RL02
2475	023254	001335				BNE	4\$:NO GO BACK FOR NEXT
2476	023256	000505				BR	6\$:DROP THE DRIVE AND EXIT
2477	023260	026427	000040	177724		CMP	BDA(R4), #177724	:AT END OF MANUFACTURING BAD?
2478	023266	001330			55\$:	BNE	4\$:BR IF NOT DONE
2479	023270	000500				BR	6\$:YES - REPORT ERROR AND EXIT
2480								
2481	023272	017401	000110		3\$:	MOV	@BBA(R4), R1	:START OF BSF ENTRY LIST
2482	023276	012164	000100			MOV	(R1)+, SERNM1(R4)	:GET LOW PART OF SERIAL #

```

2483 023302 012164 000102      MOV      (R1)+,SERNM2(R4)  ;GET HIGH PART OF SERIAL #
2484 023306 022121      CMP      (R1)+,(R1)+      ;SKIP PAST JUNK
2485 023310 012137 002350      1$:     MOV      (R1)+,TEMP1      ;GET CYLINDER
2486 023314 100444      BMI      2$               ;END OF THE ENTRYS?
2487 023316 012137 002352      MOV      (R1)+,TEMP2      ;NO - GET HEAD (0 OR 1) & SECTOR NUMBER
2488 023322 000337 002350      SWAB    TEMP1             ;PUT CYLINDER IN HIGH BYTE
2489 023326 000241      CLC
2490 023330 006037 002350      ROR     TEMP1
2491 023334 103003      BCC     111$
2492 023336 052737 100000 002350      BIS     #BIT15,TEMP1
2493 023344 013712 002350      111$:   MOV      TEMP1,(R2)       ;STORE THE CYLINDER PART
2494 023350 013737 002352 002350      MOV      TEMP2,TEMP1      ;GET SECTOR
2495 023356 042737 177700 002350      BIC     #177700,TEMP1     ;LEAVE ONLY SECTOR
2496 023364 053712 002350      BIS     TEMP1,(R2)       ;SET IN SECTOR BITS
2497 023370 006237 002352      ASR     TEMP2
2498 023374 006237 002352      ASR     TEMP2             ;POSITION THE HEAD SELECT BIT
2499 023400 042737 177677 002352      BIC     #177677,TEMP2     ;CLEAR ALL OTHER BITS
2500 023406 053722 002352      BIS     TEMP2,(R2)+      ;SET IN HEAD
2501 023412 005300      DEC     R0                ;COUNT THIS ENTRY FROM BSF
2502 023414 001335      BNE     1$               ;ALLOW MORE ENTRYS?
2503 023416 012737 003667 002246      MOV     #MBDMSC,WHY      ;MORE THAN 16 BAD SECTORS
2504 023424 000422      BR      6$               ;DROP THE DRIVE & ERROR EXIT
2505
2506 023426 005737 002354      2$:     TST     TEMP3            ;SWITCH TO FIELD BAD OR QUIT
2507 023432 001021      BNE     7$               ;QUIT, 7$
2508 023434 022764 000001 000120      CMP     #1,TDR(R4)       ;DRIVE = RL01?
2509 023442 001004      BNE     350$             ;NO - MUST BE AN RL02
2510 023444 012764 077724 000040      MOV     #77724,BDA(R4)   ;YES - POINT TO FIELD BSF 1ST SECTOR
2511 023452 000403      BR      36$
2512 023454 012764 177724 000040 350$:   MOV     #177724,BDA(R4)  ;POINT TO 1ST SECT IN FIELD FILE FOR RL02
2513 023462 012737 000001 002354 36$:   MOV     #1,TEMP3         ;INDICATE NOW DOING FIELD BSF
2514 023470 000627      BR      4$               ;PROCESS THE FIELD BSF
2515
2516      ;HERE TO DROP THE DRIVE IF MORE THAN 16. ENTRYS OR IF CAN'T FIND A BSF
2517
2518 023472 004537 023520      6$:     JSR     R5,DRDRV         ;DROP THE DRIVE
2519
2520      ;HERE TO PUT HEADS 'HOME' AND TO EXIT
2521
2522 023476 004537 025670      7$:     JSR     R5,HDHOME        ;BRINGS HEADS HOME
2523 023502 012603      MOV     (SP)+,R3
2524 023504 012602      MOV     (SP)+,R2
2525 023506 012601      MOV     (SP)+,R1
2526 023510 012600      MOV     (SP)+,R0
2527 023512 000205      RTS     R5

```



```
2529          .SBTTL  ROUTINE TO DROP DRIVE
2530 023514    STARS
(2)          :*****
2531          :DRDRV -- ROUTINE TO DROP A DRIVE FROM RUNNING
2532          :          R4 HAS BUFFER POINTER OF DRIVE TO DROP
2533          :          WE CLEAR BIT IN 'DRUT', NOT 'DRPRS'
2534 023514    STARS
(2)          :*****
2535
2536 023514    005237 002474  ODRDRV: INC      OPCALL
2537 023520    010146  DRDRV: MOV      R1,-(SP)
2538 023522    010246          MOV      R2,-(SP)          ;SAVE REGISTERS
2539 023524    010346          MOV      R3,-(SP)
2540 023526    005237 002476          INC      INCALL
2541 023532    005003          CLR      R3
2542 023534    012702 030432          MOV      #DRBUF,R2          ;START OF DRIVE BUFFERS
2543 023540    012701 000001          MOV      #1,R1          ;MASK
2544 023544    020402          1$:  CMP      R4,R2          ;IS THIS THE DRIVE?
2545 023546    001405          BEQ      2$          ;YES GO DROP IT
2546 023550    005203          INC      R3
2547 023552    006301          ASL      R1          ;NO SHIFT MASK
2548 023554    062702 000126          ADD      #PRPOS+2,R2      ;NEXT BUFFER
2549 023560    000771          BR       1$          ;GO BACK
2550
2551 023562    005737 002474          2$:  TST      OPCALL          ;CALLED VIA OPERATOR?
2552 023566    001002          BNE      6$          ;YES - SKIP CODE
2553 023570          DODU      R3          ;NO - CALLED BY DIAGNOSTIC
(3) 023570    010300          MOV      R3,R0
(3) 023572    104451          TRAP    C$DODU
2554 023574    005037 002476          6$:  CLR      INCALL
2555 023600    005037 002474          CLR      OPCALL
2556 023604    113764 002416 000070  MOVB     HOUR,DPHOUR(R4) ;TIME AT WHICH IT WAS DROPPED
2557 023612    113764 002414 000071  MOVB     MINUTE,DPMIN(R4) ;HOUR/MINUTE
2558 023620    001002          BNE      3$          ;IF MINUTE 0,
2559 023622    105264 000071          INCB     DPMIN(R4)      ;MAKE 1.
2560 023626    140137 002252          3$:  BICB     R1,DRUT        ;CLEAR THE DRIVE FROM BIT MAP
2561 023632          PRINTF   #FMT14C          ;PRINT A <CR> & <LF>
(7) 023632    012746 007562          MOV      #FMT14C,-(SP)
(6) 023636    012746 000001          MOV      #1,-(SP)
(3) 023642    010600          MOV      SP,R0
(4) 023644    104417          TRAP    C$PNTF
(4) 023646    062706 000004          ADD      #4,SP
2562 023652    004737 006220          JSR      PC,LINE2
2563 023656          PRINTF   #FMT7,#DROP,WHY
(9) 023656    013746 002246          MOV      WHY,-(SP)
(8) 023662    012746 004276          MOV      #DROP,-(SP)
(7) 023666    012746 007201          MOV      #FMT7,-(SP)
(6) 023672    012746 000003          MOV      #3,-(SP)
(3) 023676    010600          MOV      SP,R0
(4) 023700    104417          TRAP    C$PNTF
(4) 023702    062706 000010          ADD      #10,SP
2564 023706          PRINTF   #FMTS1
(7) 023706    012746 010013          MOV      #FMTS1,-(SP)
(6) 023712    012746 000001          MOV      #1,-(SP)
(3) 023716    010600          MOV      SP,R0
(4) 023720    104417          TRAP    C$PNTF
```

CZRLKBO RL01/02 PERF EXER
CZRLKB.MAC 07-DEC-79 09:46

MACY11 30A(1052) 17-DEC-79 11:31 ^{B 9} PAGE 4-26
ROUTINE TO DROP DRIVE

SEQ 0105

(4)	023722	062706	000004	ADD	#4,SP	
2565						
2566	023726	004737	014024	JSR	PC,REPORT	
2567						
2568	023732	012603		MOV	(SP)+,R3	
2569	023734	012602		MOV	(SP)+,R2	;RESTORE REGISTERS
2570	023736	012601		MOV	(SP)+,R1	
2571						
2572	023740	000205		RTS	R5	


```
2574  
2575  
2576  
2577  
2578 023742 005037 002306  
2579 023746 005737 010674  
2580 023752 001001  
2581 023754 000205  
2582 023756 005737 002310  
2583 023762 001401  
2584 023764 000205  
2585  
2586 023766  
  (3) 023766 012700 000340  
  (3) 023772 104441  
2587 023774 017402 000110  
2588 024000 016437 000042 002350  
2589 024006 005437 002350  
2590 024012 013737 010676 002352  
2591 024020 005037 002344  
2592 024024 013737 010674 002354  
2593 024032 012737 000176 002346  
2594 024040 012201  
2595 024042 005337 002350  
2596 024046 001522  
2597 024050 005301  
2598 024052 012237 002356  
2599  
2600  
2601  
2602 024056 012700 030004  
2603 024062 012703 000010  
2604 024066 022037 002356  
2605 024072 001414  
2606 024074 005303  
2607 024076 001373  
2608  
2609 024100 005237 002306  
2610 024104 024242  
2611 024106  
  (4) 024106 104456  
  (5) 024110 000264  
  (5) 024112 003631  
  (5) 024114 005716  
2612 024116 004537 027006  
2613 024122 000205  
2614  
2615 024124 005301  
2616 024126 013703 002356  
2617 024132 005337 002350  
2618 024136 012737 000020 002360  
2619 024144 005737 002350  
2620 024150 001461  
2621 024152 005737 002354  
2622 024156 001456  
2623 024160 005701
```

```
.SBTTL ROUTINE TO CHECK DATA  
;ROUTINE TO CHECK DATA ON READ  
CKDATA: CLR REGEN ;CLEAR THE REGENERATE DATA FLAG  
TST CMRD ;DO WE WANT TO CHECK ANY?  
BNE 10$ ;YES - SEE IF FORCED EXIT  
RTS R5 ;NO - EXIT NOW  
10$: TST KILLDC ;INHIBIT FLAG SET?  
BEQ 97$ ;NOPE - OK TO PROCEED  
RTS R5 ;NO, EXIT  
97$: SETPRI #340  
MOV #340,R0  
TRAP C$SPRI  
MOV @BBA(R4),R2 ;BUFFER START  
MOV BMP(R4),TEMP1 ;WORDS READ IN  
NEG TEMP1 ;MAKE POSITIVE  
MOV DELMT,TEMP2 ;# ERRORS TO BE PRINTED  
CLR DECNT ;INIT ERROR COUNT  
MOV CMRD,TEMP3 ;# WORDS TO BE COMPARED  
96$: MOV #126,TEMPO ;126 WORDS  
MOV (R2)+,R1 ;NON-ZERO WORDS  
DEC TEMP1  
BEQ CEND  
DEC R1  
MOV (R2)+,TEMP4 ;PATTERN ADDRESS  
;MAKE SURE PATTERN ADDRESS IS LEGAL  
MOV #PATLST,R0 ;GET LIST OF PATTERNS  
MOV #8,R3 ;ONLY EIGHT  
98$: CMP (R0)+,TEMP4 ;FOUND IT YET  
BEQ 99$ ;YES, CONTINUE  
DEC R3 ;NO, EXHAUST LIST YET  
BNE 98$ ;NO, GO BACK  
INC REGEN ;SET THE DATA REGENERATE FLAG  
CMP -(R2),-(R2)  
ERRHRD 180,NOREV,ERR12  
TRAP C$ERHRD  
.WORD 180  
.WORD NOREV  
.WORD ERR12  
JSR R5,STDMP  
RTS R5  
99$: DEC R1 ;ACCOUNT FOR PATTERN ADDRESS  
MOV TEMP4,R3 ;GET ADDRESS  
DEC TEMP1 ;ACCOUNT ONCE AGAIN  
MOV #16,TEMP5 ;16 ENTRIES TO PATTERN  
1$: TST TEMP1 ;ANY WORDS READIN LEFT?  
BEQ CEND ;NO, GO TO END  
TST TEMP3 ;HAVE WE EXHAUSTED COMPARE LIMIT?  
BEQ CEND ;YES GO TO END  
TST R1 ;WE CHECKING PATTERN OR ZERO FILL?
```

2624	024162	001416		BEQ	3\$:ZERO FILL SKIP
2625	024164	005301		DEC	R1	:PATTERN
2626	024166	005737	002360	TST	TEMP5	:WITHIN PATTERN
2627	024172	001005		BNE	2\$:YES SKIP
2628	024174	013703	002356	MOV	TEMP4,R3	:NO, START OVER
2629	024200	012737	000020	MOV	#16,TEMP5	:16 ENTRIES
2630	024206	012337	002402	MOV	(R3)+,GDDAT	:GET PATTERN
2631	024212	005337	002360	DEC	TEMP5	:DOWN COUNT
2632	024216	000402		BR	4\$	
2633	024220	005037	002402	CLR	GDDAT	:ZERO FILL
2634	024224	023712	002402	CMP	GDDAT,(R2)	:CORRECT DATA
2635	024230	001417		BEQ	5\$:YES YES NEXT
2636	024232	005237	002306	INC	REGEN	:NO - SET REGENERATE FLAG FOR WRT OPERATION
2637	024236	005237	002344	INC	DECNT	:COUNT THE DATA ERROR
2638	024242	005264	000074	INC	DATCER(R4)	:COUNT ERROR FOR THIS DRIVE
2639	024246	005737	002352	TST	TEMP2	:DO WE WANT TO PRINT IT
2640	024252	001406		BEQ	5\$:NO,SKIP
2641						
2642	024254			ERRHRD	185.,MDCER,ERR8	
(4)	024254	104456		TRAP	C\$ERHRD	
(5)	024256	000271		.WORD	185	
(5)	024260	003235		.WORD	MDCER	
(5)	024262	005462		.WORD	ERR8	
2643	024264	005337	002352	DEC	TEMP2	:ACCOUNT FOR PRINT
2644						
2645	024270	005337	002350	DEC	TEMP1	:WORDS READ IN
2646	024274	001407		BEQ	CEND	
2647	024276	005722		TST	(R2)+	:NEXT WORD
2648	024300	005337	002346	DEC	TEMPO	
2649	024304	001652		BEQ	96\$	
2650	024306	005337	002354	DEC	TEMP3	:WORDS TO CHECK
2651	024312	000714		BR	1\$	
2652						
2653	024314	005737	002344	TST	DECNT	:DO WE WANT TO PRINT SUMMARY
2654	024320	001406		BEQ	1\$:NO,EXIT
2655	024322	005464	000042	NEG	BMP(R4)	:MAKE POSITIVE WORD COUNT
2656	024326			ERRHRD	190.,MDCER,ERR6	:DATA ERROR SUMMARY
(4)	024326	104456		TRAP	C\$ERHRD	
(5)	024330	000276		.WORD	190	
(5)	024332	003235		.WORD	MDCER	
(5)	024334	005364		.WORD	ERR6	
2657						
2658	024336	000205		RTS	R5	

2660
2661
2662
2663
2664
2665
2666
2667 024340 010046
2668 024342 010146
2669 024344 012701 001750
2670 024350
(3) 024350 012727 000002
(3) 024354 000000
(3) 024356 013727 002116
(3) 024362 000000
(3) 024364 005367 177772
(3) 024370 001375
(3) 024372 005367 177756
(3) 024376 001367
2671 024400 032774 000200 000104
2672 024406 001006
2673 024410 005301
2674 024412 001356
2675
2676 024414
(4) 024414 104455
(5) 024416 001752
(5) 024420 002654
(5) 024422 005716
2677
2678 024424 012601
2679 024426 012600
2680 024430 000205

.SBTTL ROUTINE TO WAIT FOR CONTROLLER READY
:
:ROUTINE TO WAIT FOR CONTROLLER READY UNDER FLAG
:MODE. USED IN INITIALIZE PORTION OF PROGRAM, I.E.,
:GETTING BAD SECTOR FILE, WRITING PACK INITIALLY.
WTRDY: MOV R0,-(SP) ;SAVE REGISTERS
MOV R1,-(SP)
MOV #1000.,R1 ;WAIT A WHILE
1\$: WAITUS #2.
MOV ###2.,(PC)+
.WORD 0
MOV LSDLY,(PC)+
.WORD 0
DEC -6(PC)
BNE -4
DEC -22(PC)
BNE -20
BIT #CRDY,@DCS(R4) ;READY SET?
BNE 2\$;YES, EXIT
DEC R1 ;TIMED OUT?
BNE 1\$;NO GO BACK
ERRDF 1002.,NOCRDY,ERR12
TRAP C\$ERDF
.WORD 1002
.WORD NOCRDY
.WORD ERR12
2\$: MOV (SP)+,R1 ;RESTORE REGISTERS
MOV (SP)+,R0
RTS R5

```
2682          .SBTTL GET STATUS/DRIVE RESET ROUTINE
2683
2684          ;ROUTINE TO ISSUE DRIVE RESET
2685          ;ALSO GET STATUS, R1 HAS STATUS IF GS
2686          ;USES R3, DOES NOT SAVE IT
2687
2688 024432 016403 000104          GETDST: MOV      DCS(R4),R3
2689 024436 012763 000003 000004      MOV      #GSBIT,DA(R3)
2690 024444 000405          BR        CSTUFF
2691 024446 016403 000104          ISDRST: MOV     DCS(R4),R3
2692 024452 012763 000013 000004      MOV      #DRST,DA(R3)
2693 024460 012763 000204 000000      CSTUFF: MOV     #CRDY!GSTAT,CS(R3)
2694 024466 056463 000106 000000      BIS      DRSEL(R4),CS(R3)
2695 024474 042763 000200 000000      BIC      #CRDY,CS(R3)
2696 024502 004537 024340          JSR      R5,WTRDY
2697 024506 022763 000013 000004      CMP      #DRST,DA(R3)
2698 024514 001402          BEQ      1$
2699 024516 016301 000006          MOV      MP(R3),R1
2700 024522 000205          1$:      RTS      R5
2701
2702 024524          STARS
2703          ;:*****
2704 024524          ;:RAND -- ROUTINE TO GENERATE A RANDOM NUMBER
2705          ;:*****
2706 024524 010146          RAND:  MOV      R1,-(SP)
2707 024526 010246          MOV      R2,-(SP)
2708 024530 010346          MOV      R3,-(SP)
2709
2710 024532 013703 002262          MOV      LONUM,R3
2711 024536 013701 002260          MOV      HINUM,R1
2712 024542 012702 177771          MOV      #-7,R2
2713 024546 006303          1$:  ASL      R3
2714 024550 006101          ROL      R1
2715 024552 005202          INC      R2
2716 024554 001374          BNE      1$
2717 024556 063703 002262          ADD      LONUM,R3
2718 024562 005501          ADC      R1
2719 024564 063701 002260          ADD      HINUM,R1
2720 024570 062703 001057          ADD      #1057,R3
2721 024574 005501          ADC      R1
2722 024576 062701 047401          ADD      #47401,R1
2723 024602 010337 002260          MOV      R3,HINUM
2724 024606 010137 002262          MOV      R1,LONUM
2725 024612 012603          MOV      (SP)+,R3
2726 024614 012602          MOV      (SP)+,R2
2727 024616 012601          MOV      (SP)+,R1
2728 024620 000205          RTS      R5
```



```
2730 .SBTTL ROUTINE TO WRITE PACKS INITIALLY
2731 024622 STARS
(2) :*****
2732 :WRPACK -- ROUTINE TO WRITE PACK WITH PATTERN, ALL TRACKS WILL BE
2733 WRITTEN (EXCEPT BAD SECTOR TRACK)
2734 : FORMAT IS # OF WORDS (WORD 1), PATTERN ADDRESS (WORD 2)
2735 : PATTERN (WORDS 3 - 128)
2736 : WE WILL ATTEMPT TO WRITE MULTIPLE SECTORS AT A TIME
2737 : (MINIMUM 10 SECTORS) IF AN ERROR OCCURS WE WILL THEN
2738 : WRITE INDIVIDUAL SECTORS FOR THAT TRACK. WE DO WRITES,
2739 : READS AND INCORE COMPARISONS TO VERIFY.
2740 :
2741 : CALL: JSR R5,WRPACK ;WRITE THE PACK SELECTED
2742 024622 STARS
(2) :*****
2743
2744 024622 010046 WRPACK: MOV R0,-(SP) ;SAVE REGISTERS
2745 024624 010146 MOV R1,-(SP)
2746 024626 010246 MOV R2,-(SP)
2747 024630 010346 MOV R3,-(SP)
2748 024632 016446 000110 MOV BBA(R4),-(SP)
2749 024636 005764 000122 TST WRIPG(R4) ;SEE IF WRITE IN PROGRESS
2750 024642 001016 BNE 1$ ;JUMP IF DON'T WANT MESSAGE ON RECOVERY
2751 024644 PRINTF #FMT18,#MSWRPK ;MSG. 'WRITING PACK'
(8) 024644 012746 004400 MOV #MSWRPK,-(SP)
(7) 024650 012746 007747 MOV #FMT18,-(SP)
(6) 024654 012746 000002 MOV #2,-(SP)
(3) 024660 010600 MOV SP,R0
(4) 024662 104417 TRAP C$PNTF
(4) 024664 062706 000006 ADD #6,SP
2752 024670 000240 NOP
2753 024672 000240 NOP
2754 024674 004737 006220 JSR PC,LINE2 ;PRINT TIME-RCLS & DRIVE ID
2755 024700 004537 025670 1$: JSR R5,HDHOME ;HEADS HOME
2756
2757
2758 ;NOW ACTUALLY WRITE DATA OUT ON PACK, WILL NOT WRITE LAST
2759 ;TRACK
2760
2761
2762 024704 005037 002350 CLR TEMP1 ;TEMP1=HEAD
2763 024710 005001 CLR R1 ;R1=CYL
2764 024712 022764 000001 000120 CONWR: CMP #1,TDR(R4)
2765 024720 001007 BNE 45$
2766 024722 022701 077600 CMP #077600,R1
2767 024726 001023 BNE STWRT
2768 024730 005737 002350 TST TEMP1
2769 024734 001420 BEQ STWRT
2770 024736 000406 BR ENDWR
2771 024740 022701 177600 45$: CMP #177600,R1
2772 024744 001014 BNE STWRT ;NO GO WRITE TRACK
2773 024746 005737 002350 TST TEMP1 ;YES, CHECK IF HEAD = 1?
2774 024752 001411 BEQ STWRT ;HEAD = 0 GO WRITE
2775 024754 004537 025670 ENDWR: JSR R5,HDHOME ;HEADS HOME
2776 024760 012664 000110 MOV (SP)+,BBA(R4)
2777 024764 012603 MOV (SP)+,R3
```

```

2778 024766 012602          MOV      (SP)+,R2
2779 024770 012601          MOV      (SP)+,R1
2780 024772 012600          MOV      (SP)+,R0
2781 024774 000205          RTS      R5                ;END EXIT
2782
2783          ;THIS PORTION WILL WRITE THE PACK USING MULTIPLE SECTORS IF A
2784          ;ERROR OCCURS WE WILL GO TO 2$ AND INDIVIDUAL SECTORS.
2785
2786 024776 005002          STWRT:  CLR      R2                ;INITIAL SECTOR 0
2787 025000 012764 002436 000110  MOV      #BUF1,BBA(R4)        ;BUFFER START
2788 025006 012764 175400 000042  MOV      #-1280.,BMP(R4)     ;10 SECTORS
2789 025014 005237 002306          INC      REGEN                ;SET THE GENERATE BUFFER FLAG
2790 025020 004537 022370          JSR      R5,WRBUF            ;WRITE BUFFER INTO MEMORY
2791 025024 010164 000040          201$:  MOV      R1,BDA(R4)        ;SET UP SECTOR
2792 025030 053764 002350 000040  BIS      TEMP1,BDA(R4)
2793 025036 005764 000122          TST      WRIPG(R4)           ;WRITE IN PROGRESS?
2794 025042 001406          BEQ      762$                ;NO - JUMP OVER
2795 025044 026464 000124 000040  CMP      PRPOS(R4),BDA(R4)    ;YUP - ON CYLINDER NOW?
2796 025052 001402          BEQ      762$                ;YUP - WRITE THIS AREA
2797 025054 000137 025464          JMP      952$                ;NO - LOOK AT NEXT AREA ON DRIVE
2798 025060 050264 000040          762$:  BIS      R2,BDA(R4)
2799 025064 012764 002436 000110  MOV      #BUF1,BBA(R4)        ;SET UP TO WRITE
2800 025072 012764 000012 000044  MOV      #WRITE,FUNC(R4)     ;WRITE
2801 025100 004537 016574          JSR      R5,LDFUNC
2802 025104 004537 024340          JSR      R5,WTRDY            ;WAIT FOR READY
2803 025110 005774 000104          TST      @DCS(R4)           ;ERROR
2804 025114 100003          BPL      203$
2805 025116 004537 024446          205$:  JSR      R5,ISDRST
2806 025122 000421          BR       2$
2807
2808 025124 012764 000002 000044  203$:  MOV      #WRCHK,FUNC(R4)
2809 025132 004537 016574          JSR      R5,LDFUNC
2810 025136 004537 024340          JSR      R5,WTRDY
2811 025142 005774 000104          TST      @DCS(R4)           ;ERROR
2812 025146 100763          BMI      205$                ;YES GO DO SECTORS INDIVIDUALLY
2813
2814
2815 025150 062702 000012          ADD      #10.,R2              ;NEXT GROUP
2816 025154 022702 000050          CMP      #40.,R2            ;DONE?
2817 025160 001321          BNE      201$                ;NO, GO BACK
2818 025162 000137 025464          JMP      952$                ;YES NEXT TRACK
2819
2820          ;IF AN ERROR OCCURS THEN WE COME HERE AND DO THE TRACK SECTOR
2821          ;BY SECTOR.
2822
2823 025166 005002          2$:    CLR      R2                ;R2 = SECTOR
2824
2825 025170 012764 177600 000042          MOV      #-128.,BMP(R4)     ;LOAD WORD COUNT
2826 025176 010164 000040          3$:    MOV      R1,BDA(R4)        ;SETUP DISK ADDRESS
2827 025202 053764 002350 000040  BIS      TEMP1,BDA(R4)
2828 025210 050264 000040          BIS      R2,BDA(R4)
2829
2830 025214 012764 002436 000110          MOV      #BUF1,BBA(R4)
2831 025222 004537 022370          JSR      R5,WRBUF            ;WRITE A BUFFER
2832 025226 005037 002244          91$:   CLR      RWCNT           ;CLEAR RETRYS OUT
2833 025232 005037 002344          98$:   CLR      DECNT
  
```



```

2834 025236 012764 000012 000044 96$: MOV #WRITE, FUNC(R4) ;WRITE FUNCTION
2835 025244 004537 016574 JSR R5, LDFUNC
2836 025250 004537 024340 JSR R5, WTRDY ;WAIT FOR WRITE TO FINISH
2837
2838 025254 005774 000104 TST @DCS(R4) ;ERROR ON WRITE?
2839 025260 100021 BPL 85$ ;NO, GO READ
2840
2841 025262 016437 000040 002342 MOV BDA(R4), CHKSEC ;YES, CHECK IF SECTOR IS IN
2842 025270 004537 027216 JSR R5, CKBDSC ;BAD SECTOR FILE
2843 025274 005737 002340 TST HDRFND ;IF SET, IT WAS
2844 025300 001050 BNE 802$ ;YES GO TO NEXT SECTOR
2845
2846 025302 005237 002344 INC DECNT ;NO, GIVE IT 3 TRYS TOTAL
2847 025306 023727 002344 000003 CMP DECNT, #3. ;IT MAY HAVE BEEN NOISE.
2848 025314 001440 BEQ 801$ ;BR IF AT RETRY LIMIT - BAD SECTOR
2849 025316 004537 024446 JSR R5, ISDRST ;RESET THE DRIVE & TRY AGAIN
2850 025322 000745 BR 96$ ;TRY RECOVERY AGAIN
2851
2852 025324 005037 002242 85$: CLR RECNT ;CLEAR RETRY COUNT
2853 025330 012764 000002 000044 80$: MOV #WRCHK, FUNC(R4) ;READ/VERIFY THE 1 SECTOR WRITTEN
2854 025336 004537 016574 JSR R5, LDFUNC ;ISSUE A WRITE-CHECK FUNCTION
2855 025342 004537 024340 JSR R5, WTRDY ;WAIT FOR DRIVE READY
2856
2857 025346 005774 000104 TST @DCS(R4) ;ERROR ON READ?
2858 025352 100025 BPL 95$ ;BR IF OK ... GET THE NEXT SECTOR
2859
2860 025354 016437 000040 002342 MOV BDA(R4), CHKSEC ;CHECK IF SECTOR IS
2861 025362 004537 027216 JSR R5, CKBDSC ;A KNOWN BAD SECTOR
2862 025366 005737 002340 TST HDRFND ;IT WAS THEN
2863 025372 001013 BNE 802$ ;GO TO NEXT SECTOR
2864
2865 025374 005237 002242 INC RECNT ;GIVE IT ANOTHER CHANCE
2866 025400 023727 002242 000020 CMP RECNT, #16. ;16 RE-READS BEFORE HARD ERROR
2867 025406 001403 BEQ 801$ ;REPORT ERROR IF AT RETRY LIMIT
2868 025410 004537 024446 JSR R5, ISDRST ;RESET THE DRIVE
2869 025414 000745 BR 80$ ;AND RETRY AGAIN
2870
2871 025416 004537 025550 801$: JSR R5, INBAD ;REPORT THE BAD SECTOR
2872 025422 004537 024446 802$: JSR R5, ISDRST ;RESET THE DRIVE FOR THE NEXT OPERATION
2873
2874 025426 062702 000012 95$: ADD #10, R2 ;NEXT SECTOR (OFFSET BY 10)
2875 025432 020227 000047 CMP R2, #39. ;DONE WITH TRACK?
2876 025436 003002 BGT 951$ ;YES NEXT TRACK
2877 025440 000137 025176 JMP 3$ ;NO GO BACK FOR NEXT SECTOR
2878
2879 025444 005202 951$: INC R2 ;NEXT SECTOR
2880 025446 162702 000050 SUB #40, R2 ;DONE WITH TRACK?
2881 025452 020227 000012 CMP R2, #10. ;
2882 025456 001402 BEQ 952$ ;YES
2883 025460 000137 025176 952$: JMP 3$ ;NO
2884 025464
2885
2886 025464 005737 002350 TST TEMP1 ;WHICH SURFACE?
2887 025470 001420 BEQ 5$ ;TOP (0), BRANCH
2888
2889 025472 005037 002350 CLR TEMP1 ;BOTTOM, SWITCH TO TOP WITH

```

```

2890 025476 062701 000200          ADD    #200,R1
2891 025502 012764 000205 000040  MOV    #205,BDA(R4)      ;SEEK, GO IN ALSO
2892 025510 012764 000006 000044 4$:  MOV    #SEEK,FUNC(R4)   ;GO SEEK
2893 025516 004537 016574          JSR    R5,LDFUNC
2894 025522 004537 024340          JSR    R5,WTRDY
2895
2896 025526 000137 024712          JMP    CONWR
2897
2898 025532 012737 000100 002350 5$:  MOV    #HEAD,TEMP1      ;WAS TOP, MAKE BOTTOM.
2899 025540 012764 000021 000040  MOV    #21,BDA(R4)
2900 025546 000760          BR     4$
2901
2902
2903 025550 010146          INBAD: MOV    R1,-(SP)        ;SAVE R1
2904 025552 016403 000104          MOV    DCS(R4),R3      ;GET THE CSR ADDRESS
2905 025556 016337 000000 002420  MOV    CS(R3),E.CS     ;GET THE ERROR INFO FROM CSR
2906 025564 016337 000002 002422  MOV    BA(R3),E.BA
2907 025572 016337 000004 002424  MOV    DA(R3),E.DA
2908 025600 000240          NOP
2909 025602 000240          NOP
2910 025604 004537 024432          JSR    R5,GETDST       ;GET THE CURRENT DRIVE STATUS
2911 025610 010137 002426          MOV    R1,E.MP        ;SAVE IT AS "(RLMP)" DATA
2912 025614          ERRHRD 199.,NWRTS,ERR12
(4) 025614 104456          TRAP  C$ERRHD
(5) 025616 000307          .WORD 199
(5) 025620 002736          .WORD NWRTS
(5) 025622 005716          .WORD ERR12
2913 025624 005264 000012          INC    ERRCNT(R4)
2914 025630 005737 010722          TST   T.DRP           ;ARE WE COUNTING ERRORS
2915 025634 001413          BEQ   2$              ;NO
2916 025636 026437 000012 010662  CMP   ERRCNT(R4),ERLMT ;PAST IT
2917 025644 103407          BLO  2$              ;NO
2918 025646 012737 003322 002246  MOV   #ERLMTM,WHY
2919 025654 004537 023520          JSR   R5,DRDRV
2920 025660 012705 024754          MOV   #ENDWR,R5
2921
2922 025664 012601          2$:  MOV   (SP)+,R1        ;RESET R1
2923 025666 000205          RTS   R5
2924
2925

```



```
2927 .SBTTL HEADS HOME ROUTINE
2928 STARS
(2) :*****
2929 :HDHOME -- ROUTINE TO BRING HEADS OVER TRACK 0
2930 STARS
(2) :*****
2931
2932 HDHOME: MOV R0,-(SP) :SAVE R0
2933 MOV #RDHDR,FUNC(R4) :READ HEADER
2934 JSR R5,LDFUNC :GO DO IT.
2935 JSR R5,WTRDY
2936
2937 MOV MP(R3),R0 :GET HEADER
2938 BIC #177,R0 :ONLY CYLINDER
2939 MOV R0,LSTHDR(R4) :SAVE THIS CYL # AS THE LAST POSITION
2940 MOV R0,BDA(R4) :MOVE IT TO BUFFERED DA
2941 BIS #MK,BDA(R4) :SET MARKER FOR SEEK TO 000
2942 MOV #SEEK,FUNC(R4) :LOAD SEEK
2943 JSR R5,LDFUNC :SEEK!
2944 JSR R5,WTRDY :WAIT.
2945 CLR PRPOS(R4) :SET BUFFER TO HOME CYLINDER (000)
2946 MOV (SP)+,R0
2947 RTS R5
```

```
2949 .SBTTL RANDOM WC AND DA ROUTINE
2950 025764 STARS
(2) :*****
2951 :GWCDA -- ROUTINE TO GET RANDOM SECTOR AND WORD COUNT FOR R/W TRANSFER.
2952 :SECTOR IS CHOSEN BETWEEN MIN/MAX LIMITS, WORD COUNT IS BETWEEN
2953 :MIN/MAX WORD COUNT. WORD COUNT WILL BE ADJUSTED NOT TO CAUSE
2954 :TRACK OVERFLOW IF HIGH SECTORS ARE CHOSEN....
2955 :R4 HAS BUFFER OF DRIVE WE'RE WORKING WITH
2956 :ON EXIT - BMP(R4) HAS WORD COUNT
2957 :         - BDA(R4) HAS DISK ADDRESS
2958 025764 STARS
(2) :*****
2959
2960 025764 023737 010714 010716 GWCDA:  CMP      T.MXS,T.MNS      :MIN MAX SECTORS EQUAL
2961 025772 001003          BNE      99$          :NO, CALCULATE ONE
2962 025774 013702 010714          MOV      T.MXS,R2      :LOAD SECTOR
2963 026000 000421          BR       5$           :GO GET WC
2964 026002 004537 024524 99$:   JSR      R5,RAND      :GET RANDOM # FOR SECTOR
2965 026006 013702 002262          MOV      LONUM,R2
2966 026012 042702 177700 1$:   BIC      #177700,R2   :0-77 ONLY
2967 026016 023702 010714          CMP      T.MXS,R2      :R2 LOWER THAN MAX
2968 026022 103003          BHIS     3$           :BRANCH IF YES
2969 026024 006202          ASR      R2           :HALF IT
2970 026026 005202          INC      R2           :INC SO NOT 0
2971 026030 000770          BR       1$
2972 026032 020237 010716 3$:   CMP      R2,T.MNS      :MIN OKAY
2973 026036 103002          BHIS     5$
2974 026040 006102          ROL      R2
2975 026042 000763          BR       1$
2976
2977
2978 :NOW GET WORD COUNT
2979
2980 026044 005737 010746 5$:   TST      T.STIP        :RESTRICT THE XFER SIZE?
2981 026050 001003          BNE      95$          :BR IF YES
2982 026052 013737 002442 010702 95$:  MOV      MAXWC,T.MXB   :NO - MAKE MAXWC = BIGGEST XFER SIZE AVAIL.
2983 026060 023737 002442 010702 95$:  CMP      MAXWC,T.MXB
2984 026066 103021          BHIS     97$
2985
2986 026070          PRINTF #FMT13D,#OVER,T.MXB,MAXWC
(10) 026070 013746 002442  MOV      MAXWC,-(SP)
(9) 026074 013746 010702  MOV      T.MXB,-(SP)
(8) 026100 012746 003453  MOV      #OVER,-(SP)
(7) 026104 012746 007517  MOV      #FMT13D,-(SP)
(6) 026110 012746 000004  MOV      #4,-(SP)
(3) 026114 010600          MOV      SP,R0
(4) 026116 104417          TRAP    C$PNTF
(4) 026120 062706 000012  ADD      #12,SP
2987 026124 013737 002442 010702  MOV      MAXWC,T.MXB
2988
2989 026132 023737 010702 010724 97$:  CMP      T.MXB,T.MNB   :MIN MAX EQUAL
2990 026140 003006          BGT      6$
2991 026142 013737 010702 010724  MOV      T.MXB,T.MNB
2992
2993 026150 013703 010702          MOV      T.MXB,R3     :YES SET WC
2994 026154 000421          BR       9$
```



```
2995 026156 004537 024524      6$: JSR R5,RAND ;GET RANDOM WORD COUNT
2996 026162 013703 002262      MOV LONUM,R3
2997 026166 042703 160000      7$: BIC #160000,R3 ;MAX!!!!
2998 026172 023703 010702      CMP T.MXB,R3
2999 026176 103003      BHIS 8$
3000 026200 006203      ASR R3
3001 026202 005203      INC R3
3002 026204 000770      BR 7$
3003 026206 020337 010724      8$: CMP R3,T.MNB
3004 026212 103002      BHIS 9$
3005 026214 006103      ROL R3
3006 026216 000763      BR 7$
3007
3008 ;NOW WE HAVE SECTOR AND WORD COUNT, CHECK THAT WORD COUNT WILL FIT ON SECTOR
3009 ;IF NOT LOWER SECTOR START
3010
3011
3012 026220 012701 000050      9$: MOV #40.,R1 ;SETUP FOR FOURTY SECTORS
3013 026224 005403      NEG R3 ;MAKE WORD COUNT NEGATIVE
3014 026226 010364 000042      MOV R3,BMP(R4) ;LOAD WORD COUNT
3015 026232 005301      11$: DEC R1 ;DOWN COUNT MINIMUM START SECT NEEDED
3016 026234 062703 000200      ADD #128.,R3 ;ONE SECTOR'S WORTH
3017 026240 100774      BMI 11$ ;STILL NEED ANOTHER SECTOR
3018 026242 020201      CMP R2,R1 ;DID RANDOM SECTOR SUFFICE
3019 026244 101401      BLOS 12$ ;BRANCH IF SUFFICED
3020 026246 010102      MOV R1,R2 ;NO, THEN MAKE IT FIT
3021 026250 016464 000124 000040 12$: MOV PRPOS(R4),BDA(R4)
3022 026256 042764 000077 000040      BIC #77,BDA(R4)
3023 026264 050264 000040      BIS R2,BDA(R4)
3024 026270 000205      RTS R5
```

```
3026          .SBTTL  ROUTINE TO DUMP BUFFER ON DCK
3027 026272   STARS
(2)          :*****
3028          :DMPBUF -- ROUTINE TO DUMP BUFFER ON DCK ERROR, TWO DUMPS ARE POSSIBLE
3029          :          ONE WHERE WE CAN COMPARE WHAT IT SHOULD BE AND THE OTHER
3030          :          WHEN WE CAN'T.
3031 026272   STARS
(2)          :*****
3032
3033 026272 004737 006330   DMPBUF: JSR      PC,LINE3
3034
3035          :CALCULATE THE STARTING BUS ADDRESS FOR THE COMPARE
3036
3037 026276 012737 000200 002464   MOV      #128.,DWCNT1
3038 026304 016400 000040          MOV      BDA(R4),R0          :GET STARTING BUS ADDRESS
3039
3040 026310 013701 002424          MOV      E.DA,R1          :GET PRESENT DISK ADDRESS
3041 026314 042700 177700          BIC      #177700,R0        :SAVE SECTOR BITS
3042 026320 042701 177700          BIC      #177700,R1
3043 026324 010002          MOV      R0,R2          :SAVE A COPY
3044 026326 010103          MOV      R1,R3          :SAVE ANOTHER
3045 026330 160203          SUB      R2,R3          :GET DIFF OF SECTORS
3046 026332 005002          CLR      R2          :CALCULATE WORD COUNT
3047 026334 062702 000200 93$:   ADD      #128.,R2        :ONE SECTORS WORTH
3048 026340 005303          DEC      R3          :DONE
3049 026342 001374          BNE      93$          :NO
3050 026344 016403 000042          MOV      BMP(R4),R3      :GET WORD COUNT
3051 026350 005403          NEG      R3          :MAKE IT POSITIVE
3052 026352 020203          CMP      R2,R3          :WORKING WITH FULL SECTOR
3053 026354 003005          BGT      94$          :NO, GO CALC PARTIAL SECTOR
3054 026356 013702 002422          MOV      E.BA,R2        :PRESENT BUS ADDRESS
3055 026362 162702 000400          SUB      #400,R2        :START OF COMPARE
3056 026366 000412          BR       96$          :GO COMPARE BUFFER
3057 026370 160302          BR       94$          :GET SECTOR DIFF
3058 026372 012700 000200          MOV      #128.,R0
3059 026376 160200          SUB      R2,R0
3060 026400 010037 002464          MOV      R0,DWCNT1
3061 026404 006300          ASL      R0
3062 026406 013702 002422          MOV      E.BA,R2
3063 026412 160002          SUB      R0,R2
3064 026414          96$:   PRINTB  #FMT13,#BUSAD,R2,#CRLDA,CHKSEC
(11) 026414 013746 002342          MOV      CHKSEC,-(SP)
(10) 026420 012746 002577          MOV      #CRLDA,-(SP)
(9) 026424 010246          MOV      R2,-(SP)
(8) 026426 012746 004132          MOV      #BUSAD,-(SP)
(7) 026432 012746 007502          MOV      #FMT13,-(SP)
(6) 026436 012746 000005          MOV      #5,-(SP)
(3) 026442 010600          MOV      SP,R0
(4) 026444 104414          TRAP    C$PNTB
(4) 026446 062706 000014          ADD      #14,SP
3065 026452 012700 030004          MOV      #PATLST,R0      :CHECK PATTERN LIST
3066 026456 012701 000010          MOV      #8.,R1
3067 026462 022062 000002 1$:   CMP      (R0)+,2(R2)
3068 026466 001415          BEQ     2$
3069 026470 005301          DEC     R1
3070 026472 001373          BNE     1$
```


3071							
3072	026474				3\$:	PRINTB	#FMT14,#NOREV
(8)	026474	012746	003631			MOV	#NOREV,-(SP)
(7)	026500	012746	007544			MOV	#FMT14,-(SP)
(6)	026504	012746	000002			MOV	#2,-(SP)
(3)	026510	010600				MOV	SP,R0
(4)	026512	104414				TRAP	C\$PNTB
(4)	026514	062706	000006			ADD	#6,SP
3073	026520	000532				BR	STDMP
3074							
3075	026522	021227	000200		2\$:	CMP	(R2),#128.
3076	026526	101362				BHI	3\$
3077	026530	005037	002344			CLR	DECNT
3078	026534	013701	010744			MOV	T.CLT,R1
3079							
3080	026540	012237	002346			MOV	(R2)+,TEMPO ;NONZERO WORD COUNT
3081	026544	013737	002346	002462		MOV	TEMPO,DWCNT
3082	026552	005437	002462			NEG	DWCNT
3083	026556	012237	002350			MOV	(R2)+,TEMP1
3084	026562	162737	000002	002346		SUB	#2,TEMPO
3085	026570	012737	000002	002352		MOV	#2,TEMP2 ;WORD
3086	026576	013703	002350			MOV	TEMP1,R3 ;PATTERN ADDRESS
3087	026602	012737	000020	002360		MOV	#16,TEMP5 ;16 ENTRIES
3088	026610	005737	002346		4\$:	TST	TEMP0 ;ZERO OR PATTERN
3089	026614	001417				BEQ	6\$;ZERO BRANCH
3090	026616	005337	002346			DEC	TEMPO
3091	026622	005737	002360			TST	TEMP5 ;WITHIN LIST
3092	026626	001005				BNE	5\$
3093	026630	012737	000020	002360		MOV	#16,TEMP5
3094	026636	013703	002350			MOV	TEMP1,R3
3095	026642	012337	002402		5\$:	MOV	(R3)+,GDDAT
3096	026646	005337	002360			DEC	TEMP5
3097	026652	000402				BR	7\$
3098	026654	005037	002402		6\$:	CLR	GDDAT
3099	026660	005237	002462		7\$:	INC	DWCNT
3100	026664	021237	002402			CMP	(R2),GDDAT
3101	026670	001422				BEQ	8\$
3102							
3103	026672	005237	002344			INC	DECNT
3104	026676	005701				TST	R1
3105	026700	001416				BEQ	8\$
3106	026702	005301				DEC	R1
3107	026704					PRINTB	#FMT14B,TEMP2,GDDAT,(R2)
(10)	026704	011246				MOV	(R2)-,(SP)
(9)	026706	013746	002402			MOV	GDDAT,-(SP)
(8)	026712	013746	002352			MOV	TEMP2,-(SP)
(7)	026716	012746	007565			MOV	#FMT14B,-(SP)
(6)	026722	012746	000004			MOV	#4,-(SP)
(3)	026726	010600				MOV	SP,R0
(4)	026730	104414				TRAP	C\$PNTB
(4)	026732	062706	000012			ADD	#12,SP
3108							
3109	026736	005237	002352		8\$:	INC	TEMP2
3110	026742	005722				TST	(R2)+
3111	026744	023737	002352	002464		CMP	TEMP2,DWCNT1
3112	026752	003716				BLE	4\$

```
3113 026754          PRINTB #FMT9A,DECNT,TEMP2
(9) 026754 013746 002352  MOV     TEMP2,-(SP)
(8) 026760 013746 002344  MOV     DECNT,-(SP)
(7) 026764 012746 007263  MOV     #FMT9A,-(SP)
(6) 026770 012746 000003  MOV     #3,-(SP)
(3) 026774 010600          MOV     SP,R0
(4) 026776 104414          TRAP   C$PNTB
(4) 027000 062706 000010  ADD     #10,SP
3114
3115 027004 000205          RTS    R5
3116
3117          ;ROUTINE TO DUMP THE CONTENTS OF THE READ BUFFER ON ERROR DETECTED
3118          ;WILL ALSO TELL HOW MANY WORDS WERE IN THE XFER
3119
3120 027006 016437 000042 002346 STDMP: MOV     BMP(R4),TEMPO          ;GET NEGATIVE WORD COUNT
3121 027014 005437 002346          NEG     TEMPO                ;MAKE THE # POSITIVE
3122 027020 012737 000200 002464 MOV     #128,DWCNT1          ;SET THE SIZE OF SECTOR
3123 027026          PRINTB #FMTXS,TEMPO        ;TELL TRANSFER SIZE
(8) 027026 013746 002346  MOV     TEMPO,-(SP)
(7) 027032 012746 007754  MOV     #FMTXS,-(SP)
(6) 027036 012746 000002  MOV     #2,-(SP)
(3) 027042 010600          MOV     SP,R0
(4) 027044 104414          TRAP   C$PNTB
(4) 027046 062706 000006  ADD     #6,SP
3124 027052 013701 010744  MOV     T,CLT,R1            ;GET THE PRINT LIMIT
3125 027056 012703 000012  MOV     #10,R3              ;SETUP LINE LIMIT
3126 027062          1$: PRINTB #FMT14A,(R2)        ;PRINT A DATA WORD
(8) 027062 011246          MOV     (R2),-(SP)
(7) 027064 012746 007553  MOV     #FMT14A,-(SP)
(6) 027070 012746 000002  MOV     #2,-(SP)
(3) 027074 010600          MOV     SP,R0
(4) 027076 104414          TRAP   C$PNTB
(4) 027100 062706 000006  ADD     #6,SP
3127 027104 005722          TST    (R2)+                ;POINT TO THE NEXT DATA WORD
3128 027106 005303          DEC    R3                   ;DONE WITH THE LINE?
3129 027110 001012          BNE    2$                   ;BR IF NO
3130 027112          PRINTB #FMT14C
(7) 027112 012746 007562  MOV     #FMT14C,-(SP)
(6) 027116 012746 000001  MOV     #1,-(SP)
(3) 027122 010600          MOV     SP,R0
(4) 027124 104414          TRAP   C$PNTB
(4) 027126 062706 000004  ADD     #4,SP
3131 027132 012703 000012  MOV     #10,R3              ;RESET THE LINE LIMIT
3132 027136 005337 002464          2$: DEC    DWCNT1            ;END OF SECTOR?
3133 027142 001001          BNE    3$                   ;BR IF NO
3134 027144 000402          BR     4$                   ;YES - EXIT
3135 027146 005301          3$: DEC    R1                ;AT PRINT LIMIT?
3136 027150 001344          BNE    1$                   ;BR IF NO
3137 027152          4$: PRINTB #FMT14C
(7) 027152 012746 007562  MOV     #FMT14C,-(SP)
(6) 027156 012746 000001  MOV     #1,-(SP)
(3) 027162 010600          MOV     SP,R0
(4) 027164 104414          TRAP   C$PNTB
(4) 027166 062706 000004  ADD     #4,SP
3138 027172 000205          RTS    R5                   ;EXIT
3139
```


3140
3141
3142
3143
3144
3145
3146
3147
3148
3149
3150
3151
3152
3153
3154
3155
3156

027174 010446
027176 012704 030432
027202 005024
027204 020427 031712
027210 001374
027212 012604
027214 000205

: ROUTINE TO CLEAR ALL DRIVE INFO, USED ON START OR
: RESTART IF CALLED. CAN BE USED TO CLEAR INDIVIDUAL DRIVE
: INFO BY BITMAP FOLLOWING CALL
: CALL JSR R5,CLEAR
:

CLEAR: MOV R4,-(SP) :SAVE R4
MOV #DRBUF,R4 :GET BUFFER STARTS
2\$: CLR (R4)+ :CLEAR
CMP R4,#ENDBUF :AT END OF BUFFERS
BNE 2\$:NO, GO TO 2\$
4\$: MOV (SP)+,R4 :RESTORE CURRENT BUFFER POINTER
RTS R5 :EXIT

```
3158          .SBTTL  ROUTINE TO CHECK FOR BAD SECTOR
3159 027216    STARS
(2)          :*****
3160          :CKBDSC -- ROUTINE TO MATCH BAD SECTOR.....BDA(R4) IS SECTOR WE ARE LOOKING
3161          :          FOR IN LIST POINTED TO BY BSECT(R4).....HDRFND IS SET IF WE FIND IT.
3162 027216    STARS
(2)          :*****
3163
3164 027216 005037 002340  CKBDSC: CLR      HDRFND          :CLEAR FLAG
3165 027222 010046          MOV      R0,-(SP)         :SAVE R0
3166 027224 010246          MOV      R2,-(SP)         :SAVE R2
3167 027226 012700 000021  MOV      #17,R0          :16 ENTRIES + BSF POINTER
3168 027232 016402 000112  1$: MOV    BSECT(R4),R2    :GET WHERE WE'RE LOOKING
3169 027236 022712 177777  2$: CMP    #-1,(R2)       :END OF ENTRY LIST?
3170 027242 001411          BEQ     4$               :BRANCH IF END
3171 027244 023712 002342  CMP     CHKSEC,(R2)     :HAVE WE GOT A MATCH
3172 027250 001404          BEQ     3$               :THEN GO SET INDICATOR, ELSE
3173 027252 005722          TST     (R2)+
3174 027254 005300          DEC     R0
3175 027256 001367          BNE     2$
3176 027260 000402          BR     4$
3177 027262 005237 002340  3$: INC    HDRFND          :SET FLAG FOUND
3178 027266 012602          4$: MOV    (SP)+,R2
3179 027270 012600          MOV    (SP)+,R0
3180 027272 000205          RTS     R5
3181
3182 027274    STARS
(2)          :*****
3183          :CKBDTK -- HERE TO CHECK IF CYLINDER & HEAD SELECTED IS IN THE BAD SECTOR FILE
3184 027274    STARS
(2)          :*****
3185
3186 027274 005037 002340  CKBDTK: CLR      HDRFND          :CLEAR FLAG
3187 027300 010046          MOV      R0,-(SP)         :SAVE R0
3188 027302 010146          MOV      R1,-(SP)         :SAVE R1
3189 027304 010246          MOV      R2,-(SP)         :SAVE R2
3190 027306 012700 000021  MOV      #17,R0          :16 ENTRIES + BSF POINTER
3191 027312 016402 000112  1$: MOV    BSECT(R4),R2    :GET WHERE WE'RE LOOKING
3192 027316 022712 177777  2$: CMP    #-1,(R2)       :END OF LIST?
3193 027322 001414          BEQ     4$               :BRANCH IF END
3194 027324 011201          MOV    (R2),R1         :GET THE ENTRY FROM BAD SECT FILE
3195 027326 043701 002272  BIC     SMSK,R1         :LEAVE ONLY CYL # & HEAD
3196 027332 023701 002342  CMP     CHKSEC,R1      :HAVE WE GOT A MATCH
3197 027336 001404          BEQ     3$               :THEN GO SET INDICATOR, ELSE
3198 027340 005722          TST     (R2)+
3199 027342 005300          DEC     R0
3200 027344 001364          BNE     2$
3201 027346 000402          BR     4$
3202 027350 005237 002340  3$: INC    HDRFND          :SET FLAG FOUND
3203 027354 012602          4$: MOV    (SP)+,R2
3204 027356 012601          MOV    (SP)+,R1
3205 027360 012600          MOV    (SP)+,R0
3206 027362 000205          RTS     R5
```



```
3208 027364 STARS
(2) :.....
3209 027364 STARS
(2) :.....
3210 :BUFFER TO STORE BAD SECTOR LISTS
3211
3212 027364 000021 BSEC0: .BLKW 17.
3213 027426 000021 BSEC1: .BLKW 17.
3214 027470 000021 BSEC2: .BLKW 17.
3215 027532 000021 BSEC3: .BLKW 17.
3216 027574 000021 BSEC4: .BLKW 17.
3217 027636 000021 BSEC5: .BLKW 17.
3218 027700 000021 BSEC6: .BLKW 17.
3219 027742 000021 BSEC7: .BLKW 17.
3220 030004 STARS
(2) :.....
3221 030004 STARS
(2) :.....
3222
3223 :LIST OF PATTERNS USED IN WRITING
3224
3225 030004 030024 PATLST: PAT0 :ALL 0'S
3226 030006 030064 PAT1 :-1'S TO ALT BITS
3227 030010 030124 PAT2 :0'S TO ALT BITS
3228 030012 030164 PAT3 :SHIFTING ALT BITS
3229 030014 030224 PAT4 :WORST CASE DATA
3230 030016 030264 PAT5 :STRANGE DATA
3231 030020 030324 PAT6 :ALL 1'S
3232 030022 030364 PAT7 :STRANGE DATA
3233
3234 030024 000000 PAT0: .WORD 0
3235 030026 000000 .WORD 0
3236 030030 000000 .WORD 0
3237 030032 000000 .WORD 0
3238 030034 000000 .WORD 0
3239 030036 000000 .WORD 0
3240 030040 000000 .WORD 0
3241 030042 000000 .WORD 0
3242 030044 000000 .WORD 0
3243 030046 000000 .WORD 0
3244 030050 000000 .WORD 0
3245 030052 000000 .WORD 0
3246 030054 000000 .WORD 0
3247 030056 000000 .WORD 0
3248 030060 000000 .WORD 0
3249 030062 000000 .WORD 0
3250
3251 030064 177777 PAT1: .WORD 177777
3252 030066 177777 .WORD 177777
3253 030070 177777 .WORD 177777
3254 030072 052525 .WORD 052525
3255 030074 052525 .WORD 052525
3256 030076 052525 .WORD 052525
3257 030100 177777 .WORD 177777
3258 030102 177777 .WORD 177777
3259 030104 052525 .WORD 052525
```

3260	030106	052525	.WORD	052525
3261	030110	177777	.WORD	177777
3262	030112	052525	.WORD	052525
3263	030114	177252	.WORD	177252
3264	030116	177252	.WORD	177252
3265	030120	172765	.WORD	172765
3266	030122	172765	.WORD	172765
3267				
3268	030124	000000	PAT2: .WORD	0
3269	030126	000000	.WORD	0
3270	030130	000000	.WORD	0
3271	030132	177777	.WORD	177777
3272	030134	177777	.WORD	177777
3273	030136	177777	.WORD	177777
3274	030140	000000	.WORD	0
3275	030142	000000	.WORD	0
3276	030144	177777	.WORD	177777
3277	030146	177777	.WORD	177777
3278	030150	000000	.WORD	0
3279	030152	177777	.WORD	177777
3280	030154	000000	.WORD	0
3281	030156	177777	.WORD	177777
3282	030160	000000	.WORD	0
3283	030162	177777	.WORD	177777
3284				
3285	030164	025252	PAT3: .WORD	25252
3286	030166	052525	.WORD	52525
3287	030170	052525	.WORD	52525
3288	030172	125252	.WORD	125252
3289	030174	125252	.WORD	125252
3290	030176	125252	.WORD	125252
3291	030200	052525	.WORD	52525
3292	030202	052525	.WORD	52525
3293	030204	125252	.WORD	125252
3294	030206	125252	.WORD	125252
3295	030210	052525	.WORD	52525
3296	030212	125252	.WORD	125252
3297	030214	052525	.WORD	52525
3298	030216	125252	.WORD	125252
3299	030220	052525	.WORD	52525
3300	030222	125252	.WORD	125252
3301				
3302	030224	155555	PAT4: .WORD	155555
3303	030226	066666	.WORD	066666
3304	030230	133333	.WORD	133333
3305	030232	155555	.WORD	155555
3306	030234	066666	.WORD	066666
3307	030236	133333	.WORD	133333
3308	030240	155555	.WORD	155555
3309	030242	066666	.WORD	066666
3310	030244	133333	.WORD	133333
3311	030246	155555	.WORD	155555
3312	030250	066666	.WORD	066666
3313	030252	133333	.WORD	133333
3314	030254	155555	.WORD	155555
3315	030256	066666	.WORD	066666

3316	030260	133333	.WORD	133333
3317	030262	155555	.WORD	155555
3318				
3319	030264	121105	PAT5: .WORD	121105
3320	030266	150442	.WORD	150442
3321	030270	064221	.WORD	64221
3322	030272	132110	.WORD	132110
3323	030274	055044	.WORD	55044
3324	030276	026422	.WORD	26422
3325	030300	013211	.WORD	13211
3326	030302	105504	.WORD	105504
3327	030304	042642	.WORD	42642
3328	030306	021321	.WORD	21321
3329	030310	110550	.WORD	110550
3330	030312	044264	.WORD	44264
3331	030314	022132	.WORD	22132
3332	030316	011055	.WORD	11055
3333	030320	104426	.WORD	104426
3334	030322	042213	.WORD	42213
3335				
3336	030324	177777	PAT6: .WORD	177777
3337	030326	177777	.WORD	177777
3338	030330	177777	.WORD	177777
3339	030332	177777	.WORD	177777
3340	030334	177777	.WORD	177777
3341	030336	177777	.WORD	177777
3342	030340	177777	.WORD	177777
3343	030342	177777	.WORD	177777
3344	030344	177777	.WORD	177777
3345	030346	177777	.WORD	177777
3346	030350	177777	.WORD	177777
3347	030352	177777	.WORD	177777
3348	030354	177777	.WORD	177777
3349	030356	177777	.WORD	177777
3350	030360	177777	.WORD	177777
3351	030362	177777	.WORD	177777
3352				
3353	030364	045513	PAT7: .WORD	45513
3354	030366	122645	.WORD	122645
3355	030370	151322	.WORD	151322
3356	030372	064551	.WORD	64551
3357	030374	132264	.WORD	132264
3358	030376	055132	.WORD	55132
3359	030400	026455	.WORD	26455
3360	030402	113226	.WORD	113226
3361	030404	045513	.WORD	45513
3362	030406	122645	.WORD	122645
3363	030410	151322	.WORD	151322
3364	030412	064551	.WORD	64551
3365	030414	132264	.WORD	132264
3366	030416	055132	.WORD	55132
3367	030420	026455	.WORD	26455
3368	030422	113226	.WORD	113226
3369				
3370				
3371				

3372 030424 000240
3373 030426
(3) 030426
(3) 030426 104401
3374 030430 000000
3375
3376
3377
3378
3379
3380
3381
3382
3383 030432
3429
(1) 030432 000000
(1) 030434 000002
(1) 030436 000004
(1) 030440 000006
(1) 030442 000010
(1) 030444 000012
(1) 030446 000014
(1) 030450 000016
(1) 030452 000020
(1) 030454 000022
(1) 030456 000024
(1) 030460 000026
(1) 030462 000030
(1) 030464 000032
(1) 030466 000034
(1) 030470 000036
(1) 030472 000040
(1) 030474 000042
(1) 030476 000044
(1) 030500 000046
(1) 030502 000050
(1) 030504 000052
(1) 030506 000054
(1) 030510 000056
(1) 030512 000060
(1) 030514 000062
(1) 030516 000064
(1) 030520 000066
(1) 030522 000070
(1) 030524 000072
(1) 030526 000074
(1) 030530 000076
(1) 030532 000100
(1) 030534 000102
(1) 030536 000104
(1) 030540 000106
(1) 030542 000110
(1) 030544 000112
(1) 030546 000114
(1) 030550 000116
(1) 030552 000120

ENDOFPROGRAM: NOP
ENDTST
L10024:
TRAP CSETST
HALT

.SBTTL DRIVE INFORMATION BUFFERS
:DRIVE INFORMATION BUFFER

.LIST ME

DRBUF:

SKCNT
RXFR1
RXFR2
WXFR1
WXFR2
ERRCNT
SFTCNT
SKECNT
DERCNT
DCRCER
WCRCER
DLTCNT
OPICNT
WVFERR
NXCNT
RETRY
BDA
BMP
FUNC
BCSADR
LSTHDR
RTYPE
SKCNT1
PRFLGS
RXFR3
WXFR3
LSTDA
DIFWD
DPHOUR
TRERR
DATCER
DOWCK
SERNM1
SERNM2
DCS
DRSEL
BBA
BSECT
RSEEK
SOFTCS
TDR

:SEEK OPERATION COUNT
:READ OPERATION COUNT (BITS) LOW ORDER
:HIGH ORDER
:WRITE OPERATION COUNT (BITS) LOW ORDER
:HIGH ORDER
:ERROR COUNT - HARD
:ERROR COUNT - SOFT
:SEEK ERROR COUNT
:DRIVE ERROR COUNT
:DATA CRC ERROR COUNT
:HEADER CRC ERROR COUNT
:DATA LATE ERROR COUNT
:OPERATION INCOMPLETE ERROR COUNT
:HEADER NOT FOUND ERROR COUNT
:NON EXISTANT MEMORY ERROR COUNT
:PRESENT RETRY NUMBER
:DISK ADDRESS CONTENTS
:PRESENT MULTIPURPOSE CONTENTS
:LAST FUNCTION LOADED
:CSR IMAGE OF LAST COMMAND
:LAST POSITION ON DISK
:ERROR ON WHICH RECOVERY IS IN PROGRESS
:SEEK COUNT LOW ORDER
:PROGRAM INTERNAL FLAGS
:READ COUNT THIRD
:WRITE COUNT THIRD
:DISK ADDRESS OF SOFT ERROR
:LAST DIFFERENCE WORD OF SEEK
:TIME DRIVE WAS DROPPED
:TRACKING ERROR COUNT

:WRITE CHECK NECESSARY
:SERIAL NUMBER OF CARTRIDGE
:SERIAL NUMBER OF CARTRIDGE
:CSR ADDRESS
:DRIVE SELECT BITS(8,9,10)
:PRESENT BUS ADDRESS CONTENTS
:POINTER TO BAD SECTOR FILE

:CSR AT TIME OF SOFT ERROR
:DRIVE TYPE FLAG (RL01 =1)

(1)	030554	000122	WRIPG	:WRITE IN PROGRESS FLAG
(1)	030556	000124	PRPOS	:PRESENT POSITION ON DISK
(1)	030560	000000	SKCNT	:SEEK OPERATION COUNT
(1)	030562	000002	RXFR1	:READ OPERATION COUNT (BITS) LOW ORDER
(1)	030564	000004	RXFR2	:READ OPERATION COUNT (BITS) HIGH ORDER
(1)	030566	000006	WXFR1	:WRITE OPERATION COUNT (BITS) LOW ORDER
(1)	030570	000010	WXFR2	:WRITE OPERATION COUNT (BITS) HIGH ORDER
(1)	030572	000012	ERRCNT	:ERROR COUNT - HARD
(1)	030574	000014	SFTCNT	:ERROR COUNT - SOFT
(1)	030576	000016	SKECNT	:SEEK ERROR COUNT
(1)	030600	000020	DERCNT	:DRIVE ERROR COUNT
(1)	030602	000022	DCRCER	:DATA CRC ERROR COUNT
(1)	030604	000024	HRCRCER	:HEADER CRC ERROR COUNT
(1)	030606	000026	DLTCNT	:DATA LATE ERROR COUNT
(1)	030610	000030	OPICNT	:OPERATION INCOMPLETE ERROR COUNT
(1)	030612	000032	HNFERR	:HEADER NOT FOUND ERROR COUNT
(1)	030614	000034	NXMCNT	:NON EXISTANT MEMORY ERROR COUNT
(1)	030616	000036	RETRY	:PRESENT RETRY NUMBER
(1)	030620	000040	BDA	:DISK ADDRESS CONTENTS
(1)	030622	000042	BMP	:PRESENT MULTIPURPOSE CONTENTS
(1)	030624	000044	FUNC	:LAST FUNCTION LOADED
(1)	030626	000046	BCSADR	:CSR IMAGE OF LAST COMMAND
(1)	030630	000050	LSTHDR	:LAST POSITION ON DISK
(1)	030632	000052	RTYPE	:ERROR ON WHICH RECOVERY IS IN PROGRESS
(1)	030634	000054	SKCNT1	:SEEK COUNT LOW ORDER
(1)	030636	000056	PRFLGS	:PROGRAM INTERNAL FLAGS
(1)	030640	000060	RXFR3	:READ COUNT THIRD
(1)	030642	000062	WXFR3	:WRITE COUNT THIRD
(1)	030644	000064	LSTDA	:DISK ADDRESS OF SOFT ERROR
(1)	030646	000066	DIFWD	:LAST DIFFERENCE WORD OF SEEK
(1)	030650	000070	DPHOUR	:TIME DRIVE WAS DROPPED
(1)	030652	000072	TRERR	:TRACKING ERROR COUNT
(1)	030654	000074	DATCER	
(1)	030656	000076	DOWCK	:WRITE CHECK NECESSARY
(1)	030660	000100	SERNM1	:SERIAL NUMBER OF CARTRIDGE
(1)	030662	000102	SERNM2	:SERIAL NUMBER OF CARTRIDGE
(1)	030664	000104	DCS	:CSR ADDRESS
(1)	030666	000106	DRSEL	:DRIVE SELECT BITS(8,9,10)
(1)	030670	000110	BBA	:PRESENT BUS ADDRESS CONTENTS
(1)	030672	000112	BSECTP	:POINTER TO BAD SECTOR FILE
(1)	030674	000114	RSEEK	
(1)	030676	000116	SOFTCS	:CSR AT TIME OF SOFT ERROR
(1)	030700	000120	TDR	:DRIVE TYPE FLAG (RL01 =1)
(1)	030702	000122	WRIPG	:WRITE IN PROGRESS FLAG
(1)	030704	000124	PRPOS	:PRESENT POSITION ON DISK
(1)	030706	000000	SKCNT	:SEEK OPERATION COUNT
(1)	030710	000002	RXFR1	:READ OPERATION COUNT (BITS) LOW ORDER
(1)	030712	000004	RXFR2	:READ OPERATION COUNT (BITS) HIGH ORDER
(1)	030714	000006	WXFR1	:WRITE OPERATION COUNT (BITS) LOW ORDER
(1)	030716	000010	WXFR2	:WRITE OPERATION COUNT (BITS) HIGH ORDER
(1)	030720	000012	ERRCNT	:ERROR COUNT - HARD
(1)	030722	000014	SFTCNT	:ERROR COUNT - SOFT
(1)	030724	000016	SKECNT	:SEEK ERROR COUNT
(1)	030726	000020	DERCNT	:DRIVE ERROR COUNT

(1)	030730	000022	DCRCER	:DATA CRC ERROR COUNT
(1)	030732	000024	HRCRCR	:HEADER CRC ERROR COUNT
(1)	030734	000026	DLTCNT	:DATA LATE ERROR COUNT
(1)	030736	000030	OPICNT	:OPERATION INCOMPLETE ERROR COUNT
(1)	030740	000032	HNFERR	:HEADER NOT FOUND ERROR COUNT
(1)	030742	000034	NXMCNT	:NON EXISTANT MEMORY ERROR COUNT
(1)	030744	000036	RETRY	:PRESENT RETRY NUMBER
(1)	030746	000040	BDA	:DISK ADDRESS CONTENTS
(1)	030750	000042	BMP	:PRESENT MULTIPURPOSE CONTENTS
(1)	030752	000044	FUNC	:LAST FUNCTION LOADED
(1)	030754	000046	BCSADR	:CSR IMAGE OF LAST COMMAND
(1)	030756	000050	LSTHDR	:LAST POSITION ON DISK
(1)	030760	000052	RTYPE	:ERROR ON WHICH RECOVERY IS IN PROGRESS
(1)	030762	000054	SKCNT1	:SEEK COUNT LOW ORDER
(1)	030764	000056	PRFLGS	:PROGRAM INTERNAL FLAGS
(1)	030766	000060	RXFR3	:READ COUNT THIRD
(1)	030770	000062	WXFR3	:WRITE COUNT THIRD
(1)	030772	000064	LSTDA	:DISK ADDRESS OF SOFT ERROR
(1)	030774	000066	DIFWD	:LAST DIFFERENCE WORD OF SEEK
(1)	030776	000070	DPHOUR	:TIME DRIVE WAS DROPPED
(1)	031000	000072	TRERR	:TRACKING ERROR COUNT
(1)	031002	000074	DATCER	
(1)	031004	000076	DOWCK	:WRITE CHECK NEEDED
(1)	031006	000100	SERNM1	:SERIAL NUMBER OF CARTRIDGE
(1)	031010	000102	SERNM2	:SERIAL NUMBER OF CARTRIDGE
(1)	031012	000104	DCS	:CSR ADDRESS
(1)	031014	000106	DRSEL	:DRIVE SELECT BITS(8,9,10)
(1)	031016	000110	BBA	:PRESENT BUS ADDRESS CONTENTS
(1)	031020	000112	BSECTP	:POINTER TO BAD SECTOR FILE
(1)	031022	000114	RSEEK	
(1)	031024	000116	SOFTCS	:CSR AT TIME OF SOFT ERROR
(1)	031026	000120	TDR	:DRIVE TYPE FLAG (RL01 =1)
(1)	031030	000122	WRIPG	:WRITE IN PROGRESS FLAG
(1)	031032	000124	PRPOS	:PRESENT POSITION ON DISK
(1)	031034	000000	SKCNT	:SEEK OPERATION COUNT
(1)	031036	000002	RXFR1	:READ OPERATION COUNT (BITS) LOW ORDER
(1)	031040	000004	RXFR2	:READ OPERATION COUNT (BITS) HIGH ORDER
(1)	031042	000006	WXFR1	:WRITE OPERATION COUNT (BITS) LOW ORDER
(1)	031044	000010	WXFR2	:WRITE OPERATION COUNT (BITS) HIGH ORDER
(1)	031046	000012	ERRCNT	:ERROR COUNT - HARD
(1)	031050	000014	SFTCNT	:ERROR COUNT - SOFT
(1)	031052	000016	SKECNT	:SEEK ERROR COUNT
(1)	031054	000020	DERCNT	:DRIVE ERROR COUNT
(1)	031056	000022	DCRCER	:DATA CRC ERROR COUNT
(1)	031060	000024	HRCRCR	:HEADER CRC ERROR COUNT
(1)	031062	000026	DLTCNT	:DATA LATE ERROR COUNT
(1)	031064	000030	OPICNT	:OPERATION INCOMPLETE ERROR COUNT
(1)	031066	000032	HNFERR	:HEADER NOT FOUND ERROR COUNT
(1)	031070	000034	NXMCNT	:NON EXISTANT MEMORY ERROR COUNT
(1)	031072	000036	RETRY	:PRESENT RETRY NUMBER
(1)	031074	000040	BDA	:DISK ADDRESS CONTENTS
(1)	031076	000042	BMP	:PRESENT MULTIPURPOSE CONTENTS
(1)	031100	000044	FUNC	:LAST FUNCTION LOADED
(1)	031102	000046	BCSADR	:CSR IMAGE OF LAST COMMAND
(1)	031104	000050	LSTHDR	:LAST POSITION ON DISK

(1)	031106	000052	RTYPE	:ERROR ON WHICH RECOVERY IS IN PROGRESS
(1)	031110	000054	SKCNT1	:SEEK COUNT LOW ORDER
(1)	031112	000056	PRFLGS	:PROGRAM INTERNAL FLAGS
(1)	031114	000060	RXFR3	:READ COUNT THIRD
(1)	031116	000062	WXFR3	:WRITE COUNT THIRD
(1)	031120	000064	LSTDA	:DISK ADDRESS OF SOFT ERROR
(1)	031122	000066	DIFWD	:LAST DIFFERENCE WORD OF SEEK
(1)	031124	000070	DPHOUR	:TIME DRIVE WAS DROPPED
(1)	031126	000072	TRERR	:TRACKING ERROR COUNT
(1)	031130	000074	DATCER	
(1)	031132	000076	DOWCK	:WRITE CHECK NECESSARY
(1)	031134	000100	SERNM1	:SERIAL NUMBER OF CARTRIDGE
(1)	031136	000102	SERNM2	:SERIAL NUMBER OF CARTRIDGE
(1)	031140	000104	DCS	:CSR ADDRESS
(1)	031142	000106	DRSEL	:DRIVE SELECT BITS(8,9,10)
(1)	031144	000110	BBA	:PRESENT BUS ADDRESS CONTENTS
(1)	031146	000112	BSECTP	:POINTER TO BAD SECTOR FILE
(1)	031150	000114	RSEEK	
(1)	031152	000116	SOFTCS	:CSR AT TIME OF SOFT ERROR
(1)	031154	000120	TDR	:DRIVE TYPE FLAG (RL01 =1)
(1)	031156	000122	WRIPG	:WRITE IN PROGRESS FLAG
(1)	031160	000124	PRPOS	:PRESENT POSITION ON DISK
(1)	031162	000000	SKCNT	:SEEK OPERATION COUNT
(1)	031164	000002	RXFR1	:READ OPERATION COUNT (BITS) LOW ORDER
(1)	031166	000004	RXFR2 HIGH ORDER
(1)	031170	000006	WXFR1	:WRITE OPERATION COUNT (BITS) LOW ORDER
(1)	031172	000010	WXFR2 HIGH ORDER
(1)	031174	000012	ERRCNT	:ERROR COUNT - HARD
(1)	031176	000014	SFTCNT	:ERROR COUNT - SOFT
(1)	031200	000016	SKECNT	:SEEK ERROR COUNT
(1)	031202	000020	DERCNT	:DRIVE ERROR COUNT
(1)	031204	000022	DCRCER	:DATA CRC ERROR COUNT
(1)	031206	000024	HRCRCR	:HEADER CRC ERROR COUNT
(1)	031210	000026	DLTCNT	:DATA LATE ERROR COUNT
(1)	031212	000030	OPICNT	:OPERATION INCOMPLETE ERROR COUNT
(1)	031214	000032	HNFERR	:HEADER NOT FOUND ERROR COUNT
(1)	031216	000034	NXMCNT	:NON EXISTANT MEMORY ERROR COUNT
(1)	031220	000036	RETRY	:PRESENT RETRY NUMBER
(1)	031222	000040	BDA	:DISK ADDRESS CONTENTS
(1)	031224	000042	BMP	:PRESENT MULTIPURPOSE CONTENTS
(1)	031226	000044	FUNC	:LAST FUNCTION LOADED
(1)	031230	000046	BCSADR	:CSR IMAGE OF LAST COMMAND
(1)	031232	000050	LSTHDR	:LAST POSITION ON DISK
(1)	031234	000052	RTYPE	:ERROR ON WHICH RECOVERY IS IN PROGRESS
(1)	031236	000054	SKCNT1	:SEEK COUNT LOW ORDER
(1)	031240	000056	PRFLGS	:PROGRAM INTERNAL FLAGS
(1)	031242	000060	RXFR3	:READ COUNT THIRD
(1)	031244	000062	WXFR3	:WRITE COUNT THIRD
(1)	031246	000064	LSTDA	:DISK ADDRESS OF SOFT ERROR
(1)	031250	000066	DIFWD	:LAST DIFFERENCE WORD OF SEEK
(1)	031252	000070	DPHOUR	:TIME DRIVE WAS DROPPED
(1)	031254	000072	TRERR	:TRACKING ERROR COUNT
(1)	031256	000074	DATCER	
(1)	031260	000076	DOWCK	:WRITE CHECK NECESSARY
(1)	031262	000100	SERNM1	:SERIAL NUMBER OF CARTRIDGE

(1)	031440	000002	RXFR1	:READ OPERATION COUNT (BITS) LOW ORDER
(1)	031442	000004	RXFR2	:HIGH ORDER
(1)	031444	000006	WXFR1	:WRITE OPERATION COUNT (BITS) LOW ORDER
(1)	031446	000010	WXFR2	:HIGH ORDER
(1)	031450	000012	ERRCNT	:ERROR COUNT - HARD
(1)	031452	000014	SFTCNT	:ERROR COUNT - SOFT
(1)	031454	000016	SKECNT	:SEEK ERROR COUNT
(1)	031456	000020	DERCNT	:DRIVE ERROR COUNT
(1)	031460	000022	DCRCER	:DATA CRC ERROR COUNT
(1)	031462	000024	HRCRCER	:HEADER CRC ERROR COUNT
(1)	031464	000026	DLTCNT	:DATA LATE ERROR COUNT
(1)	031466	000030	OPICNT	:OPERATION INCOMPLETE ERROR COUNT
(1)	031470	000032	WNFERR	:HEADER NOT FOUND ERROR COUNT
(1)	031472	000034	NXMCNT	:NON EXISTANT MEMORY ERROR COUNT
(1)	031474	000036	RETRY	:PRESENT RETRY NUMBER
(1)	031476	000040	BDA	:DISK ADDRESS CONTENTS
(1)	031500	000042	BMP	:PRESENT MULTIPURPOSE CONTENTS
(1)	031502	000044	FUNC	:LAST FUNCTION LOADED
(1)	031504	000046	BCSADR	:CSR IMAGE OF LAST COMMAND
(1)	031506	000050	LSTHDR	:LAST POSITION ON DISK
(1)	031510	000052	RTYPE	:ERROR ON WHICH RECOVERY IS IN PROGRESS
(1)	031512	000054	SKCNT1	:SEEK COUNT LOW ORDER
(1)	031514	000056	PRFLGS	:PROGRAM INTERNAL FLAGS
(1)	031516	000060	RXFR3	:READ COUNT THIRD
(1)	031520	000062	WXFR3	:WRITE COUNT THIRD
(1)	031522	000064	LSTDA	:DISK ADDRESS OF SOFT ERROR
(1)	031524	000066	DIFWD	:LAST DIFFERENCE WORD OF SEEK
(1)	031526	000070	DPHOUR	:TIME DRIVE WAS DROPPED
(1)	031530	000072	TREAR	:TRACKING ERROR COUNT
(1)	031532	000074	DATCER	
(1)	031534	000076	DOWCK	:WRITE CHECK NECESSARY
(1)	031536	000100	SERNM1	:SERIAL NUMBER OF CARTRIDGE
(1)	031540	000102	SERNM2	:SERIAL NUMBER OF CARTRIDGE
(1)	031542	000104	DCS	:CSR ADDRESS
(1)	031544	000106	DRSEL	:DRIVE SELECT BITS(8,9,10)
(1)	031546	000110	BBA	:PRESENT BUS ADDRESS CONTENTS
(1)	031550	000112	BSECPY	:POINTER TO BAD SECTOR FILE
(1)	031552	000114	RSEEK	
(1)	031554	000116	SOFTCS	:CSR AT TIME OF SOFT ERROR
(1)	031556	000120	TDR	:DRIVE TYPE FLAG (RLO1 =1)
(1)	031560	000122	WRIPG	:WRITE IN PROGRESS FLAG
(1)	031562	000124	PRPOS	:PRESENT POSITION ON DISK
(1)	031564	000000	SKCNT	:SEEK OPERATION COUNT
(1)	031566	000002	RXFR1	:READ OPERATION COUNT (BITS) LOW ORDER
(1)	031570	000004	RXFR2	:HIGH ORDER
(1)	031572	000006	WXFR1	:WRITE OPERATION COUNT (BITS) LOW ORDER
(1)	031574	000010	WXFR2	:HIGH ORDER
(1)	031576	000012	ERRCNT	:ERROR COUNT - HARD
(1)	031600	000014	SFTCNT	:ERROR COUNT - SOFT
(1)	031602	000016	SKECNT	:SEEK ERROR COUNT
(1)	031604	000020	DERCNT	:DRIVE ERROR COUNT
(1)	031606	000022	DCRCER	:DATA CRC ERROR COUNT
(1)	031610	000024	HRCRCER	:HEADER CRC ERROR COUNT
(1)	031612	000026	DLTCNT	:DATA LATE ERROR COUNT
(1)	031614	000030	OPICNT	:OPERATION INCOMPLETE ERROR COUNT

(1)	031616	000032	HNFERR	:HEADER NOT FOUND ERROR COUNT
(1)	031620	000034	NXMCNT	:NON EXISTANT MEMORY ERROR COUNT
(1)	031622	000036	RETRY	:PRESENT RETRY NUMBER
(1)	031624	000040	BDA	:DISK ADDRESS CONTENTS
(1)	031626	000042	BMP	:PRESENT MULTIPURPOSE CONTENTS
(1)	031630	000044	FUNC	:LAST FUNCTION LOADED
(1)	031632	000046	BCSADR	:CSR IMAGE OF LAST COMMAND
(1)	031634	000050	LSTHDR	:LAST POSITION ON DISK
(1)	031636	000052	RTYPE	:ERROR ON WHICH RECOVERY IS IN PROGRESS
(1)	031640	000054	SKCNT1	:SEEK COUNT LOW ORDER
(1)	031642	000056	PRFLGS	:PROGRAM INTERNAL FLAGS
(1)	031644	000060	RXFR3	:READ COUNT THIRD
(1)	031646	000062	WXFR3	:WRITE COUNT THIRD
(1)	031650	000064	LSTDA	:DISK ADDRESS OF SOFT ERROR
(1)	031652	000066	DIFWD	:LAST DIFFERENCE WORD OF SEEK
(1)	031654	000070	DPHOUR	:TIME DRIVE WAS DROPPED
(1)	031656	000072	TRERR	:TRACKING ERROR COUNT
(1)	031660	000074	DATCER	
(1)	031662	000076	DOWCK	:WRITE CHECK NECESSARY
(1)	031664	000100	SERNM1	:SERIAL NUMBER OF CARTRIDGE
(1)	031666	000102	SERNM2	:SERIAL NUMBER OF CARTRIDGE
(1)	031670	000104	DCS	:CSR ADDRESS
(1)	031672	000106	DRSEL	:DRIVE SELECT BITS(8,9,10)
(1)	031674	000110	BBA	:PRESENT BUS ADDRESS CONTENTS
(1)	031676	000112	BSECTP	:POINTER TO BAD SECTOR FILE
(1)	031700	000114	RSEEK	
(1)	031702	000116	SOFTCS	:CSR AT TIME OF SOFT ERROR
(1)	031704	000120	TDR	:DRIVE TYPE FLAG (RL01 =1)
(1)	031706	000122	WRIPG	:WRITE IN PROGRESS FLAG
(1)	031710	000124	PRPOS	:PRESENT POSITION ON DISK
3430			.NLIST ME	
3431				
3432	031712	000000	ENDBUF: .WORD 0	
3433				
3434				
3435				
3436			:QUESTIONS TO GET PARAMETERS FOR HARDWARE P-TABLE	
3437				
3438	031714		BGNMOD HRDPRM	
3439	031714		BGNHRD	
(3)	031714	000030	.WORD L10030-L\$HARD/2	
3440				
3441	031716		GPRML CNTYPE,CNT,1,YES	
(4)	031716	005130	.WORD T\$CODE	
(4)	031720	031776	.WORD CNTYPE	
(4)	031722	000001	.WORD 1	
3442	031724		GPRMA CSRMSG,CSR,0,160000,177776,YES	
(4)	031724	000031	.WORD T\$CODE	
(4)	031726	032003	.WORD CSRMSG	
(4)	031730	160000	.WORD T\$LOLIM	
(4)	031732	177776	.WORD T\$HILIM	
3443	031734		GPRMA VECMSG,VECT,0,0,776,YES	
(4)	031734	001031	.WORD T\$CODE	
(4)	031736	032052	.WORD VECMSG	
(4)	031740	000000	.WORD T\$LOLIM	
(4)	031742	000776	.WORD T\$HILIM	


```
3444 031744 GPRMD DRMSG,DRBT,0,03400,0,7,YES
(4) 031744 004032 .WORD T$CODE
(4) 031746 032061 .WORD DRMSG
(4) 031750 003400 .WORD 03400
(4) 031752 000000 .WORD T$LOLIM
(4) 031754 000007 .WORD T$HILIM
3445 031756 GPRML DRTYPE,TYPDR,1,YES
(4) 031756 003130 .WORD T$CODE
(4) 031760 032030 .WORD DRTYPE
(4) 031762 000001 .WORD 1
3446 031764 GPRMD BRMSG,PRIOR,0,340,0,7,YES
(4) 031764 002032 .WORD T$CODE
(4) 031766 032017 .WORD BRMSG
(4) 031770 000340 .WORD 340
(4) 031772 000000 .WORD T$LOLIM
(4) 031774 000007 .WORD T$HILIM
3447
3448 031776 ENDHRD
(2) .EVEN
(3) 031776 L10030:
3449
3453
3454 031776 046122 030461 000 CNTYPE: .ASCIZ /RL11/
3455 032003 102 051525 040440 CSRMSG: .ASCIZ /BUS ADDRESS/
3456 032017 102 020122 042514 BRMSG: .ASCIZ /BR LEVEL/
3457 032030 051104 053111 020105 DRTYPE: .ASCIZ /DRIVE TYPE = RL01/
3458 032052 042526 052103 051117 VECMSG: .ASCIZ /VECTOR/
3459 032061 104 044522 042526 DRMSG: .ASCIZ /DRIVE/
3460
3464
3465 032070 .EVEN
3466
3467 032070 ENDMOD
3468
3469
3470
3471 ;QUESTIONS TO GET PARAMETERS FOR SOFTWARE P-TABLE
3472
3473 032070 BGNMOD SFTPRM
3474
3475 032070 BGNSFT
(3) 032070 000215 .WORD L10031-L$SOFT/2
3476
3477 032072 GPRMD RTMSG,RLT,D,177777,0,177777,YES
(4) 032072 000052 .WORD T$CODE
(4) 032074 032771 .WORD RTMSG
(4) 032076 177777 .WORD 177777
(4) 032100 000000 .WORD T$LOLIM
(4) 032102 177777 .WORD T$HILIM
3478 032104 GPRMD SRTMSG,SRLT,D,177777,0,177777,YES
(4) 032104 031052 .WORD T$CODE
(4) 032106 032614 .WORD SRTMSG
(4) 032110 177777 .WORD 177777
(4) 032112 000000 .WORD T$LOLIM
(4) 032114 177777 .WORD T$HILIM
3479 032116 GPRML FDCHK,DCKFG,1,YES
```

(4)	032116	020130	.WORD	T\$CODE
(4)	032120	033307	.WORD	FDCHK
(4)	032122	000001	.WORD	1
3480	032124		XFERF	5\$
(5)	032124	006044	.WORD	T\$CODE
3481	032126		GPRMD	CHKLMT,CLMT,D,177777,0,128.,YES
(4)	032126	032052	.WORD	T\$CODE
(4)	032130	032633	.WORD	CHKLMT
(4)	032132	177777	.WORD	177777
(4)	032134	000000	.WORD	T\$LOLIM
(4)	032136	000200	.WORD	T\$HILIM
3482	032140		GPRMD	INMSG,TYT,D,177777,1,177777,YES
(4)	032140	005052	.WORD	T\$CODE
(4)	032142	033101	.WORD	INMSG
(4)	032144	177777	.WORD	177777
(4)	032146	000001	.WORD	T\$LOLIM
(4)	032150	177777	.WORD	T\$HILIM
3483	032152		GPRML	DRPMS,DRFLG,1,YES
(4)	032152	021130	.WORD	T\$CODE
(4)	032154	033370	.WORD	DRPMS
(4)	032156	000001	.WORD	1
3484	032160		XFERF	3\$
(5)	032160	032044	.WORD	T\$CODE
3485	032162		GPRMD	ERMSG,ELT,D,177777,0,177777,YES
(4)	032162	001052	.WORD	T\$CODE
(4)	032164	032705	.WORD	ERMSG
(4)	032166	177777	.WORD	177777
(4)	032170	000000	.WORD	T\$LOLIM
(4)	032172	177777	.WORD	T\$HILIM
3486	032174		GPRMD	SFTMSG,SEL,D,177777,0,177777,YES
(4)	032174	023052	.WORD	T\$CODE
(4)	032176	032721	.WORD	SFTMSG
(4)	032200	177777	.WORD	177777
(4)	032202	000000	.WORD	T\$LOLIM
(4)	032204	177777	.WORD	T\$HILIM
3487	032206		GPRMD	DERPMS,DCD,D,177777,0,177777,YES
(4)	032206	036052	.WORD	T\$CODE
(4)	032210	033424	.WORD	DERPMS
(4)	032212	177777	.WORD	177777
(4)	032214	000000	.WORD	T\$LOLIM
(4)	032216	177777	.WORD	T\$HILIM
3488	032220		GPRMD	SEMSG,SET,D,177777,0,177777,YES
(4)	032220	002052	.WORD	T\$CODE
(4)	032222	033003	.WORD	SEMSG
(4)	032224	177777	.WORD	177777
(4)	032226	000000	.WORD	T\$LOLIM
(4)	032230	177777	.WORD	T\$HILIM
3489	032232		GPRMD	DREMSG,DET,D,177777,0,177777,YES
(4)	032232	025052	.WORD	T\$CODE
(4)	032234	033016	.WORD	DREMSG
(4)	032236	177777	.WORD	177777
(4)	032240	000000	.WORD	T\$LOLIM
(4)	032242	177777	.WORD	T\$HILIM
3490	032244		GPRML	STLMT,OPFLG,1,YES
(4)	032244	024130	.WORD	T\$CODE
(4)	032246	033333	.WORD	STLMT

(4)	032250	000001	.WORD	1
3491	032252		XFERF	2\$
(5)	032252	013044	.WORD	T\$CODE
3492	032254		GPRMD	DAMSG,DAT,D,177777,1,177776,YES
(4)	032254	003052	.WORD	T\$CODE
(4)	032256	033031	.WORD	DAMSG
(4)	032260	177777	.WORD	177777
(4)	032262	000001	.WORD	T\$LOLIM
(4)	032264	177776	.WORD	T\$HILIM
3493	032266		GPRMD	SKMSG,SKT,D,177777,1,177776,YES
(4)	032266	004052	.WORD	T\$CODE
(4)	032270	033061	.WORD	SKMSG
(4)	032272	177777	.WORD	177777
(4)	032274	000001	.WORD	T\$LOLIM
(4)	032276	177776	.WORD	T\$HILIM
3494	032300		2\$: GPRML	CHANGE,CHFLG,1,YES
(4)	032300	010130	.WORD	T\$CODE
(4)	032302	033131	.WORD	CHANGE
(4)	032304	000001	.WORD	1
3495	032306		XFERF	1\$
(5)	032306	107044	.WORD	T\$CODE
3496	032310		GPRML	STIPMS,STIP,1,YES
(4)	032310	034130	.WORD	T\$CODE
(4)	032312	032564	.WORD	STIPMS
(4)	032314	000001	.WORD	1
3497	032316		XFERF	6\$
(5)	032316	013044	.WORD	T\$CODE
3498	032320		GPRMD	MXBUF,MXB,D,177777,3,5120.,YES
(4)	032320	011052	.WORD	T\$CODE
(4)	032322	033165	.WORD	MXBUF
(4)	032324	177777	.WORD	177777
(4)	032326	000003	.WORD	T\$LOLIM
(4)	032330	012000	.WORD	T\$HILIM
3499	032332		GPRMD	MINBUF,MNB,D,177777,3.,5120.,YES
(4)	032332	022052	.WORD	T\$CODE
(4)	032334	033176	.WORD	MINBUF
(4)	032336	177777	.WORD	177777
(4)	032340	000003	.WORD	T\$LOLIM
(4)	032342	012000	.WORD	T\$HILIM
3500	032344		6\$: GPRML	RONLY,ROF,1,YES
(4)	032344	026130	.WORD	T\$CODE
(4)	032346	032653	.WORD	RONLY
(4)	032350	000001	.WORD	1
3501	032352		GPRML	RANPAT,RAN,1,YES
(4)	032352	027130	.WORD	T\$CODE
(4)	032354	032663	.WORD	RANPAT
(4)	032356	000001	.WORD	1
3502	032360		XFERT	7\$
(5)	032360	006024	.WORD	T\$CODE
3503	032362		GPRMD	ONLONE,PAT,0,17,0,7,YES
(4)	032362	030032	.WORD	T\$CODE
(4)	032364	032673	.WORD	ONLONE
(4)	032366	000017	.WORD	17
(4)	032370	000000	.WORD	T\$LOLIM
(4)	032372	000007	.WORD	T\$HILIM
3504	032374		7\$: GPRMD	CMMSG,RDT,D,177777,0,128.,YES

(4)	032374	006052	.WORD	T\$CODE
(4)	032376	033452	.WORD	CMMMSG
(4)	032400	177777	.WORD	177777
(4)	032402	000000	.WORD	T\$LOLIM
(4)	032404	000200	.WORD	T\$HILIM
3505	032406		GPRMD	DEMSG,DDT,D,177777,0,175,YES
(4)	032406	007052	.WORD	T\$CODE
(4)	032410	032735	.WORD	DEMSG
(4)	032412	177777	.WORD	177777
(4)	032414	000000	.WORD	T\$LOLIM
(4)	032416	000175	.WORD	T\$HILIM
3506	032420		GPRMD	MXHD,MXH,D,100,0,1,YES
(4)	032420	012052	.WORD	T\$CODE
(4)	032422	033207	.WORD	MXHD
(4)	032424	000100	.WORD	100
(4)	032426	000000	.WORD	T\$LOLIM
(4)	032430	000001	.WORD	T\$HILIM
3507	032432		GPRMD	MINHD,MNH,D,100,0,1,YES
(4)	032432	013052	.WORD	T\$CODE
(4)	032434	033216	.WORD	MINHD
(4)	032436	000100	.WORD	100
(4)	032440	000000	.WORD	T\$LOLIM
(4)	032442	000001	.WORD	T\$HILIM
3508	032444		GPRML	ASK,ANS,1,YES
(4)	032444	037130	.WORD	T\$CODE
(4)	032446	032524	.WORD	ASK
(4)	032450	000001	.WORD	1
3509	032452		XFERF	15\$
(5)	032452	013044	.WORD	T\$CODE
3510	032454		GPRMD	MXCYL,MXC,D,177600,0,511.,YES
(4)	032454	014052	.WORD	T\$CODE
(4)	032456	033225	.WORD	MXCYL
(4)	032460	177600	.WORD	177600
(4)	032462	000000	.WORD	T\$LOLIM
(4)	032464	000777	.WORD	T\$HILIM
3511	032466		GPRMD	MINCYL,MNC,D,177600,0,511.,YES
(4)	032466	015052	.WORD	T\$CODE
(4)	032470	033235	.WORD	MINCYL
(4)	032472	177600	.WORD	177600
(4)	032474	000000	.WORD	T\$LOLIM
(4)	032476	000777	.WORD	T\$HILIM
3512	032500		GPRMD	MXSEC,MXS,D,77,0,39.,YES
(4)	032500	016052	.WORD	T\$CODE
(4)	032502	033245	.WORD	MXSEC
(4)	032504	000077	.WORD	77
(4)	032506	000000	.WORD	T\$LOLIM
(4)	032510	000047	.WORD	T\$HILIM
3513	032512		GPRMD	MINSEC,MNS,D,77,0,39.,YES
(4)	032512	017052	.WORD	T\$CODE
(4)	032514	033266	.WORD	MINSEC
(4)	032516	000077	.WORD	77
(4)	032520	000000	.WORD	T\$LOLIM
(4)	032522	000047	.WORD	T\$HILIM
3514	032524			
3515				
3516				


```
3517 032524          ENDSFT
      (2)             .EVEN
      (3) 032524      L10031:
3518
3519
3523
3524 032524 044103 047101 042507 ASK: .ASCIZ /CHANGE VALUES OF MXCYL & MINCYL/
3525 032564 052123 050111 046125 STIPMS: .ASCIZ %STIPULATE R/W XFER SIZE%
3526 032614 042523 045505 051040 SRTMSG: .ASCIZ /SEEK RETRY LMT/
3527 032633      043 047440 020106 CHKLMT: .ASCIZ /# OF ERR DUMPED/
3528 032653      122 020104 047117 RDONLY: .ASCIZ /RD ONLY/
3529 032663      122 047101 050040 RANPAT: .ASCIZ /RAN PAT/
3530 032673      127 044510 044103 ONLONE: .ASCIZ /WHICH ONE/
3531 032705      110 042122 042440 ERMMSG: .ASCIZ /HRD ERR LMT/
3532 032721      123 052106 042440 SFTMSG: .ASCIZ /SFT ERR LMT/
3533 032735      043 047440 020106 DEMSG: .ASCIZ /# OF DATA ERR RPT'D PER BUF/
3534 032771      122 052105 054522 RTMSG: .ASCIZ /RETRY LMT/
3535 033003      123 020113 051105 SEMSG: .ASCIZ /SK ERR LMT/
3536 033016 051104 042440 051122 DREMSG: .ASCIZ /DR ERR LMT/
3537 033031      104 052101 020101 DAMSG: .ASCIZ /DATA XFER LMT (*10(10))/
3538 033061      123 020113 046514 SKMSG: .ASCIZ /SK LMT (*10(3))/
3539 033101      12 046511 020105 INMSG: .ASCIZ /TIME BETW REPORTS (MIN)/
3540 033131      103 040510 043516 CHANGE: .ASCIZ %CHANGE SEEK, R/W PARAMETERS%
3541 033165      115 054101 054040 MXBUF: .ASCIZ /MAX XFER/
3542 033176 044515 020116 043130 MINBUF: .ASCIZ /MIN XFER/
3543 033207      115 054101 044040 MXHD: .ASCIZ /MAX HD/
3544 033216 044515 020116 042110 MINHD: .ASCIZ /MIN HD/
3545 033225      115 054101 041440 MXCYL: .ASCIZ /MAX CYL/
3546 033235      115 047111 041440 MINCYL: .ASCIZ /MIN CYL/
3547 033245      123 040524 052122 MXSEC: .ASCIZ /STARTING MAX SEC/
3548 033266 052123 051101 044524 MINSEC: .ASCIZ /STARTING MIN SEC/
3549 033307      104 052101 020101 FDCHK: .ASCIZ /DATA DMP ON DCK ERR/
3550 033333      104 047522 020120 STLMT: .ASCIZ /DROP DR ON OPER LMTS REACHED/
3551 033370 051104 050117 042040 DRPMS: .ASCIZ /DROP DR ON ERR LMTS REACHED/
3552 033424 040504 040524 046440 DERPMS: .ASCIZ /DATA MISCOMPARE LIMIT/
3553 033452 047527 042122 020123 CMMSG: .ASCIZ /WORDS PER SECTOR COMPARED ON READ/
3554
3555          .EVEN
3559
3560
3561 033514          ENDMOD
3562
3563 033514          LASTAD
      (2)             .EVEN
      (4) 033514 000000 .WORD 0
      (4) 033516 000000 .WORD 0
      (3) 033520          L$LAST::
3564
3565          000001          .END
```


F\$CLEA= 000007	10#	1016	1041											
F\$DU = 000016	10#	1075	1091											
F\$END = 000041	10#	18	21	34	171	224	325	332	447	451	457	464	473	
	480	489	496	506	513	520	527	535	632	637	650	654	693	
	695	699	703	718	720	735	959	960	1007	1013	1041	1043	1046	
	1069	1071	1073	1091	1093	1099	1186	1190	1705	1714	2180	3373	3438	
	3448	3467	3473	3517	3561									
F\$HARD= 000004	10#	3439	3448	3480	3484	3491	3495	3497	3502	3509				
F\$HW = 000013	10#	639	648											
F\$INIT= 000006	10#	737	959											
F\$JMP = 000050	10#													
F\$MOD = 000000	10#	18	21	34	171	224	325	332	447	451	632	637	650	
	654	693	695	699	703	720	735	960	1013	1043	1046	1071	1073	
	1093	1099	1186	3438	3467	3473	3561							
F\$MSG = 000011	10#	455	457	461	464	467	473	477	480	485	489	493	496	
	500	506	509	513	518	520	523	527	529	535				
F\$PROT= 000021	10#	726	730											
F\$PWR = 000017	10#													
F\$RPT = 000012	10#	705	718											
F\$SEG = 000003	10#													
F\$SOFT= 000005	10#	3475	3480	3484	3491	3495	3497	3502	3509	3517				
F\$SRV = 000010	10#	1659	1705	1709	1714	1718	2180							
F\$SUB = 000002	10#													
F\$SW = 000014	10#	656	691											
F\$TEST= 000001	10#	1190	3373											
GDDAT 002402	280#	504	2630*	2633*	2634	3095*	3098*	3100	3107					
GETDST 024432	1798	2196	2213	2220	2256	2688#	2910							
GETFNC 015314	1302	1322	1330#											
GHDR 022316	2234	2252	2280#											
GLBDAT 002242	224#													
GLBEQA 002242	34#													
GLBERR 005070	451#													
GLBSUB 013474	1099#													
GLBTXT 002516	332#													
GOERRX 020422	1906	1925	2009#											
GOFIN 020426	1910	1929	2012#											
G\$BIT = 000003	120#	1467	2689											
G\$STAT = 000004	114#	1466	2693											
G\$TFNC 015746	1437	1466#												
G\$WCDA 025764	1580	1603	2960#											
G\$CNTO= 000200	10#													
G\$DELM= 000372	10#	763	787	2239	2259	2266	2270	2670						
G\$DISP= 000003	10#													
G\$EXCP= 000400	10#													
G\$HILI= 000002	10#													
G\$LOLI= 000001	10#													
G\$NO = 000000	10#													
G\$OFFS= 000400	10#	3441	3442	3443	3444	3445	3446	3477	3478	3479	3481	3482	3483	
	3485	3486	3487	3488	3489	3490	3492	3493	3494	3496	3498	3499	3500	
	3501	3503	3504	3505	3506	3507	3508	3510	3511	3512	3513			
G\$OF SI= 000376	10#	3441	3442	3443	3444	3445	3446	3477	3478	3479	3481	3482	3483	
	3485	3486	3487	3488	3489	3490	3492	3493	3494	3496	3498	3499	3500	
	3501	3503	3504	3505	3506	3507	3508	3510	3511	3512	3513			
G\$PRMA= 000001	10#	3442	3443											
G\$PRMD= 000002	10#	3444	3446	3477	3478	3481	3482	3485	3486	3487	3488	3489	3492	
	3493	3498	3499	3503	3504	3505	3506	3507	3510	3511	3512	3513		

GSPRML= 000000	10#	344,1	3445	3479	3483	3490	3494	3496	3500	3501	3508			
GSRADA= 000140	10#													
GSRADB= 000000	10#													
GSRADD= 000040	10#	3477	3478	3481	3482	3485	3486	3487	3488	3489	3492	3493	3498	
	3499	3504	3505	3506	3507	3510	3511	3512	3513					
GSRADL= 000120	10#	3441	3445	3479	3483	3490	3494	3496	3500	3501	3508			
GSRADO= 000020	10#	3442	3443	3444	3446	3503								
GSXFER= 000004	10#	3480	3484	3491	3495	3497	3502	3509						
GSYES = 000010	10#	3441	3442	3443	3444	3445	3446	3477	3478	3479	3481	3482	3483	
	3485	3486	3487	3488	3489	3490	3492	3493	3494	3496	3498	3499	3500	
	3501	3503	3504	3505	3506	3507	3508	3510	3511	3512	3513			
HCE = 040000	97#	2206												
HCRC = 004000	107#	1953	1971											
HCR CER= 000024	56#	1181	1927*	3429										
HDHOME 025670	795	2522	2755	2775	2932#									
HDRFND 002340	263#	1507	1766	1847	1904	1923	2007	2171*	2843	2862	3164*	3177*	3186*	
	3202*													
HEAD = 000100	124#	1502	1521	1549	1551	1563	2898							
HINUM 002260	234#	817*	2711	2719	2723*									
HNF = 010000	108#	1870												
HNFERR= 000032	59#	1181	1908*	3429										
HOE = 100000 G	36#													
HOUR 002416	287#	546	551	1164	2556									
HPTCOD 010640 G	637#													
HRDPRM 031714 G	3438#													
HWSEC 003720	384#	2445												
IBE = 010000 G	36#													
IDU = 000040 G	36#													
IER = 020000 G	36#													
ILLEG 003770	387#	868												
INBAD 025550	2871	2903#												
INCALL 002476	314#	744*	1034*	1077	1770	2167*	2540*	2554*						
INIEND 012674	808	931	935	958#										
INITCO 011046 G	735#													
INMSG 033101	3482	3539#												
INSMEM 004711	419#	941												
INTEN = 000100	93#	539	1023	1031	1627	1851	1918	2162						
INTERV 002406	282#	1277	1279*	1684*										
INTR1 017116 G	920	1718#	1754#											
INTR2 017126	925	1757#												
ISDRST 024446	794	1873	2210	2212	2219	2237	2265	2408	2461	2691#	2805	2849	2868	
	2872													
ISR = 000100 G	36#													
ISSUE 016542	1412	1448	1459	1468	1569	1574	1595	1612	1622#					
IXE = 004000 G	36#													
ISAU = 000041	10#	1048#	1069#											
ISAUTO= 000041	10#	972#	1007#											
ISCLN = 000041	10#	1016#	1041#											
ISDU = 000041	10#	1075#	1091#											
ISHRD = 000041	3439#	3448#												
ISINIT= 000041	10#	737#	959#											
ISMOD = 000041	10#	18#	21#	34#	171#	224#	325#	332#	447#	451#	632#	637#	650#	
	654#	693#	695#	699#	703#	720#	735#	960#	1013#	1043#	1046#	1071#	1073#	
	1093#	1099#	1186#	3438#	3467#	3473#	3561#							
ISMSG = 000041	10#	455#	457#	461#	464#	467#	473#	477#	480#	485#	489#	493#	496#	
	500#	506#	509#	513#	518#	520#	523#	527#	529#	535#				

MSHAPT	19#														
MSHNP	19#														
MSINCR	18#	34#	224#	332#	451#	455#	457#	461#	463#	464#	467#	469#	472#	473#	477#
	479#	480#	485#	488#	489#	493#	494#	496#	500#	504#	505#	506#	509#	512#	513#
	518#	519#	520#	523#	527#	529#	533#	534#	535#	546#	547#	548#	551#	552#	556#
	557#	576#	637#	639#	654#	656#	695#	703#	705#	706#	718#	726#	735#	737#	739#
	741#	749#	754#	757#	760#	762#	764#	765#	771#	772#	774#	810#	831#	868#	887#
	901#	920#	925#	938#	941#	959#	972#	978#	983#	994#	1000#	1007#	1013#	1016#	1018#
	1019#	1024#	1032#	1036#	1040#	1041#	1046#	1048#	1060#	1069#	1073#	1075#	1091#	1099#	1120#
	1124#	1164#	1165#	1171#	1174#	1176#	1177#	1178#	1179#	1180#	1181#	1182#	1190#	1202#	1250#
	1251#	1266#	1282#	1641#	1659#	1709#	1718#	1804#	1820#	1821#	1857#	1892#	1898#	2042#	2060#
	2087#	2091#	2099#	2121#	2131#	2156#	2188#	2198#	2233#	2251#	2274#	2553#	2561#	2563#	2564#
	2586#	2611#	2642#	2656#	2676#	2751#	2912#	2986#	3064#	3072#	3107#	3113#	3123#	3126#	3130#
	3137#	3373#	3438#	3439#	3473#	3475#									
MSLDRO	739#	749#	754#	762#	764#	765#	774#	810#	831#	887#	1000#	1019#	1024#	1032#	1036#
	1060#	1202#	1251#	2060#	2553#	2586#									
MSMCHI	10#														
MSMCLO	10#														
MSPOP	21#	171#	325#	447#	457#	464#	473#	480#	489#	496#	506#	513#	520#	527#	535#
	632#	648#	650#	691#	693#	699#	718#	720#	730#	959#	960#	1007#	1041#	1043#	1069#
	1071#	1091#	1093#	1186#	1705#	1714#	2180#	3373#	3448#	3467#	3517#	3561#			
MSPRIN	463#	469#	472#	479#	488#	494#	504#	505#	512#	519#	533#	534#	546#	547#	548#
	551#	552#	556#	557#	576#	706#	771#	772#	901#	941#	983#	994#	1164#	1165#	1171#
	1174#	1176#	1177#	1178#	1179#	1180#	1181#	1182#	1250#	1857#	2233#	2251#	2561#	2563#	2564#
	2751#	2986#	3064#	3072#	3107#	3113#	3123#	3126#	3130#	3137#					
MSPUSH	18#	34#	224#	332#	451#	455#	461#	467#	477#	485#	493#	500#	509#	518#	523#
	529#	637#	639#	654#	656#	695#	703#	705#	726#	735#	737#	972#	1013#	1016#	1046#
	1048#	1073#	1075#	1099#	1190#	1659#	1709#	1718#	3438#	3439#	3473#	3475#			
MSPUT	463#	469#	472#	479#	488#	494#	504#	505#	512#	519#	533#	534#	546#	547#	548#
	551#	552#	556#	557#	576#	706#	760#	771#	772#	901#	920#	925#	941#	978#	983#
	994#	1018#	1120#	1124#	1164#	1165#	1171#	1174#	1176#	1177#	1178#	1179#	1180#	1181#	1182#
	1250#	1857#	2233#	2251#	2561#	2563#	2564#	2751#	2986#	3064#	3072#	3107#	3113#	3123#	3126#
	3130#	3137#													
MSPUT1	463#	469#	472#	479#	488#	494#	504#	505#	512#	519#	533#	534#	546#	547#	548#
	551#	552#	556#	557#	576#	706#	760#	771#	772#	901#	920#	925#	941#	978#	983#
	994#	1018#	1120#	1124#	1164#	1165#	1171#	1174#	1176#	1177#	1178#	1179#	1180#	1181#	1182#
	1250#	1857#	2233#	2251#	2561#	2563#	2564#	2751#	2986#	3064#	3072#	3107#	3113#	3123#	3126#
	3130#	3137#													
MSRADI	3441#	3442#	3443#	3444#	3445#	3446#	3477#	3478#	3479#	3481#	3482#	3483#	3485#	3486#	3487#
	3488#	3489#	3490#	3492#	3493#	3494#	3496#	3498#	3499#	3500#	3501#	3503#	3504#	3505#	3506#
	3507#	3508#	3510#	3511#	3512#	3513#									
MSRNRO	749#	754#	831#	938#	1060#										
MSSETS	18#	34#	224#	332#	451#	455#	461#	467#	477#	485#	493#	500#	509#	518#	523#
	529#	637#	639#	654#	656#	695#	703#	705#	726#	735#	737#	972#	1013#	1016#	1046#
	1048#	1073#	1075#	1099#	1190#	1659#	1709#	1718#	3438#	3439#	3473#	3475#			
MSSVC	457#	463#	464#	469#	472#	473#	479#	480#	488#	489#	494#	496#	504#	505#	506#
	512#	513#	519#	520#	527#	533#	534#	535#	546#	547#	548#	551#	552#	556#	557#
	576#	706#	718#	739#	741#	749#	754#	757#	760#	762#	764#	765#	771#	772#	774#
	810#	831#	868	887#	901#	920#	925#	938#	941#	959#	978#	983#	994#	1000#	1007#
	1018#	1019#	1024#	1032#	1036#	1040#	1041#	1060#	1069#	1091#	1120#	1124#	1164#	1165#	1171#
	1174#	1176#	1177#	1178#	1179#	1180#	1181#	1182#	1202#	1250#	1251#	1266	1282#	1641	1804
	1820	1821#	1857#	1892	1898	2042	2060#	2087	2091	2099	2121	2131	2156	2188	2198
	2233#	2251#	2274#	2553#	2561#	2563#	2564#	2586#	2611	2642	2656	2676	2751#	2912	2986#
	3064#	3072#	3107#	3113#	3123#	3126#	3130#	3137#	3373#						
MSTLAB	457#	463#	464#	469#	472#	473#	479#	480#	488#	489#	494#	496#	504#	505#	506#
	512#	513#	519#	520#	527#	533#	534#	535#	546#	547#	548#	551#	552#	556#	557#

	576#	706#	718#	739#	741#	749#	754#	757#	760#	762#	764#	765#	771#	772#	774#
	810#	831#	868#	887#	901#	920#	925#	938#	941#	959#	978#	983#	994#	1000#	1007#
	1018#	1019#	1024#	1032#	1036#	1040#	1041#	1060#	1069#	1091#	1120#	1124#	1164#	1165#	1171#
	1174#	1176#	1177#	1178#	1179#	1180#	1181#	1182#	1202#	1250#	1251#	1266#	1282#	1641#	1804#
	1820#	1821#	1857#	1892#	1898#	2042#	2060#	2087#	2091#	2099#	2121#	2131#	2156#	2188#	2198#
	2233#	2251#	2274#	2553#	2561#	2563#	2564#	2586#	2611#	2642#	2656#	2676#	2751#	2912#	2986#
	3064#	3072#	3107#	3113#	3123#	3126#	3130#	3137#	3373#						
MSTSTL	457#	463#	464#	469#	472#	473#	479#	480#	488#	489#	494#	496#	504#	505#	506#
	512#	513#	519#	520#	527#	533#	534#	535#	546#	547#	548#	551#	552#	556#	557#
	576#	706#	718#	739#	741#	749#	754#	757#	760#	762#	764#	765#	771#	772#	774#
	810#	831#	868#	887#	901#	920#	925#	938#	941#	959#	978#	983#	994#	1000#	1007#
	1018#	1019#	1024#	1032#	1036#	1040#	1041#	1060#	1069#	1091#	1120#	1124#	1164#	1165#	1171#
	1174#	1176#	1177#	1178#	1179#	1180#	1181#	1182#	1202#	1250#	1251#	1266#	1282#	1641#	1804#
	1820#	1821#	1857#	1892#	1898#	2042#	2060#	2087#	2091#	2099#	2121#	2131#	2156#	2188#	2198#
	2233#	2251#	2274#	2553#	2561#	2563#	2564#	2586#	2611#	2642#	2656#	2676#	2751#	2912#	2986#
	3064#	3072#	3107#	3113#	3123#	3126#	3130#	3137#	3373#						
MSWORD	19#	697#	868#	1266#	1641#	1804#	1820#	1892#	1898#	2042#	2087#	2091#	2099#	2121#	2131#
	2156#	2188#	2198#	2611#	2642#	2656#	2676#	2912#	3441#	3442#	3443#	3444#	3445#	3446#	3477#
	3478#	3479#	3480#	3481#	3482#	3483#	3484#	3485#	3486#	3487#	3488#	3489#	3490#	3491#	3492#
	3493#	3494#	3495#	3496#	3497#	3498#	3499#	3500#	3501#	3502#	3503#	3504#	3505#	3506#	3507#
	3508#	3509#	3510#	3511#	3512#	3513#	3563								
MSXFER	3480#	3484#	3491#	3495#	3497#	3502#	3509#								
POINTE	15														
PRINTB	463	469	472	479	488	494	504	505	512	519	533	534	546	547	548
	551	552	556	557	576	2233	2251	3064	3072	3107	3113	3123	3126	3130	3137
PRINTF	771	772	901	941	983	994	1250	1857	2561	2563	2564	2751	2986		
PRINTS	706	1164	1165	1171	1174	1176	1177	1178	1179	1180	1181	1182			
READBU	757														
READEF	774	810	887												
REQTIM	214#	807	934												
SETPRI	739	762	764	1019	1202	1251	2060	2586							
SETVEC	760	920	925	978	1018	1120	1124								
STARS	1191	1200	1356	1361	1371	1375	1383	1388	1398	1401	1420	1426	1442	1444	1450
	1452	1462	1464	1470	1472	1553	1555	1631	1635	2293	2297	2374	2402	2530	2534
	2702	2704	2731	2742	2928	2930	2950	2958	3027	3031	3159	3162	3182	3184	3208
	3209	3220	3221												
SVC	8#	10													
WAITMS	179#	763	787	2239	2259	2266	2270								
WAITUS	189#	2670													
XFERF	3480	3484	3491	3495	3497	3509									
XFERT	3502														

. ABS. 033520 000

ERRORS DETECTED: 0

,CZRLKB.LST/CRF=SVC33/ML,CZRLKB.MAC
RUN-TIME: 109 107 10 SECONDS
RUN-TIME RATIO: 657/228=2.8
CORE USED: 16K (31 PAGES)