

IDENTIFICATION

PRODUCT CODE: AC-E246B-MC  
PRODUCT NAME: CZRLERO RLO1/RLV11 PERF EXERCISER  
DATE CREATED: 11-OCT-78  
MAINTAINER: DIAGNOSTIC ENGINEERING  
AUTHOR: D. DEKNIS

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

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

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

COPYRIGHT (C) 1977, 1978, DIGITAL EQUIPMENT CORPORATION

## TABLE OF CONTENTS

1.0	GENERAL INFORMATION
1.1	PROGRAM ABSTRACT
1.2	SYSTEM 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 SIX STEPS OF EXECUTION
2.1.2	SAMPLE RUN-THROUGH
2.2	HOW TO CREATE A CHAINABLE FILE
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
4.0	PERFORMANCE AND PROGRESS REPORTS
5.0	DEVICE INFORMATION TABLES
6.0	TEST SUMMARIES

## 1.0 GENERAL INFORMATION

## 1.1 PROGRAM ABSTRACT

1.1.1 STRUCTURE OF PROGRAM

THIS DIAGNOSTIC OCCUPIES 14.5K WORDS OF MEMORY AND 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 "CREATE CORE IMAGE" COMMAND BELOW FOR DETAILS OF CHAINING PROCEDURE). IT IS A SINGLE PROGRAM FROM THE STANDPOINT OF THE DIAGNOSTIC USER, BUT WE HAVE INCORPORATED INTO IT A CONTROL MODULE WHICH WILL LATER BE RELEASED INDEPENDENTLY AS A DIAGNOSTIC SUPERVISOR.

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

THE SUPERVISOR CODING FOLLOWS IMMEDIATELY THE DIAGNOSTIC TEST CODING, BUT THE SUPERVISOR LISTING HAS BEEN SUPPRESSED FOR GENERAL DISTRIBUTION. A LIMITED DISTRIBUTION HAS BEEN MADE TO FIELD SERVICE OF THE SUPERVISOR ASSEMBLY LISTING, AND IT MAY BE CONSULTED IN EVENT OF A SOFTWARE PROBLEM.

## 1.1.2 DIAGNOSTIC INFORMATION

THE RL11/RLV11 RLO1 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, THE DRIVES ARE RANDOMLY PICKED AND GIVEN A RANDOM FUNCTION OF SEEK, GET STATUS, READ HEADER, READ OR WRITE.

## 1.2 SYSTEM REQUIREMENTS

1.2.1 HARDWARE REQUIREMENTS

PDP-11/LSI-11 PROCESSOR WITH 16K OR MORE OF MEMORY  
 CONSOLE DEVICE (LA30, LA36, VT50, ETC.)  
 RL11/RLV11 CONTROLLER(S)  
 1 - 8 RLO1 DRIVES  
 1 - 8 RLO1K CARTRIDGES WITH BAD SECTOR FILE  
 KW11P, KW11L (OPTIONAL)  
 LINEPRINTER(OPTIONAL)

1.2.2 SOFTWARE REQUIREMENTS

CXRLEB0 RL11/RLV11 RLO1 EXERCISER  
 (FORMERLY MD-11-DZRUE-A)

## 1.3 RELATED DOCUMENTS AND STANDARDS

RL01 USERS MANUAL (EK-RL01-UG-PPE)  
 XXDP USERS MANUAL

## 1.4 DIAGNOSTIC HIERARCHY PREREQUISITES

THE RL01 SUBSYSTEM SHOULD HAVE SUCCESSFULLY RUN THE FOLLOWING PROGRAMS:

CZRLBA0	RL11/RLV11	RL01	CONTROLLER TEST (PART 1)
CZRLBBO	RL11/RLV11	RL01	CONTROLLER TEST (PART 2)
CVRLAAG	RLV11	RL01	DISKLESS TEST (RLV11 ONLY)
CZRLCBO	RL01	DRIVE	TEST (PART 1)
CZRLDBO	RL01	DRIVE	TEST (PART 2)

## 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 SIX STEPS OF EXECUTION

THIS DIAGNOSTIC SHOULD BE LOADED AND STARTED USING NORMAL XXDP PROCEDURES. THE START COMMAND SHOULD NOT SPECIFY AN ADDRESS, BECAUSE THE DIAGNOSTIC HAS THE PROPER TRANSFER ADDRESS CODED INTO IT.

WHEN THIS DIAGNOSTIC IS STARTED, THE FOLLOWING STEPS WILL OCCUR:

\*\*\*\*\*  
 \* STEP 1 \*  
 \*\*\*\*\*

A SHORT SERIES OF "HARDCORE QUESTIONS" WILL BE ASKED:

QUESTION	MEANING
L-CLK (L) N ?	IS THERE AN L-CLOCK?
P-CLK (L) N ?	" " " P-CLOCK?
50HZ (L) N ?	IS THE POWER 50 CYCLES (AS IN EUROPE)?
LSI (L) N ?	IS MACHINE AN LSI?
LPT (L) N ?	IS THERE A LINE PRINTER?
MEM (K) (D) 16 ?	HOW MANY K OF MEMORY ARE THERE?

THE DEFAULTS (SHOWN AFTER EACH QUESTION) CAN BE SELECTED BY HITTING CARRIAGE RETURN. IT IS POSSIBLE THAT NOT ALL OF THE QUESTIONS WILL BE ASKED: FOR EXAMPLE, IF YOU SAY "YES" TO THE L-CLOCK QUESTION, THE P-CLOCK QUESTION WILL NOT BE ASKED.

IF NEITHER P OR L CLOCK ARE ANSWERED YES THE OPERATOR WILL BE ASKED TO TYPE TWO CHARACTERS 4 SECONDS APART.

\*\*\*\*\*  
\* STEP 2 \*  
\*\*\*\*\*

WHEN YOU HAVE ANSWERED ALL THE HARDCORE QUESTIONS, THE DIAGNOSTIC WILL ISSUE THE PROMPT "DS-B>". 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 "DS-B>" 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:

LOE	LOOP ONE ERROR
HOE	HALT ON ERROR
IER	INHIBIT ERROR PRINTOUT

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

\*\*\*\*\*  
\* STEP 3 \*  
\*\*\*\*\*

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

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

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 6 \*  
 \*\*\*\*\*

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 DS-B).

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 ENLESSLY 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 OCCURED.

### 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 REISSUED.

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 2, 3, 4, 5, AND 6 AGAIN)
2. ISSUE A "RESTART" COMMAND (SAME AS START COMMAND EXCEPT THAT THE HARDWARE QUESTIONS ARE NOT ASKED)
3. ISSUE A "CONTINUE" COMMAND (EXECUTION WILL RESUME AT THE BEGINNING OF THE PARTICULAR HARDWARE TEST (MOST DIAGNOSTICS CONSIST OF A NUMBER OF THESE) THAT IT WAS IN WHEN THE ERROR HALT OCCURED. NO QUESTIONS ASKED.
4. ISSUE A "PROCEED" COMMAND: EXECUTION WILL RESUME AT THE INSTRUCTION FOLLOWING THE ERROR REPORT (THIS IS A SPECIAL COMMAND AND CAN BE ISSUED ONLY AT A HALT 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:

.R DZRKXX	BY
DZRKXX	WHOM
L-CLK (L) N ? Y	ENTERED:
50HZ (L) N ?	D
LSI (L) N ?	D
LPT (L) N ?	D,0
MEM (K) (D) 16 ?	D
DS-B>STA/PASS:1/FLAGS:HOE	D
# UNITS (D) ? 2	D,0
UNIT 1	D,0
CSR (O) ?	D,0
VECTOR (O) ?	D,0
BR LEVEL (O) ?	D,0
DRIVE (O) ? 0	D,0
UNIT 2	D,0
CSR (O) ?	D,0
VECTOR (O) ?	D,0
BR LEVEL (O) ?	D,0
DRIVE (O) ? 1	D,0
CHANGE SW (L) ? N	D,0
DZRKXX HARD ERR 00004 TST 003 SUB 002 PC:004130	D,0
ERR HLT	D
DS-B>PRO/FLAGS:IER:LOE:HOE=0	D,0
*****	
AT THIS POINT THE DIAGNOSTIC IS LOOPING ON THE	
ERROR WITHOUT PRINTING ANYTHING. YOU CAN SCOPE	
THE ERROR UNTIL YOU HAVE LOCATED IT, THEN ^C OUT	
*****	
^C	
DS-B>CON/FLAGS:HOE:IER:LOE=0	0
CHANGE SW (L) ? N	D,0
DZRKXX EOP 1	D,0
DS-B>RESTART/PASS:1	D
CHANGE SW (L) ? N	D,0
-----	D,0
-----	
-----	
-----	
-----	

## 2.2 HOW TO CREATE A CHAINABLE FILE

THE DIAGNOSTIC AS RECEIVED FROM RELEASE ENGINEERING CANNOT BE RUN IN CHAIN MODE. THAT IS WHY IT BEARS THE EXTENSION "BIN" INSTEAD OF "RIC". THERE IS A WAY, HOWEVER, TO CREATE A CHAINABLE PROGRAM FROM WHAT YOU'VE GOT.

IT CONSISTS OF RUNNING THE PROGRAM WITH THE SPECIAL COMMAND "CCI" ISSUED WHERE YOU WOULD NORMALLY ISSUE A START COMMAND (TO THE PROMPT DS-B>). THIS COMMAND CAUSES THE DIAGNOSTIC TO GO THRU ALL THE QUESTIONS AND ANSWERS AND THEN TO HALT, JUST WHERE IT WOULD ORDINARILY BEGIN EXECUTION OF THE HARDWARE TEST CODE. AT THIS POINT YOU CAN DUMP THE PROGRAM AS IT SITS IN CORE TO THE LOAD MEDIUM, WITH THE NEW EXTENSION "RIC".

HERE IS A SAMPLE DIALOGUE TO ACCOMPLISH THIS:

```
.R UPD2
RESTART: XXXXX
*CLR
*LOAD DIAG.BIN
XFER:200 CORE:0,60602
*START 200
L-CLK (L) N ?
-----
-----
```

```
DS-B>CCI
# UNITS (D) ? 4
-----
-----
```

```
CHANGE SW (L) ? N
PTAB END: 60632
```

```
*****
*AT THIS POINT THE MACHINE HALTS AND*
*YOU MUST RESTART AT ADDRESS XXXXX*
*****
```

```
*HICORE 60632
CORE: 0,60632
*DUMP DKO: DIAG.RIC
```

THE RESULT OF DOING THIS IS THAT YOU CAN NOW BUILD AN XXDP CHAIN FILE CONTAINING THE XXDP COMMAND

```
.R DIAG.RIC
```

AND THE DIAGNOSTIC WILL EXECUTE WITHOUT MANUAL INTERVENTION, USING THE ANSWERS THAT YOU GAVE IT WHEN YOU DID THE CCI COMMAND.

2.3 DETAILS OF COMMANDS AND SYNTAX

2.3.1 TABLE OF COMMAND VALIDITY

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

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

2.3.2 COMMAND SYNTAX

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

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 IF 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 THE 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 REEXECUTED. 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

\*\*\*\*\*  
 CCI/TEST:TEST-LIST/PASS:PASS-CNT/FLAGS:FLAG-LIST/EOP:EOP-INCR  
 \*\*\*\*\*

THE DIAGNOSTIC EXECUTES THRU ALL OPERATOR DIALOGUE AND HALTS AT THE HARDWARE TEST CODE. NOW THE OPERATOR CAN DUMP THE CORE IMAGE TO THE MEDIUM WITH A BIC EXTENSION.

THE BIC FILE MUST BE HANDLED DIFFERENTLY DEPENDING ON WHETHER IT IS RUN MANUALLY OR IN CHAIN MODE. IF RUN MANUALLY IT CAN BE INVOKED EITHER WITH A "START" (IN WHICH CASE IT WILL BEHAVE LIKE THE BIN FILE; THE PRE-GENERATED ANSWERS TO OPERATOR QUESTIONS WILL BE IGNORED) OR WITH A "RESTART" (IN WHICH CASE THE PRE-GENERATED OPERATOR ANSWERS WILL BE USED).

IF RUN IN CHAIN MODE, AUTOMATIC EXECUTION WILL COMMENCE IMMEDIATELY FROM THE XXDP COMMAND ".P DIAG". THE COMMAND PROMPT "DS-R>" WILL NOT BE ISSUED.

ANY SWITCHES SPECIFIED ON THE CCI COMMAND WILL CARRY OVER WHEN THE BIC FILE IS RUN IN CHAIN MODE (EXCEPT THAT UAM IS ALWAYS SET THERE) BUT WILL NOT CARRY OVER WHEN IT IS RUN MANUALLY.

TO DO A CCI ON A FULL SIZED DIAGNOSTIC (14.5K WORDS), A MACHINE SIZE LARGER THAN 16K IS REQUIRED. THE EXACT SIZE NEEDED DEPENDS ON WHICH UTILITY IS USED TO EXECUTE THE DIAGNOSTIC AT CCI TIME.

\*\*\*\*\*  
 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 ISP (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, SAY) 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.



ON THE FIRST TRIP THRU THE QUESTIONS, ALL OF THE SLOTS IN ALL OF THE P-TABLES ARE FILLED. IF THE OPERATOR TYPES IN LESS THAN N EXPLICIT VALUES IN RESPONSE TO A PARTICULAR QUESTION, THESE VALUES ARE PLACED IN THE P-TABLES (ONE VALUE GOING INTO THE PROPER SLOT OF EACH P-TABLE BEGINNING WITH THE FIRST P-TABLE) UNTIL THE STRING OF VALUES IS EXHAUSTED. THE LAST VALUE THAT SLOT IN THE REMAINING P-TABLES.

ON SUBSEQUENT TRIPS THRU THE QUESTIONS, THE SAME PROCESS IS CARRIED OUT, EXCEPT THAT THE EARLIEST P-TABLE NOT TO HAVE RECEIVED AN EXPLICIT VALUE IN ANY OF ITS SLOTS NOW ASSUMES THE ROLE THAT TABLE NUMBER ONE PLAYED IN THE FIRST TRIP.

THE SERIES OF QUESTIONS IS REISSUED UNTIL AT LEAST ONE QUESTION HAS RECEIVED N EXPLICIT VALUES FROM THE OPERATOR.

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 64 UNITS, AND THAT THERE ARE THREE HARDWARE PARAMETERS FOR EACH (THREE SLOTS IN THE P-TABLE, THREE HARDWARE QUESTIONS IN THE DIALOGUE). LET THE DESIRED VALUE FOR THE FIRST PARAMETER BE THE NUMBER 75 FOR ALL 64 TABLES. LET THE DESIRED VALUE FOR THE SECOND PARAMETER BE EQUAL TO THE UNIT NUMBER (1,2,3,...,64) EXCEPT FOR UNIT 50, WHICH SHOULD RECEIVE THE VALUE 49. LET THE DESIRED VALUE FOR THE THIRD PARAMETER BE THE NUMBER 76 FOR THE FIRST 20 UNITS AND THE NUMBER 77 FOR THE LAST 44 UNITS.

THE FOLLOWING DIALOGUE WOULD ACCOMPLISH THIS GOAL:

# UNITS (D) ? 64

UNIT 1  
 <QUESTION 1> ? 75  
 <QUESTION 2> ? 1-20  
 <QUESTION 3> ? 76

UNIT 21  
 <QUESTION 1> ?  
 <QUESTION 2> ? 21-49,,51-64  
 <QUESTION 3> ? 77

THE FIRST TIME THE SERIES IS ASKED, SLOT ONE RECEIVES A 75 IN ALL 64 TABLES. SLOT TWO RECEIVES THE VALUES 1, 2, 3, ..., 20 IN TABLES 1 THRU 20 AND A CONSTANT 20 IN TABLES 21 THRU 64. SLOT THREE RECEIVES A CONSTANT 76 IN ALL 64 TABLES.

THE SECOND TIME THRU THE SERIES, TABLES 21 THRU THE END ARE GOING TO BE AFFECTED (NOTE THAT THIS PIECE OF INFORMATION IS PRINTED OUT FOR THE OPERATOR IN THE FORM "UNIT XX" AT THE BEGINNING OF EACH SERIES). QUESTION 1 IS RESPONDED TO BY A <CR>, SO SLOT ONE STAYS A CONSTANT 75 IN TABLES 21 THRU 64, SINCE NO NEW EXPLICIT VALUES ARE TYPED IN. SLOT TWO GETS THE VALUES 21, 22, 23, ..., 49 IN TABLES 21 THRU 49, AND GETS A 49 IN SLOT 50, AND GETS THE VALUES 51, 52, 53, ..., 64 IN TABLES 51 THRU 64. SLOT THREE GETS THE VALUE 77 IN TABLES 21 THRU 64.

THE DIALOGUE IS TERMINATED WHEN THE SOFTWARE RECOGNIZES THAT 64 EXPLICIT VALUES HAVE BEEN GIVEN FOR AT LEAST ON QUESTION (NAMELY QUESTION 2).

## 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) 330?

ANSWER WITH THE INTERRUPT VECTOR OF THE CONTROLLER.

BR LEVEL (O) 5?

ANSWER WITH THE INTERRUPT PRIORITY OF THE CONTROLLER.

DRIVE (O) 0?

ANSWER WITH THE DRIVE(S) CONNECTED TO 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 X?

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 X?

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 X?

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

THIS IS THE NUMBER OF MISCOMPARES THAT WILL BE PRINTED.

LIMITS 0 - 128

## TIME BETW REPORTS (MIN) X?

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

LIMITS 1 - 65,535

DROP DR ON ERR LMTS REACHED X?

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 X?

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 X?

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 X?

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

LIMITS 1 - 65,535

SK ERR LMT X?

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

LIMITS 1 - 65,535

DR ERR LMT X?

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

LIMITS 1 - 65,535

DROP DR ON OPER LMTS REACHED X?

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)) X?

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

LIMITS 1 - 65,535

SK LMT (\*10(3)) X?

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 X?

THE NORMAL OPERATION IS TO SEEK AND TRANSFER ON THE ENTIRE CARTRIDGE, CYLINDERS 0 - 255, 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 X?

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 X?

REPRESENTS THE MAXIMUM AMOUNT OF WORDS TO READ OR WRITE

LIMITS 3 - 5120

MIN XFER X?

REPRESENTS THE MINIMUM AMOUNT OF WORDS TO READ OR WRITE

## LIMITS 3 - 5120

RD ONLY X?

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 X?

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 X?

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 - 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

WR CHK X?

DO YOU WISH TO PERFORM A WRITE CHECK AFTER EACH WRITE OPERATION

LIMITS Y OR N

WORDS PER SECTOR COMPARED ON READ X?

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 APE 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 X?

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 X?

REPRESENTS MAXIMUM HEAD TO USE IN SEEK OPERATIONS.

LIMITS 0 - 1

MIN HD X?

REPRESENTS MINIMUM HEAD TO USE IN SEEK OPERATIONS

LIMITS 0 - 1

MAX CYL X?

MAXIMUM INNER CYLINDER TO BE USED IN SEEK OPERATIONS.

LIMITS 0 - 255

MIN CYL X?

MINIMUM OUTER CYLINDER TO BE USED IN SEEK OPERATIONS.

LIMITS 0 - 255

MAX SEC X?

MAXIMUM SECTOR TO START TRANSFER ON

LIMITS 0 - 39

MIN SEC X?

MINIMUM SECTOR TO START TRANSFER ON

LIMITS 0 - 39

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

CHK DRDY X?

ON START UP IF THIS QUESTION IS ANSWERED YES THE PROGRAM WILL NOT TEST ANY DRIVES THAT DO NOT HAVE DRIVE READY HIGH.

LIMITS                    Y OR N

### 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

DR DID NOT REC'R FROM PWR UP

DRIVE DID NOT COME BACK UP AFTER A POWER FAILURE

DATA DMP - DATA CHECK/GARRBLED 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 ARE NOT LEGAL I.E. MORE THAN TWO  
CONTROLLERS  
VECTORS FOR A CONTROLLER NOT CONSISTANT  
MORE THAN TWO VECTORS.

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

RL01K HAS 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  
DRIFT TRACKING 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 WE WERE ON FOR THE LAST READ HEADER PERFORMED. EACH SEEK IS VERIFIED BY AN IMMEDIATE INITIAL READ HEADER FROM THAT POINT ANY SUBSEQUENT READ HEADER READ OR WRITE 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

#### RE ERR ENC'D

IN ATTEMPTING A RETRY OF A FUNCTION THAT WAS IN ERROR THE RETRY WAS SUCCESSFUL. ERROR INFORMATION CONSISTS OF BUS ADDRESS, DISK ADDRESS, NUMBER OF RETRIES BEFORE SUCCESS AND ERROR TYPE.

#### 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. EITHER STOP THE EXERCISER AND CHANGE CARTRIDGE, STOP THE EXERCISER AND VERIFY THE CARTRIDGE OR 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 REPORTS

## 4.1 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 PUNNING OR DROPPED DH:DM  
 PACK SERIAL #: DDDDDDDDDD  
 SEEKS: IIIII  
 BITS READ: JJJJJJJJJ (\*16)  
 BITS WRITTEN: KKKKKKKKK (\*16)

ERRORS					
DRIVE:	N	SEFK:	N	TRACK:	N
HARD:	N	SOFT:	N	DATA:	N
DCK:	N	HCRC:	N	NXM:	N
DLT:	N	OPI:	N	HNF:	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 NUMREP OF PACK  
 IIII IS TOTAL NUMBER OF SEEKS SINCE 0:00:00  
 JJJJ IS TOTAL NUMBER OF BITS READ (\*16) SINCE 0:00:00  
 KKKK IS TOTAL NUMBER OF BITS WRITTEN (\*16) SINCE 0:00:00  
 N IS NUMBER OF THAT TYPE ERROR SINCE 0:00:00

## 4.2 PROGRESS REPORTS

THE ONLY PROGRESS REPORT IS THE AUTOMATIC PERFORMANCE REPORT.

## 5.0 DEVICE INFORMATION TABLES

THE RL11/PLV11 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)  
 BIT 11 - DATA LATE (WITH BIT 10 CLEAR)  
 BIT 11 - HEADER CRC (WITH BIT 10 SET)  
 BIT 11 - 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 - MUST BE ZERO(0)  
 BIT 14-7 - CYLINDER ADDRESS FOR TRANSFER  
 BIT 6 - SURFACE FOR TRANSFER  
 BIT 5-0 - SECTOR FOR TRANSFER (0-47)

FOR SEEK FUNCTION  
-----

BIT 15 - MUST BE ZERO(0)  
 BIT 14-7 - DIFFERENCE TO NEW CYLINDER  
 BIT 6-5 - MUST BE ZERO(0)  
 BIT 4 - SURFACE  
 BIT 3 - MUST BE ZERO  
 BIT 2 - SEEK DIRECTION( 1 - IN / 0 - OUT )  
 BIT 1 - MUST BE ZERO  
 BIT 0 - MUST BE ONE(1)

FOR GET STATUS FUNCTION  
-----

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

RLMP - MULTIPURPOSE REGISTER

-----  
 FOR READ/WRITE FUNCTION  
 -----

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

FOR READ HEADER FUNCTION  
 -----

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

FOR GET STATUS FUNCTION  
 -----

HAS DRIVE STATUS

BIT 15 - WRITE DATA ERROR  
 BIT 14 - CURRENT HEAD ERROR(CHE)  
 BIT 13 - WRITE LOCK STATUS(WL)  
 BIT 12 - SEEK TIME OUT(SKTO)  
 BIT 11 - SPIN ERROR(SPE)  
 BIT 10 - WRITE GATE ERROR(WGE)  
 BIT 9 - VOLUME CHECK(VC)  
 BIT 8 - DRIVE SELECT ERROR(DSE)  
 BIT 7 - RESERVED(0)  
 BIT 6 - SURFACE  
 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 WRITE CHECK NEEDED; THEN STEP 4  
 IF SEEK NEEDS VERIFICATION; THEN STEP 12  
 IF IN PROCESS OF RETRY; THEN STEP 6  
 IF IN PROCESS OF SEEK RETRY; THEN STEP 8  
 IF GET STATUS; THEN STEP 5  
 IF SEEK; THEN STEP 7  
 IF READ; THEN STEP 13  
 IF WRITE; THEN STEP 17
5. ISSUE WRITE CHECK; GO TO STEP 1
6. ISSUE GET STATUS; GO TO STEP 1
7. ISSUE LAST FUNCTION; GO TO STEP 1
8. GET RANDOM CYLINDER AND HEAD WITHIN SOFTWARE PARAMETER LIMITS
9. CALCULATE DIFFERENCE TO NEW POSITION
10. ISSUE SEEK
11. SET POSITION VERIFICATION NEEDED FLAG
12. GO TO STEP 1
13. ISSUE READ HEADER, THEN STEP 1
14. GET RANDOM WORD COUNT WITHIN LIMITS
15. GET RANDOM SECTOR WITHIN LIMITS
16. CHECK THAT WORD COUNT AND SECTOR FIT ON TRACK IF THEN STEP 16; ELSE FIX
17. ISSUE READ; GO TO STEP 1
18. GET RANDOM WORD COUNT WITHIN LIMITS

19. GET RANDOM SECTOR WITHIN LIMITS
20. CHECK THAT WORD COUNT AND SECTOR FIT ON TRACK IF THEN STEP 20; ELSE FIX
21. SELECT RANDOM PATTERNS IN 128 WORD CHUNKS UNTIL WORD COUNT DONE AND WRITE BUFFER IN MEMORY.
22. ISSUE WRITE; 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 20; 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

31	BIT AND OFFSET DEFINITIONS
174	GLOBAL DATA AND CONSTANTS
257	GLOBAL MESSAGES
370	ERROR MESSAGES
546	SOFTWARE PARAMETERS
594	STATISTIC CODE
622	INITIALIZATION CODE
918	GLOBAL SUBROUTINES
990	PROGRAM MAIN LOOP
1091	ROUTINE TO SETUP AND ISSUE GET STATUS
1099	ROUTINE TO SETUP AND ISSUE SEEK FUNCTION
1176	ROUTINE TO LOAD READ HEADER AND ISSUE IT.
1244	ROUTINE TO LOAD WRITE DATA COMMAND
1307	ROUTINE TO LOAD READ DATA COMMAND
1325	SETUP CONTROLLER AND DRIVE INFO FOR INTERRUPT PROCESSING
1343	ROUTINE TO LOAD FUNCTION
1366	INTERRUPT SERVICE ROUTINES
1436	CONTROLLER ERROR CHECK ROUTINE
1666	COMMAND SERVICE ROUTINES
1697	SEEK
1707	READ
1725	READ HEADER
1759	GET STATUS
1784	WRITE
1842	DRIVE ERROR SERVICE
1893	RETRY LIMIT ROUTINE
2003	LIST OF FUNCTION ROUTINES
2017	RAD SECTOR FILE ROUTINE
2134	ROUTINE TO DROP DRIVE
2178	ROUTINE TO CHECK DATA
2250	ROUTINE TO WAIT FOR CONTROLLED READY
2283	GET STATUS/DRIVE RESET ROUTINE
2304	ROUTINE TO GENERATE A RANDOM NUMBER
2320	ROUTINE TO WRITE PACKS INITIALLY
2315	ROUTINE FOR SYSTEM CLOCK
2344	HEADS HOME ROUTINE
2365	RANDOM WC AND DA ROUTINE
2555	ROUTINE TO DUMP BUFFER ON DCK
2785	ROUTINE TO CHECK FOR RAD SECTOR
2977	DRIVE INFORMATION BUFFERS
3170	DIAGNOSTIC SUPERVISOR -- LOW CORE SET UP

```

1 3
2 4
3 5
4 002000
5 6
6 7
7 8
8 9
9 10 002000
10 11 000000 SVC
11 12 000000 SVCINS=0
12 13 000000 SVCTAG=0
13 14
14 15
15 16 002000 POINTER ALL
16 17
17 18
18 19 002000 BGNMOD
19 20 000000 MDHDR
20 21 000000 HEADER CZRLER,B,0,0,C,C,RL01,1
21 22 (4) 002001 103 .ASCII /C/
22 23 (4) 002002 132 .ASCII /Z/
23 24 (4) 002003 172 .ASCII /R/
24 25 (4) 002004 165 .ASCII /E/
25 26 (4) 002005 000 .RVTE /G/
26 27 (4) 002006 000 .RVTE /A/
27 28 (4) 002007 000 .RVTE /^/
28 29 (4) 002010 102 .ASCII /R/
29 30 (4) 002011 060 .ASCII /O/
30 31 (4) 002014 000000 .WORD C
31 32 (4) 002016 026322 .WORD L$HARD
32 33 (4) 002020 026446 .WORD L$SOFT
33 34 (4) 002022 007470 .WORD L$HW
34 35 (4) 002026 030132 .WORD L$SW
35 36 (4) 002030 000000 .WORD L$LAST
36 37 (4) 002032 000000 .WORD O
37 38 (4) 002034 000000 .WORD C
38 39 (4) 002036 000000 .WORD C
39 40 (4) 002040 007604 .WCPD L$DISPATCH
40 41 (4) 002042 000000 .WORD C
41 42 (4) 002044 000000 .WORD C
42 43 (4) 002046 000000 .WORD C
43 44 (4) 002050 002 .RVTE CSREVISION
44 45 (4) 002051 000000 .RVTE CSEDIT
45 46 (4) 002054 000000 .WORD O
46 47 (4) 002056 000000 .WCPD C
47 48 (4) 002060 000000 .WORD C
48 49 (4) 002062 002114 .WORD L$DVTYP
49 50 (4) 002066 007606 .WCPD L$KPT
50 51 (4) 002070 002112 .WORD L$DP
51 52 (4) 002074 011236 .WORD L$DRST
52 53 (4) 002074 011236 .WORD L$AU

```

```
(4) 002076 011322 .WORD L$DU
(5) 002100 000014 .WORD 14
(4) 002104 000280 .WORD 7
(4) 002106 011072 .WORD L$INIT
      .WORD L$CLEAN
23      002110      ENDMOD
24
25
26      002110      DEVREG
(5) 002110 000000 .WORD C
(2) 002112 000001 .RLKW
29
30      002114      DEVTYP <RLO1>
(3) 002114 046122 030460 000 .ASCIZ >RLO1>
(2) 002122      .EVEN
31
32      .SBTTL BIT AND OFFSET DEFINITIONS
33      ;DEFINITIONS
34
35      002122      BGNMOD GLREQAT
36
37      002122      EQUALS
38
39      000000      ;CONTROL AND STATUS OFFSET
40      000002      BA=2 ;BUS ADDRESS OFFSET
41      000004      DA=4 ;DISK ADDRESS OFFSET
42      000006      MA=4 ;MULTI PURPOSE OFFSET
43
44      ;CONSTANT OFFSETS FOR INDIVIDUAL DRIVE BUFFERS
45      ;THE ONLY POSITION THAT IS CRITICAL IS THAT OF
46      ;"PRPOS" IT MUST BE THE LAST ENTRY OF THE BUFFER
47
48      000000      SKCNT=0 ;SEEK OPERATION COUNT
49      000002      RXP1=2 ;READ OPERATION COUNT (BITS) LOW ORDER
50      000004      RXP2=2 ;READ OPERATION COUNT (BITS) HIGH ORDER
51      000006      WXP1=6 ;WRITE OPERATION COUNT (BITS) LOW ORDER
52      000010      WXP2=6 ;WRITE OPERATION COUNT (BITS) HIGH ORDER
53      000012      ERRCNT=12 ;ERROR COUNT - HARD
54      000014      SFTCNT=14 ;ERROR COUNT - SOFT
55      000016      SKECNT=16 ;SEEK ERROR COUNT
56      000020      DERCNT=20 ;DRIVE ERROR COUNT
57      000022      DRCRCR=22 ;DATA CRC ERROR COUNT
58      000024      HRCRCR=24 ;HEADER CRC ERROR COUNT
59      000026      DLTENT=26 ;DATA LATE ERROR COUNT
60      000030      DPICNT=30 ;OPERATION INCOMPLETE ERROR COUNT
61      000032      HNFERR=32 ;HEADER NOT FOUND ERROR COUNT
62      000034      NMEMNT=34 ;NON-EXISTANT MEMORY ERROR COUNT
63      000036      RETRY=36 ;PRESENT RETRY NUMBER
64      000040      RDA=40 ;DISK ADDRESS CONTENTS
65      000042      BMP=42 ;PRESENT MULTIPURPOSE CONTENTS
66      000044      FUNC=44 ;LAST FUNCTION LOADED
67      000046      BCSADR=46 ;CSR IMAGE OF LAST COMMAND
```

```
BIT AND OFFSET DEFINITIONS
68      000050      LSTHDR=50 ;LAST POSITION ON DISK
69      000052      RTYPE=52 ;ERROR ON WHICH RECOVERY IS BEING TRIED
70      000054      SKCNT1=54 ;LOW SEEK COUNT
71      000056      PRP1C=56 ;INTERNAL FLAGS
72      000060      RXP3=60 ;THIRD ORDER READ COUNT
73      000062      WXP3=62 ;THIRD ORDER WRITE COUNT
74      000064      LSTDA=64 ;DISK ADDRESS AT SOFT ERROR
75      000066      DIFWD=66 ;LAST DIFFERENCE WORD OF SEEK
76      000070      DPHOUR=70 ;HOUR OF DRIVE DROPPED
77      000072      DPMIN=72 ;MINUTE OF DRIVE DROPPED
78      000074      TRERR=74 ;TRACKING ERRORS COUNT
79      000076      DMCCK=76 ;DATA CHECK ERRORS
80      000078      SERNM1=78 ;SERIAL NUMBER OF CARTRIDGE
81      000080      SERNM2=80 ;SERIAL NUMBER OF CARTRIDGE
82      000082      DCS=82 ;CSR ADDRESS
83      000084      DRSP1=84 ;DRIVE SELECT BITS(8,9,10)
84      000086      BRA=86 ;PRESENT BUS ADDRESS CONTENTS
85      000088      BSECPT=88 ;POINTER TO BAD SECTOR FILE
86      000090      RSECK=88 ;SEEK IN PROCESS OF RECOVERY
87      000092      SFTCS=88 ;CSR OF SOFT ERROR
88      000094      WRPG=90 ;WRITE OPERATION IN PROGRESS AT PWR FAIL TIME
89      000096      PRPOS=92 ;PRESENT POSITION ON DISK
90      000100      SKDON=BIT0 ;DRIVE READY
91      000102      DRDY=BIT0 ;INTERRUPT ENABLE
92      000104      INTEN=BIT6 ;COMPOSITE ERROR
93      000106      ERR=BIT15 ;DRIVE ERROR
94      000108      DRP=BIT14 ;WRITE DATA ERROR
95      000110      WDE=BIT15 ;HEAD CURRENT ERROR
96      000112      HCE=BIT14 ;WRITE LOCK
97      000114      WLT=BIT13 ;SEEK TIMEOUT ERROR
98      000116      SFT=BIT12 ;SPINDLE TIMEOUT/UNDER/OVER SPEED
99      000118      WGE=BIT11 ;WRITE GATE ERROR
100      000120      VCS=BIT9 ;VOLUME CHECK
101      000122      DSE=BIT8 ;DRIVE SELECT ERROR
102      000124      NXM=BIT13 ;NON-EXISTANT MEMORY ERROR
103      000126      DLT=BIT12 ;DATA LATE
104      000128      DRCRC=BIT11 ;DATA CRC ERROR
105      000130      HRCRC=BIT11 ;HEADER CRC ERROR
106      000132      HNF=BIT11 ;HEADER NOT FOUND ERROR
107      000134      OPT=BIT10 ;OPERATION INCOMPLETE ERROR
108      000136      CRDY=BIT7 ;CONTROLLER READY
109      000138      BA17=BIT7 ;EXTENDED BUS ADDRESS BIT 17
110      000140      BA16=BIT4 ;EXTENDED BUS ADDRESS BIT 16
111      000142      WRCHK=BIT1 ;WRITE CHECK FUNCTION CODE
112      000144      GSTAT=BIT2 ;GET DRIVE STATUS FUNCTION CODE
113      000146      SEK=BIT1 ;BIT2 ;SEEK FUNCTION CODE
114      000148      RDHDR=BIT3 ;READ HEADER FUNCTION CODE
115      000150      WRTE=BIT3 ;WRITE FUNCTION CODE
116      000152      READ=BIT3 ;READ FUNCTION CODE
117      000154      DRST=BIT3 ;DRIVE RESET COMMAND CODE FOR DRIVE COMMAND WORD
118      000156      GSRIT=BIT1 ;BIT0 ;GET STATUS COMMAND CODE FOR DRIVE COMMAND WORD
119      000158      WK=BIT0 ;MARKER BIT FOR DRIVE COMMAND WORD(SEEK GET STATUS)
120      000160      SIGN=BIT2 ;DIRECTION FOR SEEK(0-AWAY FROM SPINDLE)
```

```

124      000020      SKHS=BIT4      ;HEAD SELECT FOR SEEK
125      000100      HEAD=BIT4      ;HEAD SELECT FOR READ,WRITE,GET STATUS
126
127
128      ;OFFSET FOR HARDWARE P-TABLE
129
130      000000      CSR=0
131      000020      VECT=2
132      000040      PRIOR=4
133      000060      DRPT=6
134      000010      CNT=10
135
136      ;OFFSET FOR SOFTWARE P-TABLE
137
138      000000      RL1=0
139      000020      RL2=2
140      000040      SET=4
141      000060      DAT=6
142      000010      SRT=10
143      000020      TWT=12
144      000040      RDT=14
145      000060      DDT=16
146      000020      CHPLG=20
147      000040      MXB=22
148      000060      MNB=24
149      000030      MNC=30
150      000032      MNC=32
151      000034      MXS=34
152      000036      MNS=36
153      000040      DCKFG=40
154      000042      DRPLG=42
155      000044      MNB=44
156      000046      SE1=46
157      000050      OPPLG=50
158      000052      DET=52
159      000054      RDT=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      UCD=74
168
169
170      002122      ENDMOD
171
172      ;
173
174      .SBTTL GLORAL DATA AND CONSTANTS
175
176      BGNMOD GLRDAT
177
178      RECNT: .WORD 0      ;READ ERROR COUNT
179      RWCNT: .WORD 0      ;R/W ERROR COUNT

```

```

180      002126      000000      WRV: .WORD 0      ;REASON FOR DROPPING DRIVE
181      002130      000000      DRPT: .BYTE 0      ;DRIVES UNDER TEST
182      002131      000000      DRPT: .BYTE 0      ;DRIVES PRESENT
183      002132      000000      SYSSMK: .WORD 0      ;MASK FOR 0-7 DRIVES
184      002134      176543      HINUM: .WORD 176543 ;PRIME FOR RANDOM
185      002136      123456      LONUM: .WORD 123456 ;NUMBER GENERATOR
186      002140      100077      CVMASK: .WORD 100177 ;MASK FOR CYLINDER ONLY
187      002142      000000      SECMSK: .WORD 100077 ;MASK OUT SECTOR BITS
188      002144      000000      WRINIT: .WORD 0      ;WRITE INIT FLAG
189      002146      000000      WRPOS: .WORD 0      ;UNIT IN WRITE INIT INDICATOR
190
191      ;
192      ;THE FOLLOWING LOCATIONS ARE CLEARED AS A GROUP (DOWN TO "STFLG")
193      ;THEREFORE DON'T INSERT ANY CONSTANTS
194
195      002150      174400      CNTLR1: .WORD 174400 ;CSR OF CONTROLLER 1 (LUN 0-3)
196      002152      000000      CNTLR2: .WORD 0      ;CSR OF CONTROLLER 2 (LUN 4-7)
197      002154      000000      LSTDR1: .WORD 0      ;BUFFER POINTER OF DRIVE
198      002160      000000      BCSR: .WORD 0      ;CSR FROM P-TABLE
199      002162      000000      BVECT: .WORD 0      ;VECTOR " "
200      002164      000000      BPRIOR: .WORD 0      ;DRIVE " "
201      002166      000000      BRSEL: .WORD 0      ;FLAG TO INDICATE HDR IN BAD LIST
202      002170      000000      HRFMD: .WORD 0      ;SECTOR OF ERROR - USED BY BAD SECTOR LOCATION
203      002172      000000      CHKSEC: .WORD 0      ;DATA ERROR COUNT
204      002174      000000      DECN1: .WORD 0      ;TEMP LOCATION
205      002176      000000      DECN2: .WORD 0      ;TEMP LOCATION
206      002200      000000      TEMP0: .WORD 0      ;TEMP LOCATION
207      002202      000000      TEMP1: .WORD 0      ;TEMP LOCATION
208      002204      000000      TEMP2: .WORD 0      ;TEMP LOCATION
209      002206      000000      TEMP3: .WORD 0      ;" "
210      002210      000000      TEMP4: .WORD 0      ;" "
211      002212      000000      TEMP5: .WORD 0      ;" "
212      002214      000000      TEMP6: .WORD 0      ;" "
213      002216      000000      TEMP7: .WORD 0      ;" "
214      002218      000000      TEMP8: .WORD 0      ;" "
215      002220      000000      TEMP9: .WORD 0      ;" "
216      002222      000160      VECT1: .WORD 160      ;VECTOR OF FIRST CONTROLLER
217      002224      000000      VECT2: .WORD 0      ;VECTOR " 2ND
218      002230      000000      PRIOR1: .WORD 0      ;" "
219      002232      000000      PRIOR2: .WORD 0      ;" "
220      002234      000000      GODAT: .WORD 0      ;" "
221      002236      000000      RNTMP: .WORD 0      ;" "
222      002240      000000      INTERVAL: .WORD 0      ;TIME BETWEEN REPORTS
223      002242      000000      LSTIM: .WORD 0      ;LAST TIME ON SYSTEM CLOCK
224      002244      000000      SECOND: .WORD 0      ;SECONDS OF SYSTEM CLOCK
225      002246      000000      MINUTE: .WORD 0      ;MINUTES OF SYSTEM CLOCK
226      002250      000000      HOUR: .WORD 0      ;HOURS OF SYSTEM CLOCK
227      002252      000000      E-RA: .WORD 0      ;IMAGES OF REGISTERS
228      002254      000000      E-DA: .WORD 0      ;ON INTERRUPT
229      002256      000000      E-MP: .WORD 0      ;" "
230      002260      000000      E-MP2: .WORD 0      ;" "
231      002262      000000      E-MP3: .WORD 0      ;" "
232      002264      000000      SYSCLK: .WORD 0      ;FLAG INDICATING PRESENCE OF SYSTEM CLOCK
233      002266      000000      BUF1: .WORD 0      ;BUFFER FOR FIRST CONTROLLER
234      002268      000000      BUF2: .WORD 0      ;BUFFER FOR SECOND CONTROLLER
235      002272      000000      MAXWC: .WORD 0      ;MAX WORD COUNT DETERMINED BY CORE

```

```

235 002374 000000 UUT: .WORD 0 ;NUMBER OF UNITS ON SYSTEM
236 002300 000000 PWFLG: .WORD 0 ;POWER FAIL INDICATOR
237 002300 000000 TRPFLG: .WORD 0 ;TRAP OCCURRENCE FLAG
238 002302 000000 STFLG: .WORD 0 ;START FLAG
239
240 ;
241 ;END OF MASS CLEAR
242
243 002304 000000 CNTFLG: .WORD 0 ;CONTINUE FLAG
244 002306 000000 FASCII: .WORD 0 ;ASCII MESSAGE OF FUNCTION
245 002306 000000 FASPMT: .WORD 0 ;PRINTER
246 002312 000000 PRCNT: .WORD 0 ;PRINTER COUNT
247 002314 000000 DMCNT1: .WORD 0 ;ERROR COUNT
248 002316 000004 ERRVEC: .WORD 4 ;ERROR VECTOR
249 002320 000034 ST1: .WORD 34 ;STATES ALLOWED
250 002322 000035 ST2: .WORD 35 ;STATES ALLOWED
251 002324 000000 OPCALL: .WORD 0
252 002326 000000 INCALL: .WORD 0
253
254 002330 ENDMOD
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
002330
    .SRTTL GLOBAL MESSAGES
    BGNMOD GLRXTX
    ;GLOBAL TEXT
002330 044524 042515 020072 TIME: .ASCIZ "TIME: "
002337 040 046122 051503 WRLCS: .ASCIZ "RLCS: "
002347 040 051050 041514 CRLCS: .ASCIZ " (RLCS): "
002361 040 052506 041514 MPUNC: .ASCIZ " FUNCTION: "
002367 040 051050 041114 CRLDA: .ASCIZ " (RLDA): "
002407 040 051050 041114 CRLMP: .ASCIZ " (RLMP): "
002433 104 043111 053440 DIFMSG: .ASCIZ /DIF WD: /
002444 040520 045503 051440 CART: .ASCIZ /PACK SERIAL #: /
002464 247516 041440 042122 NOCRDY: .ASCIZ /NO CRDY/
002474 051104 047040 052117 NDRDY: .ASCIZ /DR NOT RDY/
002507 104 043525 047506 NRDRDY: .ASCIZ /DR NOT RDY W/O DR ERR/
002535 102 043525 047506 PRCY: .ASCIZ /PRC/
002541 111 044516 020124 NWRTS: .ASCIZ /INIT WR OF SEC BAD/
002564 051240 041505 047524 SMSG: .ASCIZ /SECTOR: /
002584 047516 043440 047517 EXHAUS: .ASCIZ /NO GOOD HDR/
002612 047125 044504 043501 UNDRS: .ASCIZ /UNDRAGNOSABLE ERP/
002634 042523 044505 042444 MSKER: .ASCIZ /SEK ERR/
002654 042523 044505 042444 MSPER: .ASCIZ /SOFT ERR ENC'D/
002673 051104 043111 020124 DRERR: .ASCIZ /DR ERR/
002693 104 020122 051105 MDRER: .ASCIZ /DR ERR WILL NOT RESET/
002721 104 020122 052123 MDSER: .ASCIZ /DR STAT ERR/
002735 051125 046140 041440 MVCEP: .ASCIZ /VOL CHK WILL NOT CLR/
002755 104 020122 051105 WGER: .ASCIZ /WR GATE ERR WILL NOT RESET/
003015 104 020122 051105 MDCER: .ASCIZ /DATA CMP ERR/
003040 240504 044524 041440
    
```

```

295 003055 110 051101 020104 MHDER: .ASCIZ /HARD ERROR/
296 003078 040524 040524 DHPDCK: .ASCIZ /DATA DUMP - DCK/
297 003110 051124 042113 TRACK: .ASCIZ /TRACKING ERR/
298 003125 110 042122 042440 ERACKM: .ASCIZ /ERR NOT LMT EXC'D/
299 003147 043123 020113 SERLMT: .ASCIZ /SER ERR LMT EXC'D/
300 003170 040504 020124 SFMSG: .ASCIZ /SFT ERR LMT EXC'D/
301 003212 040504 020124 DCDMSG: .ASCIZ /DATA ERR LMT EXC'D/
302 003235 104 020122 051105 DRERR: .ASCIZ /DR ERR LMT EXC'D/
303 003256 052502 043106 OVER: .ASCIZ /PUFFER CHOSN TOO RIG - WAS /
304 003279 122 050505 REQ: .ASCIZ /REQ BY OPR/
305 003322 054105 050110 SRTHAU: .ASCIZ /EXH'D RETRY ON SEEK/
306 003352 042116 020113 UNDRS: .ASCIZ /DR NOT UNLD ON ERR/
307 003376 051104 042114 NOLBAD: .ASCIZ /DB WLD NOT LD/
308 003409 051105 046040 SOPLMT: .ASCIZ /OPER LMTS EXC'D/
309 003434 040557 051105 NRECV: .ASCIZ /CARRIED DATA - CAN'T CHECK IT/
310 003473 115 051174 020105 MBDMSG: .ASCIZ /MORE THAN 16 BAD SECTORS/
311 003524 047516 043040 HWSEC: .ASCIZ /NO FACTORY FILE/
312 003544 047516 043040 SWSEC: .ASCIZ /NO FIELD FILE/
313 003572 024520 046102 WPT: .ASCIZ /P-TABLE/
314 003574 046111 020114 TFLG: .ASCIZ /P-TABLE/
315 003610 053040 041505 MVEC: .ASCIZ /VECTOR/
316 003622 047526 042040 NODRIV: .ASCIZ /NO DRIVES/
317 003634 042040 044522 DRNV: .ASCIZ /DRIVE: /
318 003645 040 051514 EPS: .ASCIZ /EXP POS: /
319 003660 042440 050130 RPS: .ASCIZ /REC POS: /
320 003673 042522 020103 RPS: .ASCIZ /REC POS: /
321 003706 051104 042522 NDRDP: .ASCIZ /DR DID REC'R FROM PWP UP/
322 003737 101 020124 RUSBD: .ASCIZ /AT BUS ADDR: /
323 003755 042440 052105 MRT: .ASCIZ /RETRY: /
324 003810 052123 052122 ERT: .ASCIZ /ERROR TYPE: /
325 004004 040 044123 MST: .ASCIZ /STATUS WAS: /
326 004021 040 052105 RTI: .ASCIZ /SHOULD RE: /
327 004036 051040 044522 BXP: .ASCIZ /EXP'D: ATTEMPTED/
328 004061 040 054105 RCD: .ASCIZ /EXP'D: /
329 004072 051104 044522 DRDP: .ASCIZ /DRIVE DROPPED/
330 004103 104 044522 MTHNF: .ASCIZ /HNF/
331 004121 040 047110 MTHCRC: .ASCIZ /HCRC/
332 004144 040 051033 MTDCRC: .ASCIZ /DCK/
333 004174 042040 000103 MTDCRC: .ASCIZ /DCK/
334 004191 040 046104 MTDPI: .ASCIZ /DPI/
335 004214 047440 044520 MTNPM: .ASCIZ /NPM/
336 004253 040 054115 MTDPRV: .ASCIZ /DRV/
337 004280 042040 051525 MSTART: .ASCIZ /TESTING STARTED/
338 004365 124 051525 MSWPPP: .ASCIZ /WRITING PACK /
339 004205 127 044522
    ;
    ;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(15) BYTES PLUS ZERO
    ;FILL BYTE (TOTAL EIGHT) BYTES LONG. USED IN LINE1
    ;SUBROUTINE.....
    MTCR: .ASCIZ / MRCRK /
    MTRD: .ASCIZ / GTSTAT /
    
```

```

351 004244 051440 042505 020113 MTSK: .ASCIZ / SEEK /
352 004254 051040 044104 051104 MTRH: .ASCIZ / RDHDR /
353 004264 053400 044204 043504 MTR: .ASCIZ / WE /
354 004274 051040 040505 020104 MTRD: .ASCIZ / READ /
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
    
```

;END OF LIST NOW YOU CAN PUT ANY THING YOU WANT HERE  
 ;.....

```

365 .EVEN
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
    
```

;GENERAL ERROR REPORT  
 ;MIS-SEEK ERROR REPORT  
 ;SOFT.ERROR RECOVERABLE ERROR REPORT

```

(11) 004412 017446 000110
(12) 004422 017446 002375
(9) 004432 017446 000116
(8) 004442 012746 002347
(7) 004452 017446 005725
(6) 004462 017446 000007
(5) 004472 010600
(4) 004482 104014
(3) 004492 062706 000020
(2) 004502 016446 000052
(1) 004512 012746 003766
(10) 004522 016446 000036
(9) 004532 012746 003755
(8) 004542 012746 006042
(7) 004552 012746 000005
(6) 004562 010600
(5) 004572 104014
(4) 004582 062706 000014
(3) 004592
(2) 004602 104023
(1) 004612
(11) 004514
(12) 004524 004737 005474
(9) 004534 013746 002322
(8) 004544 013746 002320
(7) 004554 013746 002256
(6) 004564 012746 004004
(5) 004574 012746 006060
(4) 004584 010600 000006
(3) 004594 104014
(2) 004604 062706 000016
(1) 004614 104023
(11) 004566
(12) 004576 004737 005410
(9) 004586 016400 000042
(8) 004596
(7) 004606 010046
(6) 004616 013746 002174
(5) 004626 012746 008230
(4) 004636 012746 000003
(3) 004646 010600
(2) 004656 104014
(1) 004666 062706 000010
    
```

;GET STATUS ERROR REPORT  
 ;DATA ERROR SUMMARY

```

408 004624      ENDMSC
(3) 004624      L10004: EMT      CSMMSG
409 004624      104023
410
411
412
413 004626      BGNMSG  ERR9
(9) 004626      PRINTR  #FMT8,RETRY(R4),#RT1
(8) 004626      MOV      #RT1,-(SP)
(7) 004632      MOV      RETRY(R4),-(SP)
(6) 004642      MOV      #3,-(SP)
(5) 004646      MOV      SP,R0
(4) 004650      EMT      C$PNTB
(3) 004652      ADD      #10,SP
(2) 004656      JSR      PC,LINE3
(1) 004662      ENDMSC
(3) 004662      L10005: EMT      CSMMSG
416
417
418
419
420 004664      BGNMSG  ERR9
(15) 004664      PRINTB  #FMT10,#TIME,HOUR,MINUTE,SECOND,#MRLCS,DCS(R4),#DRNM,#B,DRSEL+1(R4)
(14) 004666      CLR      -(SP)
(13) 004666      BISR   DRSEL+1(R4),-(SP)
(12) 004672      MOV      #1,-(SP)
(11) 004676      MOV      DCS(R4),-(SP)
(10) 004702      MOV      #MRLCS,-(SP)
(9) 004706      MOV      SECOND,-(SP)
(8) 004712      MOV      #MRLCS,-(SP)
(7) 004716      MOV      HOUR,-(SP)
(6) 004722      MOV      #TIME,-(SP)
(5) 004726      MOV      #FMT10,-(SP)
(4) 004732      MOV      #1,-(SP)
(3) 004736      EMT      SP,#6
(2) 004740      EMT      C$PNTB
(1) 004742      ADD      #24,SP
(15) 004746      PRINTR  #FMT10A,#CRLBA,#RBA(R4),#CRLDA,#BDA(R4),#EXP,GDDAT,#PCD,(R2)
(14) 004750      MOV      #PCD,-(SP)
(13) 004754      MOV      #EXP,-(SP)
(12) 004758      MOV      #GDDAT,-(SP)
(11) 004764      MOV      #RBA(R4),-(SP)
(10) 004770      MOV      #CRLDA,-(SP)
(9) 004774      MOV      #RBA(R4),-(SP)
(8) 005000      MOV      #FMT10A,-(SP)
(7) 005004      MOV      #1,-(SP)
(6) 005010      EMT      SP,R0
(5) 005014      ADD      #24,SP
(4) 005020      EMT      C$PNTB
(3) 005024      ADD      #24,SP
(2) 005024      PRINTB  #FMT10B,R2
(1) 005024      MOV      R2,-(SP)
(6) 005032      MOV      #2,-(SP)
    
```

```

(3) 005036      MOV      SP,R0
(2) 005040      EMT      C$PNTB
(1) 005042      ADD      #6,SP
423 005046      ENDMSC
(3) 005046      L10006: EMT      CSMMSG
424 005046      104023
425
426
427
428 005050      BGNMSG  ERR9
429 005050      ERR9
(11) 005054      JSR      PC,LINE3
(10) 005054      PRINTR  #FMT13,#MST,R1,#LPS,LSTHDR(R4)
(9) 005062      MOV      LSTHDR(R4),-(SP)
(8) 005066      MOV      #LPS,-(SP)
(7) 005072      MOV      #MST,-(SP)
(6) 005076      MOV      #FMT13,-(SP)
(5) 005102      MOV      #5,-(SP)
(4) 005104      EMT      SP,R0
(3) 005106      EMT      C$PNTB
(2) 005112      ADD      #14,SP
(1) 005112      ENDMSC
(3) 005112      L10007: EMT      CSMMSG
433
434
435
436
437 005114      BGNMSG  ERR10
(13) 005114      PRINTR  #FMT11,#MPT,R1,#MRLCS,BCSR,#MVEC,RVEC
(12) 005120      MOV      #MVEC,-(SP)
(11) 005124      MOV      #BCSR,-(SP)
(10) 005130      MOV      #MRLCS,-(SP)
(9) 005134      MOV      #MPT,-(SP)
(8) 005136      MOV      #4,-(SP)
(7) 005142      MOV      #FMT11,-(SP)
(6) 005146      MOV      #7,-(SP)
(5) 005152      EMT      SP,R0
(4) 005154      EMT      C$PNTB
(3) 005156      ADD      #20,SP
(2) 005162      ENDMSC
(3) 005162      L10010: EMT      CSMMSG
440
441
442 005164      BGNMSG  ERR12
443 005164      ERR12
444 005170      JSR      PC,LINE3
445 005170      ENDMSC
(3) 005170      L10011: EMT      CSMMSG
446 005170      104023
447 005172      BGNMSG  ERR13
    
```

```
448 005172 004737 005474 JSP PC,LINE3
449 005176 016446 000040 PRINTB #FMT9,#MSG,BDA(R4)
450 005202 012746 002564 MOV #MSG,-(SP)
451 005206 012746 006435 MOV #FMT1,-(SP)
452 005210 017600 000003 MOV #3,-(SP)
453 005220 104014 EMT C$PNTB
454 005222 062706 000010 ADD #10,SP
455 005226 016437 000044 L10012: EMDMSG
456 005230 012746 004224 EMT C$MSG
457 005236 012737 004224 LINE1: MOV #FUNC(R4),FASPNT ;GET FUNCTION
458 005244 042737 000100 MOV #MTCR,FASCII ;FIRST FUNCTION ASCIZ
459 005248 005337 002310 RVC #INTEN,FASPNT ;CLEAR INTERRUPT ENABLE
460 005252 005337 002310 1S: ASR FASPNT ;ALIGN
461 005262 001404 REQ #C ;DOWN COUNT FUNCTION
462 005264 062737 000010 ADD #R-,FASCII ;FOUND?
463 005272 000771 BR IS ;NO NEXT ONE
464 005274 2S: ;LOOP
465 005274 005046 PRINTB #FMT1,#TIME,HOUR,MINUTE,SECOND,#MRLCS,DCS(R4),#DRNM,<B,DRSEL+1(R4)>
466 005276 156416 CLR -(SP)
467 005300 012746 BISR DRSEL+1(R4),(SP)
468 005302 012746 MOV #DRNM,-(SP)
469 005304 012746 MOV DCS(R4),-(SP)
470 005306 012746 MOV #MTCR,-(SP)
471 005308 013746 MOV SECOND,-(SP)
472 005310 013746 MOV MINUTE,-(SP)
473 005312 013746 MOV HOUR,-(SP)
474 005314 012746 MOV #TIME,-(SP)
475 005316 012746 MOV #FMT1,-(SP)
476 005318 012746 MOV #11,-(SP)
477 005320 104014 SP,RO
478 005322 062706 EMT C$PNTB
479 005324 000024 ADD #24,SP
480 005326 013746 PRINTB #FMT1,#FNUM,FASCII
481 005328 012746 MOV FASCII,-(SP)
482 005330 012746 MOV #FNUM,-(SP)
483 005332 012746 MOV #FMT1,-(SP)
484 005334 012746 MOV #3,-(SP)
485 005336 104014 EMT C$PNTB
486 005338 062706 ADD #10,SP
487 005402 062706 000010 RTS PC
488 005410 005046 LINE2: PPINTB #FMT9,#TIME,HOUR,MINUTE,SECOND,#MRLCS,DCS(R4),#DRNM,<B,DRSEL+1(R4)>
489 005412 156416 CLR -(SP)
490 005414 012746 BISR DRSEL+1(R4),(SP)
491 005416 012746 MOV #DRNM,-(SP)
492 005418 012746 MOV #MRLCS,-(SP)
493 005420 012746 MOV #MRLCS,-(SP)
```

```
494 005432 013746 MOV SECOND,-(SP)
495 005434 013746 MOV MINUTE,-(SP)
496 005436 013746 MOV HOUR,-(SP)
497 005438 012746 MOV #TIME,-(SP)
498 005440 012746 MOV #FMT9,-(SP)
499 005442 012746 MOV #11,-(SP)
500 005444 016600 SP,RO
501 005446 104014 EMT C$PNTB
502 005448 062706 ADD #24,SP
503 005450 000024 RTS PC
504 005474 004737 005230 LINE3: JSP PC,LINE1
505 005500 016446 000042 PRINTB #FMT2,#MRLCS,BCSADR(P4),#CRLBA,#BBA(R4),#CRLDA,RDA(R4),#CRLMP,BMP(R4)
506 005502 012746 MOV #BMP(R4),-(SP)
507 005504 016446 000040 MOV #CRLMP,-(SP)
508 005506 012746 002407 MOV #RDA(R4),-(SP)
509 005508 017446 000110 MOV #CRLDA,-(SP)
510 005510 017446 000110 MOV #BBA(R4),-(SP)
511 005512 016446 000046 MOV #CRLBA,-(SP)
512 005514 012746 002347 MOV #BCSADR(R4),-(SP)
513 005516 012746 005705 MOV #CRLCS,-(SP)
514 005518 012746 000011 MOV #FMT2,-(SP)
515 005520 016600 SP,RO
516 005522 104014 EMT C$PNTB
517 005524 062706 ADD #24,SP
518 005526 013746 PRINTB #FMT3,#CRLCS,E.CS,#CRLBA,E.RA,#CRLDA,E.DA,#CRLMP,E.MP
519 005528 012746 MOV #MP,-(SP)
520 005530 012746 MOV #CRLMP,-(SP)
521 005532 012746 MOV #E.DA,-(SP)
522 005534 012746 MOV #CRLDA,-(SP)
523 005536 013746 MOV #E.RA,-(SP)
524 005538 012746 MOV #CRLBA,-(SP)
525 005540 012746 MOV #E.CS,-(SP)
526 005542 012746 MOV #CRLCS,-(SP)
527 005544 012746 MOV #FMT3,-(SP)
528 005546 104014 SP,RO
529 005548 062706 EMT C$PNTB
530 005550 000024 ADD #24,SP
531 005552 104014 RTS PC
532 005560 013746 ;FORMAT STATEMENTS
533 005562 002256 FMT1: .ASCII /*?Z2%A:Z2%A:Z2/
534 005564 002221 FMT17: .ASCIZ /*?06?01/
535 005566 002254 FMT1A: .ASCIZ /*?T?N/
536 005568 002407 FMT2: .ASCII /*AREFORE ERR?T06/
537 005570 002252 FMT2A: .ASCIZ /*?06?T06?T06?N/
538 005572 002375 FMT3: /*AT ERR ?T06?T06?T06?N/
539 005574 002250 FMT4: .ASCIZ /*?06?T06?T06?T06?N/
540 005576 002347 FMT5: .ASCIZ /*?06?T?N/
541 005578 005750 FMT6: .ASCIZ /*?06?T06?A OR ?06?N/
542 005580 000011 FMT7: .ASCIZ /*?Z2%A:Z2%A:Z2?T06?T/
543 005582 104014 FMT7A: .ASCIZ /*01?N?A - ?T?N/
544 005584 062706
545 005586 006207
```



```

493 006162 042045 022466 022524 FMT9: -ASCIZ /*D6$T$W/
494 006172 052045 022466 022462 FMT9: -ASCIZ /*Z2$A:Z2$A:Z2$T$R06$T$R01$N/
495 006230 042045 022466 020101 FMT9A: -ASCIZ /*D6$A WORDS RAD OUT OF $D6$A WORDS READ$N/
496 006302 052045 022466 022462 FMT10: -ASCIZ /*T$Z2$A:Z2$A:Z2$T$R06$T$R01/
497 006330 052045 022466 022462 FMT10A: -ASCIZ /*T$06$T$06$N$T$06$T$06$A AT BUS ADDRESS /
498 006415 042045 022466 022524 FMT11: -ASCIZ /*T$02$T$06$T$03/
499 006415 042045 022524 033117 FMT12: -ASCIZ /*T$06$N/
500 006443 042045 022524 033117 FMT13: -ASCIZ /*T$06$T$06$N/
501 006443 052045 022524 033117 FMT13D: -ASCIZ /*Z$A$A NOW IS $Z$A$N/
502 006507 042045 022516 022524 FMT14: -ASCIZ /*N$T$N/
503 006516 047445 022466 020101 FMT14A: -ASCIZ /*D6$A /
504 006520 042045 022516 022524 FMT14C: -ASCIZ /*N/
505 006530 040445 047445 042122 FMT14C: -ASCIZ /*$WORD $D3$A S/R $D6$A WAS $D6$N?
506 006572 040445 051155 047522 FMT15: -ASCIZ /*$ERROR(S) SET:$T$N$ARECOVERY BEING ATTEMPTED/
507 006650 040445 052045 020124 PRM16: -ASCIZ /*$NOT TESTING CS= $D6$A DR= $D1$N/
508 006717 047045 052045 020101 FMT19: -ASCIZ /*$N$T/
509 006717 047045 052045 022516 FMT19: -ASCIZ /*$N$S10$A*** RLO1 PERFORMANCE REPORT ***$N$N/
510 006775 042045 051040 022516 FMT31A: -ASCIZ /*$A RUNNING$N/
511 007013 042045 020101 042040 FMT31B: -ASCIZ /*$A DROPPED $Z$A:$Z2$N/
512 007044 052045 047445 042040 FMT32: -ASCIZ /*$T$D5$D5$N/
513 007057 042045 047445 042040 FMT32A: -ASCIZ /*$SEEKS: $D6$Z3$N$ABITS READ: $D6$Z4$Z4$A (*16)$N/
514 007144 040445 044502 051524 FMT32B: -ASCIZ /*$ABITS WRITTEN: $D6$Z4$Z4$A (*16)$N/
515 007210 047045 040445 051105 FMT33: -ASCIZ /*$N$AERRORS$N$ADRIVE: $D6$A SEEK: $D6$A TRACK: $D6$A DATA: $D6$N/
516 007344 040445 044504 035103 FMT33A: -ASCIZ /*$AHARD: $D6$A SOFT: $D6$N/
517 007430 040445 046104 035103 FMT33B: -ASCIZ /*$ADCK: $D6$A HCRC: $D6$A NYM: $D6$A HNF: $D6$N/
518 007430 040445 046104 035124 FMT35: -ASCIZ /*$ADLT: $D6$A OPI: $D6$N$N/

```

```

252 007466 .EVEN
253 007466 ENDMOD
254 007466 BGNMOD HPTCODE
255 007466 BGNHW .WORD L10013-L$HW/2
256 007470 .WORD 174400 ;BUS ADDRESS
257 007472 000160 .WORD 160 ;VECTOR FOR 1ST RL CONTROLLER
258 007474 000240 .WORD 240
259 007500 000001 .WORD 0
260 007502 ENDMW
261 007502 L10013:
262 007502 ENDMOD
263 007502 .SBTTL SOFTWARE PARAMETERS
264 007502 BGNMOD SPTCODE

```

```

550 007502 RGNSW .WORD L10014-L$SW/2
551 (3) 007502 000037
552 007504 000001 LIMIT: .WORD 1 ;RETRY LIMIT
553 007504 000001 ERLMT: .WORD 1 ;ERROR LIMIT
554 007510 000003 SELMT: .WORD 3 ;SEEK ERROR LIMIT
555 007512 060650 DALMT: .WORD 25000 ;DATA XFER LIMIT (*10*3) (BITS)
556 007514 023420 SKLMT: .WORD 10000 ;SEEK LIMIT
557 007520 000160 TYMT: .WORD 120 ;TIME INTERVAL BETW/ STATISTICAL REPORT
558 007520 000030 CMRD: .WORD 24 ;COMPARE ON READ
559 007522 000003 DELMT: .WORD 3 ;ERRORS TO REPORT ON DATA COMPARE
560 007524 000000 XCHPLG: .WORD 0 ;CHANGE OTHER PARAMETERS
561 007524 024000 T.MTB: .WORD 1200 ;MAXIMUM R/W TRANSFER BUFFER
562 007530 000100 T.MNH: .WORD 0 ;MAXIMUM HEAD SELECT
563 007532 000000 T.MNH: .WORD 0 ;MINIMUM HEAD SELECT
564 007534 077600 T.MYC: .WORD 77600 ;MAXIMUM CYLINDER
565 007536 000000 T.MYS: .WORD 0 ;MINIMUM CYLINDER
566 007540 000047 T.MNS: .WORD 39 ;MAXIMUM SECTOR
567 007542 000000 T.MNS: .WORD 0 ;MINIMUM SECTOR
568 007544 000001 T.DCK: .WORD 1 ;DATA DUMP ON DATA CHECK ERROR
569 007546 000001 T.DRP: .WORD 1 ;DROP ON LIMIT REACHED
570 007550 000003 T.DRP: .WORD 3 ;MINIMUM BUFFER TRANSFER SIZE
571 007552 000012 SFLMT: .WORD 10 ;SOFT ERROR LIMIT
572 007554 000000 T.STA: .WORD 0 ;DROP DRIVE ON PERFORMANCE REACHED
573 007556 000000 DRLMT: .WORD 0 ;DRIVE ERROR LIMIT
574 007560 000000 T.POF: .WORD 0 ;READ ONLY FLAG
575 007562 000001 T.RAN: .WORD 1 ;RANDOM SELECT OF PATTERNS
576 007564 000004 T.PAT: .WORD 4 ;ONLY ONE PATTERN 4 = WORST CASE
577 007566 000001 T.SLT: .WORD 1 ;SEEK RETRY LIMIT
578 007570 000200 T.AUT: .WORD 128 ;NUMBER OF ERRORS ON DCK DUMP
579 007572 000000 T.AUT: .WORD 0 ;AUTO ON START UP
580 007574 000000 T.STIP: .WORD 0 ;RESTRICT BUFFER SIZE
581 007576 000001 T.MCK: .WORD 1 ;DD WRITE CHECK
582 007600 000012 T.DCD: .WORD 10
583 007602 ENDSW
584 (3) 007602 L10014:
585 007602 ENDMOD
586 007602 BGNMOD DSPCODE
587 007602 DISPATCH .WORD 1
588 (4) 007602 000001 .WORD 1
589 (6) 007604 012354 .WORD T1
590 007606 ENDMOD
591 .SBTTL STATISTIC CODE
592 BGNMOD RPICODE
593 BGNRPT
594 007606 PRINTS #FMTS1 ;PRINT STATISTICAL HEADER

```

```

(7) 007606 012746 006717      MOV    #FMTS1,-(SP)
(6) 007612 012746 000031      MOV    #1,-(SP)
(3) 007616 016600              MOV    SP,R0
(4) 007620 194016              EMT    CSFNIS
(4) 007622 062706 000004      ADD    #4,SP
602 007626 010446              MOV    R4,-(SP)          ;SAVE PRESENT VALUE OF R4
603
604
605 007630 012704 025056      MOV    #DRRUF,R4        ;START OF DRIVE BUFFER
606 007634 005764 000104      1S:   TST    DCS(R4)      ;IS THERE A DRIVE?
607 007640 001462              BEQ    2$               ;NO, GET NEXT ONE
608
609 007642 004737 011670      JSR    PC,PEPORT        ;TYPE OUT SUMMARY
610
611 007646 052704 002124      2S:   ADD    #PRPOS*2,R4   ;NEXT DRIVE
612 007652 020427 026316      CWP    R4,#ENDBUF      ;AT THE END?
613 007656 001366              BNE    1$               ;NO, TRY NEXT
614
615 007660 012604              MOV    (SP)+,R4        ;RESTORE R4
616
617
618 007662                      ENDRPT
619 (3) 007662                      L10015:
620 007664                      EMT    CSRPT
621
622                      ENDMOD
623
624 007664                      .SBTTL  INITIALIZATION CODE
625
626 007664                      BGNMOD  INITCODE        ;START OF INITIALIZE CODE
627
628 007664                      BGNINIT
629 (3) 007664 012700 000340      SETPRI #340            ;PRIORITY TO SEVEN
630 (3) 007664 104041              MOV    #340,R0
631 007668 104041              EMT    CSSPRI
632 007672 104033              BRESET CSRESET
633
634 007674 005037 000050      CLP    OPFLG
635 007700 005037 002326      CLP    INCALL
636 007704 005037 002302      CLP    STFLG
637 007710 005037 002304      CLP    CRTFLG          ;CLEAR CONT
638 007714              READEF #EF,PWR,R0
639 (3) 007714 012700 000034      MOV    #EF,PWR,R0
640 (3) 007720 104050              EMT    CSREFG
641 007722              BNCOMPLETE 3$
642 007724 103076              BCC    3$
643 007724 005237 002276      INC    PWRFLG          ;INDICATE POWER FAIL
644 007730 012704 025056      MOV    #DRRUF,R4
645 007734 012702 000001      MOV    #1,R2
646 007740 130237 002130      11S:  BITR   R2,DRUT
647 007744 001446              BEQ    13$
648 007746 016400 000106      MOV    DRSEL(R4),R0
649 007752 052700 000200      BIS    #200,R0
    
```

```

645 007756 010074 000104      MOV    R0,#DCS(R4)
646 007762 012701 000074      MOV    #60,R1
647 007766 032772 000001      000104 12S:  BIT    #1,#DCS(R4)
648 007774 001014              BNE    15$
649 007776              WAITMS #10
650 (3) 007776 012700 000012      MOV    #10,R0
651 010004 104025              EMT    CSWTH
652 010006 001367              DEC    R1
653 010010 012737 003796      002126      MOV    #NOPWR,WHY
654 010016 004537 002220      JSR    R5,DRDRV
655 010022 000137 010562              JMP    13$
656
657 010026 004537 021136      15S:  JSR    R5,ISRST
    
```

```

659 010032 004537 022500 JSR R5 HDHOME
660 010036 005064 000056 CLR PRFLGS(R4)
661 010042 005064 000036 CLR RETRY(R4)
662 010046 005064 000076 CLR DOWCK(R4)
663 010052 005064 000052 CLR RSEK(R4)
664 010056 005064 000114 CLR RSEK(R4)
665 010062 002704 000124 13$: ADD #RPROS+2,R4
666 010066 106302 BCC ASLR
667 010070 104337 BCC R7
668 010072 005737 002264 TST SVSCLK
669 010076 001406 BEQ 4$
670 010100 CLKON #1 RO
671 (3) 010104 104034 EMT C$KMON
672 010106 REQTIM RO
673 (3) 010106 104045 EMT C$REQTIM
674 010114 000137 011070 4$: JMP POWER
675 (3) 010120 012700 000036 3$: READEP #EF,CONTINUE ;CONTINUE FROM CONSOLE?
676 (3) 010124 104050 MOV #EF,CONTINUE,RO
677 (2) 010126 103004 EMT C$REFFG
678 (2) 010126 103004 BCC 1$ ;NO, CONTINUE W/ INIT CODE
679 010130 005237 002304 INC CNTFLG
680 010134 000137 010466 JMP END ;YES SET CONT FLAG, GO TO END OF INIT
681 010140 004537 023722 1$: JSR R5,CLEAR ;CLEAR ALL DRIVE BUFFERS
682 010152 012737 123456 002134 MOV #R5C4,RHTNUM ;SETUP RAD SECTOR POINTER
683 010160 012700 002150 002136 MOV #R5C4,LONUM ;GET NUMBER OF UNITS
684 010164 005020 CLR #CNTLRI,RO ;CLEAR FLAGS
685 010172 001374 BNE CLRDAT ;MASS CLEAR
686 010174 012704 MOV #DRBUF,R4 ;SETUP UP DRIVE BUFFER POINTER
687 010200 013703 002012 MOV #R5C4,R3 ;SETUP RAD SECTOR POINTER
688 010210 010337 002274 MOV #LUNIT,R3 ;GET NUMBER OF UNITS
689 010214 005001 MOV #R3,UUT ;SAVE LUNIT
690 010218 005001 CLR R1 ;INIT P-TABLE
691 010222 001522 1$: TST R1 ;INIT P-TABLE LEFT?
692 010222 010100 BEQ END ;NO,GO TO END
693 (3) 010222 104042 CPHARD R1,RO ;GET A P-TABLE
694 (3) 010222 104042 MOV R1,RO
695 (3) 010222 103110 EMT C$CPHARD
696 (2) 010226 103110 BNCMPLETE 12$
697 010234 012037 002169 MOV (R0)+,RCSR ;GET CSR
698 010240 012037 002164 MOV (R0)+,RPRIOR ;GET RPRIOR
699 010244 011037 002164 MOV (R0)+,RDRSEL ;GET DRIVE
700 010248 011037 002164 MOV (R0)+,RDRSEL ;GET DRIVE
701 010252 001374 TST CNTLRI ;DO WE HAVE CSR 1 YET?
702 010256 013737 BNE RPRIOR,PRIOR1 ;YES,THEN SEE IF IT'S IT
703 010264 013737 MOV RPRIOR,PRIOR1
704 010272 013737 MOV RCSR,CNTLRI ;NO,MAKE THIS ONE CSR 1
705 010272 013737 MOV RVEC,VECT1 ;MAKE THIS VECTOR VECT1
    
```

```

706 010300 023737 002160 002150 2$: CMP RCSR,CNTLRI ;IS THIS CSR CNTLRI?
707 010306 001012 5$ RNE 5$ ;NO,GO CHECK AGAINST #2
708 010310 013737 002162 002222 CMP RVEC,VECT1 ;IS VECTOR PROPER?
709 010314 001737 002266 002200 BNE 10$ ;NO,REPORT ERROR
710 010320 012737 011460 JSP #R0P1,TEMP1 ;FIRST CONTROLLER/FIRST BUFFER
711 010326 004537 011460 JSP #R5,FILINF ;FILE BUFFER
712 010330 005730 002152 5$: RR 11$ ;GO GET NEXT P-TABLE
713 010334 005730 002152 5$: RR 11$ ;GO GET NEXT P-TABLE
714 010340 001015 BNE CNTLR2 ;HAVE WE GOT CSR #2 YET?
715 010342 023737 002222 002160 CMP VECT1,BCSR ;YES, CHECK THIS ONE AGAINST IT
716 010350 014333 002160 002152 REQ 10$ ;IS THIS VECTOR SAME AS CNTLRI
717 010360 013737 002160 002152 MOV RCSR,CNTLR2 ;IFSO, DON'T ALLOW IT
718 010366 013737 002164 002224 MOV RVEC,VECT2 ;MAKE THIS ONE CSR 2
719 010374 023737 002160 002152 MOV RPRIOR,PRIOR2 ;SETUP SECOND VECTOR
720 010378 001016 6$: CMP RCSR,CNTLR2
721 010404 023737 002162 002224 BNE 10$ ;IS THIS CSR # 2?
722 010412 001012 002162 002224 CMP RVEC,VECT2 ;NO,WE'LL WE DON'T ALLOW 3
723 010414 023737 002224 002222 RNE 10$ ;DOES IT HAVE PROPER VECTOR
724 010418 023737 002224 002222 CMP VECT2,VECT1 ;NO, GO REPORT ERROR
725 010424 012737 002270 002200 MOV #R0F2,TEMP1 ;IS VECTOR OF FIRST EQUAL TO
726 010432 004537 011460 JSP #R5,FILINF ;VECTOR OF SECOND, YES REPORT ERROR
727 010436 006406 BNE 10$ ;OTHER CNTLR/OTHER BUFFER
728 (3) 010440 104462 10$: TRAP #2,,ERR10 ;LOAD BUFFER
729 (5) 010442 000240 TRAP T$PCODE ;NEXT
730 (5) 010444 002774 .WORD 16$ ;RAD P-TABLE
731 (5) 010446 002774 .WORD ILLEG ;RAD P-TABLE
732 010450 005064 000104 12$: .WORD ERRI0
733 010454 005201 11$: INC R1 ;POINT TO NEXT
734 010460 005703 000040 DEC R3 ;DOWN COUNT
735 010464 000654 ADD #2,,R2 ;NEXT BAD SECTOR FILE
736 (3) 010466 000654 BP 1$ ;DO WHILE
737 010466 012737 177770 002132 END: MOV #177770,SVSMK ;SETUP FOR EIGHT DRIVES
738 010474 003727 002274 000004 CMP UUT,#4 ;MORE THAN FOUR
739 010502 003727 000004 RGT 2$ ;YES, THEN MASK IS OKAY
740 010504 052737 000004 002132 BIS #4,SVSMK ;SETUP FOR FOUR DRIVES
741 010512 023727 000002 002132 CMP UUT,#2 ;MORE THAN TWO
742 010520 003004 000002 002132 RGT 2$ ;YES, IT'S OKAY
743 010524 052737 000002 002132 BIS #2,SVSMK ;SET FOR ONE OR TWO
744 010530 012700 000040 2$: READEP #EF,START,RO ;START COMMAND
745 010534 104050 MOV #EF,START,RO
746 010536 103002 EMT C$REFFG ;NO, CHK RESTART
747 010538 005237 002302 BNCMPLETE RESTART
748 010540 005237 002302 INC STPLG ;SET START INDICATOR
749 010544 005737 002304 RESTART: TST CNTFLG ;CONTINUING
750 010550 001026 002144 BNE 3$ ;YES GO TO 3$
751 010552 005037 002144 CLR WRINIT ;CLEAR THE WRITE INIT FLAG
752
753
    
```

```

755 ;LET'S CREATE INTERNAL BITMAP
756
757 010556 012701 000001 MOV #1,R1 ;BIT MASK
758 010562 105034 002131 CLR R1 ;CLEAR OUT DRIVES PRESENT
759 010568 012704 025056 MOV #DRPRS,R4 ;START OF DRIVE BUFFERS
760 010572 005764 000104 1S: TST DCS(R4) ;ANY CSR?
761 010576 001407 002131 BEQ ZS ;NO, NO DRIVE THEN
762 010580 005301 000124 2S: BISR R1,DRPRS ;INDICATE DRIVE IN BITMAP
763 010584 062704 000124 ADD #RPOS+2,R4 ;NEXT POSITION
764 010606 062704 000124 MOV #RPOS+2,R4 ;NEXT DRIVE BUFFER
765 010612 022704 026316 CMP #ENDBUF,R4 ;DONE
766 010616 001365 BNE 1S ;NO
767
768 010620 113737 002131 002130 MOV R DRPRS,DRUT ;SET UP DRIVES UNDER TEST
769
770 010626 3S:
771
772 010626 SETVEC VECT1,#INTR1,PRIOR1 ;SET CONTROLLER 1'S VECTOR
(7) 010626 MOV PRIOR1,-(SP)
(5) 010636 012746 014222 MOV #INTR1,-(SP)
(4) 010642 012746 000003 MOV #3,-(SP)
(3) 010646 104037 000010 EMT CS$VEC
(2) 010650 062706 000010 ADD #10,SP
773
774 010654 005737 002152 TST CNTRL2 ;RUNNING TWO CONTROLLERS?
775 010660 001413 BEQ 4S ;NO
776
777 010662 SETVEC VECT2,#INTR2,PRIOR2 ;YES SET CONTROLLER 2'S VECTOR
(7) 010662 MOV PRIOR2,-(SP)
(6) 010666 012746 014232 MOV #INTR2,-(SP)
(5) 010672 012746 002224 MOV #3,-(SP)
(4) 010676 012746 000003 MOV #3,-(SP)
(3) 010702 104037 000010 EMT CS$VEC
(2) 010704 062706 000010 ADD #10,SP
778
779 010710 005737 002304 4S: TST CNTRFLG ;CONTINUE?
780 010714 001412 BEQ FINDBF ;NO, GO PAST RESTART OF CLOCK
781
782 010716 005737 002264 TST SVSCLK ;DO WE HAVE SYSTEM CLOCK
783 010722 001462 BEQ POWER ;NO
784
785 010724 CLKON #1 ;TURN CLK ON
(3) 010724 MOV #1,R0
(2) 010732 104034 000001 EMT CS$KON
(1) 010732 104045 BEQ TIM ;REQUEST TIME
(3) 010732 104045 EMT CS$REQTIM
787 010734 104037 002240 MOV #R0,L$TTIM ;MAKE IT PRESENT TIME
788 010740 000453 RR POWER ;GO TO END
789
790 010742 012703 000050 FINDBF: CLD #40,,R3 ;MAXIMUM SECTOR IS 40
791 010746 005001 R1 ;START WC AT ZERO
792 010750 005737 002152 TST CNTRL2 ;TWO CONTROLLERS???
793 010754 001402 BEQ 1S ;NO, START WC AT 5120
794 010756 012701 000024 MOV #20,,R1 ;20 256 WORD BUFFERS
    
```

```

796 010762 062701 000024 1S: ADD #20,,R1
797 010766 010100 2S: BUFFER R1,R2 ;GET BUFFER IF AVAILABLE
(3) 010770 104030 MOV #R1,R0
(2) 010772 010002 EMT CS$BUFF
(1) 010774 001407 000002 MOV #R0,R2
799 010774 103411 BCOMPLETE 4S ;WAS AVAILABLE, THEN RR
(3) 010776 005737 TST CNTRL2
(2) 010782 001401 BEQ 3S ;TWO CONTROLLERS???
(1) 010784 005301 DEC R1 ;ONE 256 WORD BUFFER LESS
802 011006 005301 000002 3S: DEC R1 ;ONE MORE LESS
803 011010 104034 000002 SUB #2,,R3 ;IF NOT ZERO GO BACK
804 011014 001364 BNE 2S
805
806 011016 000000 HALT
807
808 011020 042701 177400 4S: BIC #177400,R1
809 011024 000304 SWAP R1
810 011032 005737 002266 MOV #R2,RUF1 ;GET BUFFER FOR FIRST CONTROLLER
811 011036 001404 TST CNTRL2 ;TWO CONTROLLERS???
812 011040 000102 BEQ 5S
813 011042 005301 002270 ADD #R2,RUF2 ;SECOND'S BUFFER
814 011046 005301 MOV #R2,RUF2
815 011050 010137 002272 5S: MOV R1,MAXWC ;CORRECT WORD COUNT
816 ;MAX WORD COUNT
817
818
819
820 011054 7S: CLKON #1 ;TURN CLOCK ON?
(3) 011054 MOV #1,R0
(2) 011060 104034 EMT CS$KON
(1) 011062 103002 BCOMPLETE POWER ;WAS THERE A CLOCK?
823 011064 005237 002264 INC SVSCLK ;YES, SET FLAG FOR ONE!
824 011070 POWER:
825
826
827
828
829 011070 L10016: ENDINIT
(3) 011070 EMT CS$INIT
(2) 011072 104011 ENDWOD
(1) 011072 RGNMOD CLNCODE
833 011072 BGNCLN
834
835
836
837 011072 SETVEC ER$VEC,#TRPHAN,#340
(7) 011072 MOV #340,-(SP)
(6) 011076 012746 011662 MOV #TRPHAN,-(SP)
(5) 011080 012746 003316 MOV #ER$VEC,-(SP)
(4) 011106 012746 000043 MOV #3,-(SP)
    
```

```

(3) 011112 104037
(2) 011114 062706 000010 EMT CSCVVEC
839 011120 012700 000340 ADD #10,SP
(3) 011120 012700 000340 MOV #340,R0 ;PRIORITY TO SEVEN
840 011124 104041 EMT CSSPRI
841 011126 032777 000200 171014 1S: BIT #CRDY,@CNTLR1 ;WAIT FOR CONTROLLER TO FINISH
842 011134 042774 000100 171004 BEQ 1S ;
843 011136 042777 000100 BIT #INTEN,@CNTLR1 ;CLEAR INTERRUPT IF PENDING
844 011144 013700 002222 CLRVEC VECT1 ;RELEASE VECTOR OF FIRST CONTROLLER
(3) 011144 013700 002222 MOV VECT1,R0
(3) 011150 104036 EMT CSCVVEC
846 011152 005737 002152 TST CNTLR2 ;TWO CONTROLLERS
847 011156 001412 BEQ 3S ;NO
848 011160 032777 000200 170764 2S: BIT #CRDY,@CNTLR2 ;WAIT FOR OTHER CONTROLLER TO FINISH
849 011166 042777 000100 170754 BEQ 2S ;
850 011170 042777 000100 BIC #INTEN,@CNTLR2 ;CLEAR OUT INTERRUPT ENABLE
851 011176 013700 002224 CLRVEC VECT2 ;YES, WELL RELEASE IT'S VECTOR
(3) 011202 104036 MOV VECT2,R0
853 011204 005037 002326 CLR INCALL
854 011210 005037 002324 CLR OPCALL
855 011214 013700 002316 MOV ERVVEC VECT2
856 011220 104036 EMT CSCVVEC
858 011222 005737 002264 TST SVSCLK
859 011226 001461 BEQ 4S
860 011230 CLKOFF
(3) 011230 104035 EMT CSKWOFF
862 011232 104033 4S: BRESET ;THIS IS FOR LSI-11 CPU'S
(3) 011232 104033 EMT CSRESET
863 011234 104012 L10017: ENDCLN
(3) 011234 104012 EMT CSCLEAN
865 011236 ENDMOD
866 011236 BGNMOD ADDCODE
867 011236 BGNAU
868 011236 MOV #DRBUF,R4 ;START OF DRIVE BUFFERS
869 011242 012701 000001 MOV #1,R1 ;MASK TO FIND DRIVE
870 011246 010002 MOV #R2 ;SAVE WHICH TO FIND
871 011250 005700 TST R0 ;THIS ONE
872 011254 001405 BEQ 2C ;YES
873 011254 062704 ADD #PRPOS+2,R4 ;NEXT
874 011260 006301 ASL R1 ;NEXT MASK
875 011262 005300 DEC R0
    
```

```

881 011264 000771
882 011266 150137 002130 2S: RR 1S ;INSEPT IN DRIVE UNDER TEST
883 011272 010200 GPHARD R2,R1
(3) 011274 104942 MOV #R1 CSCPHRD
(3) 011276 010001 MOV R0,R1
884 011304 011164 000104 MOV #R1,DCS(R4) ;SETUP TO CLEAR STATS
885 011304 011164 000100 MOV #R1,R0
886 011310 006200 ASP R0
887 011312 005024 CLR (R4)+
888 011316 001376 DEC R4
889 011320 BNE 4S
890 011320 5S: ENDAU
(3) 011320 L10020: EMT CSAU
(3) 011320 104054 ENDMOD
892 011322 BGNMOD DROPCODE
893 011322 BGNMU
894 011322 005737 002326 TST INCALL
895 011326 001015 RNE 3S
901 011330 012700 025056 MOV #DRBUF,R4
903 011334 005700 TST R0
904 011336 001404 BEQ 1S
905 011340 005300 DEC R0
906 011342 062704 ADD #PRPOS+2,R4
907 011346 006772 RP 2S
908 011350 012737 003313 002126 1S: MOV #REQ,WHY
909 011356 004537 020214 3S: JSP R5,DDPDRV
910 011362 ENDDU
(3) 011362 L10021: EMT CSDU
(3) 011362 104055 ENDMOD
912 011364 .SBTTL GLOBAL SUPROUTINES
913 011364 BGNMOD GLBSUR
914 011364 SETWCK: MOV #R1
915 011370 012700 025056 MOV #DRBUF,R2
916 011402 005002 CMP DCS(R4),DCS(R2)
917 011404 010462 RNE 2S
918 011410 062702 MOV R4,DWCK(R2)
919 011414 005301 ADD #PRPOS+2,R2
920 011416 001366 DEC R1
    
```

```

930 011420 000205          RTS          R5
931
932 011422 012701 000210    CLRWCK: MOV    #R1
933 011423 012702 025056    MOV    HDR6UP,R2
934 011424 025462 000104    000104 1S:  CMP    DCS(R4),DCS(R2)
935 011425 010072          CLR
936 011426 005952          ENE
937 011427 062702 000124    2S:  ADD    #RPOS+2,R2
938 011428 053301          DEC    R1
939 011429 053301          BNE
940 011456 006205          BTE
941
942
943
944
945
946 011460 013764 002166    ;ROUTINE TO FILL BUFFERS WITH INFO
947 011461 013764 002160    FILINF: MOV    HDRSEL,DRSEL(R4)      ;SET DRIVE SELECT BITS
948 011502 013764 006419    MOV    BCSP,DCS(R4)                ;SET CSR
949 011506 005737 007572    MOV    TEMP,BBA(R4)                ;SET R/W BUFFER
950 011512 001460 007572    MOV    R2,RSECTP(R4)               ;SETUP RAD SECTOR POINTER
951
952
953
954 011514 005037 002300    TST    T,AUT                        ;DO WE AUTOSIZE?
955
956
957
958 011514 005037 002300    BEQ    1S                            ;NO, SKIP
959
960
961
962
963
964 011520 012746 000340    CLR    TRPFLG                       ;CLEAR TRAP FLAG
965 011521 012746 011662    SEVVEC ERRVEC,#TRPHAN,#340        ;SETUP TO CATCH TRAP
966 011522 012746 000003    MOV    #340,-(SP)
967 011523 012746 000003    MOV    #TRPHAN,-(SP)
968 011524 012746 000003    MOV    ERRVEC,-(SP)
969 011525 012746 000003    MOV    #3,-(SP)
970 011526 012746 000003    MOV    #3,-(SP)
971 011527 012746 000003    MOV    #3,-(SP)
972 011528 012746 000003    MOV    #3,-(SP)
973 011529 012746 000003    MOV    #3,-(SP)
974 011530 012746 000003    MOV    #3,-(SP)
975 011531 012746 000003    MOV    #3,-(SP)
976 011532 012746 000003    MOV    #3,-(SP)
977 011533 012746 000003    MOV    #3,-(SP)
978 011534 012746 000003    MOV    #3,-(SP)
979 011535 012746 000003    MOV    #3,-(SP)
980 011536 012746 000003    MOV    #3,-(SP)
981 011537 012746 000003    MOV    #3,-(SP)
982 011538 012746 000003    MOV    #3,-(SP)
983 011539 012746 000003    MOV    #3,-(SP)
984 011540 012746 000003    MOV    #3,-(SP)
985 011541 012746 000003    MOV    #3,-(SP)
986 011542 012746 000003    MOV    #3,-(SP)
987 011543 012746 000003    MOV    #3,-(SP)
988 011544 012746 000003    MOV    #3,-(SP)
989 011545 012746 000003    MOV    #3,-(SP)
990 011546 012746 000003    MOV    #3,-(SP)
991 011547 012746 000003    MOV    #3,-(SP)
992 011548 012746 000003    MOV    #3,-(SP)
993 011549 012746 000003    MOV    #3,-(SP)
994 011550 012746 000003    MOV    #3,-(SP)
995 011551 012746 000003    MOV    #3,-(SP)
996 011552 012746 000003    MOV    #3,-(SP)
997 011553 012746 000003    MOV    #3,-(SP)
998 011554 012746 000003    MOV    #3,-(SP)
999 011555 012746 000003    MOV    #3,-(SP)
1000 011556 012746 000003    MOV    #3,-(SP)

```

```

970 011662 005237 011662    TRPHAN: INC    TRPHAN
971 011666 000002          RTI
972
973
974
975 011670
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

```

(11) 012076 016446 000004      MOV     RXFR2(R4),-(SP)
(10) 012102 016446 000060      MOV     RXFR3(R4),-(SP)
(9) 012105 016446 000054      MOV     SKCNT(R4),-(SP)
(8) 012112 016446 000090      MOV     SKCNT(R4),-(SP)
(7) 012112 012746 000006      MOV     #6,-(SP)
(6) 012122 010600 000000      MOV     SP,R0
(5) 012132 062706 000016      EMT     CS%NTS
(4) 012132 062706 000016      ADD     #16,SP
(3) 012133 062706 000016      PRINTS #FMTS2B,WXFR3(R4),WXFR2(R4),WXFR1(R4)
(2) 012133 062706 000016      MOV     WXFR1(R4),-(SP)
(1) 012143 016446 000096      MOV     WXFR2(R4),-(SP)
(9) 012146 016446 000062      MOV     WXFR3(R4),-(SP)
(8) 012152 012746 007144      MOV     #FMTS2B,-(SP)
(7) 012152 012746 000004      MOV     #4,-(SP)
(6) 012162 010600 000000      MOV     SP,R0
(5) 012162 104016 000012      EMT     CS%NTS
(4) 012162 062706 000012      ADD     #12,SP
(3) 012172 016446 000074      PRINTS #FMTS3,DERCNT(R4),SKECNT(R4),TRERR(R4),DATCER(R4)
(2) 012176 016446 000072      MOV     DATCER(R4),-(SP)
(1) 012202 016446 000016      MOV     TRERR(R4),-(SP)
(9) 012202 016446 000020      MOV     SKCNT(R4),-(SP)
(8) 012202 016446 007210      MOV     DERCNT(R4),-(SP)
(7) 012216 012746 000005      MOV     #FMTS3,-(SP)
(6) 012222 010600 000000      MOV     #5,-(SP)
(5) 012222 010600 000014      MOV     SP,R0
(4) 012222 104016 000014      EMT     CS%NTS
(3) 012222 062706 000014      ADD     #14,SP
(2) 012232 016446 000014      PRINTS #FMTS3A,ERRCNT(R4),SFTCNT(R4)
(1) 012232 016446 000017      MOV     SFTCNT(R4),-(SP)
(9) 012242 016446 000024      MOV     ERRCNT(R4),-(SP)
(8) 012246 012746 000003      MOV     #FMTS3A,-(SP)
(7) 012246 012746 000003      MOV     #3,-(SP)
(6) 012252 010600 000010      MOV     SP,R0
(5) 012252 104016 000010      EMT     CS%NTS
(4) 012252 062706 000010      ADD     #10,SP
(3) 012262 016446 000032      PRINTS #FMTS4,DCRCER(R4),HRCER(R4),NXMCNT(R4),HNFERR(R4)
(2) 012262 016446 000034      MOV     HNFERR(R4),-(SP)
(1) 012262 016446 000034      MOV     NXMCNT(R4),-(SP)
(9) 012272 016446 000024      MOV     HRCER(R4),-(SP)
(8) 012276 016446 000027      MOV     DCRCER(R4),-(SP)
(7) 012302 012746 007344      MOV     #FMTS4,-(SP)
(6) 012302 012746 000005      MOV     #5,-(SP)
(5) 012312 010600 000014      MOV     SP,R0
(4) 012312 104016 000014      EMT     CS%NTS
(3) 012312 062706 000014      ADD     #14,SP
(2) 012322 016446 000030      PRINTS #FMTS5,DLTCNT(R4),OPICNT(R4)
(1) 012322 016446 000026      MOV     OPICNT(R4),-(SP)
(9) 012322 012746 007430      MOV     DLTCNT(R4),-(SP)
(8) 012332 012746 000003      MOV     #FMTS5,-(SP)
(7) 012332 012746 000003      MOV     #3,-(SP)
(6) 012336 010600 000000      MOV     SP,R0
(5) 012336 104016 000010      EMT     CS%NTS
(4) 012336 062706 000010      ADD     #10,SP
(3) 012344 010600 000010      MOV     SP,R0
(2) 012344 104016 000010      EMT     CS%NTS
(1) 012352 00207 000207      RTS     PC
    
```

```

996      ENDMOD
997      .SBTTL PROGRAM MAIN LOOP
998
999      BGNST
1000     #MAIN PROGRAM LOOP
1001     #PROGRAM WILL RANDOMLY PICK ONE OF THE DRIVES TO
1002     #PERFORM AN OPERATION. WE WILL ALWAYS PICK ONE OF FOUR
1003     #DRIVES (NO CONTROLLERS) "DR0" WILL BE
1004     #CHECKED TO SEE IF DRIVE IS ON SYSTEM. ONCE DRIVE IS PICKED
1005     #THEN A FUNCTION WILL BE SELECTED RANDOMLY FOR THAT
1006     #DRIVE. FUNCTIONS OF CONTROLLER RESET, GET STATUS, READ, WRITE
1007     #WILL BE SELECTED EACH FUNCTION WILL HAVE IT'S OWN ROUTINE
1008     #TO GET PARAMETERS FOR THE DRIVE.
1009
1010     MTEST: TST     WPNIT          ;SEE IF WRITE INIT IN PROGRESS
1011            BEQ     161S         ;JUMP OVER IF NOT INIT FROM PWR FAIL
1012            MOV     WRINIT,R4    ;VUP - RESET R4 POINTER
1013            WRPOS,R1            ;AND THE DRIVE POINTER FOR INIT
1014            IRC     SFLG         ;RAKE OUT THE START FLAG
1015            BR     15S          ;AND CONTINUE WITH THE WRITE CODE
1016
1017     161S: MOV     #DRBUF,R4    ;GET DRIVE BUFFERS
1018            MOV     #1,R3        ;MASK
1019            MOV     R4,WRPOS     ;COPY R4
1020            BITB   R1,DRUT      ;AND R1 POINTERS
1021            BEQ     15S         ;DRIVE UNDER TEST
1022
1023     16S:  MOV     #20C,#DCS(R4) ;CHECK IF DRIVE THERE
1024            BRS   #DRSEL(R4),#DCS(R4)
1025            DFC   #R0           ;STALL
1026            RNE   13S
1027            BIT   #RDY,#DCS(R4)
1028            BNE   14S
1029
1030     13S:  MOV     #DNRDY,WHV    ;GO GET BAD SECTORS
1031            JSP   #DRDRV
1032            BR   15S
1033
1034     14S:  JSR    #PDPDSC        ;GO GET BAD SECTORS
1035            CLR   #PFLGS(R4)
1036            CLR   #DCWCK(R4)
1037            CLR   #PSEK(R4)
1038            TST   #WRPG(R4)
1039            BNE   99S
1040            TST   #SFLG
1041            BEQ   15S
1042
1043     99S:  JSR    R5,WRPACK      ;SEE IF WRITE IN PROGRESS FLG SET
1044            ;VUP - RE-INIT THIS DRIVE
1045
1046     15S:  ADD     #PRPOS+2,R4   ;NEXT DRIVE
1047
1048
1049
1050
1051
    
```

```

1052 012542 016437 002144      MOV     R4,WRINIT
1053 012546 006337 002146      ASL     WPPOS
1054 012552 106301                ASLB    R1
1055 012554 103321                BCC     R1$
                                ;COPY FOR POSSIBLE PWR FAIL
                                ;SHIFT THE POSITION FLAG ALSO
                                ;DONE?
                                ;NO GO FOR NEXT ONE
1056 012556 005037 002144      CLR     WRINIT
1057 012562 012746 004165      PRINTF #FMT14,#MSTART
                                ;CLEAR WRITE INIT FLG ... ALL DONE
                                ;
012562 012746 004165      MOV     #MSTART,-(SP)
012562 012746 006507      MOV     #FMT14,-(SP)
012562 012746 000002      MOV     #2,-(SP)
012576 016600                MOV     #2,-(SP)
012600 104017                EMT     C$ENTF
012602 062706                ADD     #4,SP
1059 012606 000000                SETPRI #0,R0
                                ;PRIORITY TO ZERO
012606 012700 000000      MOV     #0,R0
012612 104044                EMT     C$SPRI
1060 012614 004534                MOV     R5,RAND
012620 004534                MOV     R5,RAND
012624 043705                LONUM  R2
012630 012701                PEROTH: R1C LONUM R2
                                ;GET A DRIVE?(LUN)
                                ;GET THE SELECTED DRIVE (LUN)
                                ;MASK TO DRIVES ON SYSTEM
012634 005702                MOV     #1,R1
                                ;LET'S SEE IF DRIVE IS THERE
012636 001403                1$:    TST     R2
                                ;HAVE WE GOT PROPER MASK YET
012640 006301                BEQ     R2$
                                ;YES, GO TO 2$
012642 005302                ASL     R2
                                ;NO, SHIFT FOR NEXT DRIVE
012644 005702                DEC     R2
                                ;DECREMENT DRIVE NUMBER
012646 005702                BR     R2
                                ;GO CHECK NEW DRIVE NUMBER
012652 001005                2$:    TSTR   DRUT
                                ;ANY DRIVES ON LINE
                                ;YES, CHECK
012654 004421                FRFSF  170,MODRIV
                                ;NO DRIVES
012656 000252                TRAP   RCODE
012660 003622                .WORD 176
                                .WORD NODRIV
1074 012662 000137 025050      JMP     ENDOFPROGRAM
1076 012666 130137 002130      5$:    BITB  R1,DRUT
                                ;IS THIS DRIVE PRESENT?
1078 012672 001750                REQ     MAIN
                                ;NO, GO BACK TRY AGAIN
1079
                                ;WE NOW HAVE A DRIVE, CHECK TO SEE IF IT'S CONTROLLER
                                ;IS FREE BEFORE WE GO ANY FURTHER
1084 012674 004537 022356      JSR     R5,GETSYS
                                ;GET PRESENT TIME OF SYSTEM
1085 012706 004203 002236      CMP     INTERVAL,TVINT
                                ;TIME TO PRINT REPORT
1086 012710 005037 002236      BPL    CLP
                                ;NO, PERFORM FUNCTION
1087 012714 104024                CLP     INTERVAL
                                ;YES, START INTERVAL OVER
1090 012714 104024                DORPT  EMT
                                ;PRINT STATISTICAL REPORT
1092 012716 012704 025056      6$:    MOV     #DPRUF,R4
                                ;GET START OF DRIVE BUFFERS
1093 012728 043705 002136      MOV     LONUM,R2
                                ;GET RANDOM DRIVE BACK (LUN)
1094 012732 005702 002132      BIT    SYSMSK,R2
                                ;MASK TO SYSTEM SYS
1095 012734 001404                BEQ     4$
                                ;DO WE HAVE BUFFER FOR THAT DRIVE
                                ;YES, GO CHECK IT'S CONTROLLER
    
```

```

1096 012736 062704 000124      ADD     #RPROS+2,R4
1097 012744 004537 000124      DEC     R2
1098 012746 032774 000200 000104 4$:    BR     R2
                                ;NO, UPDATE FOR NEXT BUFFER
1099 012754 001717 000100 000104      BIT     #BIT7,@DCS(R4)
                                ;NOW COUNT DRIVE NUMBER (LUN)
1100 012758 001717 000100 000104      BRQ    MAIN
                                ;GO BACK AND CHECK FOR FOUND
                                ;CONTROLLER ASSOCIATED WITH DRIVE
1102 012764 001313                BIT     #BIT6,@DCS(R4)
                                ;INTERUPT BEEN SERVICED?
1103 012764 001313                BNE    MAIN
                                ;
1106
                                ;WE CAN NOW PROCEED IN GETTING A FUNCTION AND RELATED DATA
                                ;FOR THE DRIVE RANDOMLY. R4 HAS DRIVE BUFFER POINTER
1107 012766 005737 007546      TAGX:  TST     T,DRP
1108 012772 001456                BEQ     R9$
                                ;DROP ON ERROR LIMITS REACHED?
1109 013002 003404 000012 007506      CMP     ERRCNT(R4),EPLMT
                                ;NO
1110 013004 012737 003125 002126      BLD    #EPLMTM,WHY
                                ;HARD REACHED?
1111 013012 000442                MOV     R1$
1112 013022 163404 000014 007552 9$:    CMP     SPICNT(R4),SPLMT
                                ;SOFT REACHED?
1113 013024 012737 003170 002126      BLD    1C$
1114 013032 000432                MOV     R1$
                                ;
1115 013042 163404 000074 007600 10$:   CMP     DATCEP(R4),T.DCD
1116 013044 012737 003212 002126      BLD    11$
                                ;
1117 013052 000422                MOV     #DCCMSG,WHY
1118 013054 016401 000074 000115      BR     R1$
1119 013064 020137 007510      110$:  MOV     SPECNT(R4),R1
1120 013070 103404                ADD     TRERR(R4),R1
1121 013072 012737 000072                CMP     R1,SELMT
1122 013100 006407 003147 002126      BLD    #SELMT,WHY
1123 013102 026437 000020 007556 7$:    MOV     R1$
                                ;DRIVE ERROR REACHED?
1124 013110 103407                BR     R1$
1125 013120 004537 002126      7$:    RLO    #DERMSG,WHY
1126 013124 000137 012614      11$:   JSR     R5,DRDRV
                                ;DROP THIS DRIVE!!!
1127 013130 005764 000076                JMP     MAIN
                                ;GO GET ANOTHER
1128 013134 001404                8$:    TST     DOWCK(R4)
                                ;WRITE CHECK NEEDED
1129 013136 001404 000076                BEQ     R4$
                                ;NO
1130 013142 012764 000002 000044      MOV     #WRCHK,FUNC(R4)
                                ;GET ONE THAT NEEDS TO BE WRCHK'D
1131 013150 001404 000036      JMP     ISSUE
                                ;WRITE CHECK
1132 013160 001402                80$:   TST     RETRY(R4)
                                ;ISSUE IT
1133 013162 000137 014052      REQ     R5$
                                ;DOES DRIVE HAVE RETRY IN
1134 013166 005764 000114      JMP     ISSUE
                                ;PROGRESS, NO CONTINUE
                                ;GO RETRY COMMAND
1142 013172 001003                78$:   TST     R$EEK(R4)
                                ;RECOVERY FROM SEEK ERROR
1143 013174 000427                BNE    R7$
                                ;NO
1144 013176 000137 014002                BR     R7$
                                ;GO READ
1145 013202 032764 000001 000056 77$:   JMP     R$DFNC
                                ;NO, CONTINUE
1146 013210 001002 000001                BR     R7$
                                ;GO READ
1147 013212 001002 013366      77$:   BIT     $SKDON,PRFLGS(R4)
                                ;SEEK BEEN VERIFIED
1148 013214 000137 013712      BNE    R7$
                                ;NO
1149 013216 000137 013712      JMP     R$HFNC
                                ;GO, TRY TO RECOVER
1150
                                ;GO VERIFY SEEK
1151
                                ;CHECK LIMITS OF ERRORS/OPERATIONS
    
```



```

1152
1153 013222 032764 000001 000056 GETFNC: BIT #SKDON,PRFLGS(R4) ;SEEK NEED TO BE VERIFIED?
1154 013230 001402 REQ RS ;NO, CONTINUE
1155 013232 000137 013712 JMP RDHFNCR ;GO VERIFY SEEK
1156
1157
1158
1159 013236 005737 007554 RS: TST T.STA ;DO WE WISH TO DROP ON OPR LIMITS
1160 013242 001422 REQ 98$ ;NO
1161
1162 013244 026437 000000 007514 CMP SKCNT(R4),SKLMT ;PAST THE SEEK LIMIT??
1163 013252 103416 BLO 98$ ;NO THEN GO TEST
1164 013254 086400 MOV R3,R3(R4),R0 ;GET READ COUNT
1165 013260 086400 ADD WYFR3(R4),R0 ;ADD IN WRITE COUNT
1166 013264 020037 007512 CMP RC,DALMT ;LIMIT REACHED??
1167 013270 103407 BLO 98$ ;NO, THEN GO TEST
1168 013272 027737 MOV #SDPLMT,WHY ;DROPPED THE DRIVE
1169 013300 004537 JSP PS,DRDRV ;GO FOR ANOTHER DRIVE
1170 013304 000137 012614 JMP MAIN
1171
1172 013310 004537 021214 98$: JSR R5,PAND ;GET FUNCTION, LEGAL FUNCTIONS
1173 ;ARE: 1 (WRITE CHECK)
1174 ; : 2 (SEEK STATUS)
1175 ; : 3 (SEEK)
1176 ; : 4 (RD HEADER)
1177 ; : 5 (WRITE)
1178 ; : 6 (READ)
1179 ; : 7 ARE NOT LEGIT
1180
1181 013314 013702 002136 MOV LONHM,R2 ;GET IT
1182 013320 042702 177770 BIC #177770,R2 ;MASK TO 0-7
1183 013324 001001 BNE 6$ ;IF 0, MAKE 1
1184 013326 005202 INC R2
1185 013334 001001 000007 6$: CMP #7,R2 ;IS IT 7?
1186 013336 005302 BNE 5$ ;IF 7, MAKE 6
1187 013340 006302 017470 5$: ASL R2 ;SHIFT LEFT (X2)
1188 013342 000172 JMP RLIST(R2) ;GO TO FUNCTION ROUTINE
1189
1190 .SBTTL ROUTINE TO SETUP AND ISSUE GET STATUS
1191
1192 ;WE GET HERE BY FALLING THRU "LIST" WITH A RANDOM FUNCTION OF 2.
1193
1194 013346 012764 000004 000044 GSTFNC: MOV #GSTAT,FUNC(R4) ;LOAD GET STATUS
1195 013354 012764 000003 000040 MOV #GSRIT,RDA(R4) ;SET GSRIT IN COMMAND WORD
1196 013362 000137 014052 JMP ISSUE ;GO ISSUE FUNCTION
1197
1198 .SBTTL ROUTINE TO SETUP AND ISSUE SEEK FUNCTION
1199
1200 ;WE GET HERE BY FALLING THRU "LIST" WITH A RANDOM FUNCTION OF 3.
1201 ;WE WILL CALL "RAND" FOR A NEW DISK ADDRESS TO SEEK
1202 ;TD. ANY TRACK BUT LAST IS LEGAL. WE WILL ALSO INCREMENT
1203 ;IT'S SEEK COUNT
1204
1205 013366 005764 000114 SKFNC: TST RSEFK(R4) ;TRYING TO RECOVER
1206 013372 001411 REQ 98$ ;NO, CONTINUE
1207
    
```

```

1208 013374 016401 000050 MOV LSTHDR(R4),R1 ;YES SET UP FOR RESEEK
1209 013400 016402 000122 MOV PRPOS(R4),R2 ;TO CYLINDER
1210 013404 042701 000100 BIC #100,R1 ;HEAD SET IN LATER
1211 013410 042702 000100 BIC #100,R2
1212 013414 000507 BR 4$ ;SKIP RANDOM PART
1213 013416 004537 021214 98$: JSP R5,RAND ;GET A RANDOM NUMBER
1214 013422 013702 002136 MOV LONHM,R2 ;LEAVE CYL AND HEAD
1215 013426 043702 000136 BIC SECSK,R2 ;ON THAT TRACK ALREADY
1216 013432 020264 000122 CMP R2,PRPOS(R4)
1217 013436 001002 RNE 98$ ;NO, CONTINUE
    
```

```

ASSEMBLY ROUTINES MACY11 30A(1052) 30-NOV-78 18:42 PAGE 3
CZRLER.P11 30-NOV-78 18:28 ROUTINE TO SETUP AND ISSUE SEEK FUNCTION

```

1219	013440	000137	013222		JMP	GETFNC		;YES, DON'T RESEEK
1220	013444	005003		90S:	CLP	R3		
1221	013446	010780			MOV	#17,R6		;COPY
1222	013450	042700	177677		BIC	#17,R0		;LEAVE ONLY HEAD
1223	013454	023737	007534	007536	CMP	T,MXC,T,MNC		;MIN AND MAX CYLINDERS THE SAME
1224	013462	001003			95S:	MOV		;NO BRANCH AND STAY IN LIMITS
1225	013466	000430	007534		MOV	T,MXC,R2		;MAKE CYLINDER MAX/MIN
1226	013470	011776			BR			;GO CALCULATE DIFF AND SEEK
1227	013472	042702	000100		95S:	BIC	#HEAD,R2	;STRIP OUT H.S. BIT
1228	013476	023702	007534		94S:	CMP	T,MXC,R2	;IS ADDRESS LESS/EQUAL THAN MAX
1229	013502	163010			BHS	R3		;YES, CHECK LOW END
1230	013504	005203			INC	R3		
1231	013506	020327	000012		CMP	R3,#10.		
1232	013514	001741			REQ	98S		
1233	013514	000204			ASR	R2		;HALF IT AND CHECK AGAIN
1234	013516	062702	000200		91S:	#BIT7,R2		;JUST TO MAKE NON ZERO
1235	013522	000763			BR			;GO BACK AND CHECK AGAIN
1236	013524	083709	007536		93S:	CMP	T,MXC,R2	;IS MIN GREATER/EQUAL THAN ADDRESS
1237	013536	101410			BLOS	R2		;YES, CALCULATE DIFF AND SEEK
1238	013532	005203			INC	R3		
1239	013534	020327	000012		CMP	R3,#10.		
1240	013540	006175			REQ	98S		
1241	013544	000204			ASR	R2		
1242	013544	042702	100000		BIC	#BIT5,R2		;NO DOUBLE IT
1243	013550	000763			BR			;IF IT CAN'T SET
1244	013552	016401	000122		92S:	91S		;GO CHECK MAX/MIN AGAIN
1245	013552	043701	002140		MOV	PRPOS(R4),R1		;GET PRESENT DISK POSITION
1246					BIC	CYLMSK,R1		;CLEAN OUT ITS SECTOR BITS
1247	013562	016464	000122	000050	MOV	PRPOS(R4),LSTHDR(R4)		;SAVE LAST
1248	013570	010264	000122		MOV	R2,PRPOS(R4)		;NEW HEADER AFTER SEEK
1249	013574	050064	000001		BIS	R0,PRPOS(R4)		;SET IN RANDOM HEAD GOTTEN
1250	013600	023737	007530	007532	CMP	T,MXC,T,MNH		;MIN AND MAX HEAD SELECT THE SAME
1251	013606	001011			96S:	RNE		;NO, THEN WE CAN USE BOTH SURFACES
1252	013610	005737	007530		TST	T,MXH		;WHICH IS OUR SURFACE FOR USE
1253	013616	001402	000100		RNE			;TOP SURFACE BRANCH
1254	013616	044364	000100	000122	BIC	#HEAD,PRPOS(R4)		;LOWER SURFACE ONLY
1255	013624	000403			BR	96S		
1256	013625	052764	000100	000122	97S:	RIS		;HEAD,PRPOS(R4) ;TOP SURFACE ONLY
1257					96S:			
1258								
1259								;CALCULATE THE DIFFERENCE WORD AND STORE IT IN RDA
1260								
1261								
1262	013634	160102			4S:	SUB	R1,R2	;SUBTRACT PRESENT FROM NEXT
1263	013636	100009			BPL	1S		;IF POSITIVE RESULT GO TO 1S
1264	013640	005402			NEG	R2		;NEG RESULT, NEGATE IT
1265	013642	000402			BR			;GO SET DIRECTION OUT
1266	013644	052702	000004		1S:	2S		;DIRECTION OUT, MARKER
1267	013650	001011			2S:	BIS	#MK,R2	;MARKER BIT
1268	013654	032764	000100	000122	BIT	#HEAD,PRPOS(R4)		;WHICH SURFACE SELECTED?
1269	013658	001402			REQ			;TOP THEN 3S
1270	013664	052702	000020		3S:	3S		;BOTTOM SET HEAD BIT
1271	013670	010264	000040		MOV	R2,RDA(R4)		;MOVE DIFFERENCE WORD TO DA
1272	013674	010264	000066		MOV	R2,DIFWD(R4)		;LOAD DIFFERENCE WORD
1273	013678	010264	000085		MOV	#SEK,FUNC(R4)		;LOAD SEEK
1274	013708	006137	013082	000044	JMP	ISSUE		

```

ASSEMBLY ROUTINES MACY11 30A(1052) 30-NOV-78 18:42 PAGE 3-1
CZRLER.P11 30-NOV-78 18:28 ROUTINE TO SETUP AND ISSUE SEEK FUNCTION

```

1275  
1276 .SBTTL ROUTINE TO LOAD READ HEADER AND ISSUE IT.  
1277  
1278 ;WE GET HERE BY FALLING THRU "LIST" WITH A RANDOM FUNCTION OF 4.  
1279  
1280  
1281 013712 012764 000010 000044 RDHFNC: MOV #RDHDR,FUNC(R4) ;LOAD READ HEADER  
1282 013720 000137 014052 JMP ISSUE  
1283  
1284 .SBTTL ROUTINE TO LOAD WRITE DATA COMMAND  
1285  
1286  
1287 013724 022764 077700 000122 WRWFNC: CMP #77700,PRPOS(R4) ;ON LAST TRACK?  
1288 013732 001002 BNE 98S ;NO CONTINUE  
1289 013734 000137 013366 JMP SKFNC ;YES CONTINUE  
1290 013740 005737 007560 98S: TST T,ROF ;READ ONLY  
1291 013744 01402 BFG 97S ;NO  
1292 013746 000137 014002 JMP RDHFNC ;YES  
1293 013752 004537 022576 97S: JSR R5,GWCDA ;GET WORD COUNT,DA  
1294  
1295 ;WE NOW HAVE SECTOR AND WORD COUNT, LET'S WRITE BUFFER IN MEMORY  
1296 ;TO WRITE OUT TO DISK  
1297 ;FORMAT:  
1298 ; WORD 1 - # OF WORDS IN SECTOR  
1299 ; WORD 2 - ADDRESS OF PATTERN WRITTEN ON SECTOR  
1300 ; WORD 3 - 127 DATA PATTERN  
1301  
1302 013756 004537 017230 JSR R5,WRRUF ;WRITE BUFFER INTO MEMORY  
1303 013762 012764 000012 000044 MOV #WRITE,FUNC(R4) ;LOAD WRITE  
1304 013770 012764 000001 000120 MOV #1,WRIFG(R4) ;SET WRITE IN PROGRESS FLAG  
1305 013776 004137 014052 JMP ISSUE ;GO ISSUE FUNCTION  
1306  
1307 .SBTTL ROUTINE TO LOAD READ DATA COMMAND  
1308  
1309 ;THIS ROUTINE WILL FIRST CLEAR OUT THE BUFFER AREA,  
1310 ;SELECT A RANDOM NUMBER OF WORDS TO READ AND A  
1311 ;RANDOM SECTOR ON THE PRESENT CYLINDER TO READ FROM  
1312  
1313 014002 022764 077700 000122 RDDFNC: CMP #77700,PRPOS(R4) ;ON LAST TRACK?  
1314 014010 001002 BNE 99S ;NO CONTINUE  
1315 014012 000137 013366 JMP SKFNC ;YES SEEK OFF IT.  
1316 014016 004537 022568 99S: JSR R5,GWCDA ;GET WORD COUNT, DA  
1317 014022 016402 006742 97S: MOV RMO(R4),R2 ;CLEAR OUT BUFFER AREA  
1318 014026 017401 000110 MOV @RRA(R4),R1 ;SO WE KNOW READ  
1319 014032 005021 1S: CLR (R1)+ ;WORKED!!  
1320 014034 005262 INC R1  
1321 014036 001375 BNE 1S  
1322 014040 012764 000014 000044 MOV #READ,FUNC(R4) ;LOAD READ  
1323 014046 000137 014052 JMP ISSUE  
1324  
1325 .SBTTL SETUP CONTROLLER AND DRIVE INFO FOR INTERRUPT PROCESSING  
1326  
1327 ;WE COME HERE BEFORE ISSUING ANY FUNCTION SO THAT ON INTERRUPT  
1328 ;WE CAN PROPERLY PROCESS THE INTERRUPT. WE WILL CHECK WHICH  
1329 ;CONTROLLER WE ARE WORKING WITH AND STORE OFF THE DRIVE BUFFER  
1330 ;POINTER IN IT'S "LSTDR"

```

1331
1332
1333
1334 014052 026437 000104 002150 ISSUE: CMP DCS(R4),CNTLR1 ;DRIVE ON CONTROLLER 1?
1335 014060 001003 ;NO ASSUME ON CONTROLLER 2
1336 014062 010437 002154 MOV R4,LSTDRI ;PUT BUFFER POINTER IN
1337 014066 000407 BR 2S ;SKIP OVER NEXT INSTRUCTION
1338 014070 052784 000100 000044 1S: MOV R4,LSTD2 ;PUT BUFFER POINTER IN 2
1339 014102 004537 014112 2S: RTN FUNC(R4) ;ALLOW INTERRUPTS
1340 014106 000137 012614 JSR R5,LDFUNC ;NO WE ISSUE IT
1341 ;GO BACK AND DO ANOTHER
1342
1343 .SBTTL ROUTINE TO LOAD FUNCTION
1344
1345 ;CALL JSR R5,LDFUNC
1346 ;ALL INFORMATION MUST BE SET UP IN DRIVE BUFFER
1347 ;R4 HAS POINTER TO BUFFER
1348
1349 014112 016403 000104 LDFUNC: MOV DCS(R4),R3 ;GET CSR FOR DRIVE
1350 014116 032713 000200 BIT #BIT7,(R3) ;CAN WE ISSUE COMMAND?
1351 014122 001003 RNE 1S ;YES, GO ISSUE COMMAND
1352
1353 014124 ERRSF 200,PRGER ;THIS ERROR SHOULD NEVER PRINT
1354 (3) 014124 TRAP #SERCODE
1355 (5) 014126 .WORD 200
1356 (7) 014130 .WORD PRGER
1357
1358 014132 017463 000110 000002 1S: MOV @RA(R4),@A(R3) ;LOAD BUS ADDRESS REGISTER
1359 014140 016463 000040 000004 MOV @DA(R4),@A(R3) ;LOAD DISK ADDRESS REGISTER
1360 014146 016463 000042 000006 MOV @MP(R4),@MP(R3) ;LOAD MULTI-PURPOSE REGISTER
1361 014154 016463 000044 000046 MOV @RSEL(R4),@RCSADR(R4) ;GET FUNCTION
1362 014162 056464 000106 000046 RTS DRSEL(R4),RCSADR(R4) ;SET DRIVE SELECT BITS
1363 014170 052784 000201 000046 BIS #CRDY:DRDY,RCSADR(R4) ;SET CRDY:DRDY IN IMAGE
1364 014176 042764 017800 000046 BIC #DPT,RCSADR(R4) ;WE'RE CLEAR BIT 10 FOR DRIVE 7-4 (OKAY?)
1365 014184 016463 000046 000000 MOV @RCSADR(R4),CS(R3) ;LOAD CSR
1366 014212 042763 000200 000000 BIC #CRDY,CS(R5) ;ISSUE FUNCTION
1367 014220 000205 RTS R5 ;EXIT
1368
1369 .SBTTL INTEPRUPT SERVICE ROUTINES
1370
1371 014222 BGNSRV INTR1
1372
1373 ;ON INTERRUPT WE CHECK FOR ERRORS FIRST, IF NO ERRORS WE
1374 ;CHECK FUNCTION PERFORMED, WE ACT ACCORDING IF FUNCTION IS:
1375 1- WRITE CHECK - NOTHING IF NO ERROR
1376 2- GET STATUS - READ AND CHECK DRIVE STATUS
1377 3- SEEK - NOTHING RTN - SET HDR AS NEXT COMMAND
1378 4- RDHDP - COMPARE HEADER TO PRESENT POSITION
1379 5- WRITE - UPDATE XFER COUNT, EXIT
1380 6- READ - COMPARE DATA IF REQUESTED, UPDATE XFER COUNT, EXIT
1381
1382 ;ALL SUCCESSFUL EXITS FROM INTERRUPT ROUTINE TEST RETRY
1383 ;LIMIT IF RETRY IS LESS THEN LIMIT THEN LOG SOFT ERROR, CLEAR RETRY
1384 ;IF RETRY = 0, THEN NOTHING
  
```

```

1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403 014222 010446 000120 002250 INTR1: MOV R4,-(SP) ;SAVE PRESENT R4 VALUE
1404 014224 013704 000104 BR LSTDRI,R4 ;GET THE DRIVE BUFFER OF INTERRUPTING DRIVE
1405 014230 000403 MOV R4,SAVE ;GO SAVE R0-R3
1406 014232 010446 INTR2: MOV R4,-(SP) ;SAVE PRESENT R4 VALUE
1407 014234 013704 000156 SAVE: MOV LSTD2,R4 ;GET THE DRIVE BUFFER OF INTERRUPTING DRIVE
1408 014240 013746 000250 MOV E.CS,-(SP)
1409 014244 013746 000254 MOV E.DA,-(SP)
1410 014250 013746 000258 MOV E.MP,-(SP)
1411 014254 013746 000256 MOV E.MP1,-(SP)
1412 014260 013746 000260 MOV E.MP2,-(SP)
1413 014264 013746 000264 MOV F.MP,-(SP)
1414 014270 013746 000272 MOV CHKSEC,-(SP)
1415 014274 013746 000270 MOV HDRFND,-(SP)
1416 014280 013746 000276 MOV TEMP1,-(SP)
1417 014304 013746 000276 MOV TEMP2,-(SP)
1418 014310 013746 000284 MOV @CALL,-(SP)
1419 014314 013746 000282 MOV @CALL,-(SP)
1420 014320 010346 MOV R3,-(SP) ;SAVE R3
1421 014324 010346 MOV R2,-(SP) ;R2
1422 014328 010146 MOV R1,-(SP) ;R1
1423 014332 010046 MOV R0,-(SP) ;R0
1424 014336 000120 CLR WRIPC(R4) ;CLEAR THE WRITE IN PROGRESS FLAG
1425 014340 000164 MOV DCS(R4),R3 ;GET CSR FOR INTERRUPT
1426 014346 016337 000002 002250 MOV @R(R4),@R(R3) ;SAVE ALL REGISTERS NOW!!
1427 014352 016337 000004 002254 MOV @DA(R4),@DA(R3)
1428 014358 016337 000006 002258 MOV @MP(R4),@MP(R3)
1429 014364 016337 000008 002262 MOV @MP1(R4),@MP1(R3)
1430 014370 016337 000006 002262 MOV @MP2(R4),@MP2(R3)
1431 014376 016337 000006 002262 MOV @MP3(R4),@MP3(R3)
1432 014404 008702 002250 TST E.CS
1433 014408 008702 RTN IS ;ANY ERRORS?
1434 014412 000137 015532 JMP CNKFN ;YES, GO SOLVE ERROR MYSTERY
1435 ;NO, GO SEE IF WE HAVE TO DO ANYTHING
1436
1437 .SBTTL CONTROLLER ERROR CHECK ROUTINE
1438
1439 ;WE HAVE SOME SORT OF ERROR LET'S FIND OUT WHICH ONE
1440 ;IT IS.
  
```

```

1440
1441 014416 013764 002254 000064 1S:  MOV     E,DA,LSTDA(R4) ;SAVE DA FOR SOFT ERROR PRINT
1442 014424 032737 040000 002250  BIT     #DERR,E.CS ;DRIVE ERROR?
1443 014432 000137 016514 000001  BEQ     2$ ;NO, CONTINUE
1444 014434 000137 016514 000001  JMP     CKDERR ;DRIVE READY THERE
1445 014440 032737 000001 002250 2S:  BIT     #DRDY,E.CS ;YES, GO CHECK DRIVE ERROR
1446 014446 001017 000001  BEQ     23$ ;NO, CONTINUE CHECKING
1447 014450 001017 000001  JSR     R5,GETDST ;NO, GET DRIVE STATUS
1448 014454 042701 000100  BIT     #100,R1 ;GET RID OF HEAD
1449 014460 020127 000034  CMP     R1,#34 ;ALLOW ONLY SEEK TRACKING STATE
1450 014464 001410  BEQ     23$ ;WAS 34 SKIP ERROR
1451
1452 014466 005264 000012  INC     ERRCNT(R4) ;INDICATE HARD ERROR
1453 014472 104462  ERRDF 1000,NDROY,ERR9 ;INDICATE HARD ERROR
1454 014474 001750  TRAP   TSEPCODE
1455 014476 002507  .WORD 1000
1456 014500 005050  .WORD NRDY
1457 014502 000137 016344  .WORD -WORD ERR9
1458 014502 000137 016344  JMP     EXIT1
1459
1460 014506 032737 020000 002250 23S: BIT     #NXM,E.CS ;NON-EXISTANT MEMORY?
1461 014514 001407  BEQ     3$ ;NO, KEEP CHECKING
1462 014514 001407  MOV     #MTNXM,RTYPE(R4) ;ERROR MESSAGE
1463 014524 005264 000034  INC     NWCNT(R4) ;LOG ERROR
1464 014530 000137 015136  JMP     ;CHECK RETRY, EXIT BACK
1465
1466 014534 032737 014000 002250 3S:  BIT     #RT12IBIT11,E.CS ;QUALIFYING BITS SET?
1467 014542 001020  BNE     5$ ;YES, CAN'T BE OPI ALONE
1468
1469 014544 032737 002000 002250  BIT     #OPI,E.CS ;OPI SET?
1470 014552 001006  BNE     4$ ;YES, CONTINUE
1471
1472 014554 104461  ERRSF 10,UDERR,ERR1 ;WE HAVE AN UNDIAGNOSABLE CONDITION, ONLY COMPOSITE SET
1473 014554 104461  TRAP   TSEPCODE
1474 014560 002212  .WORD 10
1475 014562 004304  .WORD UDERR
1476 014564 104022  .WORD ERR1
1477 014566 000776 33S:  BREAK
1478 014566 000776  EMT   CSBRK
1479 014566 000776  RR    33$
1480
1481 014570 012764 004146 000052 4S:  MOV     #MTOPI,RTYPE(R4);SET UP FOR "OPI" PRINT
1482 014576 005264 000030  INC     DLT CNT(R4) ;LOG ERROR
1483 014602 000555  BR     111$ ;CHECK RETRY EXIT BACK
1484
1485 ;WE KNOW IT'S NOW EITHER DLT, DCRC,HNF, OR HCRC
1486 ;CHECK FOR EACH
1487
1488 014604 032737 002000 002250 5S:  BIT     #OPI,E.CS ;OPI QUALIFIER SET?
1489 014612 001060  BNE     7$ ;YES, THEN IT'S HCRC OR HNF
1490
1491 ;IT'S NOW DOWN TO DLT OR DCRC
1492
1493 014614 032737 010000 002250  BIT     #DLT,E.CS ;DATA LATE?
    
```

```

1487 014622 001406 000052 6$  BEQ     6$ ;NO, MUST BE DATA CRC
1488 014624 012764 004141  MOV     #MTDLT,RTYPE(R4);SET UP FOR "DLT" PRINT
1489 014630 000557 000026  INC     DLT CNT(R4) ;LOG ERROR
1490 014636 000557 000026  BR     111$ ;CHECK RETRY, EXIT
1491
1492 014640 013737 002254 002172 6S:  MOV     E,DA,CHKSEC ;SET UP SECTOR TO LOOK FOR
1493 014646 005364 000064  DEC     LSTDA(R4) ;DOWN COUNT FOR PRINT OUT
1494 014652 005364 000064  DEC     CHKSEC ;DOWN COUNT FOR LOOP UP
1495 014656 004537 023744  JSR     R5,CKRDSC ;CHECK BAD SECTOR LIST
1496 014662 005737 002170  TST     #6,END ;IS HEADER THERE?
1497 014666 001176  BNE     110$ ;IGNORE ERROR, RETURN
1498 014670 005264 000022  INC     DCRCR(R4) ;ACCOUNT FOR "DCRC" PRINT
1499 014674 012764 004134  MOV     #MTDCRC,RTYPE(R4);SET UP FOR "DCRC" PRINT
1500 014702 022764 000102 000044  CMP     #MTDCRC,RTYPE(R4) ;SET UP FOR "DCRC" PRINT
1501 014710 000101  BNE     118$
1502 014712 000511  BR     111$
1503
1504 014714 005737 007544 118S: TST     T,DCK ;DUMP BUFFER?
1505 014722 001506  BEQ     111$ ;NO, EXIT
1506 014722 012746 003070  PRINTF #FMT14,#DMPDCK
1507 014726 002746 006507  MOV     #DMPDCK,-(SP)
1508 014736 010600 000002  MOV     #FMT14,-(SP)
1509 014740 104017  MOV     #2,-(SP)
1510 014742 062706 000006  MOV     SP,RO
1511 014746 004537 023104  EMT   CSINTF
1512 014752 000471  JSR     R5,DMPBUF ;DUMP BUFFER
1513 014752 000471  BR     111$ ;EXIT
1514
1515 ;IT'S NOW EITHER HNF OR HCRC.
1516 ;IF HCRC AND RDHDR, DETERMINE IF BAD SECTOR BY DOING 40 RDHDRS
1517 ;IF HNF READ HEAD, CHECK IF DA IS IN BAD SECTOR FILE
1518 ;IF HNF READ HEAD TO VERIFY IF ON CORRECT CYLINDER
1519 ;THEN IF ON CORRECT CYLINDER SEE IF DA IS A BAD SECTOR
1520 ;IF NOT ON CORRECT CYLINDER REPORT WISSEK, LOG WISSEK
1521 ;AND PRESENT POSITION UPDATE.
1522
1523 014754 032737 010000 002250 7S:  BIT     #HNF,E.CS ;HEADER NOT FOUND SET?
1524 014762 001486  BEQ     112$ ;NO IT MUST BE HCRC
1525 014764 012701  MOV     #41,R1 ;ALLOW FORTY READ HEADERS TO
1526 014770 004537 021136  JSR     R5,SDRST ;FIND CYLINDER
1527 014774 016402 000186  MOV     DRSEL(R4),R2 ;READ HEADER
1528 015000 052702 000010  RIS     #RDHDR,R2
1529 015004 016403 000186  MOV     DCS(R4),R3
1530 015010 010263 000000  MOV     R2,CS(R3)
1531 015014 004537 021152  JSR     R2,WTRDY ;ISSUE READ HEADER
1532 015020 005364 000000  DEC     R1 ;DONE 40 OF THESE?
1533 015022 001427  BEQ     R1 ;YES, GIVE UP WE DON'T HAVE ALL
1534 015024 005763 000000  TST     CS(R3) ;DAY, IS ERROR SET?
1535 015030 100757  BNE     R2 ;YES, GO DO IT AGAIN
1536
1537 015032 016301 000006  MOV     MP(R3),R1 ;GET HEADER
1538 015036 043701 002142  BIC     SECS,R1 ;MASK OUT SECTOR BITS
1539 015042 020164 000122  CMP     R1,PRPOS(R4) ;IS CYLINDER HEAD CORRECT?
1540 015046 001415  BEQ     10$ ;YES, GO CHECK BAD SECTOR LIST
    
```

```

1537
1538
1539 015050 005264 000072      INC      TRERR(R4)
1540 015051 200000 000000      ERRHRD  20, TRACK, ERR2 ;TRACKING DRIFT ERROR
1541 015052 150000 000000      TRAP    TSERCODE
1542 015053 104463 000024      .WORD  10
1543 015054 000024 000000      .WORD  10
1544 015055 003116 000000      .WORD  TRACK
1545 015056 004312 000000      .WORD  ERR2
1546
1547 015064 000137 016334      JMP      SKRETPV          ;FIX TRACKING ERROR
1548
1549 015070 000000 000000      9S:     ERRHRD  30, EXHAUS, ERR1 ;WE CAN'T FIND GOOD HEADER ON THIS TRACK
1550 015071 104463 000000      TRAP    TSERCODE
1551 015072 000036 000000      .WORD  30
1552 015073 002576 000000      .WORD  EXHAUS
1553 015074 004304 000000      .WORD  ERR1
1554
1555 015100 000410 000000      BR      110S
1556
1557 015102 013737 002254 002172 10S:     MOV      E, DA, CHKSEC
1558 015103 004537 023744      JSP     R5, CKRDSC      ;GO CHECK BAD SECTOR FILE
1559 015104 005737 002170      TST     HDRFND          ;WAS IT THERE?
1560 015105 001481 000000      BEQ     11S            ;NO LOG IT EXIT
1561 015106 000577 000000      BR      G0ERRX        ;YES IGNORE ERROR
1562
1563 015124 005264 000032 000052 11S:     INC      HNFERR(R4)     ;LOG IT
1564 015125 017764 004121 000052 111S:    MOV      HNFHNP, RTYPE(R4) ;SET UP FOR "HNF" PRINT
1565 015126 000573 000000      BR      GOFIN         ;EXIT
1566
1567 ;IT WAS A HEADER CRC ERROP, FIGURE OUT IF IT WAS
1568 ;ON A READ HEADER OR READ/WRITE
1569
1570 015140 022764 000110 000044 112S:    CMP      #INTENIPDHDR, FUNC(R4) ;READ HEADER?
1571 015141 001417 000000      BEQ     13S           ;YES, GO FIND OUT MORE ABOUT IT
1572
1573 015150 013737 002254 002172      MOV      E, DA, CHKSEC
1574 015151 004537 023744      JSP     R5, CKRDSC      ;BAD SECTOR SEARCH
1575 015152 005737 002170      TST     HDRFND          ;WAS OUR DA THERE?
1576 015153 001481 000000      BEQ     12S           ;NO, MUST BE LEGIT ERROR
1577 015154 000577 000000      BR      G0ERRX        ;YES, IGNORE ERROR
1578
1579 015172 005264 000024 000052 12S:     INC      HRCRCR(R4)     ;LOG ERROR
1580 015173 015204 000550 004126 121S:    MOV      HMTXCR, RTYPE(R4)
1581
1582 015206 017401 000110 002200 13S:     MOV      @RRA(R4), R1   ;USE IT'S BUFFER TO STORE HDRS
1583 015207 012701 000050 000000      MOV      #40, R1        ;40 CONSECUTIVE HEADERS
1584 015208 012701 000050 000000      MOV      @R1, R2        ;READ HEADER
1585 015209 056402 000106 000000      BIS     DRSEL(R4), R2
1586 015210 016403 000104 000000      MOV     DCS(R4), R3
1587 015211 010263 000000 000000      MOV     R2, CS(R3)
    
```

```

1585 015240 004537 021052      JSP     R5, WTRDY      ;WAIT FOR READY
1586 015241 016321 000000      MOV     CS(R3), (R1)+  ;READ ALL REGISTERS
1587 015242 016321 000000      MOV     MP(R3), (R1)+
1588 015243 016321 000000      MOV     MP(R3), (R1)+
1589 015244 016321 000000      MOV     MP(R3), (R1)+
1590 015245 005337 002200      DEC     TEMPI
1591 015246 001353 000000      BNE     14S           ;DONE 40 YET?
1592
1593 ;WE HAVE 40 HEADERS NOW LETS SEE IF WE CAN VERIFY WHETHER
1594 ;FOR NOT A BAD SECTOR CAUSED THE ERROR. CHECK FIRST TO SEE
1595 ;IF WE HAVE ANY BAD SECTORS ON THIS TRACK.
1596
1597 015272 017402 000110 000110 99S:     MOV     @RRA(R4), R2   ;GET BUFFER START
1598 015273 012701 000050 000050      MOV     #40, R1        ;FOURTY HEADERS
1599 015274 032712 002000 002000 15S:     BIT     #OPT, (R2)     ;IS OPT SET IN CS
1600 015275 001403 000000      BEQ     16S           ;NO, WELL CAN'T BE HCRC
1601 015276 032712 004000      BEQ     17S           ;INSURE HCRC W/OPT
1602 015277 001005 000010 000010 16S:     ADD     #10, R2        ;FOUND GO SEE IF IT COMPARES
1603 015278 062702 000000      DEC     R1            ;NEXT CS IMAGE
1604 015279 001366 000000      BNE     15S           ;DONE 40
1605 015280 000721 000000      BR      12S
1606
1607 015330 020274 000110 000110 17S:     CMP     R2, @RRA(R4)   ;IS HEADER FIRST ONE?
1608 015331 001646 000000      BNE     21S           ;NO, READ PREVIOUS HEADER
1609
1610 ;YES, WE'LL HAVE TO GO THRU
1611 ;AND CHECK OTHERS BEFORE WE
1612 ;CAN SAFELY CALCULATE
1613 ;"SUSPOED" BAD SECTOR
1614
1615 015336 017401 000110 000010 18S:     MOV     @RRA(R4), R1
1616 015337 012703 000001 000010      MOV     #1, R3
1617 015338 062701 000010 002000      ADD     #10, R1
1618 015339 001416 000000      BIT     #OPT, (R1)
1619 015340 032711 004000      BEQ     19S
1620 015341 001005 000000      BIT     #HCRC, (R1)
1621 015342 001403 000017 000017      BEQ     R3
1622 015343 022703 000017 000017      INC     R3
1623 015344 001364 000000 000017      CMP     #15, R3
1624 015345 001364 000000 000017      BNE     18S
1625
1626 015376 012737 003473 002126      MOV     #RDMSC, WHY    ;DROP DRIVE DUE TO
1627 015377 004537 020220 002126      JSP     R5, DRDRV     ;MORE THAN 16 BAD SECTORS
1628 015378 000137 016344 002126      JMP     EXIT1
1629
1630 015414 005017 000002 000002 19S:     CLR     (R2)          ;CLEAR THIS CS
1631 015415 062702 000002 000002      ADD     R2, R1        ;GET IT'S HEADER ADDRESS
1632 015416 010201 000002 000002      MOV     (R1), R2      ;GET HEADER
1633 015417 042762 017700 000002      MOV     #17700, R2    ;SAVE HEADER
1634 015418 000403 000002 000002      BIC     R3, R1        ;MASK ONLY SECTOR
1635 015419 000403 000002 000002      SUB     R3, R1        ;BACK UP TO SECTOR WHICH IS BAD
1636 015420 160301 000002 000002      RMI     20S          ;IF MINUS DO MAGIC
1637 015421 000421 000002 000002      SUB     R2, R2        ;NO THEN SUBTRACT IS LEGAL
1638 015422 160302 000002 000002      BR      20S          ;BRANCH TO CHECK FILE
1639 015423 160302 000002 000002      BR      20S          ;THIS SUB PRODUCES WRONG ANSWER
    
```

```

1641 015444 062702 000050          ADD    #50,R2          ;FIX IT UP
1642 015450 006415          BR     22S            ;GO CHECK FILE
1643
1644 015452 005012          21S: CLR    (R2)        ;CLEAR THIS CS OUT
1645 015454 012707          SUB    #6,R2          ;GET PREVIOUS HEADER
1646 015456 005201          MOV    (R2),R1
1647 015462 005201          INC    R1
1648 015464 010102          MOV    R1,R2
1649 015466 012707          BIC    #177700,R1
1650 015472 000050          CMP    #40,R1
1651 015476 002402          BGT    22S
1652 015500 000050          22S: SUB    #40,R2
1653 015504 002472          MOV    R2,CHKSEC
1654 015510 014237          JSR    R5,CRDSC
1655 015514 005737          TST    HD,CRDSC
1656 015520 001664          REQ    99S
1657 015522 000137          GOERRX: JMP   ERREX
1658
1659
1660 015526 000137 016452          GOFIN: JMP   FINERR
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674 015532 016401 000044          CHKFN: MOV   FUNC(R4),R1 ;GET FUNCTION OF DRIVE
1675 015536 006301          ASR    #40,R1          ;ALIGN THE FUNCTION CODE
1676 015540 042701 000040          BIC    #40,R1          ;WRITE OUT INT. ENAB (SHIFTED)
1677 015544 005301          DEC    R1              ;WRITE CHECK??
1678 015550 001004          BNE    2S             ;NO, BRANCH
1679 015554 000137 011422          JMP    R4,CLRWCCK
1680 015560 005301          2S:  DEC    R1           ;GET STATUS?
1681 015564 002555          REQ    A$STAT         ;BRANCH IF SO
1682 015568 001470          DEC    R1              ;SEEK?
1683 015566 001416          REQ    A$EEK         ;BRANCH IF SO
1684 015570 005301          DEC    R1              ;RDHDR?
1685 015574 002470          REQ    ARDHDR        ;BRANCH IF SO
1686 015578 001004          DEC    R1              ;WRITE?
1687 015576 001004          BNE    1S             ;NO, BRANCH
1688 015600 000137 016222          JMP    AWRITE
1689 015604 005301          1S:  DEC    R1           ;READ?
1690 015606 001425          REQ    AFREAD         ;BRANCH IF SO
1691
1692 015610          ERRSF 210.,PRGER
1693 015612          TRAP  T$ERCODE
1694 015614          .WORD 210
1695          .WORD PRGER
    
```

.SBTTL COMMAND SERVICE ROUTINES

;THERE WAS NO ERROR SO.....  
 ;NOW WE WILL FIND OUT WHICH FUNCTION WE DID TO CAUSE  
 ;INTERRUPT AND ACT ACCORDINGLY.

```

1694 015616 000000          XEXIT: HALT
1695 015620 000137 016312          JMP    EXIT
1696
1697
1698
1699
1700 015624 052764 000001 000056          .SBTTL          SEEK
1701 015632 005264 000054 001750          A$EEK: BIS   #SKDON,PRFLGS(R4) ;SET SEEK VERIFY NEEDED
1702 015636 026427 000054          INC    SKCNT1(R4) ;INCREMENT COUNT
1703 015640 005264 000000          CMP    SKCNT1(R4),#1000 ;10(3) REACHED
1704 015652 005264 000054          BLT    99S           ;NO, EXIT
1705 015656 000137 016350          INC    SKCNT1(R4) ;YES, BUMP THOUSANDS
1706
1707 99S: JMP    ERREX
1708
1709
1710
1711
1712
1713 015662          .SBTTL          READ
1714 015666 012700 000340          A$READ: SETPRI #340
1715 015670 104041          MOV    #340,R0
1716 015674 004537 020500          EMT    C$SPRI
1717          JSR    R5,CRDATA ;CHECK DATA
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728 015674 016401 000042          A$WRCK: MOV   RWP(R4),R1 ;BUMP UP XFER COUNT
1729 015678 005401          NEG    R1 ;MAKE POSITIVE
1730 015702 000164 000002          ADD    R1,DXFR1(R4) ;ADD THE BITS
1731 015706 022764 000002          CMP    #10000.,RXFR1(R4) ;10(5) REACHED YET
1732 015710 101016          BHI    2S             ;NO, EXIT
1733 015714 005264 000004          INC    RXFR2(R4) ;BUMP 10(10)
1734 015718 162764 000002          INC    #10000.,RXFR1(R4) ;START 10(R) AT 0
1735 015722 022764 000004          CMP    #10000.,RXFR2(R4) ;10(10) REACHED YET
1736 015726 101005          BHI    2S             ;NO, EXIT
1737 015730 005264 000060          INC    RXFR3(R4) ;YES BUMP 65K 10(10)
1738 015734 162764 000004          CMP    #10000.,RXFR2(R4) ;MAKE 10(10) 0
1739 015738 000557          SUB    BP ;EXIT
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
    
```

```

(5) 016032 004312          .WORD FRR2
1741 016032 004312          SKRETRY=.
1742 016034 000000          3S: INC RSEEK(R4) ;SET RETRY IN PROGRESS
1743 016034 005264 000114 007566 CMP RSEEK(R4),T.SLT ;RETRY EXHAUSTED????
1745 016040 026437 000114 BLOS 4S ;NO, THEN RETRY
1746 016046 101405          ERPHRD 333, SEXHAU,ERR2
1747 016050 000000          TRAP T.SERCODE
(3) 016050 104463          .WORD 333
(5) 016052 000515          .WORD SEXHAU
(5) 016054 003375          .WORD ERR2
(5) 016056 000406          BR 1S
1749 016062 010164 000050 4S: MOV R1, LSTHDR(R4) ;SET UP RETRY
1750 016066 042764 000031 000056 BIC #SKDON,PRFLGS(R4) ;ALLOW SEEK
1752 016074 000506          BR EXIT ;EXIT
1753 016076 042764 000001 000056 1S: BIC #SKDON,PRFLGS(R4) ;SET VERIFICATION DONE
1754 016104 005064 000114 CLR RSEEK(R4)
1756 016110 010164 000122 MOV R1, ERPOS(R4) ;MAKE THIS HEADER PRESENT POSITION
1757 016114 000476          BR EXIT ;EXIT
1758          .SBTTL GET STATUS
1759          AGSTAT: MOV E.MP,R1 ;GET STATUS
1760          BIC #100,R1 ;CLEAR OUT HEAD SELECT
1761          TST T.ROF ;READ ONLY
1762          BEO
1763          BIC #WL,R1
1764          BIC #177400,R1
1765          BIT 1S ;ANY BITS WRONG
1766          BEQ 2S ;NO, CONTINUE
1767          INC ERRCNT(R4) ;STATUS BITS WRONG
1768          ERPHRD 60, MDSER,ERR4
1769          TRAP T.SERCODE
(3) 016152 104463          .WORD 60
(5) 016154 000074          .WORD MDSER
(5) 016156 002721          .WORD ERR4
(5) 016160 004514          .WORD
1772 016162 010102 177700 1S: MOV R1,R2 ;COPY STATUS WORD
1773 016164 042702 177700 BIC #177700,R2 ;GET STATE BITS
1774 016170 022702 000034 CMP #34,R2 ;COVER CLSD, HEADS OUT, BRUSHES HOME, SEEK TRACK COUNTIN
1775 016172 002702 000035 BEQ EXIT ;YES, EXIT
1776 016174 002702 000035 CMP #35,R2 ;COVER CLSD, HEADS OUT, BRUSHES HOME, SEEK LINEAR MODE
1777 016202 001443          BEQ EXIT ;YES, EXIT
1778          INC ERRCNT(R4)
1779          ERPHRD 70, MDSER,ERR4
1780          TRAP T.SERCODE
(3) 016210 104463          .WORD 70
(5) 016212 000106          .WORD MDSER
(5) 016214 001443          .WORD ERR4
(5) 016216 004514          .WORD
1781 016220 000434          BR EXIT
1782
1783
    
```

```

1784          .SBTTL WRITE
1785          AWRITE: MOV RMP(R4),R1 ;GET WORD COUNT
1786          NEG R1 ;MAKE POSITIVE
1787          ADD R1,WXFR1(R4) ;ADD THE BITS
1788          CMP #10000, WXFR1(R4) ;10(5) YET?
1789          BHI 1S ;NO, EXIT
1790          INC WXFR2(R4) ;YES BUMP 10(10)
1791          SUB #10000, WXFR1(R4) ;10(5) GUES TO ZERO
1792          CMP #10000, WXFR2(R4) ;10(10) YET?
1793          BHI 1S ;NO, EXIT
1794          INC WXFR3(R4) ;INC 65K (10)(10)
1795          INW #1000, WXFR2(R4) ;MAKE 10(10)
1796          TST T.WCK ;PERFORM WRITE CHECK
1797          BEQ EXIT
1798          JSP R5,SETWCK
1799          JSP R5,SETWCK
1800          TST RETPY(R4) ;IN PROCESS OF RETRYING?
1801          REO ERREX ;NO
1802          CMP RTYPE(R4),#VTDV
1803          BEQ EXIT1
1804          INC SFTCNT(R4) ;YES, LOG SOFT ERROR
1805          ERRSOFT 80, MDSER,ERR3 ;REPORT SOFT ERROR
1806          TRAP T.SERCODE
(3) 016334 104464          .WORD 80
(5) 016336 000120          .WORD MDSER
(5) 016340 002645          .WORD ERR3
(5) 016342 004376          .WORD
1809 016344 005064 000036 EXIT1: CLR RETPY(R4) ;CLEAR RETRY
1810          BIC #INTEN,ADCS(R4)
1811          MOV (SP)+,R1
1812          MOV (SP)+,R2
1813          MOV (SP)+,R3
1814          MOV (SP)+,R4
1815          MOV (SP)+,R5
1816          MOV (SP)+,R6
1817          MOV (SP)+,R7
1818          MOV (SP)+,R8
1819          MOV (SP)+,R9
1820          MOV (SP)+,R10
1821          MOV (SP)+,R11
1822          MOV (SP)+,R12
1823          MOV (SP)+,R13
1824          MOV (SP)+,R14
1825          MOV (SP)+,R15
1826          MOV (SP)+,R16
1827          MOV (SP)+,R17
1828          MOV (SP)+,R18
1829          MOV (SP)+,R19
1830          MOV (SP)+,R20
1831          MOV (SP)+,R21
1832          MOV (SP)+,R22
1833          MOV (SP)+,R23
    
```

```

1834 016466 000137 016350
1835 016472 005264 000012
1839 016476
(3) 016476 104463
(5) 016500 000132
(5) 016504 003055
(5) 016504 004304
1838 016506 004537 011422
1839
1840 016512 000714
1841
1842
1843
1844
1845
1846
1847 016514 005264 000020
1848 016526 004537 021122
1849
1850 016524 104463
(3) 016524 000340
(5) 016526 002664
(5) 016530 002664
1850 016532 005950
1851
1852
1853
1854 016534 032701 001000
1855 016540 001027
1856 016542 032701 010000
1857 016546 091070
1858 016550 144000
1859 016554 001130
1859 016556 032701 002000
1860 016562 001097
1861 016564 004537 021136
1862 016570 000431
1863 016572 004537 021136
1864 016574 001177 021000
1865 016600 001177 021000
1866 016606 001422
1867 016610 012737 002762 002126
1868 016616 000412
1869
1870 016620 004537 021136
1871 016624 004537 021122
1872 016630 001000 001000
1873 016634 001404
1874 016636 012737 002735 002126
1875
    
```

1S: JMP ERREX ;NOT EXCEEDED EXIT  
 INC ERRCNT(R4) ;INDICATE ERROR

ERRHRD 90, MHDR,ERR1 ;NON-RECOVERABLE ERROR  
 TRAP 30,ERRCODE  
 .WORD MHDR  
 .WORD ERR1  
 JSR R5,CLPWCK  
 BR EXIT1

.SBTTL DRIVE ERROR SERVICE  
 ;WE HAVE A DRIVE ERROR, LET'S GET THE STATUS  
 CKDERR: INC DEPCNT(R4) ;ACCOUNT FOR ERROR  
 JSR R5,GETDST ;GET DRIVE STATUS  
 ;REPORT DRIVE ERROR  
 ERRHRD 224, DRIVER,ERR9 ;DRIVE ERROR  
 TRAP 12,ERRCODE  
 .WORD 224  
 .WORD DRIVER  
 .WORD ERR9

;ACT ACCORDINGLY TO DRIVE ERROR

BIT #VC,R1 ;VOLUME CHECK?  
 BNE Q6 ;YES, GO ISSUE RESET  
 BIT #SKTO,R1 ;SEEK TIME OUT?  
 BNE 125 ;YES, ISSUE RESET  
 BIT #WDEIHCEI\$PE,R1 ;WRITE DATA, CURRENT HEAD, SPINDLE?  
 BNE 155 ;GO WAIT FOR HEADS TO UNLOAD  
 BIT #WGE,R1 ;WRITE GATE ERROR  
 BNE 205 ;WRITE ISSUE RESET  
 JSR R5,ISDRST ;ISSUE RESET  
 BR 105 ;GO CHECK DRIVE READY  
 JSR R5,ISDRST ;ISSUE RESET  
 JSR R5,GETDST ;RESET WORK?  
 BIT #WGE,R1 ;WGE CLEAR  
 BNE 105 ;YES GO CHECK DRIVE READY  
 MOV #0,GETST,WHY ;REPORT WGE DIDN'T CLR  
 BR 915 ;DROP DRIVE

9S: JSR R5,ISDRST ;ISSUE RESET  
 JSR R5,GETDST ;GET WORK  
 BIT #VC,R1 ;VOLUME CHECK?  
 BNE 105 ;YES, CHECK DRIVE READY  
 MOV #0,WHY ;DROP THE DRIVE

```

1877 016644 004537 020220
1878 016650 000137 012444
1879 016654 032763 000001 000000
1880 016662 001004
1881
1882 016664 012737 002474 002126
1883 016672 000764
1884
1885 016674
(8) 016674 012746 003015
(7) 016700 012746 006507
(6) 016704 012746 000002
(3) 016710 000600
(4) 016712 001194
(4) 016714 062706 000006
1886 016720 004537 017156
1887 016724 000137 000004
1888 016730 001177 021136
1889 016734 004537 021136
1890
1891 016740
(3) 016740 012700 035230
(3) 016744 104027
1892
1893 016746 032763 000001 000000
1894 016752 001002
1895 016756 005302
1896 016760 001365
1897
1898 016762 012737 002673 002126 141S:
1899 016770 000725
1900
1901 016772 032763
1902 017000 001370 040000 000000 14S:
1903 017002
(8) 017002 012746 003015
(7) 017006 012746 006507
(6) 017010 012746 000002
(3) 017016 000600
(4) 017020 104014 000006
1904 017024 004537 017156
1905 017026 000137 000004
1906 017032 000137 016312
1907
1908 017036 012702 000004
1909 017040 004537 021122
1910 017046 032701 000020
1911 017052 001411
1912
1913 017054 012700 000001
(3) 017060 104026
1914 017062 005302
1915 017064 001365
1916 017066 001365 003352 002126
1917 017074 000663
    
```

91S: JSR R5,DRDRV  
 JMP EDP11

10S: BIT EDP11,CS(R3) ;DRIVE READY POSTED?  
 RNE 101S ;YES, PRINT RECOVERED

MOV #0,DRDY,WHY ;NO, DROP DRIVE  
 BR 91S

101S: PRINTR #FMT14,#MRDR ;PRINT DRIVE RECOVERED  
 MOV #MRDR,-(SP)  
 MOV #FMT14,-(SP)  
 MOV #2,-(SP)  
 SP,R0  
 C\$PNTP  
 ADD #6,SP  
 JSR R5,CHDR  
 JMP FINERR

12S: MOV #4,R2 ;SEEK TIME OUT  
 13S: JSR R5,ISDRST ;ISSUE DRIVE RESET  
 WAITUS #1500, ;FOUR TIMES BEFORE  
 MOV #1500, R0 ;DROPPING DRIVE  
 EMT C\$WTU

BIT #DRDY,CS(R3) ;DRIVE READY YET?  
 BNE 145 ;YES, CHECK IF ERROR CLEARED  
 DEC R2 ;NO, HAVE WE DONE IT FOUR TIMES  
 RNE 13S ;YET

141S: MOV #MDERS,WHY ;YES, DROP DRIVE  
 BR 91S

14S: BIT #DERR,CS(R3) ;DRIVE ERROR SET STILL  
 BNE 141S ;YES, DROP DRIVE  
 PRINTR #FMT14,#MRDR  
 MOV #MRDR,-(SP)  
 MOV #FMT14,-(SP)  
 MOV #2,-(SP)  
 SP,R0  
 C\$PNTP  
 ADD #6,SP  
 JSR R5,CHDR  
 JMP EXIT

15S: MOV #4,R2 ;WAIT FOR HEADS TO UNLOAD  
 16S: JSR R5,GETDST ;GET STATUS  
 BIT #P14,R1 ;UNLOAD STATE  
 BFO 175 ;YES, CONTINUE W/ RECOVERY  
 WAITMS #1 ;WAIT A WHILE  
 MOV #R0  
 C\$WTM  
 DEC R2  
 BNE 16S ;WAIT LONG ENOUGH  
 MOV #UNLOAD,WHY ;NO, GO BACK  
 BR 91S ;DROP DRIVE



```

1917 017076 004537 021136 17S: JSR R5,ISRST ;ISSUE RESET
1918 017102 000001 000001 MOV #1,R0 ;
(3) 017106 104026 EMT CSWTH ;
1920 017118 002163 040000 000000 BIT UNDER,CS(R3) ;DRIVE ERROR CLEAR?
1921 017120 012702 000075 MOV #1,R2 ;YES, WAIT 60 SECONDS
1922 017124 012700 000012 18S: WAITMS #1,R0 ;FOR DRIVE READY TO
(3) 017130 104026 EMT CSWTH ;
1923 017132 032763 000001 000000 BIT #DRDY,CS(R3) ;COME BACK
1924 017140 001314 RNE 14S ;
1925 017144 001367 DEC 14S ;
1927 017146 012737 003376 002126 MOV #NOLOAD,WHY ;NO READY DROP DRIVE
1928 017154 000633 BR 91S ;
1930
1931 017156 012763 000210 000000 CHDR: MOV #CRDYRDHDR,CS(R3)
1932 017164 056463 000106 000000 BIS DRSEL(R4),CS(R3)
1933 017172 042763 000200 000000 BIC #20,CS(R3)
1934 017200 004537 000000 JSR ;
1935 017204 016301 000005 MOV #P(R3),R1
1936 017410 043701 002142 BIC SECMSK,R1
1937 017414 012764 001222 MOV #R1,PRPOS(R4)
1938 017220 012764 004160 000052 MOV #R1,DRV,RTYPE(R4) ;SETUP DRIVE ERROR
1939 RTS
1940
;ROUTINE TO WRITE A BUFFER INTO MEMORY. USES WORD COUNT AND BUS
;ADDRESS FROM DRIVE BUFFER (R4) WILL WRITE RANDOM FROM ONE OF
;8 PATTERNS. USED BY WRITE FUNCTION AND WRPACK ROUTINE.
1942
1943
1944
1945 017230 010346 WRBUF: MOV R3,-(SP) ;SAVE REGISTERS
1946 017232 010246 MOV R2,-(SP)
1947 017234 010146 MOV R1,-(SP)
1948 017236 010046 MOV R0,-(SP)
1949 017248 005405 000042 MOV #RMP(R4),R2 ;R2 HAS TOTAL WORDS TO SET UP FOR
1950 NEG ;POSITIVE NUMBER
1951 017246 017401 000110 MOV @RBA(R4),R1 ;WHERE BUFFER IS
1952 017252 020227 000200 2S: CMP R2,#128. ;MORE THAN 128 WORDS
1953 RGE 4S ;YES, BRANCH
1954 017260 020237 000003 CMP #3 ;GREATER THAN THREE WORDS
1955 BGE 4S ;YES, BRANCH
1956 017266 062707 000003 ADD #3,R2 ;ADD 3
1957 017270 062762 000003 3S: SUB #3,RMP(R4) ;WUP BY 3
1958 MOV #R1,R1+ ;STORE WC
1959 017302 005307 002212 DEC R2 ;ACCOUNT FOR WC
1960 017304 010237 MOV R2,TEMP6 ;LOAD DOWN COUNTER
1961 017312 010237 BR 5S ;
1962 017314 010237 000177 002212 4S: MOV #27,TEMP6 ;LOAD DOWN COUNTER
1963 017320 010237 000200 MOV #128,(R1)+ ;
1964 017324 005737 007562 5S: TST T-RAN ;RANDOM SELECT OF PATTERNS
1965 017330 010237 007564 BNE 5S ;YES
1966 017332 010237 007564 MOV #PAT,R3 ;GET PATTERN OPERATOR
1967 017336 000406 BR 5S ;WANTS TO USE
1968 017340 004537 021214 JSR R5,RAND ;GET RANDOM # FOR PATTERN

```

```

1969 017344 013703 002136 MOV LONHM,R3 ;GET RANDOM PATTERN
1971 017354 006303 177770 RIC #177770,R3 ;0.7
1972 017356 062703 024430 56S: ASL R3 ;WORD OFFSET
1973 017362 011303 ADD #PATLST,R3 ;GET PATTERN LIST
1974 017370 010337 002214 MOV #R3,R3 ;GET LIST ADDRESS
1975 017378 010337 002214 MOV #R3,TEMP7 ;STORE FOR RECALL
1976 017372 005337 002212 MOV #R3,(R1)+ ;LOAD IT
1977 017376 013703 002214 6S: DEC TEMP7 ;ACCOUNT FOR IT
1978 017410 012737 000020 002216 7S: MOV #16,TEMP8 ;16 ENTRIES
1979 017412 005337 002212 MOV #R3+(R1)+ ;STORE PATTEPN
1980 017416 001464 DEC TEMP6 ;DOWN COUNT
1981 017420 005337 002216 REQ #5 ;DONE?
1982 017424 001371 DEC TEMP8 ;DONE WITH PATTERN
1983 017426 000763 BR 7S ;NO, GO BACK
1984 017430 062762 000200 8S: SUB #128,R2 ;RESTART PATTERN
1985 017434 062762 BGT 2S ;ANOTHER SECTOR TO USE
1986 017436 012600 2S ;YES GO BACK
1987 017440 012601 MOV (SP)+,R0 ;RESTORE REGISTERS
1988 017442 012602 MOV (SP)+,R1
1989 017444 012603 MOV (SP)+,R2
1990 017446 000205 MOV (SP)+,R3
1991 RTS
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009 017470 000000 LIST: .WORD 0 ;WRITE DATA
2010 017472 013346 WRTFNC ;WRITE DATA
2011 017474 013346 GSTFNC ;GET STATUS
2012 017476 013366 SKFNC ;SEEK FUNCTION
2013 017500 013366 SKFNC ;SEEK FUNCTION
2014 017504 014002 RDDFNC ;READ DATA
2015 017504 014002 RDDFNC ;READ DATA
2016
2017
2018
2019
2020
2021
2022
2023
2024

```

```

2025 ;FIELD BAD SECTORS. R4 WILL CONTAIN THE BUFFER POINTER
2026 ;TO THE DRIVE WE WANT TO READ
2027 ;
2028 ;CALL JSP R5,DRDRSC
2029
2030
2031 017506 010046 RDBDSC: MOV R0,-(SP) ;SAVE REGISTERS
2032 017510 010046 MOV R1,-(SP) ;
2033 017514 010046 MOV R2,-(SP) ;
2034 017518 010046 MOV R3,-(SP) ;
2035 017516 004537 021136 21S: JSR R5,ISDRST
2036 017522 012764 000010 MOV #RDRDR,FUNC(R4);READ HEADER TO FIND POSITION
2037 017530 004537 014112 JSR R5,LDFUNC ;ON DISK
2038 017534 004537 021052 JSR R5,WTRDY
2039
2040 017540 016300 000006 MOV MP(R3),R0 ;GET HEADER AND CALCULATE
2041 017544 043700 002140 BIC CVMASK,R0 ;DIFFERENCE TO GET TO
2042 017550 012701 077600 MOV #77600,R1 ;BAD SECTOR FILE, AND GO
2043 017554 160001 SUR R0,R1 ;THERE
2044 017558 010164 000340 MOV #R5,RDA(R4)
2045 017562 052764 000025 MOV #R5,ISGMWK,BDA(R4)
2046 017570 012764 000006 000040 MOV #SEEK,FUNC(R4)
2047 017576 004537 014112 JSR R5,LDFUNC
2048 017582 016300 000010 JSR R5,WTRDY
2049 017606 012764 000010 000044 MOV #R5,RDA(R4)
2050 017614 004537 014112 JSR R5,LDFUNC
2051 017620 004537 021052 JSR R5,WTRDY
2052 017624 016300 000006 MOV MP(R3),R0
2053 017630 043700 000006 BIC #77600,R0
2054 017634 022700 077700 CMP #77700,R0
2055 017640 001326 RNE 21S
2056
2057 017642 012764 077700 000040 MOV #77700,BDA(R4)
2058 017650 012764 177400 000042 MOV #256,RMP(R4) ;SETUP AND READ IN THE
2059 017656 012764 000014 000044 MOV #PEAD,FUNC(R4) ;BAD SECTOR FILE ON SECTOR
2060 ;0
2061 017664 005037 002204 CLR TEMP3 ;MANUFACTURING/FIELD FILE SWITCH
2062 017670 012737 003524 002126 MOV #HWSEC,WHY ;START WITH MANUFACTURING BAD
2063 017676 016402 009112 MOV R5CEPT(R4),R2 ;INITIALIZE LIST TO ALL 1'S
2064 017682 004537 000006 MOV #16,R0 ;SIXTEEN ENTRIES
2065 017706 012722 177777 11S: MOV #1,(R2)+
2066 017712 005300 DEC R0
2067 017714 001374 BNE 11S
2068
2069 017716 016402 000112 MOV R5CEPT(R4),R2 ;GET LIST TO STORE
2070 017722 012700 000020 MOV #16,R0 ;SIXTEEN ENTRIES
2071 017726 004537 014112 4S: JSR R5,LDFUNC
2072 017732 004537 021052 JSR R5,WTRDY
2073
2074 017736 005774 000104 TST #DCS(R4) ;WAS THE PEAD GOOD?
2075 017742 100025 RPL 3S ;YES
2076
2077 017744 004537 021136 JSR R5,ISDRST
2078 017750 062764 000004 000040 ADD #RDA(R4) ;NO, NEXT SECTOR
2079 017756 005737 002204 TST TEMP3 ;MANUFACTURING OR FIELD BAD
2080 017762 001410 BEQ 5S ;MANUFACTURING
    
```

```

2081 017764 012737 003544 MOV #HWSEC,WHY ;FIELD BAD
2082 017772 022764 077750 000040 CMP #7750,BDA(R4) ;AT END OF FIELD BAD?
2083 020002 001352 4S BNE 4S ;NO, GO BACK FOR NEXT
2084 020004 026427 000040 077724 5S: MOV #RDA(R4),#77724
2085 020004 026427 000040 077724 5S: MOV #RDA(R4),#77724
2086 020012 001345 4S BNE 4S ;AT END OF MANUFACTURING BAD
2087 020014 000463 4S BNE 4S ;AT END OF BAD FACTORY SECTION
2088 ;YES, REPORT ERROR
2089
2090 020016 017401 000110 3S: MOV #RRA(R4),R1 ;START OF LIST
2091 020022 012164 000100 MOV (R1)+,SERW1(R4) ;GET LOW PART OF SERIAL #
2092 020026 002164 000102 MOV (R1)+,SERW2(R4) ;GET HIGH PART OF SERIAL #
2093 020034 014134 002200 1S: CMP (R1)+,(R1)+ ;SKIP PAST JUNK
2094 020040 106437 002200 MOV (R1)+,TEMP1 ;GET CYLINDER
2095 020042 012137 002222 MOV (R1)+,TEMP2 ;IF MINUS END OF BAD SECTORS
2096 020046 006337 002200 SWAB TEMP1 ;GET TRACK AND CYLINDER
2097 020052 006037 002200 ROR TEMP1 ;PUT CYLINDER IN HIGH BYTE
2098 020056 013712 002200 MOV TEMP1 ;ALIGN IT
2099 020062 013737 002202 MOV TEMP2,TEMP1 ;STORE OFF CYLINDER PART
2100 020064 043737 177700 002200 BIC #177700,TEMP1 ;SET SECTOR
2101 020076 053712 002200 BIC TEMP1,(R2) ;LEAVE ONLY SECTOR
2102 020102 042737 177377 002202 BIC #177377,TEMP2 ;SET IN SECTOR BITS
2103 020110 006237 002202 ASR TEMP2
2104 020114 006237 002202 ASR TEMP2
2105 020120 053737 002202 ASR TEMP2
2106 020124 005300 BIC TEMP2,(R2)+ ;SET IN HEAD
2107 020126 001342 DEC R0
2108 020128 012737 000104 1S BNE 1S
2109 020136 000472 003473 002126 MOV #MRDMSC,WHY ;MORE THAN 16 BAD SECTORS
2110 ;
2111
2112 020140 005737 002204 2S: TST TEMP3
2113 020144 014011 7S BNE 7S ;SWITCH TO FIELD BAD OR QUIT
2114 020146 012764 077724 000040 MOV #77724,RDA(R4) ;QUIT 7S
2115 020154 012737 000061 002204 9S: MOV #1,TEMP3 ;SWITCH TO FIELD BAD
2116 020162 006661 4S BR 4S ;SET TO QUIT NEXT TIME THRU
2117
2118 020164 004537 020220 6S: JSR R5,DRDRV ;DROP THE DRIVE
2119 020170 004537 022500 9S: JSR R5,DRDRMF ;BRINGS HEADS HOME
2120 020174 012653 MOV (SP)+,R3
2121 020200 012601 MOV (SP)+,R2
2122 020202 012600 MOV (SP)+,R1
2123 020204 006205 RTS ;
2124
2125 020206 004537 020220 8S: JSR R5,DRDRV
2126 020212 000770 BR 9S
2127
2128 ;
2129 ;
2130 ;
2131 ;
2132 ;
2133 ;
2134 ;
2135 ;
2136 ;
    
```

SRITL ROUTINE TO DROP DRIVE  
 ;ROUTINE TO DROP A DRIVE FROM RUNNING

```

2137 ;R4 HAS BUFFER POINTER OF DRIVE TO DROP
2138 ;WE CLEAR R1T IN "DRDRT", NOT "DRPRS"
2139 ;
2140
2141 020214 005237 002324 ODRDRV: INC OPCODE
2142 020220 010346 DRDRV: MOV R1, -(SP)
2143 020222 010346 MOV R2, -(SP) ;SAVE REGISTERS
2144 020224 010346 MOV R3, -(SP)
2145 020226 005237 INC INCALL
2146 020228 005237 CLR R3
2147 020230 010346 MOV #DRRUP, R2 ;START OF DRIVE BUFFERS
2148 020240 012701 MOV #1, R1 ;MASK
2149 020244 020400 1S: CMP R4, R2 ;IS THIS THE DRIVE?
2150 020246 005237 REQ R3 ;YES GO DROP IT
2151 020250 005237 INC R3
2152 020252 006301 ASL R1 ;NO SHIFT MASK
2153 020254 062702 ADD #RPROS+2, R2 ;NEXT BUFFER
2154 020260 000771 BR 1S ;GO BACK
2155
2156 020262 005737 2S: TST OPCODE
2157 020266 001002 RNE CS
2158 020270 010300 DDDU R3
2159 020272 104053 MOV #PC, PC
2160 020274 005037 EMT CSDDDU
2161 020304 005037 CLR INCALL
2162 020312 113764 000070 MOVB HOUR, DPHOUR(R4) ;TIME AT WHICH IT WAS DROPPED
2163 020320 001002 MINUTE, DPMIN(R4) ;HOUR/MINUTE
2164 020322 142137 000071 RNE 3S ;IF MINUTE 0,
2165 020326 142137 INCB DPMIN(R4) ;MAKE 1
2166 020330 000071 PRINTF #FMT7, TIME, HOUR, MINUTE, SECOND, #MRLCS, DCS(R4), #DRM
2167 020332 012746 MOV #DRNM, -(SP)
2168 020334 001104 MOV UCSC(R4), -(SP)
2169 020336 012746 MOV #MRC, -(SP)
2170 020340 002242 MOV SECOND, -(SP)
2171 020342 013746 MOV MINUTE, -(SP)
2172 020344 002244 MOV HOUR, -(SP)
2173 020346 002330 MOV #FMT7, -(SP)
2174 020348 006106 MOV #10, -(SP)
2175 020350 000010 MOV SP, PC
2176 020352 104010 EMT CSBNTF
2177 020400 104010 ADD #22, SP
2178 020402 062706 000022 PRINTF #FMT7A, <R, DRSEL+1(R4)>, #DROP, WHY
2179 020404 012746 MOV #R1, -(SP)
2180 020406 005046 MOV #R2, -(SP)
2181 020410 005046 CLR -(SP)
2182 020412 154416 BTR DRSEL+1(R4), (SP)
2183 020414 012746 MOV #4, -(SP)
2184 020416 005046 MOV SP, R0
2185 020418 010600 EMT CSBNTF
2186 020420 104017 ADD #12, SP
2187 020422 062706 000012 PRINTF #FMT8, -(SP)
2188 020424 012746 MOV #FMT8I, -(SP)
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
    
```

```

2201 020450 012746 (6) MOV #1, -(SP)
2202 020452 016600 (3) MOV SP, R0
2203 020454 104017 (4) EMT CSBNTF
2204 020460 062706 (4) ADD #4, SP
2205
2206 020464 004737 011670 JSP PC, REPORT
2207
2208 020470 012603 MOV (SP)+, R3
2209 020472 012603 MOV (SP)+, R2 ;RESTORE REGISTERS
2210 020474 012601 MOV (SP)+, R1
2211
2212 020476 000205 RTS R5
2213
2214 ;SBTTL ROUTINE TO CHECK DATA
2215 ;ROUTINE TO CHECK DATA ON READ
2216
2217 020500 005737 007520 CKDATA: TST CWRD
2218 020504 001001 RNE R5 ;DO WE WANT TO CHECK ANY?
2219 020506 000205 RTS R5 ;YES CONTINUE
2220 020510 012700 97S: MOV #340, R0 ;NO, EXIT
2221 020514 104041 40V CSDPT
2222 020516 017400 MOV #RBA(R4), R2 ;BUFFER START
2223 020518 000110 MOV RMP(R4), TEMP1 ;WORDS READ IN
2224 020520 000042 NEG TEMP1 ;MAKE POSITIVE
2225 020522 013737 002200 MOV DELMT, TEMP2 ;# ERRORS TO BE PRINTED
2226 020524 007522 002202 CLR DECNM ;INIT ERROR COUNT
2227 020526 007522 002204 MOV CWRD, TEMP3 ;# WORDS TO BE COMPARED
2228 020528 012737 002176 96S: MOV #26, TEMP4 ;126 WORDS
2229 020530 005337 MOV #1, R1 ;NON-ZERO WORDS
2230 020532 005337 DEC TEMP1
2231 020534 005337 BFC CFND
2232 020536 005337 DEC R1
2233 020538 012237 MOV (R2)+, TEMP4 ;PATTERN ADDRESS
2234
2235 ;MAKE SURE PATTERN ADDRESS IS LEGAL
2236
2237 020600 012700 024430 MOV #PATLIST, R0 ;GET LIST OF PATTERNS
2238 020604 012703 000010 MOV #R, R3 ;ONLY EIGHT
2239 020608 022037 002206 98S: CMP (R0)+, TEMP4 ;FOUND IT YET
2240 020614 006112 RQ CS ;YES, CONTINUE
2241 020616 005303 DEC R3 ;YES, EXHAUST LIST YET
2242 020620 001373 RNE 99S ;NO, GO BACK
2243
2244 020622 024242 2208: CMP -(R2), -(R2)
2245 020624 1000REV, ERR13 ERRHPD ;ERRHPD
2246 020626 000264 TRAP TRAP ;TRAP
2247 020628 003434 .WORD 100 ;WORD
2248 020630 005172 .WORD 100 ;WORD
2249 020632 005172 .WORD 100 ;WORD
2250 020634 004537 JSE R5, STDMP
2251 020640 000205 RTS R5
2252
2253 020642 005301 99S: DEC R1 ;ACCOUNT FOR PATTERN ADDRESS
2254 020644 013703 MOV TEMP4, R3 ;GET ADDRESS
    
```

2215	020650	005337	002200		DEC	TEMP1		;ACCOUNT ONCE AGAIN
2216	020654	012737	006020		MOV	#16,TEMP5		;16 ENTRIES TO PATTERN
2217	020662	005737	002200	002210	1S:	TST	TEMP1	;ANY WORDS READ IN LEFT?
2218	020666	001457			REQ	CEND		;NO, GO TO END
2219	020670	005737			TST	TEMP3		;HAVE WE EXHAUSTED COMPARE LIMIT?
2220	020674	001464	002204		REQ	CEND		;YES GO TO END
2221	020676	005701			TST	R1		;WE CHECKING PATTERN OR ZERO FILL?
2222	020700	001416			REQ	3C		;ZERO FILL SKIP
2223	020702	005301			DEC	R1		;PATTERN
2224	020704	005737	002210		TST	TEMP5		;WITHIN PATTERN
2225	020710	001005			BNE	2S		;YES SKIP
2226	020712	013793	002206		MOV	TEMP4,R3		;NO, START OVER
2227	020716	012737	006020	002210	MOV	#16,TEMP5		;16 ENTRIES
2228	020724	012737	002212		MOV	(R3),GDDAT		;SET PATTERN
2229	020730	005337	002210		DEC	TEMP5		;DOWN COUNT
2230	020734	000402			RP	4S		
2231	020738	005037	002232		CLB	GDDAT		;ZERO FILL
2232	020742	003772	002232		CMF	GDDAT,(R2)		;CORRECT DATA
2233	020746	001415			REQ	5S		;YES YES NEXT
2234	020750	005237	002174		INC	DECNT		;DATA ERROR
2235	020754	005264	000774		INC	DATCER(R4)		
2236	020760	005737	002202		TST	TEMP2		;DO WE WANT TO PRINT IT
2237	020764	001406			REQ	5S		;NO,SKIP
2238								
(3)	020766	104463			ERRHRD	1R0,MDCER,EPR8		
(5)	020770	000264			TRAP	TERRCODE		
(2)	020772	003040			.WORD	18C		
(2)	020774	004664			.WORD	MDCER		
2240	020776	005337	002202		.WORD	ERRB		
2241					DEC	TEMP2		;ACCOUNT FOR PRINT
2242	021002	005337	002200		DEC	TEMP1		;WORDS READ IN
2243	021006	001407			REQ	CEND		
2244	021010	005722			TST	(R2)+		;NEXT WORD
2245	021012	005337	002176		DEC	TEMP0		
2246	021016	001656			REQ	9FS		

2248	021020	005337	002204		DEC	TEMP3		;WORDS TO CHECK
2249	021024	000716			BR	1S		
2250								
2251	021026	005737	002174		CEND:	TST	DECNT	;DO WE WANT TO PRINT SUMMARY
2252	021032	001406			REQ	1S		;NO,EXIT
2253								

```

2255 021034 005464 000042      NEG     RMP(R4)      ;MAKE POSITIVE WORD COUNT
2256 021040          ERRHRD 190,MD CER,ERR6 ;DATA ERROR SUMMARY
      (3) 021040          TRAP  TSERCODE
      (5) 021042          .WORD 190
      (5) 021044          .WORD MD CER
      (5) 021046          .WORD ERR6
2257 021050 000205          1$:   RTS     R5
2258          .SBTTL  ROUTINE TO WAIT FOR CONTROLLER READY
2259          ;
2260          ;ROUTINE TO WAIT FOR CONTROLLER READY UNDER FLAG
2261          ;MODE USED IN INITIALIZE PORTION OF PROGRAM I.E.
2262          ;GETTING RAD SECTOR FILE, WRITING PACK INITIALLY
2263
2264 021052 010046          WTRD V: MOV    R0,-(SP)      ;SAVE REGISTERS
2265 021054 010146          MOV    R1,-(SP)
2266 021056 012701          MOV    #1000,R1      ;WAIT A WHILE
2267 021062          1$:   WAITUS #2
2268 021062 012700 000002          MOV    #2,R0
2269 021064 014027          EMT    CSRTF
2270 021070 032774 000200 000104          RTI    #CRDY,MD CS(R4) ;READY SET?
2271 021076 001706          BNE    #5          ;YES, EXIT
2272 021100 005301          DEC    R1          ;TIMED OUT?
2273 021102 001367          BNE    1$          ;NO GO BACK
2274
2275 021104          ERRDF 1062,MD CRDY,ERR12
2276 021104          TRAP  TSERCODE
2277 021106          .WORD 1002
2278 021110          .WORD NDCRDV
2279 021112 005164          .WORD ERR12
2280
2281 021114 012601          2$:   MOV    (SP)+,R1      ;RESTORE REGISTERS
2282 021116 012600          MOV    (SP)+,R0
2283 021120 000205          RTS     R5
2284          .SBTTL  GET STATUS/DRIVE RESET ROUTINE
2285          ;ROUTINE TO ISSUE DRIVE RESET
2286          ;ALSO GET STATUS, R1 HAS STATUS IF GS
2287          ;USES R3, DOES NOT SAVE IT
2288
2289 021122 016403 000104          GETDST: MOV   DCS(R4),R3
2290 021124 012763 000003 000004          MOV   #GSRT,DA(R3)
2291 021134 000045          MOV   #R3,R3
2292 021136 016403 000104          ISDRST: MOV   DCS(R4),R3
2293 021142 012763 000013 000004          MOV   #DRST,DA(R3)
2294 021150 012763 000204 000000          CSTUFF: MOV   #CRDY,STAT,CS(R3)
2295 021156 055633 000106 000000          BIT   DSEL(R4),CS(R3)
2296 021164 042763 000200 000000          BIC   #CRDY,CS(R3)
2297 021172 004537 021052 000013          JSP   R5,WTRD V
2298 021176 022763 000013 000004          CMP   #DRST,DA(R3)
2299 021204 001402          RFD
2300 021206 016301 000006          MOV   #R3,R1
    
```

```

2301 021212 000205          1$:   RTS     R5
2302
2303          .SBTTL  ROUTINE TO GENERATE A RANDOM NUMBER
2304
2305 021214 010146          RAND:  MOV    R1,-(SP)
2306 021218 010346          MOV    R2,-(SP)
2307 021220 010346          MOV    R3,-(SP)
2308
2309 021222 013703 002136          MOV    LONUM,R3
2310 021224 013703 002134          MOV    HNUM,R1
2311 021226 013703 177771          MOV    #7,R2
2312 021236 006303          1$:   ASL    R3
2313 021240 006101          ROL    R1
2314 021244 006303          ROL    R2
2315 021246 006303          INC    R5
2316 021248 006303          INC    R6
2317 021246 063703 002136          ADD    LONUM,R3
2318 021252 005501          ADC    R1
2319 021254 063703 002134          ADD    HNUM,R1
2320 021260 063703 001034          ADD    #1057,R3
2321 021264 005501          ADC    R1
2322 021266 062701 047401          ADD    #47401,R1
2323 021272 010146          MOV    R3,HNUM
2324 021274 010137 002136          MOV    R1,LONUM
2325 021302 012603          MOV    (SP)+,R3
2326 021304 012602          MOV    (SP)+,R2
2327 021306 012601          MOV    (SP)+,R1
2328 021310 006205          RTS     R5
2329          .SBTTL  ROUTINE TO WRITE PACKS INITIALLY
2330          ;ROUTINE TO WRITE PACK WITH PATTERN, ALL TRACKS WILL BE
2331          ;WRITTEN (EXCEPT RAD SECTOR TRACK)
2332          ;FORMAT IS # OF WORDS (WORD 1), PATTERN ADDRESS (WORD 2)
2333          ;PATTERN (WORDS 3 - 128)
2334          ;WE WILL ATTEMPT TO WRITE MULTIPLE SECTORS AT A TIME
2335          ;(MINIMUM 10 SECTORS) IF AN ERROR OCCURS WE WILL THEN
2336          ;WRITE INDIVIDUAL SECTORS FOR THAT TRACK. WE DO WRITES,
2337          ;READS AND INCOPE COMPARISONS TO VERIFY.
2338          ;
2339          ;CALL JSR R5,WRPACK
2340
2341 021312 010046          WRPACK: MOV   R0,-(SP)      ;SAVE REGISTERS
2342 021314 010146          MOV   R1,-(SP)
2343 021316 010246          MOV   R2,-(SP)
2344 021318 010346          MOV   R3,-(SP)
2345 021322 006011 000110          MOV   #R4,-(SP)
2346 021326 005764 000120          TST  WRP(R4)
2347 021332 001033          BNE  1$          ;SEE IF WRITE IN PROGRESS WAS SET
2348 021334          PRINTF #FMTR, #MSWRPK ;JUMP IF SET - DON'T PRINT MESSAGE
2349 021336          PRINTF #FMTR, #MSWRPK
2350 021338          MOV   #R4,-(SP)
2351 021340          MOV   #R4,-(SP)
2352 021342          MOV   #R4,-(SP)
2353 021344          MOV   #R4,-(SP)
2354 021346          MOV   #R4,-(SP)
2355 021348          MOV   #R4,-(SP)
2356 021350          MOV   #R4,-(SP)
2357 021352          EMT  CSRTF
    
```

```

(4) 021354 062706 000006      ADD #6,SP
2355 021360 005046      PRINTF #FMT17,#MRLCS,DCS(R4),#DRNM,CR,DRSEL+1(P4)>
(11) 021362 152416 000107      CLR -(SP)
(10) 021366 012746 003634      RT5B
(9) 021372 016446 000104      MOV DRSCL+1(R4),(SP)
(8) 021376 012746 005237      MOV #DRNM,(SP)
(7) 021402 012746 005237      MOV DCS(R4),(SP)
(6) 021406 012746 000005      MOV #MRLCS,(SP)
(5) 021412 016600 000005      MOV #FMT17,(SP)
(4) 021414 104074 000014      MOV #5,(SP)
(3) 021416 062706 000014      MOV SP,R0
(2) 021422 004537 022500      EMT CS,PTE
2359 021422 004537 022500      JSR R5,HDHOME ;HEADS HOME

;NOW ACTUALLY WRITE DATA OUT ON PACK, WILL NOT WRITE LAST
;TRACK
;
2360 021426 005037 002200      CLP TEMP1 ;TEMP1=HEAD
2361 021432 005001 000001      CLR R1 ;R1=CYL
2362 021434 027071 077600      CONWR: CMP #77600,R1 ;CYL=255?
2363 021436 027071 077600      BNE STWRT ;NO GO WRITE TRACK
2364 021442 005737 002200      TST STWRT ;YES, CHECK IF HEAD = 1?
2365 021446 001411 000000      BEQ ENDWR ;HEAD = 0 GO WRITE
2366 021450 004537 000110      ENDWR: JSR R5,HDHOME ;HEADS HOME
2367 021450 012663 000000      MOV (SP)+,BRA(R4)
2368 021460 012663 000000      MOV (SP)+,R5
2369 021462 012663 000000      MOV (SP)+,R3
2370 021464 012663 000000      MOV (SP)+,R1
2371 021466 006500 000000      MOV (SP)+,R0
2372 021470 0060205 000000      RTS R5 ;END EXIT

;THIS PORTION WILL WRITE THE PACK USING MULTIPLE SECTORS IF A
;ERROR OCCURS WE WILL GO TO 25 AND INDIVIDUAL SECTORS.
2373
2374
2375
2376
2377 021472 005002 002266 000110      STWRT: CLR R2 ;INITIAL SECTOR 0
2378 021474 012764 017500 000042      MOV #PUP1,BRA(R4) ;BUFFER START
2379 021502 044264 175400 000042      JSR R5,WRRUF,BMP(R4) ;10 SECTORS
2380 021510 044264 175400 000042      BSR R5,BDA(R4) ;WRITE BUFFER INTO MEMORY
2381 021514 010164 000040 2015: 2015: MOV R2,BDA(R4) ;SET UP SECTOR
2382 021520 053764 002200 000040      BIS TEMP1,BDA(R4)
2383 021526 005764 000120      TST WRIPC(R4)
2384 021530 005764 000120      TST R6
2385 021534 026464 000122 000040      CMP #762,R6 ;WAS WRITE IN PROGRESS SET?
2386 021542 001402 000000      BEQ 762S ;JUMP IF NOT SET
2387 021546 005764 000040      JMP PROS(R4),BDA(R4) ;JUMP IF SAME ADDRESS WHEN DIED?
2388 021550 005764 000040      BEQ 952S ;JUMP IF ON CYLINDER
2389 021554 012764 002266 000110      JMP BDA(R4) ;ELSE, LOOK AT THE NEXT CYL ADDRESS
2390 021562 012764 000012 000044      MOV #RUF1,BRA(R4) ;SET UP TO WRITE
2391 021566 004537 000012 000044      MOV #WRITE,FUNC(R4) ;WRITE
2392 021574 004537 011052      JSR R5,LDFUNC
2393 021600 005774 000104      JSR R5,WTRDY ;WAIT FOR READY
2394 021604 100003 000000      TST @DCS(R4) ;ERROR
2395 021610 004537 000000      RPL 203S
2396 021612 000421 021136 2055: JSR R5,ISDRST

```

```

2397
2398 021614 012764 000002 000044 203S: MOV #WRCHK,FUNC(R4)
2399 021622 004537 014112      JSP R5,LDFUNC
2400 021630 005774 000104      JSR R5,WTRDY
2401 021634 005774 000104      TST @DCS(R4)
2402 021636 100764      BMI 2055 ;ERROR
2403 ;YES GO DO SECTORS INDIVIDUALLY
2404
2405 021640 062702 000012      ADD #10,R2 ;NEXT GROUP
2406 021644 022702 000050      CMP #40,R2 ;DONE?
2407 021650 001321 000000      BNE 2015 ;NO, GO BACK
2408 021652 004137 022154      JMP 952S ;YES NEXT TRACK
2409
2410 ;IF AN ERROR OCCURS THEN WE COME HERE AND DO THE TRACK SECTOR
2411 ;BY SECTOR.
2412
2413 021656 005002 000002      2S: CLR R2 ;R2 = SECTOR
2414
2415 021660 012764 177600 000042      MOV #128,BMP(R4) ;LOAD WORD COUNT
2416 021662 053764 002200 000040 3S: 3S: MOV R1,BDA(R4) ;SETUP DISK ADDRESS
2417 021666 053764 002200 000040      BIS TEMP1,BDA(R4)
2418 021700 050264 000040      BIS R2,BDA(R4)
2419
2420 021704 012764 002266 000110      MOV #PUP1,BRA(R4)
2421 021712 004537 002266 000110      JSR R5,WRRUF ;WRITE A BUFFER
2422 021716 005037 002124      CLP R6CNT ;CLEAR RETRY'S OUT
2423 021722 005037 002174 000044 91S: 91S: CLP DECNT
2424 021726 012764 000012 000044 96S: 96S: MOV #WRITE,FUNC(R4) ;WRITE FUNCTION
2425 021734 004537 014112      JSR R5,LDFUNC
2426 021740 004537 021052      JSR R5,WTRDY ;WAIT FOR WRITE TO FINISH
2427
2428 021744 005774 000104      TST @DCS(R4) ;ERROR ON WRITE?
2429 021750 100023 000000      RPL 95S ;NO, GO READ
2430
2431 021752 004537 021136 002172      JSR P5,ISDRST
2432 021756 016437 000744      MOV BDA(R4),CHKSEC ;YES, CHECK IF SECTOR IS IN
2433 021764 004537 000744      JSR R5,CKRDISC ;RAD SECTOR FILE
2434 021770 005737 002170      HDE FND ;IF SET IT WAS
2435 021774 001050 000000      BNE 95S ;YES GO TO NEXT SECTOR
2436
2437 021776 005237 002174      INC DECNT ;NO, GIVE IT ONE MORE TRY
2438 022002 023727 002174 000002 96S: 96S: CMP DECNT,#2. ;IT MAY HAVE BEEN NOISE.
2439 022010 001346      RNF
2440
2441 022012 004537 022240      JSR R5,INPAD
2442 022016 000437 000000      BP 95S
2443
2444
2445
2446 022020 005037 002122      R5S: 000044  R5S: CLP RECNT ;CLEAR RETRY COUNT
2447 022024 012764 000002 000044  R0S:  R0S: MOV #WRCHK,FUNC(R4)
2448 022030 004537 014112      JSR R5,LDFUNC
2449 022036 004537 021052      JSR R5,WTRDY
2450
2451 022042 005774 000104      TST @DCS(R4) ;ERROR ON READ
2452 022046 100023 000000      RPL R1S ;NO, GO COMPARE

```

```

2453 022050 004537 021136 JSR R5,ISDRST
2454 022054 016437 000040 MOV RDA(R4),CHKSEC ;CHECK IF SECTOR IS
2455 022062 004537 023744 JSR R5,CKRDSC ;A KNOWN BAD SECTOR
2456 022072 001011 021170 RST ;IT WAS THEN
2457 022072 001011 021170 RNE ;GO TO NEXT SECTOR
2458 022074 005237 002122 INC RECNT ;GIVE IT ANOTHER CHANCE
2459 022074 005237 002122 CWF RECNT,#2.
2460 022106 001346 000002 RNE R05
2461 022110 004537 022240 JSR R5,INRAD
2462 022114 000400 000000 BR 955
2463 022116 81$:
2464 022116 062702 000017 ADD #10,R2 ;NEXT SECTOR (OFFSET BY 10)
2465 022122 026227 000047 CMP R2,#58. ;DONE WITH TRACK?
2466 022126 003002 000000 RGT 9515 ;YES NEXT TRACK
2467 022130 000137 021666 JMP 35 ;NO GO BACK FOR NEXT SECTOR
2468 022134 005202 000000 INC R2 ;NEXT SECTOR
2469 022136 162702 000050 SUB #40,R2 ;DONE WITH TRACK?
2470 022142 020202 000012 BEO R2,#10. ;YES
2471 022150 000137 021666 JMP 35 ;NO
2472 022154 952$:
2473 022154 005737 002200 TST TEMP1 ;WHICH SURFACE?
2474 022160 001420 000000 BEO 55 ;TOP (0), BRANCH
2475 022162 005037 002200 CLP TEMP1 ;BOTTOM, SWITCH TO TOP WITH
2476 022172 012764 000205 ADD #700,R1 ;NO
2477 022200 012764 000006 MOV #205,RDA(R4) ;SEEK GO IN ALSO
2478 022206 004537 014112 JSR R5,LDFUNC ;GO SEEK
2479 022212 004537 021052 JSR R5,WTRDY
2480 022216 000137 021434 JMP CONWR
2481 022222 012737 000100 MOV #HEAD,TEMP1 ;WAS TOP, MAKE BOTTOM.
2482 022230 012764 000021 MOV #21,BDA(R4)
2483 022236 000760 000040 BR 45
2484 022240 016337 000000 INBAD: MOV CS(R3),E-CS
2485 022246 016337 000002 MOV BA(R3),E-BA
2486 022254 016337 000004 MOV DA(R3),E-DA
2487 022260 016337 000006 MOV MP(R3),E-MP
2488 022270 016337 000006 MOV MP(R3),E-MP1
2489 022276 016337 000006 MOV MP(R3),E-MP2
2490 022304 104463 ERRHRD 109 ;WRTS,ERR13
2491 022306 000307 .TRAP CSFCODE
2492 022310 002541 .WORD 109
2493 022312 005172 .WORD WRTS
2494 .WORD ERR13
    
```

```

2505 022314 005264 000012 INC EPRCNT(R4)
2506 022324 001413 007545 RST 7,DPP ;ARE WE COUNTING ERRORS
2507 022326 026437 000012 CMP EPRCNT(R4),ERLMT ;PAST IT
2508 022334 103407 002126 BLO 25 ;NO
2509 022344 004537 020220 MOV #ERLMTM,WHY
2510 022344 004537 020220 JSR R5,DBDRV
2511 022350 012705 021450 MOV #ENDMR,R5
2512 022354 000205 2$:
2513 .SBTTL RTS R5
2514 ;SBTTL ROUTINE FOR SYSTEM CLOCK
2515 ;ROUTINE TO READ SYSTEM CLOCK
2516 ;USES "RECTIM" FROM DIAGNOSTIC SUPERVISOR
2517 GETSYS: TST SYSCLK ;DO WE HAVE A CLOCK
2518 RNE 45 ;YES, GO SERVICE IT
2519 BREAK ;NO, CALL SUPER FOR ^C
2520 C$RPK R5 ;EXIT
2521 RTS R0 ;GET PRESENT TIME
2522 4$:
2523 EMT C$RECTIM
2524 CMP R0,L$TTIM ;HAS IT MOVED
2525 RGT 35 ;NO MOVEMENT SINCE LAST CALL
2526 MOV R0,L$TTIM ;CALCULATE DIFFERENCE
2527 MOV R1,R0 ;AND FIX ACCORDINGLY
2528 SUB R1,R0
2529 RGT 25 ;RUMP SECONDS
2530 ADD R0,SECOND ;SECONDS OVERFLOW
2531 CMP #0,SECOND
2532 RGT 35 ;RUMP SECONDS
2533 INC #0,SECOND
2534 INC INTERVAL ;TIME BETWEEN REPORTS
2535 INC MINUTE ;RUMP MINUTES
2536 CMP #0,SECOND
2537 RGT 75 ;RUMP MINUTES
2538 INC #0,MINUTE
2539 RGT 35 ;RUMP MINUTES
2540 INC HOUR
2541 CMP #0,MINUTE
2542 RTS R5
2543 .SBTTL HEADS HOME ROUTINE
2544 ;ROUTINE TO BRING HEADS OVER TRACK 0
2545 HDHOME: MOV R0,(-SP) ;SAVE R0
2546 MOV #RDR,FUNC(R4) ;READ HEADER
2547 JSR R5,LDFUNC ;GO DO IT.
2548 JSR R5,WTRDY
2549 MOV MP(R3),R0 ;GET HEADER
2550 BIC #17,R0 ;ONLY CYLINDER
2551 MOV R0,RDA(R4) ;MOVE IT TO BUFFERED DA
2552 RIS #MK,RDA(R4) ;SET MARKER
2553 MOV #SEL,FUNC(R4) ;LOAD SEEK
2554 JSR R5,LDFUNC ;SEEK
    
```

```

2559 022554 004537 021052 JSR R5,WTRDV ;WAIT.
2560 022560 016464 000122 MOV PROS(R4),LSTHDR(R4)
2561 022566 005064 000122 CLP PROS(R4) ;SET BUFFER TO HOME
2562 022572 002500 (SP)+,R0
2563 022574 000205 RTS R5
2564
2565 .SBTTL RANDOM WC AND DA ROUTINE
2566
2567 ;ROUTINE TO GET RANDOM SECTOR AND WORD COUNT FOR R/W TRANSFER.
2568 ;SECTOR IS CHOSEN BETWEEN MIN/MAX LIMITS, WORD COUNT IS BETWEEN
2569 ;MIN/MAX WORD COUNT. WORD COUNT WILL BE ADJUSTED NOT TO CAUSE
2570 ;TRACK OVERFLOW IF HIGH SECTORS ARE CHOSEN....
2571 ;R4 HAS BUFFER OF DRIVE WE'RE WORKING WITH
2572 ;ON EXIT - BWP(R4) HAS WORD COUNT
2573 ; ON EXIT - BDA(R4) HAS DISK ADDRESS
2574
2575 022576 023737 007540 007542 GWGDA: CMP T,MXS,T,MNS ;MIN MAX SECTORS EQUAL
2576 022604 001003 BNE 99S ;NO, CALCULATE ONE
2577 022606 013702 MOV T,MXS,R2 ;LOAD SECTOR
2578 022612 000421 BR 99S ;GO GET WC
2579 022614 004537 JSR R5,RAND ;GET RANDOM # FOR SECTOR
2580 022620 013702 MOV LOGUM,R2 ;MIN/MAX WORD COUNT
2581 022624 042702 002136 1S: BIC #177700,R2 ;0-77 ONLY
2582 022630 023702 007540 CMP #1,R2 ;R2 LOWER THAN MAX
2583 022634 103003 BHS R2 ;BRANCH IF YES
2584 022636 006202 ASR R2 ;HALF IT
2585 022640 005202 INC R2 ;INC SO NOT 0
2586 022644 00A970 BR 3S ;MIN OKAY
2587 022648 002337 007542 3S: CMP R5,T,MNS
2588 022650 103502 BHS 1S
2589 022652 006102 ROL R5
2590 022654 000763 BR 1S
2591
2592 ;NOW GET WORD COUNT
2593
2594 022656 005737 007574 5S: TST T,STIP
2595 022662 001003 BNE 95S
2596 022664 013737 MOV MAXWC,T,MXB
2597 022672 023737 002272 007526 95S: CMP MAXWC,T,MXB
2598 022700 103021 BHS 97S
2600
2601 022702 PRINTF #PMT13D,#OVER,T,MXB,MAXWC
2602 (0) 022706 013746 MOV MAXWC,-(SP)
2603 (1) 022712 012746 MOV T,MXB,-(SP)
2604 (2) 022716 012746 MOV #OVER,-(SP)
2605 (3) 022722 012746 MOV #PMT13D,-(SP)
2606 (4) 022730 104017 MOV #4,-(SP)
2607 (5) 022732 062706 EMT CPRINTF
2608 022736 013737 ADD #12,SP
2609 022744 023737 007526 007550 97S: CMP T,MXB,T,MNB ;MIN MAX EQUAL
2610 022752 003006 BGT 5S
2611 022754 013737 007526 007550 MOV T,MXB,T,MNB
    
```

```

2607
2608 022762 013703 007526 MOV T,MXB,R3 ;YES SET WC
2609 022766 000421 BR 9S
2610 022770 004537 JSR R5,RAND ;GET RANDOM WORD COUNT
2611 022774 013703 007526 6S: MOV LOGUM,R3
2612 023000 042703 100000 7S: BIC #160000,R3 ;MAX!!!!
2613 023004 023703 007526 CMP T,MXB,R3
2614 023010 103003 BHS R3
2615 023014 005202 ASR R3
2616 023016 000770 INC R3
2617 023020 000770 BR 7S
2618 023024 020337 007550 8S: CMP R3,T,MNB
2619 023028 002103 BHS R3
2620 023032 002103 ROL R3
2621 023036 000763 BR 7S
2622
2623 ;NOW WE HAVE SECTOR AND WORD COUNT, CHECK THAT WORD COUNT WILL FIT ON SECTOR
2624 ;IF NOT LOWER SECTOR START
2625
2626 023038 012761 000050 9S: MOV #40,,R1 ;SETUP FOR FOURTY SECTORS
2627 023040 005403 NEG R3,RMP(R4) ;MAKE WORD COUNT NEGATIVE
2628 023044 005301 000042 11S: MOV R3,RMP(R4) ;LOAD WORD COUNT
2629 023048 005301 000200 DEC R1 ;DOWN COUNT MINIMUM START SECT NEEDED
2630 023052 100774 000200 ADD #128,,R3 ;ONE SECTOR'S WORTH
2631 023056 020201 000200 BWT R1 ;STILL NEED ANOTHER SECTOR
2632 023060 101401 RLOS R2,R1 ;DID RANDOM SECTOR SUFFICE
2633 023064 010102 CMP R2,R1 ;BRANCH IF SUFFICED
2634 023068 010102 MOV R1,R2 ;NO, THEN MAKE IT FIT
2635 023072 016464 000122 000040 12S: MOV PROS(R4),BDA(R4)
2636 023076 042764 000077 000040 BIC #77,BDA(R4)
2637 023080 050264 000040 BIS R2,BDA(R4)
2638 023102 000205 RTS R5
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656 .SBTTL ROUTINE TO DUMP BUFFER ON DCK
2657 ;ROUTINE TO DUMP BUFFER ON DCK ERROR, TWO DUMPS ARE POSSIBLE
2658 ;ONE WHERE WE CAN COMPARE WHAT IT SHOULD BE AND THE OTHER
2659 ;WHEN WE CAN'T
2660
2661 023104 004737 005230 DMPBUF: JSR PC,LINE1
2662
    
```



```

2663                                     ;
2664                                     ;
2665                                     ;
2666                                     ;
2667 023110 012737 000200 002314    MOV    #128,DWCNT1
2668 023118 016400 000040              MOV    R0A(R4),R0
2669                                     ;
2670 023122 013701 002254              MOV    E,DA,R1
2671 023125 042700 177700              BIC    #177700,R0
2672 023126 042701 177700              BIC    #177700,R1
2673 023136 010002                    MOV    R0,R2
2674 023140 010103                    MOV    R1,R3
2675 023142 065003                    SUB    R2,R3
2676 023144 016403                    CLR    R4
2677 023146 062702 000200 93S:     ADD    #12R.,R2
2678 023152 005303                    DEC    R3
2679 023154 001374                    RNE   R3
2680 023156 016403                    MOV    R0P(R4),R3
2681 023162 005403                    NEG    R3
2682 023164 020203                    CMP    R2,R3
2683 023166 003405                    RGT   R2,R3
2684 023168 002252                    MOV    E,BA,R2
2685 023174 162702 000400          SUB    #40C,R2
2686 023200 000412                    BR    R2
2687 023204 010000                    RR    R2
2688 023204 010000                    SUB    R2,R0
2689 023210 160200                    SUB    R2,R0
2690 023212 010037 002314          MOV    R0,DWCNT1
2691 023214 005303                    ASL   R0
2692 023216 002252                    MOV    E,BA,R2
2693 023224 160002                    SUB    R0,R2
2694 023226                    PRINTR #FMT13,#BUSAD,R2,#CRLDA,CHKSEC
2695 023228                    MOV    CHKSEC,-(SP)
2696 023230                    MOV    #CRLDA,-(SP)
2697 023232                    MOV    R2,-(SP)
2698 023234                    MOV    #R0SAD,-(SP)
2699 023236                    MOV    #FMT13,-(SP)
2700 023238                    MOV    #FMT13,-(SP)
2701 023240                    MOV    SP,R0
2702 023242                    EMT   C$ENTR
2703 023244                    ADD    #4,SP
2704 023246                    RR    ST0MP
2705 023306 013746 002172 3S:     PRINTR #FMT14,#NOREV
2706 023308 013746 002407          MOV    #NOREV,-(SP)
2707 023310 010246                    MOV    #FMT14,-(SP)
2708 023312 010246                    MOV    R2,-(SP)
2709 023314 012746 003737          MOV    #R0SAD,-(SP)
2710 023316 012746 000244          MOV    #FMT13,-(SP)
2711 023318 104914                    MOV    #FMT13,-(SP)
2712 023320 010600                    MOV    SP,R0
2713 023322 104914                    EMT   C$ENTR
2714 023324 000014                    ADD    #4,SP
2715 023326 024430                    MOV    #4,ST,R0
2716 023328 012701 000010          MOV    #R.,R1
2717 023330 022062 000002 1S:     CMP    (R0)+,2(R2)
2718 023332 005303                    BEQ   R1
2719 023334 001373                    DEC    R1
2720 023336 001373                    RNE   R1
2721 023338                    PRINTR #FMT14,#NOREV
2722 023340 012746 003434 3S:     MOV    #NOREV,-(SP)
2723 023342 012746 006507          MOV    #FMT14,-(SP)
2724 023344 012746 000002          MOV    #2,-(SP)
2725 023346 104914                    MOV    SP,R0
2726 023348 104914                    EMT   C$ENTR
2727 023350 062706 000006          ADD    #6,SP
2728 023352 000532                    RR    ST0MP
    
```

```

2704                                     ;
2705 023334 021227 000200 2S:     CMP    (R2),#128.
2706 023340 101362                    RHI   3
2707 023342 005037 002174          CLR    DECNT
2708 023346 013701 007570          MOV    T,CLT,R1
2709                                     ;
2710 023352 012237 002176              MOV    (R2)+,TEMP0
2711 023356 013737 002176 002312  MOV    TEMP0,DWCNT
2712 023358 005303 002312          NEG    DWCNT
2713 023360 012237 002300          MOV    (R2)+,TEMP1
2714 023374 162737 000002 002176  SUB    R2+,TEMP1
2715 023402 012737 000002 002202  MOV    E$,TEMP2
2716 023410 013739 002200          MOV    TEMP1,R3
2717 023414 013739 002200          MOV    TEMP1,R3
2718 023422 005737 002176 002210 4S:     TST   TEMP5
2719 023426 001417                    BEQ   R0
2720 023430 005337 002176          DEC    TEMP0
2721 023434 005337 002210          TST   TEMP5
2722 023440 001005                    BRE   5
2723 023442 012737 000020 002210  MOV    #16.,TEMP5
2724 023450 013703 002200          MOV    TEMP1,R3
2725 023454 005337 002210          MOV    #3+,GDDAT
2726 023460 005337 002216          DEC    TEMP5
2727 023464 000402                    BR    7S
2728 023466 002037 002212 6S:     CLP   GDDAT
2729 023470 005337 002212 7S:     INC   DWCNT
2730 023476 021237 002232          CMP    (R2),GDDAT
2731 023502 001422                    BEQ   R0
2732                                     ;
2733 023504 005237 002174          INC   DECNT
2734 023510 005701                    TST   R1
2735 023512 001416                    REQ   R0
2736 023514 005301                    DEC   R1
2737 023516                    PRINTR #FMT14B,TEMP2,GDDAT,(R2)
2738 023520 011246 002232          MOV    (R2)+,-(SP)
2739 023524 013746 002202          MOV    GDDAT,-(SP)
2740 023528 013746 002202          MOV    TEMP2,-(SP)
2741 023534 012746 000004          MOV    #FMT14B,-(SP)
2742 023538 010600                    MOV    #4,-(SP)
2743 023540 104914                    MOV    SP,R0
2744 023544 062706 000012          EMT   C$ENTR
2745 023546 062706                    ADD    #12,SP
2746 023550 005237 002202 8S:     INC   TEMP2
2747 023554 005739 002202          TST   (R2)+
2748 023558 005739 002202 002314  CMP    TEMP2,DWCNT1
2749 023564 003716                    BLE   R0
2750 023566                    PRINTR #FMT9A,DECNT,TEMP2
2751 023568 013746 002203          MOV    TEMP2,-(SP)
2752 023572 012746 000230          MOV    DECNT,-(SP)
2753 023576 012746 000003          MOV    #FMT9A,-(SP)
2754 023580 012746 000003          MOV    #3,-(SP)
2755 023584 010600                    MOV    SP,R0
2756 023588 104914                    EMT   C$ENTR
2757 023592 062706                    ADD    #10,SP
    
```



2846	024500	000000		
2847	024500	000000	.WORD	0
2848	024504	000000	.WORD	0
2849	024506	000000	.WORD	0
2850				
2851	024510	177777	PAT1:	.WORD 177777
2852	024512	177777		.WORD 177777
2853	024514	177777		.WORD 177777
2854	024516	052525		.WORD 052525
2855	024520	052525		.WORD 052525
2856	024522	052525		.WORD 052525
2857	024524	177777		.WORD 177777
2858	024526	177777		.WORD 177777
2859	024530	052525		.WORD 052525
2860	024532	052525		.WORD 052525
2861	024534	177777		.WORD 177777
2862	024536	177777		.WORD 177777
2863	024540	177777		.WORD 177777
2864	024542	177777		.WORD 177777
2865	024544	172765		.WORD 172765
2866	024546	172765		.WORD 172765
2867				
2868	024550	000000	PAT2:	.WORD 0
2869	024552	000000		.WORD 0
2870	024554	000000		.WORD 0
2871	024556	177777		.WORD 177777
2872	024560	177777		.WORD 177777
2873	024562	177777		.WORD 177777
2874	024564	000000		.WORD 0
2875	024566	000000		.WORD 0
2876	024570	177777		.WORD 177777
2877	024572	177777		.WORD 177777
2878	024574	000000		.WORD 0
2879	024576	177777		.WORD 177777
2880	024600	000000		.WORD 0
2881	024602	177777		.WORD 177777
2882	024604	000000		.WORD 0
2883	024606	177777		.WORD 177777
2884				
2885	024610	025252	PAT3:	.WORD 25252
2886	024612	025252		.WORD 25252
2887	024614	052525		.WORD 52525
2888	024616	125252		.WORD 125252
2889	024620	125252		.WORD 125252
2890	024622	052525		.WORD 52525
2891	024624	052525		.WORD 52525
2892	024626	052525		.WORD 52525
2893	024630	125252		.WORD 125252
2894	024632	125252		.WORD 125252
2895	024634	052525		.WORD 52525
2896	024636	125252		.WORD 125252
2897	024640	052525		.WORD 52525
2898	024642	125252		.WORD 125252
2899	024644	052525		.WORD 52525
2900	024646	125252		.WORD 125252
2901				

2902	024650	155555	PAT4:	.WORD 155555
2903	024652	133333		.WORD 133333
2904	024654	066666		.WORD 066666
2905	024656	155555		.WORD 155555
2906	024660	133333		.WORD 133333
2907	024662	066666		.WORD 066666
2908	024664	066666		.WORD 066666
2909	024666	133333		.WORD 133333
2910	024670	066666		.WORD 066666
2911	024672	155555		.WORD 155555
2912	024674	133333		.WORD 133333
2913	024676	066666		.WORD 066666
2914	024700	155555		.WORD 155555
2915	024702	133333		.WORD 133333
2916	024704	066666		.WORD 066666

```

2918 024706 155555 .WORD 155555
2919
2920 024710 121105 PAT5: .WORD 121105
2921 024712 157442 .WORD 157442
2922 024714 064221 .WORD 064221
2923 024716 132110 .WORD 132110
2924 024720 055044 .WORD 055044
2925 024722 013211 .WORD 013211
2926 024724 105504 .WORD 105504
2927 024726 105504 .WORD 105504
2928 024730 042642 .WORD 042642
2929 024732 013211 .WORD 013211
2930 024734 116550 .WORD 116550
2931 024736 044264 .WORD 044264
2932 024740 022132 .WORD 022132
2933 024742 011052 .WORD 011052
2934 024744 104426 .WORD 104426
2935 024746 042213 .WORD 042213
2936
2937 024750 177777 PAT6: .WORD 177777
2938 024752 177777 .WORD 177777
2939 024754 177777 .WORD 177777
2940 024756 177777 .WORD 177777
2941 024758 177777 .WORD 177777
2942 024760 177777 .WORD 177777
2943 024762 177777 .WORD 177777
2944 024764 177777 .WORD 177777
2945 024766 177777 .WORD 177777
2946 024768 177777 .WORD 177777
2947 024770 177777 .WORD 177777
2948 024772 177777 .WORD 177777
2949 024774 177777 .WORD 177777
2950 025000 177777 .WORD 177777
2951 025002 177777 .WORD 177777
2952 025004 177777 .WORD 177777
2953 025006 177777 .WORD 177777
2954
2955 025010 045513 PAT7: .WORD 045513
2956 025012 122645 .WORD 122645
2957 025014 151322 .WORD 151322
2958 025016 081527 .WORD 081527
2959 025018 132264 .WORD 132264
2960 025020 132264 .WORD 132264
2961 025022 055132 .WORD 055132
2962 025024 026455 .WORD 026455
2963 025026 113226 .WORD 113226
2964 025028 045513 .WORD 045513
2965 025030 122645 .WORD 122645
2966 025032 151322 .WORD 151322
2967 025034 081527 .WORD 081527
2968 025036 132264 .WORD 132264
2969 025038 026455 .WORD 026455
2970 025040 113226 .WORD 113226
2971
2972 025050 000240 ENDOFPROGRAM: NOP
2973
    
```

```

2974 025052 ENDTST:
2975 025054 L10022: FMT CSETST
2976 025054 HALT
2977
2978 .SBTTL DRIVE INFORMATION BUFFERS
2979 ;DRIVE INFORMATION BUFFER
2980
2981
2982 .LIST ME
2983
2984 025056 DRBUF:
3030
(1) 025056 000000 ;SKCNT
(1) 025060 000002 ;RXFR1
(1) 025062 000004 ;RXFR2
(1) 025064 000006 ;WXFR1
(1) 025066 000008 ;WXFR2
(1) 025070 000012 ;ERRCNT
(1) 025072 000014 ;SFTCNT
(1) 025074 000016 ;SKECNT
(1) 025076 000018 ;DRCNT
(1) 025100 000022 ;DRCCEP
(1) 025102 000024 ;HRCCEP
(1) 025104 000026 ;DLTCNT
(1) 025106 000028 ;OPICNT
(1) 025110 000032 ;HWFERR
(1) 025112 000034 ;NXCNT
(1) 025114 000036 ;RETRY
(1) 025120 000042 ;RPA
(1) 025122 000044 ;RMP
(1) 025124 000046 ;RUNC
(1) 025126 000050 ;RCSADR
(1) 025128 000052 ;LSTHDR
(1) 025130 000054 ;RYPE
(1) 025132 000056 ;SKCNT1
(1) 025134 000058 ;PPFLGS
(1) 025136 000060 ;RXFR3
(1) 025140 000062 ;LSTDA
(1) 025142 000064 ;DIFWD
(1) 025144 000066 ;DPHOUR
(1) 025150 000072 ;TRERR
(1) 025152 000074 ;DATCEP
(1) 025154 000076 ;DOWCK
(1) 025156 000100 ;SEPNM1
(1) 025160 000102 ;SEPNM2
(1) 025162 000104 ;DCS
(1) 025164 000106 ;DRESEL
(1) 025166 000110 ;RA
(1) 025170 000112 ;BSECP
(1) 025172 000114 ;RSEK
(1) 025174 000116 ;SOFTCS
(1) 025176 000120 ;WIFG
(1) 025200 000122 ;PPPOS
(1) 025056 000000 ;SEEK OPERATION COUNT
(1) 025060 000002 ;READ OPERATION COUNT (BITS) LOW ORDER
(1) 025062 000004 ;" " " HIGH ORDER
(1) 025064 000006 ;WRITE OPERATION COUNT (BITS) LOW ORDER
(1) 025066 000008 ;" " " HIGH ORDER
(1) 025070 000012 ;ERROR COUNT - HARD
(1) 025072 000014 ;ERROR COUNT - SOFT
(1) 025074 000016 ;SEEK ERROR COUNT
(1) 025076 000018 ;DRIVE ERROR COUNT
(1) 025100 000022 ;DATA CRC ERROR COUNT
(1) 025102 000024 ;HEADER CRC ERROR COUNT
(1) 025104 000026 ;DATA LATE ERROR COUNT
(1) 025106 000028 ;OPERATION INCOMPLETE ERROR COUNT
(1) 025110 000032 ;HEADER NOT FOUND ERROR COUNT
(1) 025112 000034 ;NON EXISTANT MEMDRY ERROR COUNT
(1) 025114 000036 ;PRESENT RETRY NUMBER
(1) 025120 000042 ;" DISK ADDRESS CONTENTS
(1) 025122 000044 ;" MULTIPURPOSE CONTENTS
(1) 025124 000046 ;LAST FUNCTION LOADED
(1) 025126 000050 ;CSR IMAGE OF LAST COMMAND
(1) 025128 000052 ;LAST POSITION ON DISK
(1) 025130 000054 ;SEROP ON WHICH RECOVERY IS IN PROGRESS
(1) 025132 000056 ;SEEK COUNT LOW ORDER
(1) 025134 000058 ;PROGRAM INTERNAL FLAGS
(1) 025136 000060 ;READ COUNT THIRD
(1) 025140 000062 ;WRITE COUNT THIRD
(1) 025142 000064 ;DISK ADDRESS OF SOFT ERROR
(1) 025144 000066 ;LAST DIFFERENCE WORD OF SEEK
(1) 025150 000072 ;TIME DRIVE WAS DROPPED
(1) 025152 000074 ;TRACKING ERROR COUNT
(1) 025154 000076 ;WRITE CHECK NECESSARY
(1) 025156 000100 ;SERIAL NUMBER OF CARTRIDGE
(1) 025160 000102 ;SERIAL NUMBER OF CARTRIDGE
(1) 025162 000104 ;CSR ADDRESS
(1) 025164 000106 ;DRIVE SELECT BITS(8,9,10)
(1) 025166 000110 ;PRESENT BUS ADDRESS CONTENTS
(1) 025170 000112 ;POINTER TO BAD SECTOR FILE
(1) 025172 000114 ;
(1) 025174 000116 ;CSR AT TIME OF SOFT ERROR
(1) 025176 000120 ;WRITE IN PROGRESS DURING PWR FAIL
(1) 025200 000122 ;PRESENT POSITION ON DISK
    
```

025202	000000	SKCNT	;	SEEK OPERATION COUNT
025204	000002	RXPFR1	;	READ OPERATION COUNT (BITS) LOW ORDER
025206	000004	WXFR1	;	" " " HIGH ORDER
025208	000006	WYFR1	;	WRITE OPERATION COUNT (BITS) LOW ORDER
025210	000010	WXFR2	;	" " " HIGH ORDER
025212	000012	WYFR2	;	" " " HIGH ORDER
025214	000014	ERRCNT	;	ERROR COUNT - HARD
025216	000016	SPICNT	;	ERROR COUNT - SOFT
025218	000018	SKECNT	;	SEEK ERROR COUNT
025220	000020	DERCNT	;	DRIVE ERROR COUNT
025222	000022	DCRCER	;	DATA CRC ERROR COUNT
025224	000024	HRCRER	;	HEADER CRC ERROR COUNT
025226	000026	DLTCNT	;	DATA LATE ERROR COUNT
025228	000028	OPICNT	;	OPERATION INCOMPLETE ERROR COUNT
025230	000030	HNFERR	;	HEADER NOT FOUND ERROR COUNT
025232	000032	NXMCNT	;	NON EXISTANT MEMORY ERROR COUNT
025234	000034	RETRY	;	PRESENT RETRY NUMBER
025236	000036	RDA	;	" DISK ADDRESS CONTENTS
025238	000038	BWP	;	PRESENT MULTIPURPOSE CONTENTS
025240	000040	FUNC	;	LAST FUNCTION LOADED
025242	000042	BCSADR	;	CSR IMAGE OF LAST COMMAND
025244	000044	LSTHDR	;	LAST POSITION ON DISK
025246	000046	RTYPE	;	ERROR ON WHICH RECOVERY IS IN PROGRESS
025248	000048	SKCNT1	;	SEEK COUNT LOW ORDER
025250	000050	PRFLGS	;	PROGRAM INTERNAL FLAGS
025252	000052	RXPFR3	;	READ COUNT THIRD
025254	000054	WXFR3	;	WRITE COUNT THIRD
025256	000056	LSTDA	;	DISK ADDRESS OF SOFT ERROR
025258	000058	DIFWD	;	LAST DIFFERENCE WORD OF SEEK
025260	000060	DPHOUR	;	TIME DRIVE WAS DROPPED
025262	000062	TREPR	;	TRACKING ERROR COUNT
025264	000064	DATCER	;	
025266	000066	DOWCK	;	WRITE CHECK NECESSARY
025268	000068	SERNM1	;	SERIAL NUMBER OF CARTRIDGE
025270	000070	SERNM2	;	SERIAL NUMBER OF CARTRIDGE
025272	000072	DCS	;	CSR ADDRESS
025274	000074	DRSEL	;	DRIVE SELECT BITS(8,9,10)
025276	000076	BRA	;	PRESENT BUS ADDRESS CONTENTS
025278	000078	BSECT	;	POINTER TO BAD SECTOR FILE
025280	000080	RSECK	;	
025282	000082	SOFTCS	;	CSR AT TIME OF SOFT ERROR
025284	000084	WRIPG	;	WRITE IN PROGRESS DURING PWR FAIL
025286	000086	PRPOS	;	PRESENT POSITION ON DISK
025288	000088		;	
025290	000090	SKCNT	;	SEEK OPERATION COUNT
025292	000092	RXPFR1	;	READ OPERATION COUNT (BITS) LOW ORDER
025294	000094	WXFR1	;	" " " HIGH ORDER
025296	000096	WYFR1	;	WRITE OPERATION COUNT (BITS) LOW ORDER
025298	000100	WXFR2	;	" " " HIGH ORDER
025300	000102	WYFR2	;	" " " HIGH ORDER
025302	000104	ERRCNT	;	ERROR COUNT - HARD
025304	000106	SPICNT	;	ERROR COUNT - SOFT
025306	000108	SKECNT	;	SEEK ERROR COUNT
025308	000110	DERCNT	;	DRIVE ERROR COUNT
025310	000112	DCRCER	;	DATA CRC ERROR COUNT

025352	000024	HRCRER	;	HEADER CRC ERROR COUNT
025354	000026	DLTCNT	;	DATA LATE ERROR COUNT
025356	000028	OPICNT	;	OPERATION INCOMPLETE ERROR COUNT
025358	000030	HNFERR	;	HEADER NOT FOUND ERROR COUNT
025360	000032	NXMCNT	;	NON EXISTANT MEMORY ERROR COUNT
025362	000034	RETRY	;	PRESENT RETRY NUMBER
025364	000036	RDA	;	" DISK ADDRESS CONTENTS
025366	000038	BWP	;	PRESENT MULTIPURPOSE CONTENTS
025368	000040	FUNC	;	LAST FUNCTION LOADED
025370	000042	BCSADR	;	CSR IMAGE OF LAST COMMAND
025372	000044	LSTHDR	;	LAST POSITION ON DISK
025374	000046	RTYPE	;	ERROR ON WHICH RECOVERY IS IN PROGRESS
025376	000048	SKCNT1	;	SEEK COUNT LOW ORDER
025378	000050	PRFLGS	;	PROGRAM INTERNAL FLAGS
025380	000052	RXPFR3	;	READ COUNT THIRD
025382	000054	WXFR3	;	WRITE COUNT THIRD
025384	000056	LSTDA	;	DISK ADDRESS OF SOFT ERROR
025386	000058	DIFWD	;	LAST DIFFERENCE WORD OF SEEK
025388	000060	DPHOUR	;	TIME DRIVE WAS DROPPED
025390	000062	TREPR	;	TRACKING ERROR COUNT
025392	000064	DATCER	;	
025394	000066	DOWCK	;	WRITE CHECK NECESSARY
025396	000068	SERNM1	;	SERIAL NUMBER OF CARTRIDGE
025398	000070	SERNM2	;	SERIAL NUMBER OF CARTRIDGE
025400	000072	DCS	;	CSR ADDRESS
025402	000074	DRSEL	;	DRIVE SELECT BITS(8,9,10)
025404	000076	BRA	;	PRESENT BUS ADDRESS CONTENTS
025406	000078	BSECT	;	POINTER TO BAD SECTOR FILE
025408	000080	RSECK	;	
025410	000082	SOFTCS	;	CSR AT TIME OF SOFT ERROR
025412	000084	WRIPG	;	WRITE IN PROGRESS DURING PWR FAIL
025414	000086	PRPOS	;	PRESENT POSITION ON DISK
025416	000088		;	
025418	000090	SKCNT	;	SEEK OPERATION COUNT
025420	000092	RXPFR1	;	READ OPERATION COUNT (BITS) LOW ORDER
025422	000094	WXFR1	;	" " " HIGH ORDER
025424	000096	WYFR1	;	WRITE OPERATION COUNT (BITS) LOW ORDER
025426	000100	WXFR2	;	" " " HIGH ORDER
025428	000102	WYFR2	;	" " " HIGH ORDER
025430	000104	ERRCNT	;	ERROR COUNT - HARD
025432	000106	SPICNT	;	ERROR COUNT - SOFT
025434	000108	SKECNT	;	SEEK ERROR COUNT
025436	000110	DERCNT	;	DRIVE ERROR COUNT
025438	000112	DCRCER	;	DATA CRC ERROR COUNT
025440	000114	HRCRER	;	HEADER CRC ERROR COUNT
025442	000116	DLTCNT	;	DATA LATE ERROR COUNT
025444	000118	OPICNT	;	OPERATION INCOMPLETE ERROR COUNT
025446	000120	HNFERR	;	HEADER NOT FOUND ERROR COUNT
025448	000122	NXMCNT	;	NON EXISTANT MEMORY ERROR COUNT
025450	000124	RETRY	;	PRESENT RETRY NUMBER
025452	000126	RDA	;	" DISK ADDRESS CONTENTS
025454	000128	BWP	;	PRESENT MULTIPURPOSE CONTENTS
025456	000130	FUNC	;	LAST FUNCTION LOADED
025458	000132	BCSADR	;	CSR IMAGE OF LAST COMMAND
025460	000134	LSTHDR	;	LAST POSITION ON DISK
025462	000136	RTYPE	;	ERROR ON WHICH RECOVERY IS IN PROGRESS

(1)	025526	000354	SKCNT1	;SEEK COUNT LOW ORDER
(1)	025528	000354	PRPOS	;PROGRAM INTERNAL FLAGS
(1)	025530	000060	RXRFR3	;READ COUNT THIRD
(1)	025532	000062	WXRFR3	;WRITE COUNT THIRD
(1)	025534	000064	LSTDA	;DISK ADDRESS OF SOFT ERROR
(1)	025536	000066	DIFWD	;LAST DIFFERENCE WORD OF SEEK
(1)	025542	000070	DPHOUR	;TIME DRIVE WAS DROPPED
(1)	025544	000072	TRERR	;TRACKING ERROR COUNT
(1)	025546	000074	DATCER	
(1)	025550	000076	DOWNCK	
(1)	025552	000100	SERNM1	;WRITE CHECK NECESSARY
(1)	025554	000100	SERNM2	;SERIAL NUMBER OF CARTRIDGE
(1)	025556	000104	DCS	;SERIAL NUMBER OF CARTRIDGE
(1)	025560	000104	DRSEL	;CSR ADDRESS
(1)	025562	000110	RRR	;DRIVE SELECT BITS(8,9,10)
(1)	025564	000112	BSECPT	;PRESENT BUS ADDRESS CONTENTS
(1)	025566	000114	RSECK	;POINTER TO RAD SECTOR FILE
(1)	025570	000116	SOFTCS	
(1)	025572	000120	WRTPG	;CSR AT TIME OF SOFT ERROR
(1)	025574	000122	PRPOS	;WRITE IN PROGRESS DURING PWR FAIL
(1)				;PRESENT POSITION ON DISK
(1)	025576	000000	SKCNT	;SEEK OPERATION COUNT
(1)	025600	000002	RXRFR1	;READ OPERATION COUNT (BITS) LOW ORDER
(1)	025602	000004	RXRFR2	; " " " " HIGH ORDER
(1)	025604	000006	WXRFR1	;WRITE OPERATION COUNT (BITS) LOW ORDER
(1)	025606	000008	WXRFR2	; " " " " HIGH ORDER
(1)	025610	000010	ERRCNT	;ERROR COUNT - HARD
(1)	025612	000012	SFTCNT	;ERROR COUNT - SOFT
(1)	025614	000014	SEKCNT	;SEEK ERROR COUNT
(1)	025616	000020	DERCNT	;DRIVE ERROR COUNT
(1)	025620	000022	DCRCER	;DATA CRC ERROR COUNT
(1)	025622	000024	HRCER	;HEADER CRC ERROR COUNT
(1)	025624	000026	DLCNT	;DATA LATE ERROR COUNT
(1)	025626	000030	OPICNT	;OPERATION INCOMPLETE ERROR COUNT
(1)	025630	000032	HNFERR	;HEADER NOT FOUND ERROR COUNT
(1)	025632	000034	NXMCNT	;NON EXISTANT MEMORY ERROR COUNT
(1)	025634	000036	RETRY	;PRESENT RETRY NUMBER
(1)	025636	000040	RDA	; " DISK ADDRESS CONTENTS
(1)	025640	000042	BMP	;PRESENT MULTIPURPOSE CONTENTS
(1)	025642	000044	FUNC	;LAST FUNCTION LOADED
(1)	025644	000046	BCSADR	;CSR IMAGE OF LAST COMMAND
(1)	025646	000050	LSTHDR	;LAST POSITION ON DISK
(1)	025650	000052	RTYPE	;ERROR ON WHICH RECOVERY IS IN PROGRESS
(1)	025652	000054	SKCNT1	;SEEK COUNT LOW ORDER
(1)	025654	000056	PRFLGS	;PROGRAM INTERNAL FLAGS
(1)	025656	000060	RXRFR3	;READ COUNT THIRD
(1)	025660	000062	WXRFR3	;WRITE COUNT THIRD
(1)	025662	000064	LSTDA	;DISK ADDRESS OF SOFT ERROR
(1)	025664	000066	DIFWD	;LAST DIFFERENCE WORD OF SEEK
(1)	025666	000070	DPHOUR	;TIME DRIVE WAS DROPPED
(1)	025670	000072	TRERR	;TRACKING ERROR COUNT
(1)	025674	000074	DATCER	
(1)	025676	000076	DOWNCK	
(1)	025700	000100	SERNM1	;WRITE CHECK NECESSARY
(1)			SERNM2	;SERIAL NUMBER OF CARTRIDGE
(1)			SERNM2	;SERIAL NUMBER OF CARTRIDGE

(1)	025702	000104	DCS	;CSR ADDRESS
(1)	025704	000106	DRSEL	;DRIVE SELECT BITS(8,9,10)
(1)	025706	000110	RRR	;PRESENT BUS ADDRESS CONTENTS
(1)	025712	000114	BSECPT	;POINTER TO RAD SECTOR FILE
(1)	025714	000116	RSECK	
(1)	025716	000120	SOFTCS	;CSR AT TIME OF SOFT ERROR
(1)			WRTPG	;WRITE IN PROGRESS DURING PWR FAIL
(1)			PRPOS	;PRESENT POSITION ON DISK
(1)	025722	000000	SKCNT	;SEEK OPERATION COUNT
(1)	025724	000002	RXRFR1	;READ OPERATION COUNT (BITS) LOW ORDER
(1)	025726	000004	RXRFR2	; " " " " HIGH ORDER
(1)	025730	000006	WXRFR1	;WRITE OPERATION COUNT (BITS) LOW ORDER
(1)	025732	000008	WXRFR2	; " " " " HIGH ORDER
(1)	025734	000010	ERRCNT	;ERROR COUNT - HARD
(1)	025736	000012	SFTCNT	;ERROR COUNT - SOFT
(1)	025740	000014	SEKCNT	;SEEK ERROR COUNT
(1)	025742	000020	DERCNT	;DRIVE ERROR COUNT
(1)	025744	000022	DCRCER	;DATA CRC ERROR COUNT
(1)	025746	000024	HRCER	;HEADER CRC ERROR COUNT
(1)	025750	000026	DLCNT	;DATA LATE ERROR COUNT
(1)	025752	000030	OPICNT	;OPERATION INCOMPLETE ERROR COUNT
(1)	025754	000032	HNFERR	;HEADER NOT FOUND ERROR COUNT
(1)	025756	000034	NXMCNT	;NON EXISTANT MEMORY ERROR COUNT
(1)	025760	000036	RETRY	;PRESENT RETRY NUMBER
(1)	025762	000040	RDA	; " DISK ADDRESS CONTENTS
(1)	025764	000042	BMP	;PRESENT MULTIPURPOSE CONTENTS
(1)	025766	000044	FUNC	;LAST FUNCTION LOADED
(1)	025770	000046	BCSADR	;CSR IMAGE OF LAST COMMAND
(1)	025772	000050	LSTHDR	;LAST POSITION ON DISK
(1)	025774	000052	RTYPE	;ERROR ON WHICH RECOVERY IS IN PROGRESS
(1)	025776	000054	SKCNT1	;SEEK COUNT LOW ORDER
(1)	026000	000056	PRFLGS	;PROGRAM INTERNAL FLAGS
(1)	026002	000060	RXRFR3	;READ COUNT THIRD
(1)	026004	000062	WXRFR3	;WRITE COUNT THIRD
(1)	026006	000064	LSTDA	;DISK ADDRESS OF SOFT ERROR
(1)	026010	000066	DIFWD	;LAST DIFFERENCE WORD OF SEEK
(1)	026014	000070	DPHOUR	;TIME DRIVE WAS DROPPED
(1)	026016	000072	TRERR	;TRACKING ERROR COUNT
(1)	026020	000074	DATCER	
(1)	026022	000076	DOWNCK	
(1)	026024	000100	SERNM1	;WRITE CHECK NECESSARY
(1)	026026	000100	SERNM2	;SERIAL NUMBER OF CARTRIDGE
(1)	026028	000104	DCS	;SERIAL NUMBER OF CARTRIDGE
(1)	026030	000106	DRSEL	;CSR ADDRESS
(1)	026032	000110	RRR	;DRIVE SELECT BITS(8,9,10)
(1)	026034	000112	BSECPT	;PRESENT BUS ADDRESS CONTENTS
(1)	026036	000114	RSECK	;POINTER TO RAD SECTOR FILE
(1)	026040	000116	SOFTCS	
(1)	026042	000120	WRTPG	;CSR AT TIME OF SOFT ERROR
(1)	026044	000122	PRPOS	;WRITE IN PROGRESS DURING PWR FAIL
(1)				;PRESENT POSITION ON DISK
(1)	026046	000000	SKCNT	;SEEK OPERATION COUNT
(1)	026050	000002	RXRFR1	;READ OPERATION COUNT (BITS) LOW ORDER

```
(1) 026052 000004 RXFR2 ; " " " " HIGH ORDER
(1) 026054 000006 WFR2 ;WRITE OPERATION COUNT (BITS) LOW ORDER
(1) 026056 000010 WFR2 ; " " " " HIGH ORDER
(1) 026060 000014 ERRCNT ;ERROR COUNT - HARD
(1) 026062 000018 SFTCNT ;ERROR COUNT - SOFT
(1) 026064 000022 SKCNT ;SEEK ERROR COUNT
(1) 026066 000026 DERCNT ;DRIVE ERROR COUNT
(1) 026070 000030 DCRCER ;DATA CRC ERROR COUNT
(1) 026072 000034 DTLCNT ;HEADER CRC ERROR COUNT
(1) 026074 000038 HRCER ;DATA LATE ERROR COUNT
(1) 026076 000042 DTLCNT ;OPERATION INCOMPLETE ERROR COUNT
(1) 026100 000046 HNFERR ;HEADER NOT FOUND ERROR COUNT
(1) 026102 000050 NREXT ;NON EXISTANT MEMORY ERROR COUNT
(1) 026104 000054 RETRY ;PRESENT RETRY NUMBER
(1) 026106 000058 BDA ; " DISK ADDRESS CONTENTS
(1) 026110 000062 BMP ;PRESENT MULTIPURPOSE CONTENTS
(1) 026112 000066 FUNC ;LAST FUNCTION LOADED
(1) 026114 000070 RCSADR ;CSR IMAGE OF LAST COMMAND
(1) 026116 000074 LSTHDR ;LAST POSITION ON DISK
(1) 026118 000078 SKCNT1 ;ERROR ON WHICH RECOVERY IS IN PROGRESS
(1) 026120 000082 PRFLCS ;SEEK COUNT LOW ORDER
(1) 026122 000086 RXFR3 ;PROGRAM INTERNAL FLAGS
(1) 026124 000090 WFR3 ;READ COUNT THIRD
(1) 026126 000094 WFR3 ;WRITE COUNT THIRD
(1) 026130 000098 LSTDA ;DISK ADDRESS OF SOFT ERROR
(1) 026132 000102 DIFWD ;LAST DIFFERENCE WORD OF SEEK
(1) 026134 000106 DPHOUR ;TIME DRIVE WAS DROPPED
(1) 026136 000110 TRERR ;TRACKING ERROR COUNT
(1) 026140 000114 DATCER ;
(1) 026142 000118 DWCK ;WRITE CHECK NECESSARY
(1) 026144 000122 SEPNM1 ;SERIAL NUMBER OF CARTRIDGE
(1) 026146 000126 SEPNM2 ;SERIAL NUMBER OF CARTRIDGE
(1) 026150 000130 DCS ;CSR ADDRESS
(1) 026154 000134 DSEL ;DRIVE SELECT BITS(8,9,10)
(1) 026156 000138 RBA ;PRESENT BUS ADDRESS CONTENTS
(1) 026160 000142 BSECT ;POINTER TO BAD SECTOR FILE
(1) 026164 000146 RSEEK ;
(1) 026168 000150 SOFTCS ;CSR AT TIME OF SOFT ERROR
(1) 026170 000154 WPIP3 ;WRITE IN PROGRESS DURING PWR FAIL
(1) 026172 000158 PRPOS ;PRESENT POSITION ON DISK

(1) 026172 000000 SKCNT ;SEEK OPERATION COUNT
(1) 026174 000004 RXFR1 ;READ OPERATION COUNT (BITS) LOW ORDER
(1) 026176 000008 RXFR1 ; " " " " HIGH ORDER
(1) 026200 000012 WFR1 ;WRITE OPERATION COUNT (BITS) HIGH ORDER
(1) 026202 000016 WFR1 ; " " " " HIGH ORDER
(1) 026204 000020 ERRCNT ;ERROR COUNT - HARD
(1) 026206 000024 SFTCNT ;ERROR COUNT - SOFT
(1) 026210 000028 SKCNT ;SEEK ERROR COUNT
(1) 026212 000032 DERCNT ;DRIVE ERROR COUNT
(1) 026214 000036 DCRCER ;DATA CRC ERROR COUNT
(1) 026216 000040 HRCER ;HEADER CRC ERROR COUNT
(1) 026220 000044 DTLCNT ;DATA LATE ERROR COUNT
(1) 026222 000048 DTLCNT ;OPERATION INCOMPLETE ERROR COUNT
(1) 026224 000052 HNFERR ;HEADER NOT FOUND ERROR COUNT
```

```
(1) 026226 000034 NREXT ;NON EXISTANT MEMORY ERROR COUNT
(1) 026230 000038 RETRY ;PRESENT RETRY NUMBER
(1) 026232 000042 RBA ; " DISK ADDRESS CONTENTS
(1) 026234 000046 BMP ;PRESENT MULTIPURPOSE CONTENTS
(1) 026240 000050 FUNC ;LAST FUNCTION LOADED
(1) 026242 000054 RCSADR ;CSR IMAGE OF LAST COMMAND
(1) 026244 000058 LSTHDR ;LAST POSITION ON DISK
(1) 026246 000062 SKCNT1 ;ERROR ON WHICH RECOVERY IS IN PROGRESS
(1) 026250 000066 PRFLCS ;SEEK COUNT LOW ORDER
(1) 026252 000070 RXFR3 ;PROGRAM INTERNAL FLAGS
(1) 026254 000074 WFR3 ;READ COUNT THIRD
(1) 026256 000078 WFR3 ;WRITE COUNT THIRD
(1) 026260 000082 LSTDA ;DISK ADDRESS OF SOFT ERROR
(1) 026262 000086 DIFWD ;LAST DIFFERENCE WORD OF SEEK
(1) 026264 000090 DPHOUR ;TIME DRIVE WAS DROPPED
(1) 026266 000094 TRERR ;TRACKING ERROR COUNT
(1) 026270 000098 DATCER ;
(1) 026272 000102 DWCK ;WRITE CHECK NECESSARY
(1) 026274 000106 SEPNM1 ;SERIAL NUMBER OF CARTRIDGE
(1) 026276 000110 SEPNM2 ;SERIAL NUMBER OF CARTRIDGE
(1) 026300 000114 DCS ;CSR ADDRESS
(1) 026302 000118 DSEL ;DRIVE SELECT BITS(8,9,10)
(1) 026304 000122 RBA ;PRESENT BUS ADDRESS CONTENTS
(1) 026310 000126 BSECT ;POINTER TO BAD SECTOR FILE
(1) 026312 000130 RSEEK ;
(1) 026314 000134 SOFTCS ;CSR AT TIME OF SOFT ERROR
(1) 026316 000138 WPIP3 ;WRITE IN PROGRESS DURING PWR FAIL
(1) 026318 000142 PRPOS ;PRESENT POSITION ON DISK

3031 .NLIST ME
3032 ENDBUF: .WORD 0
3033
3034 026320 BGNMOD HRDPPM
3035
3036 026320 BGNHRD
3037 .WORD L1024-LSHARD/2
3038
3039 026322 GPRML CNTYPE,CNT,1,YES
3040 (4) 026324 .WORD TSCODE
(4) 026326 .WORD CNTYPE
(4) 026328 .WORD
3041 (4) 026330 GPRMA CSRMSG,CSR,0,160000,177776,YES
(4) 026332 .WORD TSCODE
(4) 026334 .WORD CSRMSG
3042 (4) 026336 .WORD TSLOLIM
(4) 026340 .WORD TSHILIM
(4) 026342 GPRMA VECMSG,VECT,0,0,776,YES
(4) 026344 .WORD TSCODE
(4) 026346 .WORD VECMSG
(4) 026348 .WORD TSLOLIM
3043 (4) 026350 .WORD TSHILIM
(4) 026352 GPRPD BRMSG,PRIOR,0,340,C,7,YES
(4) 026354 .WORD TSCODE
(4) 026356 .WORD BRMSG
```

```

(4) 026354 000340 .WORD 340
(4) 026356 000000 .WORD TSLOLIM
(4) 026360 000007 .WORD TSHILIM
3044 026362 033032 GPRMD DRPT,D,03400,0,7,YES
(4) 026364 026435 .WORD TSCODE
(4) 026366 000000 .WORD DRMSG
(4) 026370 000000 .WORD C3400
(4) 026372 000007 .WORD TSLOLIM
(4) 026372 000007 .WORD TSHILIM
3045
3046 026374 ENDRD
(2) .EVEN
(3) 026374 L10024:
3047
3048
3049 026374 046122 030461 000 CNTYPE: .ASCIZ /RL1/
3050 026401 102 051525 040440 CSRMSG: .ASCIZ /BUS ADDRESS/
3051 026415 102 020122 042514 BRMSG: .ASCIZ /RR LEVEL/
3052 026426 042526 052103 051117 VECMSG: .ASCIZ /VECTOR/
3053 026435 104 044522 042526 DRMSG: .ASCIZ /DRIVE/
3054
3055
3056 026444 .EVEN
3057
3058 026444 ENDMOD
3059 026444 BGNMOD SFTPPM
3060
3061 026444 BGNST
(3) 026444 000217 .WORD L10025-LSSOFT/2
3062
3063 026446 GPRMD RTMSG,RLT,D,177777,0,177777,YES
(4) 026446 000052 .WORD TSCODE
(4) 026450 027311 .WORD RTMSG
(4) 026452 177777 .WORD SFTMSG
(4) 026454 000000 .WORD TSLOLIM
(4) 026456 177777 .WORD TSHILIM
3071 026460 GPRMD SRTMSG,SRLT,D,177777,0,177777,YES
(4) 026462 027134 .WORD TSCODE
(4) 026464 177777 .WORD SFTMSG
(4) 026466 000000 .WORD 177777
(4) 026470 177777 .WORD TSLOLIM
3072 026472 GPRMD FCHK, DCKFG,1,YES
(4) 026474 020130 .WORD TSCODE
(4) 026476 027616 .WORD TSHILIM
(4) 026500 000001 .WORD FCHK
3073 026500 XFER 1
(5) 026500 006044 .WORD TSCODE
3074 026502 GPRMD CHKMT,CLMT,D,177777,0,128.,YES
(4) 026504 032052 .WORD TSCODE
(4) 026506 177777 .WORD SFTMSG
(4) 026510 000000 .WORD 177777
(4) 026514 000200 .WORD TSLOLIM
3075 026514 000200 5S: GPRMD TSHILIM
IMSG,RYT,D,177777,1,177777,YES
    
```

```

(4) 026514 005052 .WORD TSCODE
(4) 026520 027477 .WORD INMSG
(4) 026522 000001 .WORD 177777
3076 026524 177777 .WORD TSLOLIM
(4) 026526 021130 .WORD TSHILIM
(4) 026530 027677 GPRMD DRPMS,DRFLG,1,YES
(4) 026532 000001 .WORD TSCODE
(4) 026534 000001 .WORD DRPMS
3077 026536 XFER 3
(5) 026538 032044 .WORD TSCODE
3078 026540 GPRMD DRMSG,ELT,D,177777,0,177777,YES
(4) 026542 001052 .WORD TSCODE
(4) 026544 177777 .WORD DRMSG
(4) 026546 000000 .WORD 177777
(4) 026548 177777 .WORD TSLOLIM
3079 026550 GPRMD SFTMSG,SEL,D,177777,0,177777,YES
(4) 026552 023052 .WORD TSCODE
(4) 026554 027541 .WORD SFTMSG
(4) 026556 177777 .WORD 177777
(4) 026558 000000 .WORD TSLOLIM
3080 026560 199900 .WORD TSHILIM
(4) 026562 036052 GPRMD DRPMS,DCD,D,177777,0,177777,YES
(4) 026564 027777 .WORD TSCODE
(4) 026566 177777 .WORD DRPMS
(4) 026568 000000 .WORD 177777
(4) 026570 000000 .WORD TSLOLIM
(4) 026572 177777 .WORD TSHILIM
3081 026574 GPRMD SEMSG,SET,D,177777,0,177777,YES
(4) 026576 024323 .WORD TSCODE
(4) 026578 177777 .WORD SEMSG
(4) 026600 177777 .WORD 177777
(4) 026602 000000 .WORD TSLOLIM
(4) 026604 177777 .WORD TSHILIM
3082 026606 GPRMD DRMSG,DET,D,177777,0,177777,YES
(4) 026608 025052 .WORD TSCODE
(4) 026610 027326 .WORD DRMSG
(4) 026612 177777 .WORD 177777
(4) 026614 000000 .WORD TSLOLIM
(4) 026616 177777 .WORD TSHILIM
3083 026620 024130 3S: GPRMD STLMT,OPFLG,1,YES
(4) 026622 027642 .WORD TSCODE
(4) 026624 000001 .WORD STLMT
(4) 026626 013044 .WORD 1
(5) 026630 XFER 2
3085 026630 GPRMD DRMSG,DAT,D,177777,1,177776,YES
(4) 026632 003052 .WORD TSCODE
(4) 026634 177354 .WORD DRMSG
(4) 026636 000001 .WORD DAMSG
(4) 026640 177777 .WORD 177777
(4) 026642 177777 .WORD TSLOLIM
(4) 026644 004052 GPRMD SKMSG,SKT,D,177777,1,177776,YES
(4) 026644 027401 .WORD TSCODE
(4) 026644 027401 .WORD SKMSG
    
```



```

(4) 026646 177777 .WORD 177777
(4) 026650 000001 .WORD TSLOLIM
(4) 026652 177776 .WORD TSHILIM
3087 026654 010130 25: GPRML CHANGE,CHFLG,1,YES
(4) 026656 027451 .WORD TSCODE
(4) 026660 000001 .WORD CHANGE
3088 026662 106044 XFFFF 15
(5) 026664 106044 .WORD TSCODE
3089 026664 034130 GPRML STIPMS,STIP,1,YES
(4) 026666 027104 .WORD TSCODE
(4) 026670 000001 .WORD STIPMS
3090 026672 XFFFF 65
(5) 026674 013044 .WORD TSCODE
3091 026674 011052 GPRML MXBUF,MXB,D,177777,3,5120.,YES
(4) 026676 027505 .WORD TSCODE
(4) 026700 177777 .WORD MXBUF
(4) 026702 000003 .WORD 177777
(4) 026704 012000 .WORD TSLOLIM
3092 026706 GPRML MNRRUF,MNR,D,177777,3.,5120.,YES
(4) 026708 022252 .WORD TSCODE
(4) 026710 027516 .WORD MNRRUF
(4) 026712 177777 .WORD 177777
(4) 026714 000003 .WORD TSLOLIM
(4) 026716 012000 .WORD TSHILIM
3093 026720 026130 65: GPRML RDONLV,ROF,1,YES
(4) 026722 027173 .WORD TSCODE
(4) 026724 000001 .WORD RDONLV
3094 026726 027130 GPRML RANPAT,RAN,1,YES
(4) 026728 027203 .WORD TSCODE
(4) 026730 000001 .WORD RANPAT
3095 026732 XFFFF 45
(5) 026734 006024 .WORD TSCODE
3096 026736 GPRML ONLONE,PAT,0,17,0,7,YES
(4) 026738 030032 .WORD TSCODE
(4) 026740 027213 .WORD ONLONE
(4) 026742 000007 .WORD 17
(4) 026744 000000 .WORD TSLOLIM
(4) 026746 000007 .WORD TSHILIM
3097 026750 035130 45: GPRML WCKMSG,WCK,1,YES
(4) 026752 027733 .WORD TSCODE
(4) 026754 000001 .WORD WCKMSG
3098 026756 006052 75: GPRML CMMSG,ROD,D,177777,0,128.,YES
(4) 026758 027776 .WORD TSCODE
(4) 026760 027776 .WORD CMMSG
(4) 026762 177777 .WORD 177777
(4) 026764 000000 .WORD TSLOLIM
(4) 026766 000200 .WORD TSHILIM
3099 026770 GPRML DEMSG,DDT,D,177777,0,175,YES
(4) 026772 007052 .WORD TSCODE
(4) 026774 027255 .WORD DEMSG
    
```

```

(4) 026774 177777 .WORD 177777
(4) 026776 000100 .WORD TSLOLIM
(4) 027000 000175 .WORD TSHILIM
3100 027002 012052 GPRML WHD,MXH,D,100,0,1,YES
(4) 027004 027527 .WORD TSCODE
(4) 027006 000100 .WORD WHD
(4) 027010 000000 .WORD 100
(4) 027012 000001 .WORD TSLOLIM
3101 027014 013052 GPRML MNHD,MNH,D,100,0,1,YES
(4) 027016 027536 .WORD TSCODE
(4) 027020 000100 .WORD MNHD
(4) 027022 000000 .WORD 100
(4) 027024 000001 .WORD TSLOLIM
3102 027026 GPRML MNCVLM,MNC,D,77600,0,255.,YES
(4) 027028 014052 .WORD TSCODE
(4) 027030 027545 .WORD MNCVLM
(4) 027032 000000 .WORD 77600
(4) 027034 000000 .WORD TSLOLIM
(4) 027036 000377 .WORD TSHILIM
3103 027040 GPRML MNCVLM,MNC,D,77600,0,255.,YES
(4) 027042 015052 .WORD TSCODE
(4) 027044 027555 .WORD MNCVLM
(4) 027046 000000 .WORD 77600
(4) 027048 000377 .WORD TSLOLIM
3104 027052 GPRML MNS,D,77,0,39.,YES
(4) 027054 016052 .WORD TSCODE
(4) 027056 000377 .WORD MNS
(4) 027060 000000 .WORD 77
(4) 027062 000047 .WORD TSHILIM
3105 027064 GPRML MNS,D,77,0,39.,YES
(4) 027066 017052 .WORD TSCODE
(4) 027070 000377 .WORD MNS
(4) 027072 000000 .WORD 77
(4) 027074 000047 .WORD TSLOLIM
3106 027076 GPRML AUTOMS,AUTO,1,YES
(4) 027100 033130 .WORD TSCODE
(4) 027102 000001 .WORD AUTOMS
3108 027104 .WORD 1
3109 (2) 027104 L10025: ENDSFT
(3) 027104 .EVEN
3110
3111
3112
3116 027104 052123 050111 046125 STIPMS: .ASCIZ *STIPULATE R/W XFER SIZE*
3117 027134 042523 045505 051040 SRTMSG: .ASCIZ /SEFK RETRY LMT/
3118 027134 743 047480 049106 CHKLMT: .ASCIZ /# OF ERR DUMPED/
3119 027134 020104 049117 RDONLV: .ASCIZ /RD ONLY/
3120 027203 122 047101 050040 RANPAT: .ASCIZ /RAN PAT/
    
```

```

ASSEMBLY ROUTINES          MACY11 30A(1052) 30-NOV-78 18:42 PAGE 7-12
CZRLEB.P11                18:28          DRIVE INFORMATION BUFFERS
3121 027223 127 044510 044103 ONLONE: .ASCIZ /WHICH ONE/
3122 027225 120 042122 042440 BRMSG: .ASCIZ /HRD ERR LMT/
3123 027231 123 052106 042440 SFTMSG: .ASCIZ /SFT ERR LMT/
3124 027255 043 047440 020106 DEMSG: .ASCIZ /# OF DATA ERR RPT'D PER BUF/
3125 027311 122 052105 054522 RTMSG: .ASCIZ /RETRY LMT/
3126 027333 124 026103 051102 SEMSG: .ASCIZ /SK ERR LMT/
3127 027336 051104 042440 051122 DRMSG: .ASCIZ /DR ERR LMT/
3128 027351 104 052101 020101 DAMSG: .ASCIZ /DATA XFER LMT (*10(10))/
3129 027401 123 020113 046514 SKMSG: .ASCIZ /SK LMT (*10(3))/
3130 027421 124 046511 040102 LNMMSG: .ASCIZ /TIME BETW REPORTS (MIN)/
3131 027451 103 040510 043516 CHANGE: .ASCIZ *CHANGE SEEK, R/W PARAMETERS*
3132 027505 115 054101 054040 MYBUF: .ASCIZ /MAX XFER/
3133 027516 044515 020116 043130 MYRBUF: .ASCIZ /MIN XFER/
3134 027537 115 054101 043630 MYHD: .ASCIZ /MAX HD/
3135 027536 044515 020116 042110 MYNHD: .ASCIZ /MIN HD/
3136 027545 115 054101 041440 MYCVL: .ASCIZ /MAX CVL/
3137 027555 115 047111 041440 MYNVL: .ASCIZ /MIN CVL/
3138 027555 115 054101 051440 MYSEC: .ASCIZ /MAX SEC/
3139 027575 115 047111 051440 MYNSEC: .ASCIZ /MIN SEC/
3140 027605 103 045510 042040 AUTOMS: .ASCIZ /CHK DRDY/
3141 027616 040504 040524 042040 FDCHK: .ASCIZ /DATA DWP ON DCK ERR/
3142 027642 051104 050117 040040 STLM: .ASCIZ /DROP DR ON OPER LMTS REACHED/
3143 027677 104 047522 020120 DRPMS: .ASCIZ /DROP DR ON ERR LMTS REACHED/
3144 027733 127 020122 044103 WCKMSG: .ASCIZ /WR CHK/
3145 027742 040504 040524 046446 DRPMS: .ASCIZ /DATA MISCOMPARE LIMIT/
3146 027770 047522 042122 020123 CMMSG: .ASCIZ /WORDS PER SECTOR COMPARED ON READ/
3147
3148 .EVEN
3149
3150
3151
3152 030032 .EVENMOD
3153
3154
3155
3156
3157
3158 030132 .=30132
3159
3160 ;AREA RESERVED AS PATCH AREA FOR DIAGNOSTICS
3161 ;=30132 WAS SELECTED AS "LASTAD" TO PROVIDE APT TO LSI-11 COMPATIBILITY.
3162 ;BIT 7 OF "LASTAD" MUST BE CLEARED TO ACHIEVE A VALID MAILBOX ADDRESS
3163 ;WHEN RUNNING ON THE LSI-11 UNDER APT.
3164
3165 030132 LASTAD
3166 030132 L$LAST: .EVEN
3167
3168

```

SEQ 0114

```

ASSEMBLY ROUTINES          MACY11 30A(1052) 30-NOV-78 18:42 PAGE 8
CZRLEB.SUP                09:53          DIAGNOSTIC SUPERVISOR -- LOW CORE SET UP
3170
3171 060726 000000 .SPTTL  DIAGNOSTIC SUPERVISOR -- LOW CORE SET UP
3172 060730 000000 .WORD  0 ;SPACE FOR USER POOL POINTER
3173 060732 000000 .WORD  0 ;SIZE
3174 060734 000000 .WORD  0 ;CHECKSUM (NOT CURRENTLY USED)
3175 060740 000000 .WORD  0 ;SIZE OF H.W. PTAB. ALLOCATION
3176 .END.SUPV=+2
3177 .END 260
3178

```

SEQ 0115





