

RL11,RLV11

PERFORMANCE EXERCISER
CZRLEB0

AH-E247B-MC

COPYRIGHT © 77-78

FICHE 1 OF 1

JAN 1979

digital

MADE IN USA

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

IDENTIFICATION

B 1

SEQ 0001

PRODUCT CODE: AC-E246B-MC
PRODUCT NAME: CZRLEBO RL01/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 B>). 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-DZRLE-A)

1.3 RELATED DOCUMENTS AND STANDARDS

RL01 USERS MANUAL (EK-RI 01-UG-PRE)
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)
CVRLAA0	RLV11 RL01 DISKLESS TEST (RLV11 ONLY)
CZRLCBO	RL01 DRIVE TEST (PART 1)
CZRLDB0	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 'HARDWARE 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:

	BY WHOM ENTERED:
.R DZRKXX	O
DZRKXX	D
L-CLK (L) N ? Y	D,O
50HZ (L) N ?	D
LSI (L) N ?	D
LPT (L) N ?	D
MEM (K) (D) 16 ?	D
DS-B>STA/PASS:1/FLAGS:HOE	D,O
# UNITS (D) ? 2	D,O
UNIT 1	D
CSR (O) ?	D,O
VECTOR (O) ?	D,O
BR LEVEL (O) ?	D,O
DRIVE (O) ? 0	D,O
UNIT 2	D
CSR (O) ?	D,O
VECTOR (O) ?	D,O
BR LEVEL (O) ?	D,O
DRIVE (O) ? 1	D,O
CHANGE SW (L) ? N	D,O
DZRKXX HARD ERR 00004 TST 003 SUB 002 PC:004130	D
ERR HLT	D
DS-B>PRO/FLAGS:IER:LOE:HOE 0	D,O

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

^C	O
DS-B>CON/FLAGS:HOE:IER:LOE 0	D,O
CHANGE SW (L) ? N	D,O
DZRKXX EOP 1	D
DS-B>RESTART/PASS:1	D,O
CHANGE SW (L) ? N	D,O

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 'BIC'. 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 'BIC'.

HERE IS A SAMPLE DIALOGUE TO ACCOMPLISH THIS:

```
.R UPD2
RESTART: XXXXXX
*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 XXXXXX*
*****
```

```
*HICORE 60632
CORE: 0,60632
*DUMP DK0: DIAG.BIC
```

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

```
.R DIAG.BIC
```

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 'R/JN 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/EOP:EOP-INCR
*****
```

THE DIAGNOSTIC IN CORE IS EXECUTED IN ACCORDANCE WITH THE SWITCHES SPECIFIED. THE MESSAGE '# UNITS?' IS PRINTED. THE START COMMAND MAY BE ISSUED WHEN DIAGNOSTIC COMMAND MODE HAS BEEN ENTERED VIA ONE OF THE FOLLOWING: A) OPERATOR TYPED 'RUN DIAGNOSTIC' B) DIAGNOSTIC FINISHED EXECUTING C) ERROR WAS ENCOUNTERED WITH HOE FLAG SET D) OPERATOR ENTERED CONTROL/C.

AFTER THE OPERATOR RESPONDS TO '# UNITS?', THE HARDWARE DIALOGUE IS INITIATED. WHEN IT IS COMPLETED, THE QUESTIONS 'CHANGE SW?' IS ISSUED, AND THE ANSWERS, IF GIVEN, BECOME THE NEW DEFAULTS. THEREFORE IT IS NECESSARY TO RELOAD THE PROGRAM IN ORDER TO RETURN TO THE LOAD DEFAULTS.

THE SWITCH ARGUMENTS ARE AS FOLLOWS:

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

'PASS-CNT' IS A DECIMAL NUMBER INDICATING THE DESIRED NUMBER OF PASSES. A PASS IS DEFINED AS THE EXECUTION OF THE FULL DIAGNOSTIC (ALL SELECTED TESTS) AGAINST ALL UNITS SUBMITTED. THE DEFAULT IS NON-ENDING 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 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. DEFALT FOR PASS-CNT IS THE UNSATISFIED PASS-CN* 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 '.R DIAG'. THE COMMAND PROMPT 'DS-B>' 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 ISR (INHIBIT STATISTICAL REPORTING) FLAG IS CLEARED.

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

THE HARDWARE P-TABLES FOR ALL UNITS UNDER TEST ARE PRINTED OUT IN THE FORMAT IN WHICH THEY WERE ENTERED. ANY UNITS THAT WERE DROPPED BY THE OPERATOR 'DRGP' 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 YFS 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/WITTEN (*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 ARE COMPARED. THE FEWER WORDS COMPARED IN CORE ON EACH READ THE FASTER THROUGHPUT THE EXERCISER WILL HAVE.

LIMITS 0 - 128

OF DATA ERR RPT'D PER BUF 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

RLO1K 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 FORM THAT SECTOR.

3.2 ERROR HALTS

ERROR HALTS ARE SUPPORTED PER DESCRIBED IN THE PREVIOUS SECTION WITH /FLAG:HOF. 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 RUNNING OR DROPPED DH:DM
PACK SERIAL #: DDDDDDDDD
SEEKS: I IIII
BITS READ: JJJJJJJJJ (*16)
BITS WRITTEN: KKKKKKKKK (*16)

ERRORS
DRIVE: N SEEK: N TRACK: N DATA: N
HARD: N SOFT: N
DCK: N HCRC: N NXM: N HNF: N
DLT: N OPI: N

WHERE:

HH IS HOURS SINCE START/RESTART
MM IS MINUTES SINCE START/RESTART
SS IS SECONDS SINCE START/RESTART
XXXXXX IS ADDRESS OF CONTROLLER
Y IS DRIVE NUMBER
DH IS HOUR AT WHICH DRIVE WAS DROPPED
DM IS MINUTE AT WHICH DRIVE WAS DROPPED
DDDDDDDDDD - IS 10 DIGIT OCTAL SERIAL NUMBER OF PACK
IIII IS TOTAL NUMBER OF SEEKS SINCE 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/RLV11 CONTROLLER HAS THE FOLLOWING FOUR(4) REGISTERS FOR CONTROL OF THE SUBSYSTEM.

RLCS - CONTROL AND STATUS REGISTER (XXXXX0)

BIT 15 - COMPOSITE ERROR
BIT 14 - DRIVE ERROR
BIT 13 - NON EXISTANT MEMORY ERROR

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

BIT 0 - DRIVE READY

RLBA - BUS ADDRESS REGISTER (XXXXX2)

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

RLDA - DISK ADDRESS REGISTER (XXXXX4)

FOR READ/WRITE FUNCTIONS

BIT 15 - 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 COMPIMENT)

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 29; ELSE STEP 22
22. IF HNF, THEN STEP 30; ELSE STEP 23
23. YOU SHOULD NEVER GET HERE
24. IF DOING READ/WRITE THEN STEP 25 IF DOING READ HEADER THEN STEP 26
25. CHECK IF DA IS BAD SECTOR THEN STEP 4A; ELSE STEP 18.
26. READ 40 HEADERS, IF ALL GOOD THEN STEP 27; ELSE STEP 28
27. REPORT SOFT HEADER CRC; GO TO 4A
28. FIGURE OUT BAD HEADER IF IN FILE THEN STEP 4A; ELSE STEP 18
29. CHECK IF DA-1 IS IN FILE IF THEN STEP 4A; ELSE STEP 18
30. READ HEADER. IF ON CORRECT TRACK THEN STEP 31; ELSE STEP 32
31. CHECK IF DA IS IN FILE IF THEN STEP 4A, ELSE STEP 18
32. REPORT TRACKING; FIX POSITION, GO TO STEP 4
33. ACT UPON:
 - VC
 - SKTO
 - SPE
 - WGE
 - WDE
 - CHE

34. GO TO STEP 4

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
999	PROGRAM MAIN LOOP
1191	ROUTINE TO SETUP AND ISSUE GET STATUS
1199	ROUTINE TO SETUP AND ISSUE SEEK FUNCTION
1276	ROUTINE TO LOAD READ HEADER AND ISSUE IT.
1284	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
1993	RETRY LIMIT ROUTINE
2004	LIST OF FUNCTION ROUTINES
2017	BAD SECTOR FILE ROUTINE
2134	ROUTINE TO DROP DRIVE
2178	ROUTINE TO CHECK DATA
2260	ROUTINE TO WAIT FOR CONTROLLER READY
2283	GET STATUS/DRIVE RESET ROUTINE
2304	ROUTINE TO GENERATE A RANDOM NUMBER
2330	ROUTINE TO WRITE PACKS INITIALLY
2515	ROUTINE FOR SYSTEM CLOCK
2544	HEADS HOME ROUTINE
2565	RANDOM WC AND DA ROUTINE
2655	ROUTINE TO DUMP BUFFER ON DCK
2780	ROUTINE TO CHECK FOR BAD SECTOR
2977	DRIVE INFORMATION BUFFERS
370	DIAGNOSTIC SUPERVISOR -- LOW CORE SET JP

1			.ENABLE AMA
2			.ENABLE ABS
3			.NLIST ME,CND,MD
4			
5	002000		.=2000
6			
7			
9			
10	002000		SVC
11		000000	SVCINS=0
12		000000	SVCTAG=0
13			
14			
15			
16	002000		POINTER ALL
17			
18			
19	002000		BGNMOD MDHEDR
20	002000		HEADER CZRLE,B,0,0,0,0,RL01,1
(4)	002000	103	.ASCII /C/
(4)	002001	132	.ASCII /Z/
(4)	002002	122	.ASCII /R/
(4)	002003	114	.ASCII /L/
(4)	002004	105	.ASCII /E/
(6)	002005	000	.BYTE 0
(6)	002006	000	.BYTE 0
(5)	002007	000	.BYTE 0
(4)	002010	102	.ASCII /B/
(4)	002011	060	.ASCII /O/
(4)	002012	000000	.WORD 0
(4)	002014	000000	.WORD 0
(4)	002016	026322	.WORD L\$HARD
(4)	002020	026446	.WORD L\$SOFT
(4)	002022	007470	.WORD L\$HW
(4)	002024	007504	.WORD L\$SW
(4)	002026	030132	.WORD L\$LAST
(4)	002030	000000	.WORD 0
(4)	002032	000000	.WORD 0
(4)	002034	000001	.WORD 1
(4)	002036	000000	.WORD 0
(4)	002040	007604	.WORD L\$DISPATCH
(4)	002042	000000	.WORD 0
(4)	002044	000000	.WORD 0
(4)	002046	000000	.WORD 0
(4)	002050	002	.BYTE C\$REVISION
(3)	002051	002	.BYTE C\$FDIT
(4)	002052	000000	.WORD 0
(4)	002054	000000	.WORD 0
(4)	002056	000000	.WORD 0
(5)	002060	000000	.WORD 0
(4)	002062	000000	.WORD 0
(4)	002064	002114	.WORD L\$DVTYP
(4)	002066	007606	.WORD L\$RPT
(4)	002070	002112	.WORD L\$DR
(4)	002072	002112	.WORD L\$DRST
(4)	002074	011236	.WORD L\$AU

```

(4) 002076 011322      .WORD  LSDU
(5) 002100 000014      .WORD  14
(4) 002102 000000      .WORD  0
(4) 002104 007664      .WORD  L$INIT
(4) 002106 011072      .WORD  L$CLEAN
21
22 002110                ENDMOD
23
24
25
26 002110                DEVREG
(5) 002110 000000      .WORD  0
(2) 002112 000001      .BLKW
27
28 002114                DEVTYP  <RL01>
(3) 002114 046122 030460 000  .ASCIZ  /RL01/
(2) 002114 002122
29
30
31      .SBTTL  BIT AND OFFSET DEFINITIONS
32
33      ;DEFINITIONS
34
35
36 002122                BGNMOD  GLBEQAT
37
38 002122                EQUALS
39
40      000000            CS=0                ;CONTROL AND STATUS OFFSET
41      000002            BA=2                ;BUSADDRESS OFFSET
42      000004            DA=4                ;DISK ADDRESS OFFSET
43      000006            MP=6                ;MULTI PURPOSE OFFSET
44      ;CONSTANT OFFSETS FOR INDIVIDUAL DRIVE BUFFERS
45      ;THE ONLY POSITION THAT IS CRITICAL IS THAT OF
46      ;"PRPOS" IT MUST (MUST) BE THE LAST ENTRY OF THE BUFFER
47
48      000000            SKCNT=0              ;SEEK OPERATION COUNT
49      000002            RXFR1=2              ;READ OPERATION COUNT (BITS) LOW ORDER
50      000004            RXFR2=4              ;HIGH ORDER
51      000006            WXFR1=6              ;WRITE OPERATION COUNT (BITS) LOW ORDER
52      000010            WXFR2=10             ;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            DCRCER=22           ;DATA CRC ERROR COUNT
58      000024            HRCRCER=24          ;HEADER CRC ERROR COUNT
59      000026            DLTCNT=26           ;DATA LATE ERROR COUNT
60      000030            OPICNT=30           ;OPERATION INCOMPLETE ERROR COUNT
61      000032            HNFERR=32           ;HEADER NOT FOUND ERROR COUNT
62      000034            NXMCNT=34           ;NON EXISTANT MEMORY ERROR COUNT
63      000036            RETRY=36            ;PRESENT RETRY NUMBER
64      000040            BDA=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

```

68	000050	LSTHDR=50	:LAST POSITION ON DISK
69	000052	RTYPE=52	:ERROR ON WHICH RECOVERY IS BEING TRIED
70	000054	SKCNT1=54	:LOW SEFK COUNT
71	000056	PRFLGS=56	:INTERNAL FLAGS
72	000060	RXFR3=60	:THIRD ORDER READ COUNT
73	000062	WXFR3=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	000071	DPMIN=71	:MINUTE OF DRIVE DROPPED
78	000072	TRERR=72	:TRACKING ERRORS COUNT
79	000074	DATCER=74	:DATA CMP ERRORS
80	000076	DOWCK=76	:PERFORM WRITE CHECK
81	000100	SERNM1=100	:SERIAL NUMBER OF CARTRIDGE
82	000102	SERNM2=102	:SERIAL NUMBER OF CARTRIDGE
83	000104	DCS=104	:CSR ADDRESS
84	000106	DRSEL=106	:DRIVE SELECT BITS(8,9,10)
85	000110	BBA=110	:PRESENT BUS ADDRESS CONTENTS
86	000112	BSECTP=112	:POINTER TO BAD SECTOR FILE
87	000114	RSEEK=114	:SEEK IN PROCESS OF RECOVERY
88	000116	SOFTCS=116	:CSR OF SOFT ERROR
89	000120	WRIPG=120	:WRITE OPERATION IN PROGRESS AT PWR FAIL TIME
90	000122	PRPOS=122	:PRESENT POSITION ON DISK
91			
92	000001	SKDON=BIT0	
93	000001	DRDY=BIT0	:DRIVE READY
94	000100	INTEN=BIT6	:INTERRUPT ENABLE
95	100000	ERR=BIT15	:COMPOSITE ERROR
96	040000	DERR=BIT14	:DRIVE ERROR
97	100000	WDE=BIT15	:WRITE DATA ERROR
98	040000	HCE=BIT14	:HEAD CURRENT ERROR
99	020000	WL=BIT13	:WRITE LOCK
100	010000	SKTO=BIT12	:SEEK TIMEOUT ERROR
101	004000	SPE=BIT11	:SPINDLE TIMEOUT/UNDER/OVER SPEED
102	002000	WGE=BIT10	:WRITE GATE ERROR
103	001000	VC=BIT9	:VOLUME CHECK
104	000400	DSE=BIT8	:DRIVE SELECT ERROR
105	020000	NXM=BIT13	:NON-EXISTANT MEMORY ERROR
106	010000	DLT=BIT12	:DATA LATE
107	004000	DCRC=BIT11	:DATA CRC ERROR
108	004000	HCRC=BIT11	:HEADER CRC ERROR
109	010000	HNF=BIT12	:HEADER NOT FOUND ERROR
110	002000	OPI=BIT10	:OPERATION INCOMPLETE ERROR
111	000200	CRDY=BIT7	:CONTROLLER READY
112	000040	BA17=BIT5	:EXTENDED BUS ADDRESS BIT 17
113	000020	BA16=BIT4	:EXTENDED BUS ADDRESS BIT 16
114	000002	WRCHK=BIT1	:WRITE CHECK FUNCTION CODE
115	000004	GSTAT=BIT2	:GET DRIVE STATUS FUNCTION CODE
116	000006	SEEK=BIT1!BIT2	:SEEK FUNCTION CODE
117	000010	RDHDR=BIT3	:READ HEADER FUNCTION CODE
118	000012	WRITE=BIT3!BIT1	:WRITE FUNCTION CODE
119	000014	READ=BIT3!BIT2	:READ FUNCTION CODE
120	000013	DRST=BIT3!BIT1.BIT0	:DRIVE RESET COMMAND CODE FOR DRIVE COMMAND WORD
121	000003	GSBIT=BIT1!BIT0	:GET STATUS COMMAND CODE FOR DRIVE COMMAND WORD
122	000001	MK=BIT0	:MARKER BIT FOR DRIVE COMMAND WORD(SEEK,GET STATUS)
123	000004	SIGN=BIT2	:DIRECTION FOR SEEK(0=AWAY FROM SPINDLE)

```
124      000020      SKHS=BIT4      ;HEAD SELECT FOR SEEK
125      000100      HEAD=BIT6      ;HEAD SELECT FOR READ,WRITE,GET STATUS
126
127      ;OFFSET FOR HARDWARE P-TABLE
128
129      000000      CSR=0
130      000002      VECT=2
131      000004      PRIOR=4
132      000006      DRBT=6
133      000010      CNT=10
134
135      ;OFFSET FOR SOFTWARE P-TABLE
136
137      000000      RLT=0
138      000002      ELT=2
139      000004      SET=4
140      000006      DAT=6
141      000010      SKT=10
142      000012      TYT=12
143      000014      RDT=14
144      000016      DDT=16
145      000020      CHFLG=20
146      000022      MXB=22
147      000024      MXH=24
148      000026      MNH=26
149      000030      MXC=30
150      000032      MNC=32
151      000034      MXS=34
152      000036      MNS=36
153      000040      DCKFG=40
154      000042      DRFLG=42
155      000044      MNB=44
156      000046      S L-46
157      000050      OPFLG=50
158      000052      DET=52
159      000054      ROF-54
160      000056      RAN=56
161      000060      PAT 60
162      000062      SRLT=62
163      000064      CLMT=64
164      000066      AUTO=66
165      000070      STIP=70
166      000072      WCK-72
167      000074      DCD=74
168
169
170 002122      ;
171      ;
172      ;
173      ;
174      .SBTTL GLOBAL DATA AND CONSTANTS
175
176 002122      BGNMOD GLBDAT
177
178 002122 000000      RECNT: .WORD 0      ;READ ERROR COUNT
179 002124 000000      RWCNT: .WORD 0      ;R/W ERROR COUNT
```

```

180 002126 000000 WHY: .WORD 0 ;REASON FOR DROPPING DRIVE
181 002130 000 DRUT: .BYTE 0 ;DRIVES UNDER TEST
182 002131 000 DRPRS: .BYTE 0 ;DRIVES PRESENT
183 002132 000000 SYMSK: .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 100177 CYLSK: .WORD 100177 ;MASK FOR CYLINDER ONLY
187 002142 100077 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 ;THE FOLLOWING LOCATIONS ARE CLEARED AS A GROUP (DOWN TO 'STFLG')
192 ;THEREFORE DON'T INSERT ANY CONSTANTS
193
194 002150 174400 CNTLR1: .WORD 174400 ;CSR OF CONTROLLER 1 (LUN 0-3)
195 002152 000000 CNTLR2: .WORD 0 ;CSR OF CONTROLLER 2 (LUN 4-7)
196 002154 000000 LSTDR1: .WORD 0 ;BUFFER POINTER OF DRIVE
197 002156 000000 LSTDR2: .WORD 0 ;BUFFER POINTER OF DRIVE
198 002160 000000 BCSR: .WORD 0 ;CSR FROM P-TABLE
199 002162 000000 BVEC: .WORD 0 ;VECTOR ""
200 002164 000000 BPRIOR: .WORD 0
201 002166 000000 BDRSEL: .WORD 0 ;DRIVE "" ""
202 002170 000000 HDRFND: .WORD 0 ;FLAG TO INDICATE HDR IN BAD LIST
203 002172 000000 CHKSEC: .WORD 0 ;SECTOR OF ERROR - USED BY BAD SECTOR LOCATION
204 002174 000000 DECNT: .WORD 0 ;DATA ERROR COUNT
205 002176 000000 TEMPO: .WORD 0 ;TEMP LOCATION
206 002200 000000 TEMP1: .WORD 0 ;TEMP LOCATION
207 002202 000000 TEMP2: .WORD 0 ;TEMP LOCATION
208 002204 000000 TEMP3: .WORD 0 ;"" ""
209 002206 000000 TEMP4: .WORD 0 ;"" ""
210 002210 000000 TEMP5: .WORD 0 ;"" ""
211 002212 000000 TEMP6: .WORD 0 ;"" ""
212 002214 000000 TEMP7: .WORD 0 ;"" ""
213 002216 000000 TEMP8: .WORD 0 ;"" ""
214 002220 000000 TEMP9: .WORD 0 ;"" ""
215 002222 000160 VECT1: .WORD 160 ;VECTOR OF FIRST CONTROLLER
216 002224 000000 VECT2: .WORD 0 ;VECTOR "" 2ND
217 002226 000000 PRIOR1: .WORD 0
218 002230 000000 PRIOR2: .WORD 0
219 002232 000000 GDDAT: .WORD 0
220 002234 000000 RNTEMP: .WORD 0 ;"" ""
221 002236 000000 INTERVAL: .WORD 0 ;TIME BETWEEN REPORTS
222 002240 000000 LSTTIM: .WORD 0 ;LAST TIME ON SYSTEM CLOCK
223 002242 000000 SECOND: .WORD 0 ;SECONDS OF SYSTEM CLOCK
224 002244 000000 MINUTE: .WORD 0 ;MINUTES OF SYSTEM CLOCK
225 002246 000000 HOUR: .WORD 0 ;HOURS OF SYSTEM CLOCK
226 002250 000000 E.CS: .WORD 0 ;IMAGES OF REGISTERS
227 002252 000000 E.BA: .WORD 0 ;ON INTERRUPT
228 002254 000000 E.DA: .WORD 0
229 002256 000000 E.MP: .WORD 0
230 002260 000000 E.MP1: .WORD 0
231 002262 000000 E.MP2: .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 002270 000000 BUF2: .WORD C ;BUFFER FOR SECOND CONTROLLER
235 002272 000000 MAXWC: .WORD C ;MAX WORD COUNT DETERMINED BY CORE

```



```

236 002274 000000      UUT:      .WORD    0      ;NUMBER OF UNITS ON SYSTEM
237 002276 000000      PWRFLC:  .WORD    0      ;POWER FAIL INDICATOR
238 002300 000000      TRPFLG:  .WORD    0      ;TRAP OCCURANCE FLAG
239 002302 000000      STFLG:   .WORD    0      ;START FLAG
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 002310 000000      FASPNT:  .WORD    0      ;POINTER
246 002312 000000      DWCNT:   .WORD    0      ;ERROR COUNT
247 002314 000000      DWCNT1: .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      .NCALL:  .WORD    0
253
254 002330                      ENDMOD
255
256
257                      .SBTTL  GLOBAL MESSAGES
258
259 002330      BGNMOD  GLBTXT
260
261                      ;GLOBAL TEXT
262
263
264
265
266
267
268 002330 044524 042515 020072  TIME:    .ASCIZ  'TIME: '
269 002337      040 046122 051503  MRLCS:  .ASCIZ  'RLCS: '
270 002347      040 051050 041514  CRLCS:  .ASCIZ  '(RLCS): '
271 002361      040 052506 041516  MFUNC:  .ASCIZ  'FUNCTION: '
272 002375      040 051050 041114  CRLBA:  .ASCIZ  '(RLBA): '
273 002407      040 051050 042114  CRLDA:  .ASCIZ  '(RLDA): '
274 002421      040 051050 046514  CRLMP:  .ASCIZ  '(RLMP): '
275
276 002433      104 043111 053440  DIFMSG:  .ASCIZ  '/DIF WD: /
277 002444 040520 045503 051440  CART:   .ASCIZ  '/PACK SERIAL #: /
278 002464 047516 041440 042122  NOCRDY: .ASCIZ  '/NO CRDY/
279 002474 051104 047040 052117  DNRDY:  .ASCIZ  '/DR NOT RDY/
280 002507      104 020122 047516  NORDY:  .ASCIZ  '%DR NOT RDY W/O DR ERR%
281 002535      102 043525      000  PRGER:  .ASCIZ  '/BUG/
282 002541      111 044516 020124  NWRTS:  .ASCIZ  '/INIT WR OF SEC BAD/
283 002564 051440 041505 047524  SMSG:   .ASCIZ  '/ SECTOR: /
284 002576 047516 043440 047517  EXHAUS: .ASCIZ  '/NO GOOD HDR/
285 002612 047125 044504 043501  UDERR:  .ASCIZ  '/UNDIAGNOSABLE ERR/
286 002634 042523 045505 042440  MSKER:  .ASCIZ  '/SEEK ERR/
287 002645      123 043117 020124  MSFER:  .ASCIZ  '/SOFT ERR ENC'D/
288 002664 051104 042440 051122  DRIVER: .ASCIZ  '/DR ERR/
289 002673      104 020122 051105  MDERS:  .ASCIZ  '/DR ERR WILL NOT RESET/
290 002721      104 020122 052123  MDSER:  .ASCIZ  '/DR STAT ERR/
291 002735      126 046117 041440  MVCER:  .ASCIZ  '/VOL CHK WILL NOT CLR/
292 002762 051127 043440 052101  WGEST:  .ASCIZ  '/WR GATE ERR WILL NOT RESET/
293 003015      104 020122 051105  MRDER:  .ASCIZ  '/D' ERR - RECOVERED/
294 003040 040504 040524 041440  MDCER:  .ASCIZ  '/DATA CMP ERR/

```

295	003055	110	051101	020104	MHDR:	.ASCIZ	/HARD ERROR/
296	003070	040504	040524	042040	DMPDCK:	.ASCIZ	/DATA DUMP - DCK/
297	003110	051124	041501	044513	TRACK:	.ASCIZ	/TRACKING ERR/
298	003125	110	042122	042440	ERLMTM:	.ASCIZ	/HRD ERR LMT EXC'D/
299	003147	123	020113	051105	SERLMT:	.ASCIZ	/SK ERR LMT EXC'D/
300	003170	043123	020124	051105	SFMSG:	.ASCIZ	/SFT ERR LMT EXC'D/
301	003212	040504	040524	042440	DCDMSG:	.ASCIZ	/DATA ERR LMT EXC'D/
302	003235	104	020122	051105	DERMSG:	.ASCIZ	/DR ERR LMT EXC'D/
303	003256	052502	043106	051105	OVER:	.ASCIZ	/BUFFER CHOSEN TOO BIG - WAS /
304	003313	122	050505	041040	REQ:	.ASCIZ	/REQ BY OPR/
305	003326	054105	023510	020104	SEXHAU:	.ASCIZ	/EXH'D RETRY ON SEEK/
306	003352	042110	020123	047516	UNLOAD:	.ASCIZ	/HDS NOT UNLD ON ERR/
307	003376	051104	053440	042114	NOLOAD:	.ASCIZ	/DR WLD NOT LD/
308	003414	050117	051105	046040	SOPLMT:	.ASCIZ	/OPER LMTS EXC'D/
309	003434	040507	041122	046102	NOREV:	.ASCIZ	/GARBLED DATA - CAN'T CHECK IT/
310	003473	115	051117	020105	MBDMSC:	.ASCIZ	/MORE THAN 16 BAD SECTORS/
311	003524	047516	043040	041501	HWSEC:	.ASCIZ	/NO FACTORY FILE/
312	003544	047516	043040	042511	SWSEC:	.ASCIZ	/NO FIELD FILE/
313	003562	026520	040524	046102	MPT:	.ASCIZ	/P-TABLE: /
314	003574	046111	020114	026520	ILLEG:	.ASCIZ	/ILL P-TABLE/
315	003610	053040	041505	047524	MVEC:	.ASCIZ	/ VECTOR: /
316	003622	047516	042040	044522	NODRIV:	.ASCIZ	/NO DRIVES/
317	003634	042040	044522	042526	DRNM:	.ASCIZ	/ DRIVE: /
318	003645	040	051514	020124	LPS:	.ASCIZ	/ LST POS: /
319	003660	042440	050130	050040	EPS:	.ASCIZ	/ EXP POS: /
320	003673	040	042522	020103	RPS:	.ASCIZ	/ REC POS: /
321	003706	051104	042040	042111	NOPIR:	.ASCIZ	/DR DID REC'R FROM PWR UP/
322	003737	101	020124	052502	BUSAD:	.ASCIZ	/AT BUS ADDR: /
323	003755	122	052105	054522	MRT:	.ASCIZ	/RETRY: /
324	003766	042440	051122	051117	ERT:	.ASCIZ	/ ERROR TYPE: /
325	004004	052123	052101	051525	MST:	.ASCIZ	/STATUS WAS: /
326	004021	040	044123	052517	MST1:	.ASCIZ	/ SHOULD BE: /
327	004036	051040	052105	044522	RT1:	.ASCIZ	/ RETRIES ATTEMPTED/
328	004061	040	054105	023520	EXP:	.ASCIZ	/ EXP'D: /
329	004072	051040	041505	042047	RCD:	.ASCIZ	/ REC'D: /
330	004103	104	044522	042526	DROP:	.ASCIZ	/DRIVE DROPPED/
331	004121	040	047110	000106	MTHNF:	.ASCIZ	/ HNF/
332	004126	044040	051103	000103	MTHCRC:	.ASCIZ	/ HCRC/
333	004134	042040	045503	000	MTDCRC:	.ASCIZ	/ DCK/
334	004141	040	046104	000124	MTDLT:	.ASCIZ	/ DLT/
335	004146	047440	044520	000	MTOPI:	.ASCIZ	/ OPI/
336	004153	040	054116	000115	MTNXM:	.ASCIZ	/ NXM/
337	004160	042040	053122	000	MTDRV:	.ASCIZ	/ DRV/
338	004165	124	051505	044524	MSTART:	.ASCIZ	/TESTING STARTED/
339	004205	127	044522	044524	MSWRPK:	.ASCIZ	/WRITING PACK /

340					:		
341					:	THIS LIST OF ASCII TEXT IS USED AS A TABLE FOR PRINTING	
342					:	FUNCTIONS IN ERROR MESSAGES TABLE IS 'MTCR - MTRD'	
343					:	THE ORDER IS IMPORTANT AS WELL AS THE LENGTH OF EACH	
344					:	ASCII STRING. EACH STRING IS SEVEN(10) BYTES PLUS ZERO	
345					:	FILL BYTE (TOTAL 8(EIGHT) BYTES) LONG. USED IN LINE1	
346					:	SUBROUTINE.....	
347					:	
348					:		
349	004224	053440	041522	045510	MTCR:	.ASCIZ	/ 'RCHK /
350	004234	043440	051524	040524	MTGS:	.ASCIZ	/ GTSTAT/

```

351 004244 051440 042505 020113 MTSK: .ASCIZ / SEEK /
352 004254 051040 044104 051104 MTRH: .ASCIZ / RDHDR /
353 004264 053440 044522 042524 MTWR: .ASCIZ / WRITE /
354 004274 051040 040505 020104 MTRD: .ASCIZ / READ /
355
356
357 .....:END OF LIST NOW YOU CAN PUT ANY THING YOU WANT HERE.....
358 .....
359
360
361
365
366 .EVEN
367
368 004304 ENDMOD
369
370 .SBTTL ERROR MESSAGES
371
372 004304 BGNMOD GLBERR
373
374 ;GENERAL ERROR REPORT
375
376 004304 BGNMSG ERR1
377 004304 004737 005474 JSR PC,LINE3
378 004310 ENDMSG
(3) 004310 L10000:
(3) 004310 104023 EMT CSMSG
379
380 ;MIS-SEEK ERROR REPORT
381
382 004312 BGNMSG ERR2
383 004312 004737 005474 JSR PC,LINE3
384 004316 PRINTB #FMT4,#DIFMSG,DIFWD(R4),#LPS,LSTHDR(R4),#EPS,PRPOS(R4),#RPS,R1
(15) 004316 010146 MOV R1,-(SP)
(14) 004320 012746 003673 MOV #RPS,-(SP)
(13) 004324 016446 000122 MOV PRPOS(R4),-(SP)
(12) 004330 012746 003660 MOV #EPS,-(SP)
(11) 004334 016446 000050 MOV LSTHDR(R4),-(SP)
(10) 004340 012746 003645 MOV #LPS,-(SP)
(9) 004344 016446 000066 MOV DIFWD(R4),-(SP)
(8) 004350 012746 002433 MOV #DIFMSG,-(SP)
(7) 004354 012746 006013 MOV #FMT4,-(SP)
(6) 004360 012746 000011 MOV #11,-(SP)
(3) 004364 010600 MOV SP,R0
(4) 004366 104014 EMT CSNTB
(4) 004370 062706 000024 ADD #24,SP
385 004374 L10001:
(3) 004374 104023 EMT CSMSG
386
387 ;SOFT.ERROR RECOVERABLE ERROR REPORT
388 004376 BGNMSG ERR3
389 004376 004737 005230 JSR PC,LINE1
390 004402 PRINTB #FMT2A,#CRLCS,SOFTCS(R4),#CRLBA,@BBA(R4),#CRLDA,LSTDA(R4)
(13) 004402 016446 000064 MOV LSTDA(R4),-(SP)
(12) 004406 012746 002407 MOV #CRLDA,-(SP)

```

(11)	004412	017446	000110	MOV	BBB(R4),-(SP)
(10)	004416	012746	002375	MOV	#CRLBA, -(SP)
(9)	004422	016446	000116	MOV	SOFTCS(R4), -(SP)
(8)	004426	012746	002347	MOV	#CRLCS, -(SP)
(7)	004432	012746	005726	MOV	#FMT2A, -(SP)
(6)	004436	012746	000007	MOV	#7, -(SP)
(3)	004442	010600		MOV	SP,R0
(4)	004444	104014		EMT	C\$PNTB
(4)	004446	062706	000020	ADD	#20,SP
391	004452			PRINTB	#FMT5,#MRT,RETRY(R4),#ERT,RTYPE(R4)
(11)	004452	016446	000052	MOV	RTYPE(R4), -(SP)
(10)	004456	012746	003766	MOV	#ERT, -(SP)
(9)	004462	016446	000036	MOV	RETRY(R4), -(SP)
(8)	004466	012746	003755	MOV	#MRT, -(SP)
(7)	004472	012746	006044	MOV	#FMT5, -(SP)
(6)	004476	012746	000005	MOV	#5, -(SP)
(3)	004502	010600		MOV	SP,R0
(4)	004504	104014		EMT	C\$PNTB
(4)	004506	062706	000014	ADD	#14,SP
392	004512			ENDMSG	
(3)	004512				
(3)	004512	104023		EMT	C\$MSG
393					
394					
395					
396	004514				
397	004514	004737	005474	BGNMSG	ERR4
398	004520			JSR	PC,LINE3
(12)	004520	013746	002322	PRINTB	#FMT6,#MST,E.MP,#MST1,ST1,ST2
(11)	004524	013746	002320	MOV	ST2, -(SP)
(10)	004530	012746	004021	MOV	ST1, -(SP)
(9)	004534	013746	002256	MOV	#MST1, -(SP)
(8)	004540	012746	004004	MOV	E.MP, -(SP)
(7)	004544	012746	006060	MOV	#MST, -(SP)
(6)	004550	012746	000006	MOV	#FMT6, -(SP)
(3)	004554	010600		MOV	#6, -(SP)
(4)	004556	104014		MOV	SP,R0
(4)	004560	062706	000016	EMT	C\$PNTB
399	004564			ADD	#16,SP
(3)	004564			ENDMSG	
(3)	004564	104023		EMT	C\$MSG
400					
401					
402					
403					
404	004566				
405	004566	004737	005410	BGNMSG	ERR6
406	004572	016400	000042	JSR	PC,LINE2
407	004576			MOV	BMP(R4),R0
(9)	004576	010046		PRINTB	#FMT9A,DECNT,R0
(8)	004600	013746	002174	MOV	R0, -(SP)
(7)	004604	012746	006230	MOV	DECNT, -(SP)
(6)	004610	012746	000003	MOV	#FMT9A, -(SP)
(3)	004614	010600		MOV	#3, -(SP)
(4)	004616	104014		MOV	SP,R0
(4)	004620	062706	000010	EMT	C\$PNTB
				ADD	#10,SP

;GET STATUS ERROR REPORT

;DATA ERROR SUMMARY

```

408 004624          ENDMSG
(3) 004624          L10004: EMT      C$MSG
(3) 004624 104023
409
410                                     ;NON RECOVERABLE ERROR REPORT
411
412 004626          BGNMSG  ERR7
413 004626          PRINTB  #FMT8,RETRY(R4),#RT1
(9) 004626 012746 004036  MOV      #RT1,-(SP)
(8) 004632 016446 000036  MOV      RETRY(R4),-(SP)
(7) 004636 012746 006162  MOV      #FMT8,-(SP)
(6) 004642 012746 000003  MOV      #3,-(SP)
(3) 004646 010600          MOV      SP,R0
(4) 004650 104014          EMT      C$PNTB
(4) 004652 062706 000010  ADD      #10,SP
414 004656 004737 005474  JSR      PC,LINE3
415 004662          ENDMSG
(3) 004662          L10005: EMT      C$MSG
(3) 004662 104023
416
417                                     ;BAD DATA COMPARE ERROR REPORT
418
419 004664          BGNMSG  ERR8
420 004664          PRINTB  #FMT10,#TIME,HOUR,MINUTE,SECOND,#MRLCS,DCS(R4),#DRNM,<B,DRSEL+1(R4)>
(15) 004664 005046          CLR      -(SP)
(15) 004666 156416 000107  BISB    DRSEL+1(R4),(SP)
(14) 004672 012746 003634  MOV      #DRNM,-(SP)
(13) 004676 016446 000104  MOV      DCS(R4),-(SP)
(12) 004702 012746 002337  MOV      #MRLCS,-(SP)
(11) 004706 013746 002242  MOV      SECOND,-(SP)
(10) 004712 013746 002244  MOV      MINUTE,-(SP)
(9) 004716 013746 002246  MOV      HOUR,-(SP)
(8) 004722 012746 002330  MOV      #TIME,-(SP)
(7) 004726 012746 006302  MOV      #FMT10,-(SP)
(6) 004732 012746 000011  MOV      #11,-(SP)
(3) 004736 010600          MOV      SP,R0
(4) 004740 104014          EMT      C$PNTB
(4) 004742 062706 000024  ADD      #24,SP
421 004746          PRINTB  #FMT10A,#CRLBA,@BBA(R4),#CRLDA,BDA(R4),#EXP,GDDAT,#RCD,(R2)
(15) 004746 011246          MOV      (R2),-(SP)
(14) 004750 012746 004072  MOV      #RCD,-(SP)
(13) 004754 013746 002232  MOV      GDDAT,-(SP)
(12) 004760 012746 004061  MOV      #EXP,-(SP)
(11) 004764 016446 000040  MOV      BDA(R4),-(SP)
(10) 004770 012746 002407  MOV      #CRLDA,-(SP)
(9) 004774 017446 000110  MOV      @BBA(R4),-(SP)
(8) 005000 012746 002375  MOV      #CRLBA,-(SP)
(7) 005004 012746 006336  MOV      #FMT10A,-(SP)
(6) 005010 012746 000011  MOV      #1,-(SP)
(3) 005014 010600          MOV      SP,R0
(4) 005016 104014          EMT      C$PNTB
(4) 005020 062706 000024  ADD      #24,SP
422 005024          PRINTB  #FMT10B,R2
(8) 005024 010246          MOV      R2,-(SP)
(7) 005026 012746 006407  MOV      #FMT10B,-(SP)
(6) 005032 012746 000002  MOV      #2,-(SP)
  
```

423	(3)	005036	010600		MOV	SP,R0	
	(4)	005040	104014		EMT	C\$PNTB	
	(4)	005042	062706	000006	ADD	#6,SP	
		005046			ENDMSG		
	(3)	005046		L10006:			
	(3)	005046	104023		EMT	C\$MSG	
							;DRIVE ERROR
		005050		BGNMSG	ERR9		
		005050	004737	005474	JSR	PC,LINE3	
		005054			PRINTB	#FMT13,#MST,R1,#LPS,LSTHDR(R4)	
	(11)	005054	016446	000050	MOV	LSTHDR(R4),-(SP)	
	(10)	005060	012746	003645	MOV	#LPS,-(SP)	
	(9)	005064	010146		MOV	R1,-(SP)	
	(8)	005066	012746	004004	MOV	#MST,-(SP)	
	(7)	005072	012746	006445	MOV	#FMT13,-(SP)	
	(6)	005076	012746	000005	MOV	#5,-(SP)	
	(3)	005102	010600		MOV	SP,R0	
	(4)	005104	104014		EMT	C\$PNTB	
	(4)	005106	062706	000014	ADD	#14,SP	
		005112			ENDMSG		
	(3)	005112		L10007:			
	(3)	005112	104023		EMT	C\$MSG	
							;INVALID ENTRY IN P-TABLE REPORT
		005114		BGNMSG	ERR10		
		005114			PRINTB	#FMT11,#MPT,R1,#MRLCS,BCSR,#MVEC,BVEC	
	(13)	005114	013746	002162	MOV	BVEC,-(SP)	
	(12)	005120	012746	003610	MOV	#MVEC,-(SP)	
	(11)	005124	013746	002160	MOV	BCSR,-(SP)	
	(10)	005130	012746	002337	MOV	#MRLCS,-(SP)	
	(9)	005134	010146		MOV	R1,-(SP)	
	(8)	005136	012746	003562	MOV	#MPT,-(SP)	
	(7)	005142	012746	006415	MOV	#FMT11,-(SP)	
	(6)	005146	012746	000007	MOV	#7,-(SP)	
	(3)	005152	010600		MOV	SP,R0	
	(4)	005154	104014		EMT	C\$PNTB	
	(4)	005156	062706	000020	ADD	#20,SP	
		005162			ENDMSG		
	(3)	005162		L10010:			
	(3)	005162	104023		EMT	C\$MSG	
		005164		BGNMSG	ERR12		
		005164	004737	005474	JSR	PC,LINE3	
		005170			ENDMSG		
	(3)	005170		L10011:			
	(3)	005170	104023		EMT	C\$MSG	
		005172		BGNMSG	ERR13		

```

448 005172 004737 005474      JSR    PC,LINE3
449 005176                    PRINTB  #FMT12,#SMSG,BDA(R4)
(9) 005176 016446 000040      MOV    BDA(R4),-(SP)
(8) 005202 012746 002564      MOV    #SMSG,-(SP)
(7) 005206 012746 006435      MOV    #FMT12,-(SP)
(6) 005212 012746 000003      MOV    #3,-(SP)
(3) 005216 010600              MOV    SP,R0
(4) 005220 104014              EMT    C$PNTB
(4) 005222 062706 000010      ADD    #10,SP
450
451 005226                    ENDMSG
(3) 005226                    L10012:
(3) 005226 104023              EMT    C$MSG
452
453 005230 016437 000044 002310  LINE1:  MOV    FUNC(R4),FASPNT      ;GET FUNCTION
454 005236 012737 004224 002306      MOV    #MTCR,FASCII      ;FIRST FUNCTION ASCIIZ
455 005244 042737 000100 002310      BIC    #INTEN,FASPNT     ;CLEAR INTERRUPT ENABLE
456 005252 006237 002310              ASR    FASPNT            ;ALIGN
457 005256 005337 002310              1$:  DEC    FASPNT        ;DOWN COUNT FUNCTION
458 005262 001404              BEQ    2$               ;FOUND?
459 005264 062737 000010 002306      ADD    #8.,FASCII:      ;NO NEXT ONE
460 005272 000771              BR     1$               ;LCOP
461
462 005274                    2$:
463
464 005274                    PRINTB  #FMT1,#TIME,HOUR,MINUTE,SECOND,#MRLCS,DCS(R4),#DRNM,<B,DRSEL+1(R4)>
(15) 005274 005046      CLR    -(SP)
(15) 005276 156416 000107      BLSB  DRSEL+1(R4),(SP)
(14) 005302 012746 003634      MOV    #DRNM,-(SP)
(13) 005306 016446 000104      MOV    DCS(R4),-(SP)
(12) 005312 012746 002337      MOV    #MRLCS,-(SP)
(11) 005316 013746 002242      MOV    SECOND,-(SP)
(10) 005322 013746 002244      MOV    MINUTE,-(SP)
(9) 005326 013746 002246      MOV    HOUR,-(SP)
(8) 005332 012746 002330      MOV    #TIME,-(SP)
(7) 005336 012746 005642      MOV    #FMT1,-(SP)
(6) 005342 012746 000011      MOV    #11,-(SP)
(3) 005346 010600              MOV    SP,R0
(4) 005350 104014              EMT    C$PNTB
(4) 005352 062706 000024      ADD    #24,SP
465 005356                    PRINTB  #FMT1A,#MFUNC,FASCII
(9) 005356 013746 002306      MOV    FASCII,-(SP)
(8) 005362 012746 002361      MOV    #MFUNC,-(SP)
(7) 005366 012746 005676      MOV    #FMT1A,-(SP)
(6) 005372 012746 000003      MOV    #3,-(SP)
(3) 005376 010600              MOV    SP,R0
(4) 005400 104014              EMT    C$PNTB
(4) 005402 062706 000010      ADD    #10,SP
466 005406 000207      RTS    PC
467
468 005410                    LINE2:  PRINTB  #FMT9,#TIME,HOUR,MINUTE,SECOND,#MRLCS,DCS(R4),#DRNM,<B,DRSEL+1(R4)>
(15) 005410 005046      CLR    -(SP)
(15) 005412 156416 000107      BLSB  DRSEL+1(R4),(SP)
(14) 005416 012746 003634      MOV    #DRNM,-(SP)
(13) 005422 016446 000104      MOV    DCS(R4),-(SP)
(12) 005426 012746 002337      MOV    #MRLCS,-(SP)

```

(11) 005432 013746 002242
(10) 005436 013746 002244
(9) 005442 013746 002246
(8) 005446 012746 002330
(7) 005452 012746 006172
(6) 005456 012746 000011
(3) 005462 010600
(4) 005464 104014
(4) 005466 062706 000024
469 005472 000207

MOV SECOND,-(SP)
MOV MINUTE,-(SP)
MOV HOUR,-(SP)
MOV #TIME,-(SP)
MOV #FMT9,-(SP)
MOV #11,-(SP)
MOV SP,R0
EMT C\$PNTB
ADD #24,SP
RTS PC

470
471 005474 004737 005230
472 005500
(15) 005500 016446 000042
(14) 005504 012746 002421
(13) 005510 016446 000040
(12) 005514 012746 002407
(11) 005520 017446 000110
(10) 005524 012746 002375
(9) 005530 016446 000046
(8) 005534 012746 002347
(7) 005540 012746 005705
(6) 005544 012746 000011
(3) 005550 010600
(4) 005552 104014
(4) 005554 062706 000024

LINE3: JSR PC,LINE1
PRINTB #FMT2,#CRLCS,BCSADR(R4),#CRLBA,@BBA(R4),#CRLDA,BDA(R4),#CRLMP,BMP(R4)
MOV BMP(R4),-(SP)
MOV #CRLMP,-(SP)
MOV BDA(R4),-(SP)
MOV #CRLDA,-(SP)
MOV @BBA(R4),-(SP)
MOV #CRLBA,-(SP)
MOV BCSADR(R4),-(SP)
MOV #CRLCS,-(SP)
MOV #FMT2,-(SP)
MOV #11,-(SP)
MOV SP,R0
EMT C\$PNTB
ADD #24,SP

473 005560
(15) 005560 013746 002256
(14) 005564 012746 002421
(13) 005570 013746 002254
(12) 005574 012746 002407
(11) 005600 013746 002252
(10) 005604 012746 002375
(9) 005610 013746 002250
(8) 005614 012746 002347
(7) 005620 012746 005750
(6) 005624 012746 000011
(3) 005630 010600
(4) 005632 104014
(4) 005634 062706 000024
474 005640 000207

PRINTB #FMT3,#CRLCS,E.CS,#CRLBA,E.BA,#CRLDA,E.DA,#CRLMP,E.MP
MOV E.MP,-(SP)
MOV #CRLMP,-(SP)
MOV E.DA,-(SP)
MOV #CRLDA,-(SP)
MOV E.BA,-(SP)
MOV #CRLBA,-(SP)
MOV E.CS,-(SP)
MOV #CRLCS,-(SP)
MOV #FMT3,-(SP)
MOV #11,-(SP)
MOV SP,R0
EMT C\$PNTB
ADD #24,SP
RTS PC

475
476
477

;FORMAT STATEMENTS

481
482 005642 052045 055045 022462 FMT1: .ASCII /%T%Z2%A:%Z2%A:%Z2/
483 005663 045 022524 033117 FMT17: .ASCIZ /%T%06%T%01/
484 005676 052045 052045 047045 FMT1A: .ASCIZ /%T%T%N/
485 005705 045 041101 043105 FMT2: .ASCII /%ABEFORE EKR%T%06/
486 005726 052045 047445 022466 FMT2A: .ASCIZ /%T%06%T%06%T%06%N/
487 005750 040445 052101 042440 FMT3: .ASCIZ /%AAT ERR %T%06%T%06%T%06%T%06%N/
488 006013 045 022524 033117 FMT4: .ASCIZ /%T%06%T%06%N%T%06%T%06%N/
489 006044 052045 047445 022466 FMT5: .ASCIZ /%T%06%T%T%N/
490 006060 052045 047445 022466 FMT6: .ASCIZ /%T%06%T%06%A OR %06%N/
491 006106 047045 052045 055045 FMT7: .ASCIZ /%N%T%Z2%A:%Z2%A:%Z2%T%06%T/
492 006141 045 030517 047045 FMT7A: .ASCIZ /%01%N%T%A - %T%N/


```
493 006162 042045 022466 022524 FMT8: .ASCIZ /%D6%T%N/
494 006172 052045 055045 022462 FMT9: .ASCIZ /%T%Z2%A:%Z2%A:%Z2%T%06%T%01%N/
495 006230 042045 022466 020101 FMT9A: .ASCIZ /%D6%A WORDS BAD OUT OF %D6%A WORDS READ%N/
496 006302 052045 055045 022462 FMT10: .ASCIZ /%T%Z2%A:%Z2%A:%Z2%T%06%T%01/
497 006336 052045 047445 022466 FMT10A: .ASCIZ /%T%06%T%06%N%T%06%T%06%A AT BUS ADDRESS /
498 006407 045 033117 047045 FMT10B: .ASCIZ /%06%N/
499 006415 045 022524 031117 FMT11: .ASCIZ /%T%02%T%06%T%03/
500 006435 045 022524 033117 FMT12: .ASCIZ /%T%06%N/
501 006445 045 022524 033117 FMT13: .ASCIZ /%T%06%T%06%N/
502 006462 052045 055045 022464 FMT13D: .ASCIZ /%T%Z4%A NOW IS %Z4%N/
503 006507 045 022516 022524 FMT14: .ASCIZ /%N%T%N/
504 006516 047445 022466 020101 FMT14A: .ASCIZ /%06%A /
505 006525 045 000116 FMT14C: .ASCIZ /%N/
506 006530 040445 047527 042122 FMT14B: .ASCIZ ?%AWORD %D3%A. S/B %06%A WAS %06%N?
507 006572 040445 051105 047522 FMT15: .ASCIZ /%AERROR(S) SET:%T%N%ARECOVERY BEING ATTEMPTED/
508 006650 040445 047516 020124 FRMT16: .ASCIZ /%ANOT TESTING CS= %06%A DR= %01%N/
509 006712 047045 052045 000 FMT18: .ASCIZ /%N%T/
510 006717 045 022516 022516 FMTS1: .ASCIZ /%N%N%S10%A*** RL01 PERFORMANCE REPORT ***%N%N/
511 006775 045 020101 051040 FMTS1A: .ASCIZ /%A RUNNING%N/
512 007013 045 020101 042040 FMTS1B: .ASCIZ /%A DROPPED %Z2%A:%Z2%N/
513 007044 052045 047445 022465 FMTS2: .ASCIZ /%T%05%05%N/
514 007057 045 051501 042505 FMTS2A: .ASCIZ /%ASEEKS: %D6%Z3%N%ABITS READ: %D6%Z4%Z4%A (*16)%N/
515 007144 040445 044502 051524 FMTS2B: .ASCIZ /%ABITS WRITTEN: %D6%Z4%Z4%A (*16)%N/
516 007210 047045 040445 051105 FMTS3: .ASCIZ /%N%AEERRORS%N%ADRIVE: %D6%A SEEK: %D6%A TRACK: %D6%A DATA: %D6%N/
517 007311 045 044101 051101 FMTS3A: .ASCIZ /%AHARD: %D6%A SOFT: %D6%N/
518 007344 040445 041504 035113 FMTS4: .ASCIZ /%ADCK: %D6%A HCRC: %D6%A NXM: %D6%A HNF: %D6%N/
519 007430 040445 046104 035124 FMTS5: .ASCIZ /%ADLT: %D6%A OPI: %D6%N%N/
520
524
525
526
527
528 007466 .EVEN
529
530 007466 ENDMOD
531
532 007466 BGNMOD HPTCODE
533
534 007466 BGNHW
(3) 007466 000005 .WORD L10013-L$HW/2
535
536 007470 174400 .WORD 174400 ;BUS ADDRESS
537 007472 000160 .WORD 160 ;VECTOR FOR 1ST RL CONTROLLER
538 007474 000240 .WORD 240
539 007476 000000 .WORD 0
540 007500 000001 .WORD 1
541
542 007502 ENDMOD
(3) 007502 L10013:
543
544 007502 ENDMOD
545
546 .SBTTL SOFTWARE PARAMETERS
547
548 007502 BGNMOD SPTCODE
549
```

```

550 007502          BGNSW
(3) 007502 000037      .WORD  L10014-L$SW/2
551
552 007504 000001      LIMIT:  .WORD  1          :RETRY LIMIT
553 007506 000003      ERLMT:  .WORD  3          :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 007516 000170      TYINT:  .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      XCHFLG: .WORD  0          :CHANGE OTHER PARAMETERS
561 007526 002400      T.MXB:  .WORD  1280.    :MAXIMUM R/W TRANSFER BUFFER
562 007530 000100      T.MXH:  .WORD  100      :MAXIMUM HEAD SELECT
563 007532 000000      T.MNH:  .WORD  0          :MINIMUM HEAD SELECT
564 007534 077600      T.MXC:  .WORD  77600    :MAXIMUM CYLINDER
565 007536 000000      T.MNC:  .WORD  0          :MINIMUM CYLINDER
566 007540 000047      T.MXS:  .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.MNB:  .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 000003      DRLMT:  .WORD  3          :DRIVE ERROR LIMIT
574 007560 000000      T.ROF:  .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.CLT:  .WORD  128.    :NUMBER OF ERRORS ON DCK DUMP
579 007572 000000      T.AUT:  .WORD  0          :AUTO ON START UP
580 007574 000000      T.ST!P: .WORD  0          :RESTRICT BUFFER SIZE
581 007576 000001      T.WCK:  .WORD  1          :DO WRITE CHECK
582 007600 000012      T.DCD:  .WORD  10.
583
584 007602          ENDSW
(3) 007602          L10014:
585
586 007602          ENDMOD
587
588 007602          BGNMOD  DSPCODE
589
590 007602          DISPATCH  1
(4) 007602 000001      .WORD  1
(6) 007604 012354      .WORD  T1
591
592 007606          ENDMOD
593
594          .SBTTL  STATISTIC CODE
595
596 007606          BGNMOD  RPTCODE
597
598 007606          BGNRPT
599
600
601 007606          PRINTS  #FMIS1          :PRINT STATISTICAL HEADER
  
```

```

(7) 007606 012746 006717      MOV    #FMTS1,-(SP)
(6) 007612 012746 000001      MOV    #1,-(SP)
(3) 007616 010600              MOV    SP,R0
(4) 007620 104016              EMT    C$PNTS
(4) 007622 062706 000004      ADD    #4,SP
602
603 007626 010446              MOV    R4,-(SP)          ;SAVE PRESENT VALUE OF R4
604
605 007630 012704 025056      MOV    #DRBUF,R4        ;START OF DRIVE BUFFER
606 007634 005764 000104      TST    DCS(R4)          ;IS THERE A DRIVE?
607 007640 001402              BEQ    2$                ;NO, GET NEXT ONE
608
609 007642 004737 011670      JSR    PC,REPORT        ;TYPE OUT SUMMARY
610
611 007646 062704 000124      ADD    #PRPOS+2,R4      ;NEXT DRIVE
612 007652 020427 026316      CMP    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
(3) 007662                      L10015:
(3) 007662 104025              EMT    C$RPT
619
620 007664                      ENDMOD
621
622                      .SBTTL  INITIALIZATION CODE
623
624 007664                      BGNMOD  INITCODE        ;START OF INITIALIZE CODE
625
626 007664                      BGNINIT
627
628 007664                      SETPRI  #340             ;PRIORITY TO SEVEN
(3) 007664 012700 000340      MOV    #340,R0
(3) 007670 104041              EMT    C$SPRI
629 007672                      BRESET
(3) 007672 104033              EMT    C$RESET
630
631
632 007674 005037 000050      CLR    OPFLG
633 007700 005037 002326      CLR    INCALL
634 007704 005037 002302      CLR    STFLG
635 007710 005037 002304      CLR    CNTFLG          ;CLEAR CONT
636 007714                      READEF  #EF.PWR
(3) 007714 012700 000034      MOV    #EF.PWR,R0
(3) 007720 104050              EMT    C$REFG
637 007722                      BNCOMPLETE 3$
(2) 007722 103076              BCC    3$
638 007724 005237 002276      INC    PWRFLG          ;INDICATE POWER FAIL
639 007730 012704 025056      MOV    #DRBUF,R4
640 007734 012702 000001      MOV    #1,R2
641 007740 130237 002130      BITB   R2,DRUT        11$:
642 007744 001446              BEQ    13$
643 007746 016400 000106      MOV    DRSEL(R4),R0
644 007752 052700 000200      BIS    #200,R0

```

```
645 007756 010074 000104      MOV    R0,@DCS(R4)
646 007762 012701 000074      MOV    #60,R1
647 007766 032774 000001 000104 12$:  BIT    #1,@DCS(R4)
648 007774 001014      BNE    15$
649 007776      WAITMS #10.
(3) 007776 012700 000012      MOV    #10,R0
(3) 010002 104026      EMT    CSWTM
650 010004 005301      DEC    R1
651 010006 001367      BNE    12$
652
653 010010 012737 003706 002126      MOV    #NOPWR,WHY
654 010016 004537 020220      JSR    R5,DRDRV
655 010022 000137 010062      JMP    13$
656
657 010026 004537 021136      JSR    R5,ISDRST
```

```

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      RTYPE(R4)
664 010056 005064 000114      CLR      RSEEK(R4)
665 010062 062704 000124      13$:    ADD      #PRPOS+2,R4
666 010066 106302                ASLB     R2
667 010070 103323                BCC     11$
668 010072 005737 002264      TST     SYSCLK
669 010076 001406                BEQ     4$
670 010100                CLKON   #1
(3) 010100 012700 000001      MOV     #1,R0
(3) 010104 104034                EMT     C$KWON
671 010106                REQTIM R0
(3) 010106 104045                EMT     C$REQTIM
672 010110 010037 002240      MOV     R0,LSTTIM
673 010114 000137 011070      4$:    JMP     POWER
674 010120                3$:    READEF #EF.CONTINUE           ;CONTINUE FROM CONSOLE?
(3) 010120 012700 000036      MOV     #EF.CONTINUE,R0
(3) 010124 104050                EMT     C$REFG
675 010126                BNCOMPLETE 1$           ;NO, CONTINUE W/ INIT CODE
(2) 010126 103004                BCC     1$
676
677 010130 005237 002304      INC     CNTFLG           ;YES SET CONT FLAG, GO TO END OF INIT
678 010134 000137 010466      JMP     END
679
680 010140 004537 023722      1$:    JSR     R5,CLEAR           ;CLEAR ALL DRIVE BUFFERS
681 010144 012737 176543 002134  MOV     #176543,HINUM     ;PRIME RANDOM GENERATOR
682 010152 012737 123456 002136  MOV     #123456,LONUM
683 010160 012700 002150      2$:    MOV     #CNTLR1,R0
684 010164 005020      CLRDAT: CLR     (R0)+
685 010166 020027 002304      CMP     R0,#STFLG+2
686 010172 001374                BNE
687
688 010174 012704 025056      MOV     #DRBUF,R4           ;SETUP UP DRIVE BUFFER POINTER
689 010200 012702 024030      MOV     #BSECO,R2           ;SETUP BAD SECTOR POINTER
690 010204 013703 002012      MOV     L$UNIT,R3           ;GET NUMBER OF UNITS
691 010210 010337 002274      MOV     R3,UUT
692 010214 005001                CLR     R1
693 010216 005703      1$:    TST     R3
694 010220 001522                BEQ     END
695 010222                GPHARD R1,R0
(3) 010222 010100      MOV     R1,R0
(3) 010224 104042                EMT     C$GPHRD
696 010226                BNCOMPLETE 12$
(2) 010226 103110                BCC     12$
697 010230 012037 002160      MOV     (R0)+,BCSR           ;GET CSR
698 010234 012037 002162      MOV     (R0)+,BVEC           ;GET VECTOR
699 010240 012037 002164      MOV     (R0)+,BPRIOR         ;GET BPRIOR
700 010244 011037 002166      MOV     (R0),BDRSEL          ;GET DRIVE
701 010250 005737 002150      TST     CNTLR1
702 010254 001011                BNE     2$
703 010256 013737 002164 002226  MOV     BPRIOR,PRIOR1
704 010264 013737 002160 002150  MOV     BCSR,CNTLR1
705 010272 013737 002162 002222  MOV     BVEC,VECT1
  
```

```

706 010300 023737 002160 002150 2$:    CMP      BCSR,CNTRL1      ;IS THIS CSR CNTRL1?
707 010306 001012          BNE      5$              ;NO,GO CHECK AGAINST #2
708 010310 023737 002162 002222          CMP      BVEC,VECT1     ;IS VECTOR PROPER?
709 010316 001050          BNE      10$             ;NO, REPORT ERROR
710 010320 012737 002266 002200          MOV      #BUF1,TEMP1    ;FIRST CONTROLLER/FIRST BUFFER
711 010326 004537 011460          JSR      R5,FILINF      ;FILL BUFFER
712 010332 000450          BR       11$             ;GO GET NEXT P-TABLE
713 010334 005737 002152          5$:    TST      CNTRL2      ;HAVE WE GOT CSR #2 YET?
714 010340 001015          BNE      6$              ;YES, CHECK THIS ONE AGAINST IT
715 010342 023737 002222 002160          CMP      VECT1,BCSR     ;IS THIS VECTOR SAME AS CNTRL1
716 010350 001433          BEQ      10$             ;IFSO, DON'T ALLOW IT
717 010352 013737 002160 002152          MOV      BCSR,CNTRL2   ;MAKE THIS ONE CSR 2
718 010360 013737 002162 002224          MOV      BVEC,VECT2    ;SETUP SECOND VECTOR
719 010366 013737 002164 002230          MOV      BPRIOR,PRIOR2
720 010374 023737 002160 002152 6$:    CMP      BCSR,CNTRL2   ;IS THIS CSR # 2?
721 010402 001016          BNE      10$             ;NO, WELL WE DON'T ALLOW 3
722 010404 023737 002162 002224          CMP      BVEC,VECT2    ;DOES IT HAVE PROPER VECTOR
723 010412 001012          BNE      10$             ;NO, GO REPORT ERROR
724 010414 023737 002224 002222          CMP      VECT2,VECT1   ;IS VECTOR OF FIRST EQUAL TO
725 010422 001406          BEQ      10$             ;VECTOR OF SECOND, YES REPORT ERROR
726 010424 012737 002270 002200          MOV      #BUF2,TEMP1   ;OTHER CNTRLR/OTHER BUFFER
727 010432 004537 011460          JSR      R5,FILINF      ;LOAD BUFFER
728 010436 000406          BR       11$             ;NEXT
729 010440          10$:    ERRDF 160.,ILLEG,ERR10 ;BAD P-TABLE
(3) 010440 104462          TRAP    T$ERCODE
(5) 010442 000240          .WORD  160
(5) 010444 003574          .WORD  ILLEG
(5) 010446 005114          .WORD  ERR10
730 010450 005064 000104          12$:    CLR      DCS(R4)
731 010454 005201          11$:    INC      R1          ;POINT TO NEXT
732 010456 005303          DEC      R3              ;DOWN COUNT
733 010460 062702 000040          ADD     #32.,R2         ;NEXT BAD SECTOR FILE
734 010464 000654          BR       1$              ;DO WHILE
735
736
737 010466          END:
738
739 010466 012737 177770 002132          MOV     #177770,SYSMSK ;SETUP FOR EIGHT DRIVES
740 010474 023727 002274 000004          CMP     UUT,#4          ;MORE THAN FOUR
741 010502 003012          BGT     2$              ;YES, THEN MASK IS OKAY
742 010504 052737 000004 002132          BIS     #4,SYSMSK      ;SETUP FOR FOUR DRIVES
743 010512 023727 002274 000002          CMP     UUT,#2          ;MORE THAN TWO
744 010520 003003          BGT     2$              ;YES, IT'S OKAY
745 010522 052737 000002 002132          BIS     #2,SYSMSK      ;SET FOR ONE OR TWO
746 010530          2$:    READEF #EF.START    ;START COMMAND
(3) 010530 012700 000040          MOV     #EF.START,RO
(3) 010534 104050          EMT     C$REFG
747 010536          BNCOMPLETE RESTART    ;NO, CHK RESTART
(2) 010536 103002          BCC     RESTART
748 010540 005237 002302          INC     STFLG           ;SET START INDICATOR
749
750 010544 005737 002304          RESTART: TST     CNTRFLG ;CONTINUING
751 010550 001026          BNE     3$              ;YES GO TO 3$
752 010552 005037 002144          CLR     WRINIT          ;CLEAR THE WRITE INIT FLAG
753
754

```

```

755          :LET'S CREATE INTERNAL BITMAP
756
757 010556 012701 000001      MOV      #1,R1          ;BIT MASK
758 010562 105037 002131      CLR      DRPRS         ;CLEAR OUT DRIVES PRESENT
759 010566 012704 025056      MOV      #DRBUF,R4     ;START OF DRIVE BUFFERS
760 010572 005704 000104      1$:    TST      DCS(R4)   ;ANY CSR?
761 010576 001402             BEQ      2$            ;NO, NO DRIVE THEN
762 010600 150137 002131      BIS      R1,DRPRS      ;INDICATE DRIVE IN BITMAP
763 010604 006301             2$:    ASL      R1          ;NEXT POSITION
764 010606 062704 000124      ADD      #PRPOS+2,R4   ;NEXT DRIVE BUFFER
765 010612 022704 026316      CMP      #ENDBUF,R4   ;DONE
766 010616 001365             BNE      1$          ;NO
767
768 010620 113737 002131 002*30  MOV      DRPRS,DRPTR   ;SET UP DRIVES UNDER TEST
769
770 010626             3$:
771
772 010626             SETVEC  VECT1,#INTR1,PRIOR1 ;SET CONTROLLER 1'S VECTOR
(7) 010626 013746 002226      MOV      PRIOR1,-(SP)
(6) 010632 012746 014222      MOV      #INTR1,-(SP)
(5) 010636 013746 002222      MOV      VECT1,-(SP)
(4) 010642 012746 000003      MOV      #3,-(SP)
(3) 010646 104037             EMT      C$SVEC
(2) 010650 062706 000010      ADD      #10,SP
773
774 010654 005737 002152      TST      CNTLR2       ;RUNNING TWO CONTROLLERS?
775 010660 001413             BEQ      4$          ;NO
776
777 010662             SETVEC  VECT2,#INTR2,PRIOR2 ;YES SET CONTROLLER 2'S VECTOR
(7) 010662 013746 002230      MOV      PRIOR2,-(SP)
(6) 010666 012746 014232      MOV      #INTR2,-(SP)
(5) 010672 013746 002224      MOV      VECT2,-(SP)
(4) 010676 012746 000003      MOV      #3,-(SP)
(3) 010702 104037             EMT      C$SVEC
(2) 010704 062706 000010      ADD      #10,SP
778
779 010710 005737 002304      4$:    TST      CNTFLG   ;CONTINUE?
780 010714 001412             BEQ      FINDBF     ;NO, GO PAST RESTART OF CLOCK
781
782 010716 005737 002264      TST      SYSCLK      ;DO WE HAVE SYSTEM CLOCK
783 010722 001462             BEQ      5$          ;NO
784
785 010724             CLKON  #1          ;TURN CLK ON
(3) 010724 012700 000001      MOV      #1,R0
(3) 010730 104034             EMT      C$KWON
786 010732             REQTIM R0          ;REQUEST TIME
(3) 010732 104045             EMT      C$REQTIM
787 010734 010037 002240      MOV      R0,LSTTIM   ;MAKE IT PRESENT TIME
788 010740 000453             BR       POWER      ;GO TO END
789
790
791 010742 012703 000050      FINDBF: MOV      #40.,R3   ;MAXIMUM SECTOR IS 40
792 010746 005001             CLR      R1          ;START WC AT ZERO
793 010750 005737 002152      TST      CNTLR2     ;TWO CONTROLLERS????
794 010754 001402             BEQ      1$          ;NO, START WC AT 5120
795 010756 012701 000024      MOV      #20.,R1    ;20 256 WORD BUFFERS

```

```

796 010762 062701 000024 1$: ADD #20,R1 ;WC TO 5120 PLUS 5120
797 010766 2$: BUFFER R1,R2 ;GET BUFFER IF AVAILABLE
(3) 010766 010100 MOV R1,R0
(3) 010770 104030 EMT C$BUFF
(3) 010772 010002 MOV R0,R2
798 010774 BCOMPLETE 4$ ;WAS AVAILABLE, THEN BR
(2) 010774 103411 BCS 4$
799 010776 005737 002152 TST CNTLR2 ;TWO CONTROLLERS??
800 011002 001401 BEQ 3$ ;NO
801 011004 005301 DEC R1 ;ONE 256 WORD BUFFER LESS
802 011006 005301 3$: DEC R1 ;ONE MORE LESS
803 011010 162703 000002 SUB #2,R3
804 011014 001364 BNE 2$ ;IF NOT ZERO GO BACK
805
806 011016 000000 HALT
807
808 011020 042701 177400 4$: BIC #177400,R1
809 011024 000301 SWAB R1
810 011026 010237 002266 MOV R2,BUF1 ;GET BUFFER FOR FIRST CONTROLLER
811 011032 005737 002152 TST CNTLR2 ;TWO CONTROLLERS??
812 011036 001404 BEQ 5$ ;NO
813 011040 060102 ADD R1,R2 ;SECOND'S BUFFER
814 011042 010237 002270 MOV R2,BUF2
815 011046 006201 ASR R1 ;CORRECT WORD COUNT
816 011050 010137 002272 5$: MOV R1,MAXWC ;MAX WORD COUNT
817
818
819
820 011054 7$: CLKON #1 ;TURN CLOCK ON?
(3) 011054 012700 000001 MOV #1,R0
(3) 011060 104034 EMT C$KWON
821 011062 BNCOMPLETE POWER ;WAS THERE A CLOCK?
(2) 011062 103002 BCC POWER
822
823 011064 005237 002264 INC SYSCLK ;YES, SET FLAG FOR ONE!
824
825 011070 POWER:
826
827
828
829 011070 ENDINIT
(3) 011070 L10016: EMT C$INIT
(3) 011070 104011
830
831 011072 ENDMOD
832
833 011072 BGNMOD CLNCODE
834
835
836 011072 BGNCLN
837
838 011072 SETVEC ERRVEC,#TRPHAN,#340
(7) 011072 012746 000340 MOV #340,-(SP)
(6) 011076 012746 011662 MOV #TRPHAN,-(SP)
(5) 011102 013746 002316 MOV ERRVEC,-(SP)
(4) 011106 012746 000003 MOV #3,-(SP)

```



```

(3) 011112 104037      EMT      C$SVEC
(2) 011114 062706 000010  ADD      #10,SP
839 011120      SETPRI  #340           ;PRIORITY TO SEVEN
(3) 011120 012700 000340  MOV      #340,R0
(3) 011124 104041      EMT      C$SPRI
840
841 011126 032777 000200 171014 1$:  BIT      #CRDY,@CNTLR1      ;WAIT FOR CONTROLLER TO FINISH
842 011134 001774      BEQ      1$
843 011136 042777 000100 171004  BIC      #INTEN,@CNTLR1    ;CLEAR INTERRUPT IF PENDING
844 011144      CLRVEC  VECT1           ;RELEASE VECTOR OF FIRST CONTROLLER
(3) 011144 013700 002222  MOV      VECT1,R0
(3) 011150 104036      EMT      C$CVEC
845
846 011152 005737 002152      TST      CNTLR2           ;TWO CONTROLLERS
847 011156 001412      BEQ      3$           ;NO
848
849 011160 032777 000200 170764 2$:  BIT      #CRDY,@CNTLR2      ;WAIT FOR OTHER CONTROLLER TO FINISH
850 011166 001774      BEQ      2$
851 011170 042777 000100 170754  BIC      #INTEN,@CNTLR2    ;CLEAR OUT INTERRUPT ENABLE
852 011176      CLRVEC  VECT2           ;YES, WELL RELEASE IT'S VECTOR
(3) 011176 013700 002224  MOV      VECT2,R0
(3) 011202 104036      EMT      C$CVEC
853
854 011204 005037 002326      CLR      INCALL           ;THIS IS FOR LSI-11 CPU'S
855 011210 005037 002324      CLR      OPCALL
856 011214      CLRVEC  ERRVEC
(3) 011214 013700 002316  MOV      ERRVEC,R0
(3) 011220 104036      EMT      C$CVEC
857
858 011222 005737 002264      TST      SYSCLK
859 011226 001401      BEQ      4$
860
861 011230      CLKOFF
(3) 011230 104035      EMT      C$KWOFF
862
863 011232      BRESET  ;THIS IS FOR LSI-11 CPU'S
(3) 011232 104033      EMT      C$RESET
864 011234      ENDCLN
(3) 011234      L10017:
(3) 011234 104012      EMT      C$CLEAN
865
866 011236      ENDMOD
867
868
869 011236      BGNMOD  ADDCODE
870
871 011236      BGNAU
872
873 011236 012704 025056  MOV      #DRBUF,R4           ;START OF DRIVE BUFFERS
874 011242 012701 000001  MOV      #1,R1             ;MASK TO FIND DRIVE
875 011246 010002      MOV      R0,R2           ;SAVE WHICH TO FIND
876 011250 005700      TST      R0             ;THIS ONE
877 011252 001405      BEQ      2$           ;YES
878 011254 062704 000124  ADD      #PRPOS+2,R4       ;NEXT
879 011260 006301      ASL      R0             ;NEXT MASK
880 011262 005300      DEC      R0
  
```

```

881 011264 000771          BR      1$
882 011266 150137 002130 2$:  BISB   R1,DRUT          ;INSERT IN DRIVE UNDER TEST
883 011272          GPHARD  R2,R1
(3) 011272 010200          MOV    R2,R0
(3) 011274 104042          EMT    C$GPHRD
(3) 011276 010001          MOV    R0,R1
884 011300 011164 000104  MOV    (R1),DCS(R4)
885 011304 012700 000100  MOV    #SERM1,R0          ;SETUP TO CLEAR STATS
886 011310 006200          ASR    R0
887 011312 005024          4$:  CLR    (R4)+
888 011314 005300          DEC    R0
889 011316 001375          BNE   4$
890 011320          5$:
891
892 011320          ENDAU
(3) 011320          L10020:
(3) 011320 104054          EMT    C$AU
893
894 011322          ENDMOD
895
896 011322          BGNMOD  DROP CODE
897
898 011322          BGNDU
899
900 011322 005737 002326          TST    INCALL
901 011326 001015          BNE   3$
902 011330 012704 025056          MOV    #DRBUF,R4
903 011334 005700          2$:  TST    R0
904 011336 001404          BEQ   1$
905 011340 005300          DEC    R0
906 011342 062704 000124          ADD    #PRPOS+2,R4
907 011346 000772          BR     2$
908
909 011350 012737 003313 002126 1$:  MOV    #REQ,WHY
910 011356 004537 020214          JSR   R5,ODRDRV
911 011362          3$:
912
913
914 011362          ENDDU
(3) 011362          L10021:
(3) 011362 104055          EMT    C$DU
915
916 011364          ENDMOD
917
918          .SBTTL  GLOBAL SUBROUTINES
919
920 011364          BGNMOD  GLBSUB
921
922 011364 012701 000010          SETWCK: MOV    #8,R1
923 011370 012702 025056          MOV    #DRBUF,R2
924 011374 026462 000104 000104 1$:  CMP    DCS(R4),DCS(R2)
925 011402 001002          BNE   2$
926 011404 010462 000076          MOV    R4,DOWCK(R2)
927 011410 062702 000124          2$:  ADD    #PRPOS+2,R2
928 011414 005301          DEC    R1
929 011416 001366          BNE   1$

```

```

930 011420 000205          RTS      R5
931
932 011422 012701 000010  CLRWCK: MOV      #8,R1
933 011426 012702 025056          MOV      #DRBUF,R2
934 011432 026462 000104 000104 1$:  CMP      DCS(R4),DCS(R2)
935 011440 001002          BNE      2$
936 011442 005062 000076          CLR      DOWCK(R2)
937 011446 062702 000124          2$:  ADD      #PRPOS+2,R2
938 011452 005301          DEC      R1
939 011454 001366          BNE      1$
940 011456 000205          RTS      R5
941
942
943          ;ROUTINE TO FILL BUFFERS WITH INFO
944
945 011460 013764 002166 000106  FILINF: MOV      BDRSEL,DRSEL(R4)          ;SET DRIVE SELECT BITS
946 011466 013764 002160 000104          MOV      BCSR,DCS(R4)          ;SET CSR
947 011474 013764 002200 000110          MOV      TEMP1,BBA(R4)          ;SET R/W BUFFER
948 011502 010264 000112          MOV      R2,BSECT(R4)          ;SETUP BAD SECTOR POINTER
949 011506 005737 007572          TST      T.AUT          ;DO WE AUTOSIZE?
950 011512 001460          BEQ      1$          ;NO, SKIP
951
952 011514 005037 002300          CLR      TRPFLG          ;CLEAR TRAP FLAG
953 011520          SETVEC  ERRVEC,#TRPHAN,#340          ;SETUP TO CATCH TRAP
(7) 011520 012746 000340          MOV      #340,-(SP)
(6) 011524 012746 011662          MOV      #TRPHAN,-(SP)
(5) 011530 013746 002316          MOV      ERRVEC,-(SP)
(4) 011534 012746 000003          MOV      #3,-(SP)
(3) 011540 104037          EMT      C$SVEC
(2) 011542 062706 000010          ADD      #10,SP
954 011546 005774 000104          TST      @DCS(R4)
955 011552 005737 002300          TST      TRPFLG          ;DID TRAP OCCUR
956 011556 001012          BNE      3$          ;YES IGNORE DRIVE
957 011560 016400 000106          MOV      DRSEL(R4),R0          ;YES, FIND OUT IF DRIVE
958 011564 052700 000200          BIS      #200,R0          ;HAS DRIVE READY POSTED
959 011570 010074 000104          MOV      R0,@DCS(R4)
960 011574 032774 000001 000104          BIT      #1,@DCS(R4)          ;IS DRIVE READY HIGH?
961 011602 001021          BNE      2$          ;YES, CHECK NEXT
962
963 011604          3$:  PRINTF  #FRMT16,DCS(R4),<B,DRSEL+1(R4)>
(9) 011604 005046          CLR      -(SP)
(9) 011606 156416 000107          BISB    DRSEL+1(R4),(SP)
(8) 011612 016446 000104          MOV      DCS(R4),-(SP)
(7) 011616 012746 006650          MOV      #FRMT16,-(SP)
(6) 011622 012746 000003          MOV      #3,-(SP)
(3) 011626 010600          MOV      SP,R0
(4) 011630 104017          EMT      C$PNTF
(4) 011632 062706 000010          ADD      #10,SP
964
965 011636 005337 002274          DEC      UUT          ;ONE LESS DRIVE NOW
966 011642 005064 000104          CLR      DCS(R4)          ;TAKE DRIVE OUT OF BUFFER
967 011646          2$:  CLRVEC  ERRVEC          ;RELEASE THE VECTOR
(3) 011646 013700 002316          MOV      ERRVEC,R0
(3) 011652 104036          EMT      C$CVEC
968 011654 062704 000124          1$:  ADD      #PRPOS+2,R4          ;UPDATE POINTER
969 011660 000205          RTS      R5
  
```

```

970 011662 005237 011662      TRPHAN: INC      TRPHAN
971 011666 000002                RTI
972
973                               ;ROUTINE TO PRINT STATISTICAL REPORT OF DRIVE(S)
974
975 011670      REPORT:
976
977 011670      PRINTS #FMT1,#TIME,HOUR,MINUTE,SECOND,#MRLCS,DCS(R4),#DRNM,<B,DRSEL+1(R4)>
(15) 011670 005046      CLR      -(SP)
(15) 011672 156416 000107      BISB   DRSEL+1(R4),(SP)
(14) 011676 012746 003634      MOV    #DRNM,-(SP)
(13) 011702 016446 000104      MOV    DCS(R4),-(SP)
(12) 011706 012746 002337      MOV    #MRLCS,-(SP)
(11) 011712 013746 002242      MOV    SECOND,-(SP)
(10) 011716 013746 002244      MOV    MINUTE,-(SP)
(9) 011722 013746 002246      MOV    HOUR,-(SP)
(8) 011726 012746 002330      MOV    #TIME,-(SP)
(7) 011732 012746 005642      MOV    #FMT1,-(SP)
(6) 011736 012746 000011      MOV    #1,-(SP)
(3) 011742 010600                MOV    SP,R0
(4) 011744 104016                EMT    C$PNTS
(4) 011746 062706 000024      ADD    #24,SP
978
979 011752 005764 000070      TST    DPHOUR(R4)                ;DO WE HAVE ANY DROPPED TIME
980 011756 001417                BEQ    1$                        ;NO, THEN PRINT RUNNING
981
982 011760      PRINTS #FMTS1B,<B,DPHOUR(R4)>,<B,DPMIN(R4)>
(9) 011760 005046      CLR      -(SP)
(9) 011762 156416 000071      BISB   DPMIN(R4),(SP)
(8) 011766 005046      CLR      -(SP)
(8) 011770 156416 000070      BISB   DPHOUR(R4),(SP)
(7) 011774 012746 007013      MOV    #FMTS1B,-(SP)
(6) 012000 012746 000003      MOV    #3,-(SP)
(3) 012004 010600                MOV    SP,R0
(4) 012006 104016                EMT    C$PNTS
(4) 012010 062706 000010      ADD    #10,SP
983 012014 000410      BR     2$
984
985 012016      1$: PRINTS #FMTS1A
(7) 012016 012746 006775      MOV    #FMTS1A,-(SP)
(6) 012022 012746 000001      MOV    #1,-(SP)
(3) 012026 010600                MOV    SP,R0
(4) 012030 104016                EMT    C$PNTS
(4) 012032 062706 000004      ADD    #4,SP
986 012036      2$:
987 012036      PRINTS #FMTS2,#CART,SERNM2(R4),SERNM1(R4)
(10) 012036 016446 000100      MOV    SERNM1(R4),-(SP)
(9) 012042 016446 000102      MOV    SERNM2(R4),-(SP)
(8) 012046 012746 002444      MOV    #CART,-(SP)
(7) 012052 012746 007044      MOV    #FMTS2,-(SP)
(6) 012056 012746 000004      MOV    #4,-(SP)
(3) 012062 010600                MOV    SP,R0
(4) 012064 104016                EMT    C$PNTS
(4) 012066 062706 000012      ADD    #12,SP
988 012072      PRINTS #FMTS2A,SKCNT(R4),SKCNT1(R4),RXFR3(R4),RXFR2(R4),RXFR1(R4)
(12) 012072 016446 000002      MOV    RXFR1(R4),-(SP)
    
```

(11)	012076	016446	000004	MOV	RXFR2(R4),-(SP)
(10)	012102	016446	000060	MOV	RXFR3(R4),-(SP)
(9)	012106	016446	000054	MOV	SKCNT1(R4),-(SP)
(8)	012112	016446	000000	MOV	SKCNT(R4),-(SP)
(7)	012116	012746	007057	MOV	#FMTS2A, -(SP)
(6)	012122	012746	000006	MOV	#6, -(SP)
(3)	012126	010600		MOV	SP,R0
(4)	012130	104016		EMT	CSPNTS
(4)	012132	062706	000016	ADD	#16, SP
989	012136			PRINTS	#FMTS2B, WXFR3(R4), WXFR2(R4), WXFR1(R4)
(10)	012136	016446	000006	MOV	WXFR1(R4), -(SP)
(9)	012142	016446	000010	MOV	WXFR2(R4), -(SP)
(8)	012146	016446	000062	MOV	WXFR3(R4), -(SP)
(7)	012152	012746	007144	MOV	#FMTS2B, -(SP)
(6)	012156	012746	000004	MOV	#4, -(SP)
(3)	012162	010600		MOV	SP,R0
(4)	012164	104016		EMT	CSPNTS
(4)	012166	062706	000012	ADD	#12, SP
990	012172			PRINTS	#FMTS3, DERCNT(R4), SKECNT(R4), TRERR(R4), DATCER(R4)
(11)	012172	016446	000074	MOV	DATCER(R4), -(SP)
(10)	012176	016446	000072	MOV	TRERR(R4), -(SP)
(9)	012202	016446	000016	MOV	SKECNT(R4), -(SP)
(8)	012206	016446	000020	MOV	DERCNT(R4), -(SP)
(7)	012212	012746	007210	MOV	#FMTS3, -(SP)
(6)	012216	012746	000005	MOV	#5, -(SP)
(3)	012222	010600		MOV	SP,R0
(4)	012224	104016		EMT	CSPNTS
(4)	012226	062706	000014	ADD	#14, SP
991	012232			PRINTS	#FMTS3A, ERRCNT(R4), SFTCNT(R4)
(9)	012232	016446	000014	MOV	SFTCNT(R4), -(SP)
(8)	012236	016446	000012	MOV	ERRCNT(R4), -(SP)
(7)	012242	012746	007311	MOV	#FMTS3A, -(SP)
(6)	012246	012746	000003	MOV	#3, -(SP)
(3)	012252	010600		MOV	SP,R0
(4)	012254	104016		EMT	CSPNTS
(4)	012256	062706	000010	ADD	#10, SP
992	012262			PRINTS	#FMTS4, DCR CER(R4), HCR CER(R4), NXMCNT(R4), HNFERR(R4)
(11)	012262	016446	000032	MOV	HNFERR(R4), -(SP)
(10)	012266	016446	000034	MOV	NXMCNT(R4), -(SP)
(9)	012272	016446	000024	MOV	HCR CER(R4), -(SP)
(8)	012276	016446	000022	MOV	DCR CER(R4), -(SP)
(7)	012302	012746	007344	MOV	#FMTS4, -(SP)
(6)	012306	012746	000005	MOV	#5, -(SP)
(3)	012312	010600		MOV	SP,R0
(4)	012314	104016		EMT	CSPNTS
(4)	012316	062706	000014	ADD	#14, SP
993	012322			PRINTS	#FMTS5, DLTCNT(R4), OPI CNT(R4)
(9)	012322	016446	000030	MOV	OPI CNT(R4), -(SP)
(8)	012326	016446	000026	MOV	DLTCNT(R4), -(SP)
(7)	012332	012746	007430	MOV	#FMTS5, -(SP)
(6)	012336	012746	000003	MOV	#3, -(SP)
(3)	012342	010600		MOV	SP,R0
(4)	012344	104016		EMT	CSPNTS
(4)	012346	062706	000010	ADD	#10, SP
994	012352	000207		RTS	PC
995					

```

996
997 012354          ENDMOD
998
999          .SBTTL PROGRAM MAIN LOOP
1000
1001 012354          BGNTST
1002          :MAIN PROGRAM LOOP
1003          :PROGRAM WILL RANDOMLY PICK ONE OF THE DRIVES TO
1004          :PERFORM AN OPERATION. WE WILL ALWAYS PICK ONE OF FOUR
1005          :OR EIGHT DRIVES (ONE OR TWO CONTROLLERS) 'DRUT' WILL BE
1006          :CHECKED TO SEE IF DRIVE IS ON SYSTEM. ONCE DRIVE IS PICKED
1007          :THEN A FUNCTION WILL BE SELECTED RANDOMLY FOR THAT
1008          :DRIVE. FUNCTIONS OF CONTROLLER RESET, GET STATUS, SEEK, READ, WRITE
1009          :WILL BE SELECTED, EACH FUNCTION WILL HAVE IT'S OWN ROUTINE
1010          :TO GET PARAMETERS FOR THE DRIVE.
1011
1012 012354 005737 002144          MTEST: TST      WRINIT          ;SEE IF WRITE INIT IN PROGRESS
1013 012360 001407                BEQ      161$          ;JUMP OVER IF NOT INIT FROM PWR FAIL
1014 012362 013704 002144          MOV      WRINIT,R4        ;YUP - RESET R4 POINTER
1015 012366 013701 002146          MOV      WRPOS,R1        ;AND THE DRIVE POINTER FOR INIT
1016 012372 005237 002302          INC      STFLG          ;FAKE OUT THE START FLAG
1017 012376 000410                BR       16$            ;AND CONTINUE WITH THE WRITE CODE
1018
1019 012400 012704 025056          161$: MOV      #DRBUF,R4        ;GET DRIVE BUFFERS
1020 012404 012701 000001          MOV      #1,R1          ;MASK
1021 012410 010437 002144          MOV      R4,WRINIT      ;COPY R4
1022 012414 010137 002146          MOV      R1,WRPOS        ;AND R1 POINTERS
1023 012420 130137 002130          16$: BITB     R1,DRUT      ;DRIVE UNDER TEST
1024 012424 001444                BEQ      15$            ;NO
1025
1026 012426 012774 000200 000104          MOV      #200,@DCS(R4)   ;CHECK IF DRIVE THERE
1027 012434 056474 000106 000104          BIS      DRSEL(R4),@DCS(R4)
1028 012442 012700 000000          66$: MOV      #0.,R0      ;STALL
1029 012446 005300                13$: DEC      R0
1030 012450 001376                BNE     13$
1031 012452 032774 000001 000104          BIT      #DRDY,@DCS(R4)
1032 012460 001006                BNE     14$
1033
1034 012462 012737 002474 002126          MOV      #DNRDY,WHY
1035 012470 004537 020220          JSR      R5,DRDRV
1036 012474 000420                BR       15$
1037
1038 012476 004537 017506          14$: JSR      R5,RDBDSC      ;GO GET BAD SECTORS
1039 012502 005064 000056          CLR      PRFLGS(R4)
1040 012506 005064 000076          CLR      DOWCK(R4)
1041 012512 005064 000114          CLR      RSEEK(R4)
1042 012516 005764 000120          TST      WRIPG(R4)
1043 012522 001003                BNE     99$            ;SEE IF WRITE IN PROGRESS FLG SET
1044 012524 005737 002302          TST      STFLG          ;YUP - RE-INIT THIS DRIVE
1045 012530 001402                BEQ     15$
1046
1047 012532                99$:
1048 012532 004537 021312          JSR      R5,WRPACK
1049
1050
1051 012536 062704 000124          15$: ADD      #PRPOS+2,R4      ;NEXT DRIVE

```

```

1052 012542 010437 002144      MOV      R4,WRINIT      ;COPY FOR POSSIBLE PWR FAIL
1053 012546 006337 002146      ASL      WRPOS          ;SHIFT THE POSITION FLAG ALSO
1054 012552 106301              ASLB     R1              ;DONE?
1055 012554 103321              BCC     16$             ;NO GO FOR NEXT ONE
1056
1057 012556 005037 002144      CLR      WRINIT          ;CLEAR WRITE INIT FLG ... ALL DONE
1058 012562              12$: PRINTF #FMT14,#MSTART
      (8) 012562 012746 004165      MOV      #MSTART,-(SP)
      (7) 012566 012746 006507      MOV      #FMT14,-(SP)
      (6) 012572 012746 000002      MOV      #2,-(SP)
      (3) 012576 010600              MOV      SP,R0
      (4) 012600 104017              EMT     C$PNTF
      (4) 012602 062706 000006      ADD     #6,SP
1059 012606              SETPRI #0                ;PRIORITY TO ZERO
      (3) 012606 012700 000000      MOV      #0,R0
      (3) 012612 104041              EMT     C$SPRI
1060 012614 004537 021214      MAIN: JSR     R5,RAND      ;GET A DRIVE?(LUN)
1061 012620 013702 002136      MOV     LONUM,R2        ;GET THE SELECTED DRIVE (LUN)
1062 012624 043702 002132      PEROTH: BIC    SYMSK,R2  ;MASK TO DRIVES ON SYSTEM
1063 012630 012701 000001      MOV     #1,R1           ;LET'S SEE IF DRIVE IS THERE
1064 012634 005702              1$:  TST     R2           ;HAVE WE GOT PROPER MASK YET
1065 012636 001403              BEQ     2$              ;YES, GO TO 2$
1066 012640 006301              ASL     R1              ;NO, SHIFT FOR NEXT DRIVE
1067 012642 005302              DEC     R2              ;DECREMENT DRIVE NUMBER
1068 012644 000773              BR     1$              ;GO CHECK NEW DRIVE NUMBER
1069 012646 105737 002130      2$:  TSTB    DRUT        ;ANY DRIVES ON LINE
1070 012652 001005              BNE     5$              ;YES, CHECK
1071
1072 012654              ERRSF   170.,NODRIV     ;NO DRIVES
      (3) 012654 104421              TRAP   T$ERCODE
      (5) 012656 000252              .WORD  170
      (5) 012660 003622              .WORD  NODRIV
1073
1074 012662 000137 025050              JMP     END$PROGRAM
1075
1076 012666 130137 002130      5$:  BITB    R1,DRUT     ;IS THIS DRIVE PRESENT?
1077 012672 001750              BEQ     MAIN           ;NO, GO BACK TRY AGAIN
1078
1079              ;WE NOW HAVE A DRIVE, CHECK TO SEE IF IT'S CONTROLLER
1080              ;IS FREE BEFORE WE GO ANY FURTHER
1081
1082
1083
1084 012674 004537 022356              JSR     R5,GETSYS      ;GET PRESENT TIME OF SYSTEM
1085 012700 023737 002236 007516      CMP     INTERVAL,TYINT ;TIME TO PRINT REPR:
1086 012706 002403              BLT     6$              ;NO, PERFORM FUNCTION
1087 012710 005037 002236      CLR     INTERVAL      ;YES, START INTERVAL OVER
1088
1089 012714              DORPT                    ;PRINT STATISTICAL REPORT
      (3) 012714 104024              EMT     C$DRPT
1090
1091 012716 012704 025056      6$:  MOV     #DRBUF,R4    ;GET START OF DRIVE BUFFERS
1092 012722 013702 002136      MOV     LONUM,R2        ;GET RANDOM DRIVE BACK (LUN)
1093 012726 043702 002132      BIC    SYMSK,R2        ;MASK TO SYSTEM SYS
1094 012732 005702              3$:  TST     R2           ;DO WE HAVE BUFFER FOR THAT DRIVE
1095 012734 001404              BEQ     4$              ;YES, GO CHECK IT'S CONTROLLER
  
```

```

1096 012736 062704 000124      ADD    #PRPOS+2,R4      ;NO, UPDATE FOR NEXT BUFFER
1097 012742 005302      DEC    R2              ;DOWN COUNT DRIVE NUMBER (LUN)
1098 012744 000772      BR     3$              ;GO BACK AND CHECK FOR FOUND
1099 012746 032774 000200 000104 4$:  BIT    #BIT7,@DCS(R4) ;CONTROLLER ASSOCIATED WITH DRIVE
1100 012754 001717      BEQ   MAIN
1101 012756 032774 000100 000104  BIT    #BIT6,@DCS(R4) ;INTERRUPT BEEN SERVICED?
1102 012764 001313      BNE   MAIN
1103
1104      ;WE CAN NOW PROCEED IN GETTING A FUNCTION AND RELATED DATA
1105      ;FOR THE DRIVE RANDOMLY. R4 HAS DRIVE BUFFER POINTER
1106
1107 012766 005737 007546      TAGX:  TST    T.DRP      ;DROP ON ERROR LIMITS REACHED?
1108 012772 001456      BEQ   8$              ;NO
1109 012774 026437 000012 007506      CMP    ERRCNT(R4),ERLMT ;HARD REACHED?
1110 013002 103404      BLO   9$
1111 013004 012737 003125 002126      MOV    #ERLMTM,WHY
1112 013012 000442      BR     11$
1113 013014 026437 000014 007552 9$:  CMP    SFTCNT(R4),SFLMT ;SOFT REACHED?
1114 013022 103404      BLO   10$
1115 013024 012737 003170 002126      MOV    #SFEMSG,WHY
1116 013032 000432      BR     11$
1117 013034 026437 000074 007600 10$:  CMP    DATCER(R4),T.DCD
1118 013042 103404      BLO   110$
1119 013044 012737 003212 002126      MOV    #DCDMSG,WHY
1120 013052 000422      BR     11$
1121 013054 016401 000016      110$: MOV    SKECNT(R4),R1
1122 013060 066401 000072      ADD    TRERR(R4),R1
1123 013064 020137 007510      CMP    R1,SELMT
1124 013070 103404      BLO   7$
1125 013072 012737 003147 002126      MOV    #SERLMT,WHY
1126 013100 000407      BR     11$
1127 013102 026437 000020 007556 7$:  CMP    DERCNT(R4),DRLMT ;DRIVE ERROR REACHED?
1128 013110 103407      BLO   8$
1129 013112 012737 003235 002126      MOV    #DERMSG,WHY
1130 013120 004537 020220      11$:  JSR   R5,DRDRV      ;DROP THIS DRIVE!!!
1131 013124 000137 012614      JMP   MAIN            ;GO GET ANOTHER
1132
1133 013130 005764 000076      8$:  TST    DOWCK(R4)      ;WRITE CHECK NEEDED
1134 013134 001407      BEQ   80$            ;NO
1135 013136 016404 000076      MOV    DOWCK(R4),R4  ;GET ONE THAT NEEDS TO BE WRCHK'D
1136 013142 012764 000002 000044      MOV    #WRCHK,FUNC(R4) ;WRITE CHECK
1137 013150 000137 014052      JMP   ISSUE          ;ISSUE IT
1138 013154 005764 000036      80$: TST    RETRY(R4)     ;DOES DRIVE HAVE RETRY IN
1139 013160 001402      BEQ   78$            ;PROGRESS, NO CONTINUE
1140 013162 000137 014052      JMP   ISSUE          ;GO RETRY COMMAND
1141
1142 013166 005764 000114      78$: TST    RSEEK(R4)   ;RECOVERY FROM SEEK ERROR
1143 013172 001003      BNE   77$            ;NO
1144 013174 000412      BR     GETFNC         ;NO, CONTINUE
1145 013176 000137 014002      JMP   RDDFNC         ;GO READ
1146 013202 032764 000001 000056 77$:  BIT    #SKDON,PRFLGS(R4) ;SEEK BEEN VERIFIED
1147 013210 001002      BNE   79$            ;NO
1148 013212 000137 013366      JMP   SKFNC          ;GO, TRY TO RECOVER
1149 013216 000137 013712      79$:  JMP   RDHFNC         ;GO VERIFY SEEK
1150
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 BEQ 8$ ;NO, CONTINUE
1155 013232 000137 013712 JMP RDHFNC ;GO VERIFY SEEK
1156
1157
1158
1159 013236 005737 007554 8$: TST T,STA ;DO WE WISH TO DROP ON OPR LIMITS
1160 013242 001422 BEQ 98$ ;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 016400 000060 MOV RXFR3(R4),R0 ;GET READ COUNT
1165 013260 066400 000062 ADD WXFR3(R4),R0 ;ADD IN WRITE COUNT
1166 013264 020037 007512 CMP R0,DALMT ;LIMIT REACHED??
1167 013270 103407 BLO 98$ ;NO, THEN GO TEST
1168 013272 012737 003414 002126 MOV #SOPLMT,WHY
1169 013300 004537 020220 JSR R5,DRDRV ;DROP THE DRIVE
1170 013304 000137 012614 JMP MAIN ;GO FOR ANOTHER DRIVE
1171
1172 013310 004537 021214 98$: JSR R5,RAND ;GET FUNCTION, LEGAL FUNCTIONS
1173 ;ARE: 1 (WRITE CHECK)
1174 ; : 2 (GET STATUS)
1175 ; : 3 (SEEK)
1176 ; : 4 (RD HEADER)
1177 ; : 5 (WRITE)
1178 ; : 6 (READ)
1179 ;0 & 7 ARE NOT LEGIT
1180 013314 013702 002136 MOV LONUM,R2 ;GET IT
1181 013320 042702 177770 BIC #177770,R2 ;MASK TO 0-7
1182 013324 001001 BNE 6$ ;IF 0, MAKE 1
1183 013326 005202 INC R2
1184 013330 022702 000007 6$: CMP #7,R2 ;IS IT 7?
1185 013334 001001 BNE 5$ ;IF 7, MAKE 6
1186 013336 005302 DEC R2
1187 013340 006302 5$: ASL R2 ;SHIFT LEFT (X2)
1188 013342 000172 017470 JMP @LIST(R2) ;GO TO FUNCTION ROUTINE
1189
1190
1191 .SBTTL ROUTINE TO SETUP AND ISSUE GET STATUS
1192
1193 ;WE GET HERE BY FALLING THRU 'LIST' WITH A RANDOM FUNCTION OF 2.
1194
1195 013346 012764 000004 000044 GSTFNC: MOV #GSTAT,FUNC(R4) ;LOAD GET STATUS
1196 013354 012764 000003 000040 MOV #GSBIT,BDA(R4) ;SET GSBIT IN COMMAND WORD
1197 013362 000137 014052 JMP ISSUE ;GC ISSUE FUNCTION
1198
1199 .SBTTL ROUTINE TO SETUP AND ISSUE SEEK FUNCTION
1200
1201 ;WE GET HERE BY FALLING THRU 'LIST' WITH A RANDOM FUNCTION OF 3.
1202 ;WE WILL CALL 'RAND' FOR A NEW DISK ADDRESS TO SEEK
1203 ;TO ANY TRACK BUT LAST IS LEGAL. WE WILL ALSO INCREMENT
1204 ;IT'S SEEK COUNT
1205
1206 013366 005764 000114 SKFNC: TST RCEEK(R4) ;TRYING TO RECOVER
1207 013372 001411 BEQ 98$ ;NO, CONTINUE

```

```
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      JSR     R5,RAND        ;GET A RANDOM NUMBER
1214 013422 013702 002136      MOV     LONUM,R2      ;
1215 013426 043702 002142      BIC     SECMSK,R2     ;LEAVE CYL AND HEAD
1216 013432 020264 000122      CMP     R2,PRPOS(R4)  ;ON THAT TRACK ALREADY
1217 013436 001002              BNE     90$           ;NO, CONTINUE
```

98\$:

90\$

```

1219 013440 000137 013222          JMP      GETFNC          ;YES, DON'T RESEEK
1220 013444 005003          90$:    CLR      R3
1221 013446 010200          MOV     R2,R0          ;COPY
1222 013450 042700 177677          BIC     #177677,R0     ;LEAVE ONLY HEAD
1223 013454 023737 007534 007536          CMP     T,MXC,T,MNC   ;MIN AND MAX CYLINDERS THE SAME
1224 013462 001003          BNE     95$           ;NO, BRANCH AND STAY IN LIMITS
1225 013464 013702 007534          MOV     T,MXC,R2      ;MAKE CYLINDER MAX/MIN
1226 013470 000430          BR      92$           ;GO CLCULATE DIFF AND SEEK
1227 013472 042702 000100          95$:    BIC     #HEAD,R2   ;STRIP OUT H.S. BIT
1228 013476 023702 007534          94$:    CMP     T,MXC,R2 ;IS ADDRESS LESS/EQUAL THAN MAX
1229 013502 103010          BHIS   93$           ;YES, CHECK LOW END
1230 013504 005203          INC     R3
1231 013506 020327 000012          CMP     R3,#10.
1232 013512 001741          BEQ     98$
1233 013514 006202          ASR     R2            ;HALF IT AND CHECK AGAIN
1234 013516 062702 000200          91$:    ADD     #BIT7,R2 ;JUST TO MAKE NON ZERO
1235 013522 000763          BR      95$           ;GO BACK AND CHECK AGAIN
1236 013524 023702 007536          93$:    CMP     T,MNC,R2 ;IS MIN GREATER/EQUAL THAN ADDRESS
1237 013530 101410          BLOS   92$           ;YES, CALCULATE DIFF AND SEEK
1238 013532 005203          INC     R3
1239 013534 020327 000012          CMP     R3,#10.
1240 013540 001726          BEQ     98$
1241 013542 006302          ASL     R2            ;NO, DOUBLE IT
1242 013544 042702 100000          BIC     #BIT15,R2     ;BIT 15 CAN'T SET
1243 013550 000762          BR      91$           ;GO CHECK MAX/MIN AGAIN
1244 013552 016401 000122          92$:    MOV     PRPOS(R4),R1 ;GET PRESENT DISK POSITION
1245 013556 043701 002140          BIC     CYLMSK,R1    ;CLEAN OUT ITS SECTOR BITS
1246
1247 013562 016464 000122 000050          MOV     PRPOS(R4),LSTHDR(R4) ;SAVE LAST
1248 013570 010264 000122          MOV     R2,PRPOS(R4) ;NEW HEADER AFTER SEEK
1249 013574 050064 000122          BIS     R0,PRPOS(R4) ;SET IN RANDOM HEAD GOTTEN
1250 013600 023737 007530 007532          CMP     T,MXH,T,MNH   ;MIN AND MAX HEAD SELECT THE SAME
1251 013606 001012          BNE     96$           ;NO, THEN WE CAN USE BOTH SURFACES
1252 013610 005737 007530          TST     T,MXH         ;WHICH IS OUR SURFACE FOR USE
1253 013614 001004          BNE     97$           ;TOP SURFACE BRANCH
1254 013616 042764 000100 000122          BIC     #HEAD,PRPOS(R4) ;LOWER SURFACE ONLY
1255 013624 000403          BR      96$
1256 013626 052764 000100 000122          97$:    BIS     #HEAD,PRPOS(R4) ;TOP SURFACE ONLY
1257 013634
1258
1259
1260
1261
1262 013634 160102          4$:    SUB     R1,R2      ;SUBTRACT PRESENT FROM NEXT
1263 013636 100002          BPL     1$           ;IF POSITIVE RESULT GO TO 1$
1264 013640 005402          NEG     R2           ;NEG RESULT, NEGATE IT
1265 013642 000402          BR      2$           ;GO SET DIRECTION OUT
1266 013644 052702 000004          1$:    BIS     #SIGN,R2 ;DIRECTION OUT, MARKER
1267 013650 052702 000001          2$:    BIS     #MK,R2  ;MARKER BIT
1268 013654 032764 000100 000122          BIT     #HEAD,PRPOS(R4) ;WHICH SURFACE SELECTED?
1269 013662 001402          BEQ     3$           ;TOP, THEN 3$
1270 013664 052702 000020          BIS     #SKHS,R2     ;BOTTOM SET HEAD BIT
1271 013670 010264 000040          3$:    MOV     R2,BDA(R4) ;MOVE DIFFERENCE WORD TO DA
1272 013674 010264 000066          MOV     R2,DIFWD(R4) ;LOAD DIFFERENCE WORD
1273 013700 012764 000006 000044          MOV     #SEEK,FUNC(R4) ;LOAD SEEK
1274 013706 000137 014052          JMP     ISSUE

```

```

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 WRTFNC: CMP    #77700,PRPOS(R4) ;ON LAST TRACK?
1288 013732 001002      BNE    98$      ;NO, CONTINUE
1289 013734 000137 013366      JMP    SKFNC    ;YES, WE'LL SEEK OFF IT!.
1290 013740 005737 007560 98$:  TST    T,ROF    ;READ ONLY
1291 013744 001402      BEQ    97$      ;NO
1292 013746 000137 014002      JMP    RDDFNC   ;YES
1293 013752 004537 022576 97$:  JSR    R5,GWCDA ;GET WORD COUNT,DA
1294      ;
1295      ;WE NOW HAVE SECTOR AND WORD COUNT, LET'S WRITE BUFFER IN MEMORY
1296      ;TO WRITE OUT TO DISK
1297      ;FORMAT:      WORD 1 - # OF WORDS IN SECTOR
1298      ;              :      WORD 2 - ADDRESS OF PATTERN WRITTEN ON SECTOR
1299      ;              :      WORD 3 - 127 DATA PATTERN
1300      ;
1301
1302 013756 004537 017230      JSR    R5,WRBUF ;WRITE BUFFER INTO MEMORY
1303 013762 012764 000012 000044 MOV    #WRITE,FUNC(R4) ;LOAD WRITE
1304 013770 012764 000001 000120 MOV    #1,WRIPG(R4) ;SET WRITE IN PROGRESS FLAG
1305 013776 000137 014052      JMP    ISSUE    ;GO ISSUE FUNCTION
1306
1307      .SBTTL  ROUTINE TO LOAD READ DATA COMMAND
1308
1309      ;THIS ROUTINE WILL FIRST CLEAR OUT THE BUFFER AREA,
1310      ;SELECT A RANDOM NUMBER OF WORDS TO READ AND A
1311      ;RANDOM SECTOR ON THE PRESENT CYLINDER TO READ FROM
1312
1313 014002 022764 077700 000122 RDDFNC: CMP    #77700,PRPOS(R4) ;ON LAST TRACK?
1314 014010 001002      BNE    99$      ;NO CONTINUE
1315 014012 000137 013366      JMP    SKFNC    ;YES SEEK OFF IT.
1316 014016 004537 022576 99$:  JSR    R5,GWCDA ;GET WORD COUNT, DA
1317 014022 016402 000042 97$:  MOV    BMP(R4),R2 ;CLEAR OUT BUFFER AREA
1318 014026 017401 000110      MOV    @BBA(R4),R1 ;SO WE KNOW READ
1319 014032 005021 1$:  CLR    (R1)+    ;WORKED!!
1320 014034 005202      INC    R2
1321 014036 001375      BNE    1$
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 014052 026437 000104 002150 ISSUE: CMP DCS(R4),CNTLR1 ;DRIVE ON CONTROLLER 1?
1334 014060 001003 BNE 1$ ;NO, ASSUME ON CONTROLLER 2
1335 014062 010437 002154 MOV R4,LSTDR1 ;PUT BUFFER POINTER IN 1
1336 014066 000402 BR 2$ ;SKIP OVER NEXT INSTRUCTION
1337 014070 010437 002156 1$: MOV R4,LSTDR2 ;PUT BUFFER POINTER IN 2
1338 014074 052764 000100 000044 2$: BIS #INTEN,FUNC(R4) ;ALLOW INTERRUPTS
1339 014102 004537 014112 JSR R5,LDFUNC ;NO WE ISSUE IT
1340 014106 000137 012614 JMP MAIN ;GO BACK AND DO ANOTHER
1341
1342
1343
1344
1345
1346
1347
1348

```

```

.SBTTL ROUTINE TO LOAD FUNCTION
:CALL JSR R5,LDFUNC
:ALL INFORMATION MUST BE SET UP IN DRIVE BUFFER
,R4 HAS POINTER TO BUFFER

```

```

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 BNE 1$ ;YES, GO ISSUE COMMAND
1352
1353 014124 ERRSF 200,PRGER ;THIS ERROR SHOULD NEVER PRINT
(3) 014124 104421 TRAP T$ERCODE
(5) 014126 000310 .WORD 200
(5) 014130 002535 .WORD PRGER
1354
1355 014132 017463 000110 000002 1$: MOV @BBA(R4),BA(R3) ;LOAD BUS ADDRESS REGISTER
1356 014140 016463 000040 000004 MOV BDA(R4),DA(R3) ;LOAD DISK ADDRESS REGISTER
1357 014146 016463 000042 000006 MOV BMP(R4),MP(R3) ;LOAD MULTI-PURPOSE REGISTER
1358 014154 016464 000044 000046 MOV FUNC(R4),BCSADR(R4) ;GET FUNCTION
1359 014162 056464 000106 000046 BIS DRSEL(R4),BCSADR(R4) ;SET DRIVE SELECT BITS
1360 014170 052764 000201 000046 BIS #CRDY!DRDY,BCSADR(R4) ;SET CRDY:DRDY IN IMAGE
1361 014176 042764 002000 000046 BIC #OPI,BCSADR(R4) ;WE'RE CLEAR BIT 10 FOR DRIVE 7-4 (OKAY?)
1362 014204 016463 000046 000000 MOV BCSADR(R4),CS(R3) ;LOAD CSR
1363 014212 042763 000200 000000 BIC #CRDY,CS(R3) ;ISSUE FUNCTION
1364 014220 000205 RTS R5 ;EXIT
1365
1366
1367
1368 014222
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383

```

```

.SBTTL INTERRUPT SERVICE ROUTINES
BGNSRV INTR1
:ON INTERRUPT WE CHECK FOR ERRORS FIRST, IF NO ERRORS WE
:CHECK FUNCTION PREFORMED. WE ACT ACCORDING IF FUNCTION IS:
: 1- WRITE CHECK - NOTHING IF NO ERROR
: 2- GET STATUS - READ AND CHECK DRIVE STATUS
: 3- SEEK - NOTHING RTI; SET RD HDR AS NEXT COMMAND
: 4- RDHDR - COMPARE HEADER TO PRESENT POSITION
: 5- WRITE - UPDATE XFER COUNT, EXIT
: 6- READ - COMPARE DATA IF REQUESTED, UPDATE XFER COUNT, EXIT
:ALL SUCCESSFUL EXITS FROM INTERRUPT ROUTINE TEST RETRY
:LIMIT IF RETRY IS LESS THEN LIMIT THEN LOG SOFT ERROR, CLEAR RETRY
: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
 1404 014224 013704 002154
 1405 014230 000403
 1406 014232 010446
 1407 014234 013704 002156
 1408 014240 013746 002250
 1409 014244 013746 002252
 1410 014250 013746 002254
 1411 014254 013746 002256
 1412 014260 013746 002260
 1413 014264 013746 002262
 1414 014270 013746 002172
 1415 014274 013746 002170
 1416 014300 013746 002200
 1417 014304 013746 002126
 1418 014310 013746 002324
 1419 014314 013746 002326
 1420 014320 010346
 1421 014322 010246
 1422 014324 010146
 1423 014326 010046
 1424 014330 005064 000120
 1425 014334 016403 000104
 1426 014340 016337 000000 002250
 1427 014346 016337 000002 002252
 1428 014354 016337 000004 002254
 1429 014362 016337 000006 002256
 1430 014370 016337 000006 002260
 1431 014376 016337 000006 002262
 1432 014404 005737 002250
 1433 014410 100402
 1434 014412 000137 015532
 1435
 1436
 1437
 1438
 1439

```

:ON ERRORS - IF DRIVE ERROR - UNDER NON-INTERRUPT
DO: GET STATUS - INVESTIGATE ERROR TYPE

DO: DRIVE RESET - IF ERROR OCCURS AGAIN - FATAL ERROR
IF NO ERROR, EXIT
DRIVE ERROR IS LOGGED UNDER ALL CIRCUMSTANCES

IF DCRC, HCRC, HNF CHECK BAD SECTOR LIST, IF IN LIST
IGNORE ERROR EXIT AS NORMAL, IF NOT IN LIST
INCREMENT RETRY; IF RETRY LIMIT EXCEEDED
LOG HARD ERROR, ELSE RETRY FUNCTION

IF OPI,NXM INCREMENT RETRY CHECK RETRY LIMIT
IF RETRY EXCEEDED LOG HARD ERROR EXIT
IF RETRY NOT EXCEEDED RETRY FUNCTION

INTR1: MOV R4,-(SP) ;SAVE PRESENT R4 VALUE
MOV LSTDR1,R4 ;GET THE DRIVE BUFFER OF INTERRUPTING DRIVE
BR SAVE ;GO SAVE RC-R3

INTR2: MOV R4,-(SP) ;SAVE PRESENT R4 VALUE
MOV LSTDR2,R4 ;GET THE DRIVE BUFFER OF INTERRUPTING DRIVE

SAVE: MOV E.CS,-(SP)
MOV E.BA,-(SP)
MOV E.DA,-(SP)
MOV E.MP,-(SP)
MOV E.MP1,-(SP)
MOV E.MP2,-(SP)
MOV CHKSEC,-(SP)
MOV HDRFND,-(SP)
MOV TEMP1,-(SP)
MOV WHY,-(SP)
MOV OPCALL,-(SP)
MOV INCALL,-(SP)
MOV R3,-(SP) ;SAVE R3
MOV R2,-(SP) ;R2
MOV R1,-(SP) ;R1
MOV R0,-(SP) ;R0
CLR WRIPG(R4) ;CLEAR THE WRITE IN PROGRESS FLAG
MOV DCS(R4),R3 ;GET CSR FOR INTERRUPT
MOV CS(R3),E.CS ;SAVE ALL REGISTERS NOW.!
MOV BA(R3),E.BA
MOV DA(R3),E.DA
MOV MP(R3),E.MP
MOV MP(R3),E.MP1
MOV MP(R3),E.MP2
TST E.CS ;ANY ERRORS?
BMI 1$ ;YES, GO SOLVE ERROR MYSTERY
JMP CHKFNC ;NO, GO SEE IF WE HAVE TO DO ANYTHING

.SBTTL CONTROLLER ERROR CHECK ROUTINE

;WE HAVE SOME SORT OF ERROR LET'S FIND OUT WHICH ONE
;IT IS.
  
```

```

1440
1441 014416 013764 002254 000064 1$: MOV E.DA,LSTDA(R4) ;SAVE DA FOR SOFT ERROR PRINT
1442 014424 032737 040000 002250 BIT #DERR,E.CS ;DRIVE ERROR?
1443 014432 001402 BEQ 2$ ;NO, CONTINUE
1444 014434 000137 016514 JMP CKDERR ;YES, GO CHECK DRIVE ERROR
1445 014440 032737 000001 002250 2$: BIT #DRDY,E.CS ;DRIVE READY THERE
1446 014446 001017 BNE 23$ ;YES, CONTINUE CHECKING
1447 014450 004537 021122 JSR R5,GETDST ;NO,GET DRIVE STATUS
1448 014454 042701 000100 BIC #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 ERRDF 1000.,NORDY,ERR9
(3) 014472 104462 TRAP T$ERCODE
(5) 014474 001750 .WORD 1000
(5) 014476 002507 .WORD NORDY
(5) 014500 005050 .WORD ERR9
1454
1455 014502 000137 016344 JMP EXIT1
1456
1457 014506 032737 020000 002250 23$: BIT #NXM,E.CS ;NON-EXISTANT MEMORY?
1458 014514 001407 BEQ 3$ ;NO, KEEP CHECKING
1459 014516 012764 004153 000052 MOV #MTNXM,RTYPE(R4) ;ERROR MESSAGE
1460 014524 005264 000034 INC NXMCNT(R4) ;LOG ERROR
1461 014530 000137 015136 JMP 111$ ;CHECK RETRY, EXIT BACK
1462
1463 014534 032737 014000 002250 3$: BIT #BIT12!BIT11,E.CS ;QUALIFING BITS SET?
1464 014542 001020 BNE 5$ ;YES, CAN'T BE OPI ALONE
1465
1466 014544 032737 002000 002250 BIT #OPI,E.CS ;OPI SET?
1467 014552 001006 BNE 4$ ;YES, CONTINUE
1468
1469 014554 ERRSF 10.,UDERR,ERR1 ;WE HAVE AN UNDIAGNOSABLE CONDITION, ONLY COMPOSITE SET
(3) 014554 104461 TRAP T$ERCODE
(5) 014556 000012 .WORD 10
(5) 014560 002612 .WORD UDERR
(5) 014562 004304 .WORD ERR1
1470 014564 33$: BREAK
(3) 014564 104022 EMT ($BRK
1471 014566 000776 BR 33$
1472
1473
1474 014570 012764 004146 000052 4$: MOV #MTOPI,RTYPE(R4);SET UP FOR 'OPI' PRINT
1475 014576 005264 000030 INC OPICNT(R4) ;LOG ERROR
1476 014602 000555 BR 111$ ;CHECK RETRY EXIT BACK
1477
1478 ;WE KNOW IT'S NOW EITHER DLT, DCRC,HNF, OR HCRC
1479 ;CHECK FOR EACH
1480
1481 014604 032737 002000 002250 5$: BIT #OPI,E.CS ;OPI QUALIFIER SET?
1482 014612 001060 BNE 7$ ;YES, THEN IT'S HCRC OR HNF
1483
1484 ;IT'S NOW DOWN TO DLT OR DCRC
1485
1486 014614 032737 010000 002250 BIT #DLT,E.CS ;DATA LATE?

```

```

1487 014622 001406          BEQ      6$          ;NO, MUST BE DATA CRC
1488 014624 012764 004141 000052    MOV      #MTDLT,RTYPE(R4);SET UP FOR 'DLT'PRINT
1489 014632 005264 000026          INC      DLT CNT(R4)   ;LOG ERROR
1490 014636 000537          BR       111$        ;CHECK RETRY, EXIT
1491
1492 014640 013737 002254 002172 6$:    MOV      E.DA,CHKSEC   ;SET UP SECTOR TO LOOK FOR
1493 014646 005364 000064          DEC      LSTDA(R4)    ;DOWN COUNT FOR PRINT OUT
1494 014652 005337 002172          DEC      CHKSEC       ;DOWN COUNT FOR LOOP UP
1495 014656 004537 023744          JSR      R5,CKBDSC    ;CHECK BAD SECTOR LIST
1496 014662 005737 002170          TST      HDRFND       ;WAS HEADER THERE?
1497 014666 001115          BNE     110$         ;IGNORE ERROR, RETURN
1498 014670 005264 000022          INC      DCRCER(R4)   ;ACCOUNT FOR ERROR
1499 014674 012764 004134 000052    MOV      #MTDCRC,RTYPE(R4);SET UP FOR 'DCRC' PRINT
1500 014702 022764 000102 000044    CMP      #INTEN!WRCHK,FUNC(R4)
1501 014710 001001          BNE     118$
1502 014712 000511          BR       111$
1503
1504 014714 005737 007544          118$:   TST      T.DCK         ;DUMP BUFFER?
1505 014720 001506          BEQ     111$         ;NO, EXIT
1506 014722          PRINTF #FMT14,#DMPDCK
      (8) 014722 012746 003070    MOV      #DMPDCK,-(SP)
      (7) 014726 012746 006507    MOV      #FMT14,-(SP)
      (6) 014732 012746 000002    MOV      #2,-(SP)
      (3) 014736 010600          MOV      SP,R0
      (4) 014740 104017          EMT      C$PNTF
      (4) 014742 062706 000006    ADD      #6,SP
1507 014746 004537 023104          JSR      R5,DMPBUF    ;DUMP BUFFER
1508
1509 014752 000471          BR       111$        ;EXIT
1510
1511          ;IT'S NOW EITHER HNF OR HCRC.
1512          ;IF HCRC AND RDHDR, DETERMINE IF BAD SECTOR BY DOING 40 RDHDRS
1513          ;IF HCRC AND R/W, CHECK IF DA IS IN BAD SECTOR FILE
1514          ;IF HNF READ HEADER TO VERIFY IF ON CORRECT CYLINDER
1515          ;THEN IF ON CORRECT CYLINDER SEE IF DA IS A BAD SECTOR
1516          ;IF NOT ON CORRECT CYLINDER REPORT MISSEK, LOG MISEEK
1517          ;AND PRESENT POSITION UPDATE.
1518
1519 014754 032737 010000 002250 7$:    BIT      #HNF,E.CS    ;HEADER NOT FOUND SET?
1520 014762 001466          BEQ     112$        ;NO IT MUST BE HCRC
1521 014764 012701 000051          MOV      #41,R1      ;ALLOW FOURTY READ HEADERS TO
1522 014770 004537 021136          8$:    JSR      R5,ISDRST
1523 014774 016402 000106          MOV      DRSEL(R4),R2 ;FIND CYLINDER
1524 015000 052702 000010          BIS      #RDHDR,R2   ;READ HEADER
1525 015004 016403 000104          MOV      DCS(R4),R3
1526 015010 010263 000000          MOV      R2,CS(R3)   ;ISSUE READ HEADER
1527 015014 004537 021052          JSR      R5,WTRDY    ;WAIT
1528 015020 005301          DEC      R1          ;DONE 40 OF THESE?
1529 015022 001422          BEQ     9$          ;YES, GIVE UP WE DON'T HAVE ALL
1530 015024 005763 000000          TST     CS(R3)      ;DAY, IS ERROR SET?
1531 015030 100757          BMI     8$          ;YES, GO DO IT AGAIN
1532
1533 015032 016301 000006          MOV      MP(R3),R1    ;GET HEADER
1534 015036 043701 002142          BIC     SECMSK,R1    ;MASK OUT SECTOR BITS
1535 015042 020164 000122          CMP     R1,PRPOS(R4) ;IS CYLINDER HEAD CORRECT?
1536 015046 001415          BEQ     10$         ;YES, GO CHECK BAD SECTOR LIST
    
```



```

1537
1538
1539 015050 005264 000072      INC      TRERR(R4)
1540 015054      ERRHRD 20.,TRACK,ERR2 ;TRACKING DRIFT ERROR
      (3) 015054 104463      TRAP      T$ERCODE
      (5) 015056 000024      .WORD    20
      (5) 015060 003110      .WORD    TRACK
      (5) 015062 004312      .WORD    ERR2
1541
1542
1543 015064 000137 016034      JMP      SKRETRY      ;FIX TRACKING ERROR
1544
1545
1546 015070      9$:      ERRHRD 30.,EXHAUS,ERR' ;WE CAN'T FIND GOOD HEADER ON THIS TRACK
      (3) 015070 104463      TRAP      T$ERCODE
      (5) 015072 000036      .WORD    30
      (5) 015074 002576      .WORD    FXHAUS
      (5) 015076 004304      .WORD    ERR1
1547
1548 015100 000410      BR      110$
1549
1550 015102 013737 002254 002172 10$:      MOV      E.DA,CHKSEC
1551 015110 004537 023744      JSR      R5,CKBDSC      ;GO CHECK BAD SECTOR FILE
1552 015114 005737 002170      TST      HDRFND      ;WAS IT THERE
1553 015120 001401      BEQ      11$          ;NO, LOG IT EXIT
1554 015122 000577      110$:     BR      GOERRX      ;YES IGNORE ERROR
1555
1556 015124 005264 000032      11$:      INC      HNFERR(R4)      ;LOG IT
1557 015130 012764 004121 000052      MOV      #MTHNF,RTYPE(R4);SET UP FOR 'HNF' PRINT
1558 015136 000573      111$:     BR      GOFIN       ;EXIT
1559
1560
1561
1562      ;IT WAS A HEADER CRC ERROR, FIGURE OUT IF IT WAS
1563      ;ON A READ HEADER OR READ/WRITE
1564
1565
1566 015140 022764 000110 000044 112$:     CMP      #INTEN.RDHDR,FUNC(R4) ;READ HEADER?
1567 015146 001417      BEQ      13$          ;YES, GO FIND OUT MORE ABOUT IT
1568      ;NO, IT MUST BE R/W
1569 015150 013737 002254 002172      MOV      E.DA,CHKSEC
1570 015156 004537 023744      JSR      R5,CKBDSC      ;BAD SECTOR SEARCH
1571 015162 005737 002170      TST      HDRFND      ;WAS OUR DA THERE?
1572 015166 001401      BEQ      12$          ;NO, MUST BE LEGIT ERROR
1573 015170 000554      BR      GOERRX      ;YES, IGNORE ERROR
1574
1575 015172 005264 000024      12$:      INC      HRCRC(R4)      ;LOG ERROR
1576 015176 012764 004126 000052      MOV      #MTHCRC,RTYPE(R4)
1577 015204 000550      BR      GOFIN
1578
1579 015206 017401 000110      13$:      MOV      #BSA(R4),R1      ;USE IT'S BUFFER TO STORE HDRS
1580 015212 012737 000050 002200      MOV      #40,TEMP1      ;40 CONSECUTIVE HEADERS
1581 015220 012702 000010      14$:      MOV      #RDHDR,R2      ;READ HEADER
1582 015224 056402 000106      BIS      DRSEL(R4),R2
1583 015230 016403 000104      MOV      DCS(R4),R3
1584 015234 010263 000000      MOV      R2,CS(R3)

```

```

1585 015240 004537 021052      JSR      R5,WTRDY      ;WAIT FOR READY
1586 015244 016321 000000      MOV      CS(R3),(R1)+  ;READ ALL REGISTERS
1587 015250 016321 000006      MOV      MP(R3),(R1)+  ;
1588 015254 016321 000006      MOV      MP(R3),(R1)+  ;
1589 015260 016321 000006      MOV      MP(R3),(R1)+  ;
1590 015264 005337 002200      DEC      TEMP1        ;DONE 40 YET?
1591 015270 001353              BNE      14$          ;NO, GO BACK
1592
1593
1594      ;WE HAVE 40 HEADERS NOW LETS SEE IF WE CAN VERIFY WHETHER
1595      ;OR NOT A BAD SECTOR CAUSED THE ERROR. CHECK FIRST TO SEE
1596      ;IF WE HAVE ANY BAD SECTORS ON THIS TRACK.
1597 015272 017402 000110      99$:     MOV      @BBA(R4),R2    ;GET BUFFER START
1598 015276 012701 000050      MOV      #40.,R1      ;FOURTY HEADERS
1599 015302 032712 002000      15$:     BIT      #OPI,(R2)    ;IS OPI SET IN CS
1600 015306 001403              BEQ      16$          ;NO, WELL CAN'T BE HCRC
1601 015310 032712 004000      BIT      #HCRC,(R2)   ;INSURE HCRC W/OPI
1602 015314 001005              BNE      17$          ;FOUND GO SEE IF IT COMPARES
1603 015316 062702 000010      16$:     ADD      #10,R2      ;NEXT CS IMAGE
1604 015322 005301              DEC      R1           ;DONE 40
1605 015324 001366              BNE      15$
1606 015326 000721              BR       12$
1607
1608 015330 020274 000110      17$:     CMP      R2,@BBA(R4)   ;IS HEADER FIRST ONE?
1609 015334 001046              BNE      21$          ;NO, READ PREVIOUS HEADER
1610
1611
1612
1613
1614 015336 017401 000110      MOV      @BBA(R4),R1
1615 015342 012703 000001      MOV      #1,R3
1616 015346 062701 000010      18$:     ADD      #10,R1
1617 015352 032711 002000      BIT      #OPI,(R1)
1618 015356 001416              BEQ      19$
1619 015360 032711 004000      BIT      #HCRC,(R1)
1620 015364 001413              BEQ      19$
1621 015366 005203              INC      R3
1622 015370 022703 000017      CMP      #15.,R3
1623 015374 001364              BNE      18$
1624
1625
1626 015376 012737 003473 002126      MOV      #MBDMSC,WHY   ;DROP DRIVE DUE TO
1627 015404 004537 020220      JSR      R5,DRDRV     ;MORE THAN 16 BAD SECTORS
1628 015410 000137 016344              JMP      EXIT1
1629
1630
1631 015414 005012              19$:     CLR      (R2)          ;CLEAR THIS CS
1632 015416 062701 000002      ADD      #2,R1        ;GET IT'S HEADER ADDRESS
1633 015422 011102              MOV      (R1),R2      ;GET HEADER
1634 015424 010201              MOV      R2,R1        ;SAVE HEADER
1635 015426 042702 177700      BIC      #177700,R2   ;MASK ONLY SECTOR
1636 015432 160301              SUB      R3,R1        ;BACK UP TO SECTOR WHICH IS BAD
1637 015434 100402              BMI      20$          ;IF MINUS DO MAGIC
1638 015436 160302              SUB      R3,R2        ;NO THEN SUBTRACT IS LEGAL
1639 015440 000421              BR       22$          ;BRANCH TO CHECK FILE
1640 015442 160302      20$:     SJR      R3,R2      ;THIS SUB PRODUCES WRONG ANSWER

```

```

1641 015444 062702 000050          ADD    #50,R2          ;FIX IT UP
1642 015450 000415          BR     22$           ;GO CHECK FILE
1643
1644 015452 005012          21$: CLR    (R2)          ;CLEAR THIS CS OUT
1645 015454 162702 000006          SUB    #6,R2          ;GET PREVIOUS HEADER
1646 015460 011201          MOV    (R2), R1
1647 015462 005201          INC    R1
1648 015464 010102          MOV    R1,R2
1649 015466 042701 177700          BIC    #177700,R1
1650 015472 022701 000050          CMP    #40.,R1
1651 015476 002402          BLT    22$
1652 015500 162702 000050          SUB    #40.,R2
1653 015504 010237 002172          22$: MOV    R2,CHKSEC
1654 015510 004537 023744          JSR    R5,CKBDSC
1655 015514 005737 002170          TST    HDRFND
1656 015520 001664          BEQ    99$
1657 015522 000137 016350          GERRX: JMP    RREX
1658
1659
1660 015526 000137 016452          GOFIN: JMP    FINERR
1661
1662
1663
1664
1665
1666          .SBTTL  COMMAND SERVICE ROUTINES
1667
1668          ;THERE WAS NO ERROR SO.....
1669          ;NOW WE WILL FIND OUT WHICH FUNCTION WE DID TO CAUSE
1670          ;INTERRUPT AND ACT ACCORDINGLY.
1671          ;
1672
1673 015532 016401 000044          CHKFN: MOV    FUNC(R4),R1      ;GET FUNCTION OF DRIVE
1674 015536 006201          ASR    R1                ;ALIGN THE FUNCTION CODE
1675 015540 042701 000040          BIC    #40,R1            ;WIPE OUT INT. ENAB (SHIFTED)
1676 015544 005301          DEC    R1                ;WRITE CHECK??
1677 015546 001004          BNE    2$                ;NO, BRANCH
1678 015550 004537 011422          JSR    R5,CLRWCK
1679 015554 000137 015674          JMP    AFWRCK
1680 015560 005301          2$: DEC    R1                ;GET STATUS?
1681 015562 001555          BEQ    AGSTAT            ;BRANCH IF SO
1682 015564 005301          DEC    R1                ;SEEK?
1683 015566 001416          BEQ    ASEEK            ;BRANCH IF SO
1684 015570 005301          DEC    R1                ;RDHDR?
1685 015572 001470          BEQ    ARDHDR           ;BRANCH IF SO
1686 015574 005301          DEC    R1                ;WRITE?
1687 015576 001002          BNE    1$                ;NO, BRANCH
1688 015600 000137 016222          JMP    AWRITE
1689 015604 005301          1$: DEC    R1                ;READ?
1690 015606 001425          BEQ    AFREAD           ;BRANCH IF SO
1691
1692 015610          ERRSF  210.,PRGER
   (3) 015610 104421          TRAP  T$ERCODE
   (5) 015612 000322          .WORD 210
   (5) 015614 002535          .WORD PRGER
1693

```

```

1694 015616 000000
1695 015620 000137 016312 XEXIT: HALT
1696 015620 000137 016312 XEXIT: JMP EXIT
1697 .SBTTL SEEK
1698
1699 015624 052764 000001 000056 ASEEK: BIS #SKDON,PRFLGS(R4) ;SET SEEK VERIFY NEEDED
1700 015632 005264 000054 INC SKCNT1(R4) ;INCREMENT COUNT
1701 015636 026427 000054 001750 CMP SKCNT1(R4),#1000 ;10(3) REACHED
1702 015644 002404 BLT 99$ ;NO, EXIT
1703 015646 005264 000000 INC SKCNT(R4) ;YES, BUMP THOUSANDS
1704 015652 005064 000054 CLR SKCNT1(R4)
1705 015656 000137 016350 99$: JMP ERREX
1706
1707 .SBTTL READ
1708
1709 015662 AFREAD: SETPRI #340
(3) 015662 012700 000340 MOV #340,R0
(3) 015666 104041 EMT C$SPRI
1710 015670 004537 020500 JSR R5,CKDATA ;CHECK DATA
1711 015674 AFWRCK:
1712 015674 016401 000042 1$: MOV BMP(R4),R1 ;BUMP UP XFER COUNT
1713 015700 005401 NEG R1 ;MAKE POSITIVE
1714 015702 060164 000002 ADD R1,RXFR1(R4) ;ADD THE BITS
1715 015706 022764 023420 000002 CMP #10000.,RXFR1(R4) ;10(8) REACHED YET
1716 015714 101016 BHI 2$ ;NO, EXIT
1717 015716 005264 000004 INC RXFR2(R4) ;BUMP 10(10)
1718 015722 162764 023420 000002 SUB #10000.,RXFR1(R4) ;START 10(8) AT 0
1719 015730 022764 023420 000004 CMP #10000.,RXFR2(R4) ;10(10) REACHED YET
1720 015736 101005 BHI 2$ ;NO, EXIT
1721 015740 005264 000060 INC RXFR3(R4) ;YES BUMP 65K 10(10)
1722 015744 162764 023420 000004 SUB #10000.,RXFR2(R4) ;MAKE 10(10) 0
1723 015752 000557 2$: BR EXIT ;EXIT
1724
1725 .SBTTL READ HEADER
1726
1727 015754 013701 002256 ARDHDR: MOV E.MP,R1 ;GET HEADER
1728 015760 043701 002142 BIC SECMSK,R1 ;MASK OUT SECTOR BITS
1729 015764 026401 000122 CMP PRPOS(R4),R1 ;IS HEADER CORRECT?
1730 015770 001442 BEQ 1$ ;YES, CONTINUE
1731
1732 015772 032764 000001 000056 BIT #SKDON,PRFLGS(R4) ;IS THIS MIS-SEEK OR TRACKING ERROR
1733 016000 001407 BEQ 2$ ;BRANCH IF TRACKING
1734
1735 016002 005264 000016 INC SKCNT(R4) ;ACCOUNT FOR SEEK ERROR
1736 016006 ERRHRD 50.,MSKER,ERR2
(3) 016006 104463 TRAP T$ERCODE
(5) 016010 000062 .WORD 50
(5) 016012 002634 .WORD MSKER
(5) 016014 004312 .WORD ERR2
1737 016016 000406 BR 3$ ;BRANCH AROUND TRACKING ERROR REPORT
1738
1739 016020 005264 000072 2$: INC TRERR(R4) ;ACCOUNT FOR TRACKING ERROR
1740 016024 ERRHRD 55.,TRACK,ERP2 ;TRACKING ERROR
(3) 016024 104463 TRAP T$ERCODE
(5) 016026 000067 .WORD 55
(5) 016030 003110 .WORD TRACK

```

```

(5) 016032 004312          .WORD  ERR2
1741
1742          016034          SKRETRY=.
1743
1744 016034 005264 000114 007566 3$:  INC  RSEEK(R4) ;SET RETRY IN PROGRESS
1745 016040 026437 000114          CMP  RSEEK(R4),T.SLT ;RETRY EXHAUSTED?????
1746 016046 101405          BLOS 4$ ;NO, THEN RETRY
1747
1748 016050          ERRHRD 333.,SEXHAU,ERR2
(3) 016050 104463          TRAP T$ERCODE
(5) 016052 000515          .WORD 333
(5) 016054 003326          .WORD SEXHAU
(5) 016056 004312          .WORD ERR2
1749 016060 000406          BR 1$
1750
1751 016062 010164 000050 4$:  MOV  R1,LSTHDR(R4) ;SET UP RETRY
1752 016066 042764 000001 000056 BIC  #SKDON,PRFLGS(R4) ;ALLOW SEEK
1753 016074 000506          BR  EXIT ;EXIT
1754 016076 042764 000001 000056 1$:  BIC  #SKDON,PRFLGS(R4) ;SET VERIFICATION DONE
1755 016104 005064 000114          CLR  RSEEK(R4)
1756 016110 010164 000122          MOV  R1,PRPOS(R4) ;MAKE THIS HEADER PRESENT POSITION
1757 016114 000476          BR  EXIT ;EXIT
1758
1759          .SBTTL          GET STATUS
1760
1761 016116 013701 002256          AGSTAT: MOV  E.MP,R1 ;GET STATUS
1762 016122 042701 000100          BIC  #100,R1 ;CLEAR OUT HEAD SELECT
1763 016126 005737 007560          TST  T.ROF ;READ ONLY
1764 016132 001402          BEQ  2$
1765 016134 042701 020000          BIC  #WL,R1
1766 016140 032701 177400          2$:  BIT  #177400,R1 ;ANY BITS WRONG
1767 016144 001406          BEQ  1$ ;NO, CONTINUE
1768
1769 016146 005264 000012          INC  ERRCNT(R4) ;STATUS BITS WRONG
1770 016152          ERRHRD 60.,MDSER,ERR4
(3) 016152 104463          TRAP T$ERCODE
(5) 016154 000074          .WORD 60
(5) 016156 002721          .WORD MDSER
(5) 016160 004514          .WORD ERR4
1771
1772 016162 010102          1$:  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 016174 001446          BEQ  EXIT ;YES, EXIT
1776 016176 022702 000035          CMP  #35,R2 ;COVER CLSD, HEADS OUT, BRUSHES HOME, SEEK LINEAR MODE
1777 016202 001443          BEQ  EXIT ;YES, EXIT
1778
1779 016204 005264 000012          INC  ERRCNT(R4)
1780 016210          ERRHRD 70.,MDSER,ERR4
(3) 016210 104463          TRAP T$ERCODE
(5) 016212 000106          .WORD 70
(5) 016214 002721          .WORD MDSER
(5) 016216 004514          .WORD ERR4
1781
1782 016220 000434          BR  EXIT
1783

```

```

1784          .SBTTL          WRITE
1785
1786 016222 016401 000042      AWRITE: MOV      BMP(R4),R1      ;GET WORD COUNT
1787 016226 005401          NEG      R1              ;MAKE POSITIVE
1788 016230 060164 000006      ADD      R1,WXFR1(R4)    ;ADD THE BITS
1789 016234 022764 023420 000006  CMP      #10000.,WXFR1(R4) ;10(5) YET?
1790 016242 101016          BHI     1$              ;NO, EXIT
1791 016244 005264 000010      INC      WXFR2(R4)      ;YES BUMP 10(10)
1792 016250 162764 023420 000006  SUB      #10000.,WXFR1(R4) ;10(5) GOES TO ZERO
1793 016256 022764 023420 000010  CMP      #10000.,WXFR2(R4) ;10(10) YET?
1794 016264 101005          BHI     1$              ;NO EXIT
1795 016266 005264 000062      INC      WXFR3(R4)      ;INC 65K (10)(10)
1796 016272 162764 023420 000010  SUB      #10000.,WXFR2(R4) ;MAKE 10(10)
1797 016300 005737 007576      1$:   TST      T.WCK          ;PERFORM WRITE CHECK
1798 016304 001402          BEQ     EXIT
1799 016306 004537 011364          JSR     R5,SETWCK
1800
1801 016312 005764 000036      EXIT:  TST      RETRY(R4)      ;IN PROCESS OF RETRYING?
1802 016316 001414          BEQ     ERREX          ;NO
1803 016320 026427 000052 004160  CMP      RTYPE(R4),#MTDRV
1804 016326 001406          BEQ     EXIT1
1805 016330 005264 000014      INC      SFTCNT(R4)      ;YES, LOG SOFT ERROR
1806
1807 016334          ERRSOFT 80.,MSFER,ERR3 ;REPORT SOFT ERROR
   (3) 016334 104464      TRAP   T$ERCODE
   (5) 016336 000120      .WORD  80
   (5) 016340 002645      .WORD  MSFER
   (5) 016342 004376      .WORD  ERR3
1808
1809 016344 005064 000036      EXIT1: CLR      RETRY(R4)      ;CLEAR RETRY
1810
1811 016350 042774 000100 000104  ERREX: BIC      #INTEN,@DCS(R4)
1812 016356 012600          MOV     (SP)+,R0
1813 016360 012601          MOV     (SP)+,R1
1814 016362 012602          MOV     (SP)+,R2
1815 016364 012603          MOV     (SP)+,R3
1816 016366 012637 002326      MOV     (SP)+,INCALL
1817 016372 012637 002324      MOV     (SP)+,OPCALL
1818 016376 012637 002126      MOV     (SP)+,WHY
1819 016402 012637 002200      MOV     (SP)+,TEMP1
1820 016406 012637 002170      MOV     (SP)+,HDRFND
1821 016412 012637 002172      MOV     (SP)+,CHKSEC
1822 016416 012637 002262      MOV     (SP)+,E.MP2
1823 016422 012637 002260      MOV     (SP)+,E.MP1
1824 016426 012637 002256      MOV     (SP)+,E.MP
1825 016432 012637 002254      MOV     (SP)+,E.DA
1826 016436 012637 002252      MOV     (SP)+,E.BA
1827 016442 012637 002250      MOV     (SP)+,E.CS
1828 016446 012604          MOV     (SP)+,R4
1829 016450          ENDSRV
   (3) 016450          L10023:
   (2) 016450 000002      RTI
1830
1831 016452 004537 017450      FINERR: JSR     R5,RCNT      ;CHECK TO SEE IF WE HAVE EXCEEDED
1832 016456 000405          BR     1$              ;RETRY LIMIT, IF SO 1$ AND REPORT HARD
1833 016460 013764 002250 000116  MOV     E.CS,SOFTCS(R4)
  
```

```

1834 016466 000137 016350          JMP      ERREX      ;NOT EXCEEDED EXIT
1835 016472 005264 000012    1$:    INC      ERRCNT(R4) ;INDICATE ERROR
1836
1837 016476          ERRHRD  90.,MHDER,ERR1 ;NON-RECOVERABLE ERROR
   (3) 016476 104463      TRAP    T$ERCODE
   (5) 016500 000132      .WORD   90
   (5) 016502 003055      .WORD   MHDER
   (5) 016504 004304      .WORD   ERR1
1838 016506 004537 011422    JSR     R5,CLRWCK
1839
1840 016512 000714          BR      EXIT1
1841
1842          .SBTTL  DRIVE ERROR SERVICE
1843
1844          ;WE HAVE A DRIVE ERROR, LET'S GET THE STATUS
1845
1846 016514 005264 000020    (KDERR: INC      DERCNT(R4) ;ACCOUNT FOR ERROR
1847 016520 004537 021122    JSR     R5,GETDST ;GET DRIVE STATUS
1848          ;REPORT DRIVE ERROR
1849 016524          ERRHRD  224.,DRVER,ERR9 ;DRIVE ERROR
   (3) 016524 104463      TRAP    T$ERCODE
   (5) 016526 000340      .WORD   224
   (5) 016530 002664      .WORD   DRVER
   (5) 016532 005050      .WORD   ERR9
1850
1851          ;ACT ACCORDINGLY TO DRIVE ERROR
1852
1853 016534 032701 001000          BIT     #VC,R1 ;VOLUME CHECK?
1854 016540 001027          BNE     9$ ;YES, GO ISSUE RESET
1855 016542 032701 010000          BIT     #SKTO,R1 ;SEEK TIME OUT?
1856 016546 001070          BNE     12$ ;YES, ISSUE RESET
1857 016550 032701 144000          BIT     #WDE!HCE!SPE,R1 ;WRITE DATA, CURRENT HEAD, SPINDLE?
1858 016554 001130          BNE     15$ ;GO WAIT FOR HEADS TO UNLOAD
1859 016556 032701 002000          BIT     #WGE,R1 ;WRITE GATE ERROR
1860 016562 001003          BNE     20$ ;YES, ISSUE RESET
1861 016564 004537 021136          JSR     R5,ISDRST ;ISSUE RESET
1862 016570 000431          BR      10$ ;GO CHECK DRIVE READY
1863 016572 004537 021136    20$:  JSR     R5,ISDRST ;ISSUE RESET
1864 016576 004537 021122    JSR     R5,GETDST ;RESET WORK?
1865 016602 032701 002000          BIT     #WGE,R1 ;WGE CLEAR
1866 016606 001422          BEQ     10$ ;YES GO CHECK DRIVE READY
1867 016610 012737 002762 002126  MOV     #WGEST,WHY ;REPORT WGE DIDN'T CLR
1868 016616 000412          BR      91$ ;DROP DRIVE
1869
1870 016620 004537 021136    9$:   JSR     R5,ISDRST ;ISSUE RESET
1871 016624 004537 021122    JSR     R5,GETDST ;RESET WORK
1872 016630 032701 001000          BIT     #VC,R1 ;VOL CHK CLEAR
1873 016634 001407          BEQ     10$ ;YES, CHECK DRIVE READY
1874 016636 012737 002735 002126  MOV     #MV CER,WHY ;DROP THE DRIVE
1875

```

```

1877 016644 004537 020220          91$: JSR    R5,DRDRV
1878 016650 000137 016344          JMP    EXIT1
1879 016654 032763 000001 000000 10$: BIT    #DRDY,CS(R3)      ;DRIVE READY POSTED?
1880 016662 001004          BNE    101$              ;YES, PRINT RECOVERED
1881
1882 016664 012737 002474 002126          MOV    #DNRDY,WHY
1883 016672 000764          BR     91$              ;NO, DROP DRIVE
1884
1885 016674          101$: PRINTB #FMT14,#MRDER      ;PRINT DRIVE RECOVERED
(8) 016674 012746 003015          MOV    #MRDER,-(SP)
(7) 016700 012746 006507          MOV    #FMT14,-(SP)
(6) 016704 012746 000002          MOV    #2,-(SP)
(3) 016710 010600          MOV    SP,R0
(4) 016712 104014          EMT    C$PNTB
(4) 016714 062706 000006          ADD    #6,SP
1886 016720 004537 017156          JSR    R5,GHDR
1887 016724 000137 016452          JMP    FINERR
1888 016730 012702 000004          12$: MOV    #4,R2      ;SEEK TIME OUT
1889 016734 004537 021136          13$: JSR    R5,ISDRST  ;ISSUE DRIVE RESET
1890                                ;FOUR TIMES BEFORE
1891 016740          WAITUS #15000.
(3) 016740 012700 035230          MOV    #15000.,R0
(3) 016744 104027          EMT    C$WTU      ;DROPPING DRIVE
1892
1893 016746 032763 000001 000000          BIT    #DRDY,CS(R3)      ;DRIVE READY YET?
1894 016754 001006          BNE    14$              ;YES, CHECK IF ERROR CLEARED
1895 016756 005302          DEC    R2              ;NO, HAVE WE DONE IT FOUR TIMES
1896 016760 001365          BNE    13$              ;YET
1897
1898 016762 012737 002673 002126 141$: MOV    #MDERS,WHY
1899 016770 000725          BR     91$              ;YES, DROP DRIVE
1900
1901 016772 032763 040000 000000 14$: BIT    #DERR,CS(R3)      ;DRIVE ERROR SET STILL
1902 017000 001370          BNE    141$              ;YES, DROP DRIVE
1903 017002          PRINTB #FMT14,#MRDER
(8) 017002 012746 003015          MOV    #MRDER,-(SP)
(7) 017006 012746 006507          MOV    #FMT14,-(SP)
(6) 017012 012746 000002          MOV    #2,-(SP)
(3) 017016 010600          MOV    SP,R0
(4) 017020 104014          EMT    C$PNTB
(4) 017022 062706 000006          ADD    #6,SP
1904 017026 004537 017156          JSR    R5,GHDR
1905 017032 000137 016312          JMP    EXIT
1906
1907 017036 012702 000004          15$: MOV    #4,R2      ;WAIT FOR HEADS TO UNLOAD
1908 017042 004537 021122          16$: JSR    R5,GETDST  ;GET STATUS
1909 017046 032701 000020          BIT    #BIT4,R1      ;UNLOAD STATE
1910 017052 001411          BEQ    17$              ;YES, CONTINUE W/ RECOVERY
1911 017054          WAITMS #1.           ;WAIT A WHILE
(3) 017054 012700 000001          MOV    #1.,R0
(3) 017060 104026          EMT    C$WTM
1912 017062 005302          DEC    R2              ;WAIT LONG ENOUGH
1913 017064 001366          BNE    16$              ;NO, GO BACK
1914 017066 012737 003352 002126          MOV    #UNLOAD,WHY
1915 017074 000663          BR     91$              ;DROP DRIVE
1916

```



```

1917 017076 004537 021136      17$: JSR    R5,ISDRST      ;ISSUE RESET
1918 017102                :      :
(3) 017102 012700 000001      :      :
(3) 017106 104026                :      :
1919 017110 032763 040000 000000      :      :
1920 017116 001321                :      :
1921 017120 012702 000075      :      :
1922 017124                18$: WAITMS #10.      ;FOR DRIVE READY TO
(3) 017124 012700 000012      :      :
(3) 017130 104026                :      :
1923 017132 032763 000001 000000      :      :
1924 017140 001314                :      :
1925 017142 005302                :      :
1926 017144 001367                :      :
1927 017146 012737 003376 002126      :      :
1928 017154 000633                :      :
1929
1930
1931 017156 012763 000210 000000      GHDR: MOV    #CRDY!RDHDR,CS(R3)
1932 017164 056463 000106 000000      :      :
1933 017172 042763 000200 000000      :      :
1934 017200 004537 021052                :      :
1935 017204 016301 000006      :      :
1936 017210 043701 002142                :      :
1937 017214 010164 000122                :      :
1938 017220 012764 004160 000052      :      :
1939 017226 000205                :      :
1940
1941
1942
1943
1944
1945 017230 010346                :      :
1946 017232 010246                :      :
1947 017234 010146                :      :
1948 017236 010046                :      :
1949 017240 016402 000042      :      :
1950 017244 005402                :      :
1951 017246 017401 000110      :      :
1952 017252 020227 000200      2$: MOV    @BBA(R4),R1      ;WHERE BUFFER IS
1953 017256 002015                :      :
1954 017260 020227 000003      :      :
1955 017264 002005                :      :
1956 017266 062702 000003      :      :
1957 017272 162764 000003 000042      :      :
1958 017300 010221                3$: MOV    R2,(R1)+      ;STORE WC
1959 017302 005302                :      :
1960 017304 010237 002212      :      :
1961 017310 000405                :      :
1962 017312 012737 000177 002212      4$: MOV    #127.,TEMP6      ;LOAD DOWN COUNTER
1963 017320 012721 000200      :      :
1964 017324 005737 007562      5$: MOV    #128.,(R1)+      :
1965 017330 001003                :      :
1966 017332 013703 007564      :      :
1967 017336 000406                :      :
1968 017340 004537 021214      55$: JSR    R5,RAND        ;GET RANDOM # FOR PATTERN

```

;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.

```

WRBUF: MOV    R3,-(SP)      ;SAVE REGISTERS
        MOV    R2,-(SP)
        MOV    R1,-(SP)
        MOV    R0,-(SP)
        MOV    BMP(R4),R2      ;R2 HAS TOTAL WORDS TO SET UP FOR
        NEG    R2              ;POSITIVE NUMBER
        MOV    @BBA(R4),R1      ;WHERE BUFFER IS
        CMP    R2,#128.        ;MORE THAN 128 WORDS
        BGE    4$              ;YES, BRANCH
        CMP    R2,#3           ;GREATER THAN THREE WORDS
        BGE    3$              ;YES, BRANCH
        ADD    #3,R2           ;ADD 3
        SUB    #3,BMP(R4)      ;WC UP BY 3
        MOV    R2,(R1)+        ;STORE WC
        DEC    R2              ;ACCOUNT FOR WC
        MOV    R2,TEMP6        ;LOAD DOWN COUNTER
        BR     5$
        MOV    #127.,TEMP6      ;LOAD DOWN COUNTER
        MOV    #128.,(R1)+
        TST    T,RAN           ;RANDOM SELECT OF PATTERNS
        BNE    55$            ;YEA
        MOV    T,PAT,R3        ;NO GET PATTERN OPERATOR
        BR     50$            ;WANTS TO USE
        JSR    R5,RAND        ;GET RANDOM # FOR PATTERN

```

```

1969 017344 013703 002136      MOV      LONUM,R3      ;GET RANDOM PATTERN
1970 017350 042703 177770      BIC      #177770,R3   ;0,7
1971 017354 006303          56$:    ASL      R3          ;WORD OFFSET
1972 017356 062703 024430      ADD      #PATLST,R3   ;GET PATTERN LIST
1973 017362 011303          MOV      (R3),R3      ;GET LIST ADDRESS
1974 017364 010337 002214      MOV      R3,TEMP7     ;STOR FOR RECALL
1975 017370 010321          MOV      R3,(R1)+     ;LOAD IT
1976 017372 005337 002212      DEC      TEMP6        ;ACCOUNT FOR IT
1977 017376 013703 002214      6$:    MOV      TEMP7,R3   ;PATTERN START
1978 017402 012737 000020 002216  7$:    MOV      #16.,TEMP8  ;16 ENTRIES
1979 017410 012321          MOV      (R3)+,(R1)+ ;STORE PATTERN
1980 017412 005337 002212      DEC      TEMP6        ;DOWN COUNT
1981 017416 001404          BEQ      8$          ;DONE?
1982 017420 005337 002216      DEC      TEMP8        ;DONE WITH PATTERN
1983 017424 001371          BNE      7$          ;NO, GO BACK
1984 017426 000763          BR       6$          ;RESTART PATTERN
1985 017430 162702 000200      8$:    SUB      #128.,R2   ;ANOTHER SECTOR TO USE
1986 017434 003306          BGT      2$          ;YES GO BACK
1987 017436 012600          MOV      (SP)+,R0     ;RESTORE REGISTERS
1988 017440 012601          MOV      (SP)+,R1
1989 017442 012602          MOV      (SP)+,R2
1990 017444 012603          MOV      (SP)+,R3
1991 017446 000205          RTS      R5

1992
1993      .SBTTL  RETRY LIMIT ROUTINE
1994
1995      ;RETRY BUMP, TWO RETURNS - CALL +2 - RETRY EXCEEDED
1996      ;CALL +4 - CONTINUE RETRY
1997
1998 017450 026437 000036 007504  RCNT:    CMP      RETRY(R4),LIMIT ;LIMIT REACHED?
1999 017456 001403          BEQ      1$          ;YES TAKE FIRST RETURN
2000 017460 005264 000036          INC      RETRY(R4)    ;ACCOUNT FOR RETRY
2001 017464 005725          TST      (R5)+        ;NEXT RETURN
2002 017466 000205      1$:    RTS      R5          ;RETURN
2003
2004      .SBTTL  LIST OF FUNCTION ROUTINES
2005
2006      ;WE GO THRU THIS LIST WHEN CALLED IN 'GETFNC'
2007      ;LIST IS IN NUMERICAL ORDER 1-6 (CONTROLLER RESET - READ)
2008
2009 017470 000000          LIST:    .WORD 0
2010 017472 013724          WRTFNC          ;WRITE DATA
2011 017474 013346          GSTFNC          ;GET STATUS
2012 017476 013366          SKFNC          ;SEEK FUNCTION
2013 017500 013366          SKFNC          ;SEEK FUNCTION
2014 017502 014002          RDDFNC          ;READ DATA
2015 017504 014002          RDDFNC          ;READ DATA
2016
2017      .SBTTL  BAD SECTOR FILE ROUTINE
2018
2019      ;ROUTINE TO RECOVER BAD SECTOR FILE AND SAVE IT FOR
2020      ;COMPARISON UPON ERROR ON READS/Writes. WE WILL ONLY
2021      ;RESERVE SPACE FOR 16 BAD SECTORS PER DRIVE.
2022      ;WE WILL ISSUE A DRIVE RESET FIRST, READ HEADER, POSITION
2023      ;TO LAST TRACK (Cylinder 255, SURFACE 1) AND READ IN
2024      ;THE FIRST SECTOR FOR FACTORY BAD, AND THE 20TH FOR

```

```

2025 ;FIELD BAD SECTORS. R4 WILL CONTAIN THE BUFFER POINTER
2026 ;TO THE DRIVE WE WANT TO READ
2027 .
2028 ;CALL JSR R5,RDBDSC
2029
2030
2031 RDBDSC: MOV R0,-(SP) ;SAVE REGISTERS
2032 MOV R1,-(SP) ;
2033 MOV R2,-(SP) ;
2034 MOV R3,-(SP) ;
2035 017506 010046 021136 000044 21$: JSR R5,ISDRST
2036 017510 010146 000010 000044 MOV #RDHDR,FUNC(R4);READ HEADER TO FIND POSITION
2037 017512 010246 014112 JSR R5,LDFUNC ;ON DISK
2038 017514 010346 021052 JSR R5,WTRDY
2039
2040 017516 004537 021136 MOV MP(R3),R0 ;GET HEADER AND CALCULATE
2041 017522 012764 000010 000044 BIC CYLSK,R0 ;DIFFERENCE TO GET TC
2042 017530 004537 014112 MOV #77600,R1 ;BAD SECTER FILE, AND GO
2043 017534 004537 021052 SUB R0,R1 ;THERE
2044 017540 016300 000006 MOV R1,BDA(R4)
2045 017544 043700 002140 BIS #SKHS!SIGN!MK,BDA(R4)
2046 017550 012701 077600 MOV #SEEK,FUNC(R4)
2047 017554 160001 SUB R0,R1
2048 017556 010164 000040 MOV R1,BDA(R4)
2049 017562 052764 000025 000040 BIS #SKHS!SIGN!MK,BDA(R4)
2050 017570 012764 000006 000044 MOV #SEEK,FUNC(R4)
2051 017576 004537 014112 JSR R5,LDFUNC
2052 017602 004537 021052 JSR R5,WTRDY
2053 017606 012764 000010 000044 MOV #RDHDR,FUNC(R4)
2054 017614 004537 014112 JSR R5,LDFUNC
2055 017620 004537 021052 JSR R5,WTRDY
2056 017624 016300 000006 MOV MP(R3),R0
2057 017630 042700 000077 BIC #77,R0
2058 017634 022700 077700 CMP #77700,R0
2059 017640 001326 BNE 21$
2060
2061 017642 012764 077700 000040 MOV #77700,BDA(R4) ;SETUP AND READ IN THE
2062 017650 012764 177400 000042 MOV #-256.,BMP(R4) ;BAD SECTOR FILE ON SECTOR
2063 017656 012764 000014 000044 MOV #READ,FUNC(R4) ;0
2064
2065 017664 005037 002204 CLR TEMP3 ;MANUFACTURING/FIELD FILE SWITCH
2066 017670 012737 003524 002126 MOV #HWSEC,WHY ;START WITH MANUFACTURING BAD
2067 017676 016402 000112 MOV BSECT(R4),R2 ;INITIALIZE LIST TO ALL 1'S
2068 017702 012700 000020 MOV #16.,R0 ;SIXTEEN ENTRIES
2069 017706 012722 177777 11$: MOV #-1,(R2)+
2070 017712 005300 DEC R0
2071 017714 001374 BNE 11$
2072
2073 017716 016402 000112 MOV BSECT(R4),R2 ;GET LIST TO STORE
2074 017722 012700 000020 MOV #16.,R0 ;SIXTEEN ENTRIES
2075 017726 004537 014112 4$: JSR R5,LDFUNC
2076 017732 004537 021052 JSR R5,WTRDY
2077
2078 017736 005774 000104 TST @DCS(R4) ;WAS THE READ GOOD?
2079 017742 100025 BPL 3$ ;YES
2080
2081 017744 004537 021136 JSR R5,ISDRST
2082 017750 062764 000004 000040 ADD #4,BDA(R4) ;NO, NEXT SECTOR
2083 017756 005737 002204 TST TEMP3 ;MANUFACTURING OR FIELD BAD
2084 017762 001410 BEQ 5$ ;MANUFACTURING

```

```

2081 017764 012737 003544 002126      MOV      #SWSEC,WHY      ;FIELD BAD
2082 017772 022764 077750 000040      CMP      #77750,BDA(R4) ;AT END OF FIELD BAD?
2083 020000 001352                      BNE      4$             ;NO, GO BACK FOR NEXT
2084 020002 000470                      BR       6$
2085 020004 026427 000040 077724 5$:  CMP      BDA(R4),#77724 ;AT END OF MANUFACTURING BAD
2086 020012 001345                      BNE      4$             ;AT END OF BAD FACTORY SECTION
2087 020014 000463                      BR       6$             ;YES, REPORT ERROR
2088
2089 020016 017401 000110      3$:  MOV      @BDA(R4),R1     ;START OF LIST
2090 020022 012164 000100      MOV      (R1)+,SERNM1(R4) ;GET LOW PART OF SERIAL #
2091 020026 012164 000102      MOV      (R1)+,SERNM2(R4) ;GET HIGH PART OF SERIAL #
2092 020032 022121                      CMP      (R1)+,(R1)+    ;SKIP PAST JUNK
2093 020034 012137 002200      1$:  MOV      (R1)+,TEMP1     ;GET CYLINDER
2094 020040 100437                      BMI      2$             ;IF MINUS END OF BAD SECTORS
2095 020042 012137 002202      MOV      (R1)+,TEMP2     ;GET TRACK AND CYLINDER
2096 020046 000337 002200      SWAB     TEMP1           ;PUT CYLINDER IN HIGH BYTE
2097 020052 006037 002200      ROR      TEMP1           ;ALIGN IT
2098 020056 013712 002200      MOV      TEMP1,(R2)      ;STORE OFF CYLINDER PART
2099 020062 013737 002202 002200      MOV      TEMP2,TEMP1     ;GET SECTOR
2100 020070 042737 177700 002200      BIC      #177700,TEMP1    ;LEAVE ONLY SECTOR
2101 020076 053712 002200      BIS      TEMP1,(R2)      ;SET IN SECTOR BITS
2102 020102 042737 177377 002202      BIC      #177377,TEMP2
2103 020110 006237 002202      ASR      TEMP2
2104 020114 006237 002202      ASR      TEMP2
2105 020120 053722 002202      BIS      TEMP2,(R2)+    ;SET IN HEAD
2106 020124 005300      DEC      R0
2107 020126 001342      BNE      1$
2108 020130 012737 003473 002126      MOV      #MBDMSC,WHY     ;MORE THAN 16 BAD SECTORS
2109 020136 000412      BR       6$
2110
2111 020140 005737 002204      2$:  TST      TEMP3           ;SWITCH TO FIELD BAD OR QUIT
2112 020144 001011      BNE      7$             ;QUIT, 7$
2113 020146 012764 077724 000040      MOV      #77724,BDA(R4) ;SWITCH TO FIELD BAD
2114 020154 012737 000001 002204      MOV      #1,TEMP3        ;SET TO QUIT NEXT TIME THRU
2115 020162 000661      BR       4$
2116
2117 020164 004537 020220      6$:  JSR      R5,DRDRV       ;DROP THE DRIVE
2118 020170 004537 022500      7$:  JSR      R5,HDHOME      ;BRINGS HEADS HOME
2119 020174 012603      9$:  MOV      (SP)+,R3
2120 020176 012602      MOV      (SP)+,R2
2121 020200 012601      MOV      (SP)+,R1
2122 020202 012600      MOV      (SP)+,R0
2123 020204 000205      RTS      R5
2124
2125 020206 004537 020220      8$:  JSR      R5,DRDRV       ;DROP THE DRIVE
2126 020212 000770      BR       9$
2127
2128      ;
2129
2130
2131
2132
2133
2134      .SBTTL  ROUTINE TO DROP DRIVE
2135      ;
2136      ;ROUTINE TO DROP A DRIVE FROM RUNNING
  
```

```

2137 ;R4 HAS BUFFER POINTER OF DRIVE TO DROP
2138 ;WE CLEAR BIT IN 'DRUT', NOT 'DRPRS'
2139 ;
2140
2141 020214 005237 002324 ODRDRV: INC OPCALL
2142 020220 010146 DRDRV: MOV R1,-(SP)
2143 020222 010246 MOV R2,-(SP) ;SAVE REGISTERS
2144 020224 010346 MOV R3,-(SP)
2145 020226 005237 002326 INC INCALL
2146 020232 005003 CLR R3
2147 020234 012702 025056 MOV #DRBUF,R2 ;START OF DRIVE BUFFERS
2148 020240 012701 000001 MOV #1,R1 ;MASK
2149 020244 020402 1$: CMP R4,R2 ;IS THIS THE DRIVE?
2150 020246 001405 BEQ 2$ ;YES GO DROP IT
2151 020250 005203 INC R3
2152 020252 006301 ASL R1 ;NO SHIFT MASK
2153 020254 062702 000124 ADD #PRPOS+2,R2 ;NEXT BUFFER
2154 020260 000771 BR 1$ ;GO BACK
2155
2156 020262 005737 002324 2$: TST OPCALL
2157 020266 001002 BNE 6$
2158 020270 DODU R3
(3) 020270 010300 MOV R3,R0
(3) 020272 104053 EMT C$DODU
2159 020274 005037 002326 6$: CLR INCALL
2160 020300 005037 002324 CLR OPCALL
2161 020304 113764 002246 000070 MOVB HOUR,DPHOUR(R4) ;TIME AT WHICH IT WAS DROPPED
2162 020312 113764 002244 000071 MOVB MINUTE,DPMIN(R4) ;HOUR/MINUTE
2163 020320 001002 BNE 3$ ;IF MINUTE 0,
2164 020322 105264 000071 INCB DPMIN(R4) ;MAKE 1.
2165 020326 140137 002130 3$: BICB R1,DRUT ;CLEAR THE DRIVE FROM BIT MAP
2166 020332 PRINTF #FMT7,#TIME,HOUR,MINUTE,SECOND,#MRLCS,DCS(R4),#DRNM
(14) 020332 012746 003634 MOV #DRNM,-(SP)
(13) 020336 016446 000104 MOV DCS(R4),-(SP)
(12) 020342 012746 002337 MOV #MRLCS,-(SP)
(11) 020346 013746 002242 MOV SECOND,-(SP)
(10) 020352 013746 002244 MOV MINUTE,-(SP)
(9) 020356 013746 002246 MOV HOUR,-(SP)
(8) 020362 012746 002330 MOV #TIME,-(SP)
(7) 020366 012746 006106 MOV #FMT7,-(SP)
(6) 020372 012746 000010 MOV #10,-(SP)
(3) 020376 010600 MOV SP,R0
(4) 020400 104017 EMT C$PNTF
(4) 020402 062706 000022 ADD #22,SP
2167 020406 PRINTF #FMT7A,<B,DRSEL+1(R4)>,#DROP,WHY
(10) 020406 013746 002126 MOV WHY,-(SP)
(9) 020412 012746 004103 MOV #DROP,-(SP)
(8) 020416 005046 CLR -(SP)
(8) 020420 156416 000107 BISB DRSEL+1(R4),(SP)
(7) 020424 012746 006141 MOV #FMT7A,-(SP)
(6) 020430 012746 000004 MOV #4,-(SP)
(3) 020434 010600 MOV SP,RC
(4) 020436 104017 EMT C$PNTF
(4) 020440 062706 000012 ADD #12,SP
2168 020444 PRINTF #FMTS1
(7) 020444 012746 006717 MOV #FMTS1,-(SP)

```

```

(6) 020450 012746 000001      MOV    #1,-(SP)
(3) 020454 010600              MOV    SP,R0
(4) 020456 104017              EMT   C$PNTF
(4) 020460 062706 000004      ADD    #4,SP
2169
2170 020464 004737 011670      JSR   PC,REPORT
2171
2172 020470 012603              MOV    (SP)+,R3
2173 020472 012602              MOV    (SP)+,R2      ;RESTORE REGISTERS
2174 020474 012601              MOV    (SP)+,R1
2175
2176 020476 000205              RTS   R5
2177
2178      .SBTTL  ROUTINE TO CHECK DATA
2179      ;ROUTINE TO CHECK DATA ON READ
2180
2181      CKDATA: TST   CMRD      ;DO WE WANT TO CHECK ANY?
2182 020500 005737 007520      BNE   97$             ;YES CONTINUE
2183 020504 001001              RTS   R5             ;NO, EXIT
2184 020506 000205
2185 020510      97$: SETPRI #340
(3) 020510 012700 000340      MOV    #340,R0
(3) 020514 104041              EMT   C$SPRI
2186 020516 017402 000110      MOV    @BBA(R4),R2   ;BUFFER START
2187 020522 016437 000042 002200      MOV    BMP(R4),TEMP1 ;WORDS READ IN
2188 020530 005437 002200      NEG   TEMP1         ;MAKE POSITIVE
2189 020534 013737 007522 002202      MOV    DELMT,TEMP2   ;# ERRORS TO BE PRINTED
2190 020542 005037 002174      CLR   DECNT         ;INIT ERROR COUNT
2191 020546 013737 007520 002204      MOV    CMRD,TEMP3    ;# WORDS TO BE COMPARED
2192 020554 012737 000176 002176 96$: MOV    #126.,TEMP0    ;126 WORDS
2193 020562 012201              MOV    (R2)+,R1     ;NON-ZERO WORDS
2194 020564 005337 002200      DEC   TEMP1
2195 020570 001516              BEQ   CEND
2196 020572 005301              DEC   R1
2197 020574 012237 002206      MOV    (R2)+,TEMP4   ;PATTERN ADDRESS
2198
2199      ;MAKE SURE PATTERN ADDRESS IS LEGAL
2200
2201 020600 012700 024430      MOV    #PATLST,R0    ;GET LIST OF PATTERNS
2202 020604 012703 000010      MOV    #8.,R3        ;ONLY EIGHT
2203 020610 022037 002206 98$: CMP    (R0)+,TEMP4    ;FOUND IT YET
2204 020614 001412              BEQ   99$            ;YES, CONTINUE
2205 020616 005303              DEC   R3             ;NO, EXHAUST LIST YET
2206 020620 001373              BNE   98$            ;NO, GO BACK
2207
2208 020622 024242              CMP    -(R2),-(R2)
2209 020624              ERRHRD 180.,NOREV,ERR13
(3) 020624 104463              TRAP  T$ERCODE
(5) 020626 000264              .WORD 180
(5) 020630 003434              .WORD NOREV
(5) 020632 005172              .WORD ERR13
2210 020634 004537 023620      JSR   R5,STDMP
2211 020640 000205              RTS   R5
2212
2213 020642 005301 99$: DEC   R1             ;ACCOUNT FOR PATTERN ADDRESS
2214 020644 013703 002206      MOV    TEMP4,R3     ;GET ADDRESS

```

```

2215 020650 005337 002200          DEC    TEMP1          ;ACCOUNT ONCE AGAIN
2216 020654 012737 000020 002210  MOV    #16.,TEMP5    ;16 ENTRIES TO PATTERN
2217 020662 005737 002200          TST    TEMP1         ;ANY WORDS READIN LEFT?
2218 020666 001457                BEQ    CEND          ;NO, GO TO END
2219 020670 005737 002204          TST    TEMP3         ;HAVE WE EXHAUSTED COMPARE LIMIT?
2220 020674 001454                BEQ    CEND          ;YES GO TO END
2221 020676 005701                TST    R1            ;WE CHECKING PATTERN OR ZERO FILL?
2222 020700 001416                BEQ    3$            ;ZERO FILL SKIP
2223 020702 005301                DEC    R1            ;PATTERN
2224 020704 005737 002210          TST    TEMP5         ;WITHIN PATTERN
2225 020710 001005                BNE    2$            ;YES SKIP
2226 020712 013703 002206          MOV    TEMP4,R3      ;NO, START OVER
2227 020716 012737 000020 002210  MOV    #16.,TEMP5    ;16 ENTRIES
2228 020724 012337 002232          MOV    (R3)+,GDDAT   ;GET PATTERN
2229 020730 005337 002210          DEC    TEMP5         ;DOWN COUNT
2230 020734 000402                BR     4$            ;
2231 020736 005037 002232          CLR    GDDAT         ;ZERO FILL
2232 020742 023712 002232          CMP    GDDAT,(R2)    ;CORRECT DATA
2233 020746 001415                BEQ    5$            ;YES YES NEXT
2234 020750 005237 002174          INC    DECNT         ;DATA ERROR
2235 020754 005264 000074          INC    DATCER(R4)   ;
2236 020760 005737 002202          TST    TEMP2         ;DO WE WANT TO PRINT IT
2237 020764 001406                BEQ    5$            ;NO,SKIP
2238
2239 020766                ERRHRD 180.,MDCER,ERR8
   (3) 020766 104463          TRAP  T$ERCODE
   (5) 020770 000264          .WORD 180
   (5) 020772 003040          .WORD MDCER
   (5) 020774 004664          .WORD ERR8
2240 020776 005337 002202          DEC    TEMP2         ;ACCOUNT FOR PRINT
2241
2242 021002 005337 002200          5$: DEC    TEMP1         ;WORDS READ IN
2243 021006 001407                BEQ    CEND          ;
2244 021010 005722                TST    (R2)+         ;NEXT WORD
2245 021012 005337 002176          DEC    TEMP0         ;
2246 021016 001656                BEQ    96$          ;

```

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


```

2255 021034 005464 000042      NEG      BMP(R4)      ;MAKE POSITIVE WORD COUNT
2256 021040      ERRHRD 190.,MDCER,ERR6 ;DATA ERROR SUMMARY
(3) 021040 104463      TRAP     T$ERCODE
(5) 021042 000276      .WORD   190
(5) 021044 003040      .WORD   MDCER
(5) 021046 004566      .WORD   ERR6
2257
2258 021050 000205      1$:     RTS      R5
2259
2260      .SBTTL  ROUTINE TO WAIT FOR CONTROLLER READY
2261
2262      ;
2263      ;ROUTINE TO WAIT FOR CONTROLLER READY UNDER FLAG
2264      ;MODE. USED IN INITIALIZE PORTION OF PROGRAM I.E.
2265      ;GETTING BAD SECTOR FILE, WRITING PACK INITIALLY
2266
2267 021052 010046      WTRDY:  MOV      R0,-(SP)      ;SAVE REGISTERS
2268 021054 010146      MOV      R1,-(SP)
2269 021056 012701 001750      MOV      #1000.,R1      ;WAIT A WHILE
2270 021062      1$:     WAITUS  #2.
(3) 021062 012700 000002      MOV      #2.,R0
(3) 021066 104027      EMT      C$WTU
2271 021070 032774 000200 000104      BIT      #CRDY,@DCS(R4) ;READY SET?
2272 021076 001006      BNE      2$             ;YES, EXIT
2273 021100 005301      DEC      R1             ;TIMED OUT?
2274 021102 001367      BNE      1$             ;NO GO BACK
2275
2276 021104      ERRDF  1002.,NOCRDY,ERR12
(3) 021104 104462      TRAP     T$ERCODE
(5) 021106 001752      .WORD   1002
(5) 021110 002464      .WORD   NOCRDY
(5) 021112 005164      .WORD   ERR12
2277
2278 021114 012601      2$:     MOV      (SP)+,R1      ;RESTORE REGISTERS
2279 021116 012600      MOV      (SP)+,R0
2280 021120 000205      RTS      R5
2281
2282
2283      .SBTTL  GET STATUS/DRIVE RESET ROUTINE
2284
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 021126 012763 000003 000004      MOV      #GSBIT,DA(R3)
2291 021134 000405      BR       CSTUFF
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!GSTAT,CS(R3)
2295 021156 056463 000106 000000      BIS      DRSEL(R4),CS(R3)
2296 021164 042763 000200 000000      BIC      #CRDY,CS(R3)
2297 021172 004537 021052      JSR      R5,WTRDY
2298 021176 022763 000013 000004      CMP      #DRST,DA(R3)
2299 021204 001402      BEQ     1$
2300 021206 016301 000006      MOV      MP(R3),R1
  
```

```

2301 021212 000205      1$:      RTS      R5
2302
2303
2304      .SBTTL  ROUTINE TO GENERATE A RANDOM NUMBER
2305
2306 021214 010146      RAND:  MOV      R1,-(SP)
2307 021216 010246      MOV      R2,-(SP)
2308 021220 010346      MOV      R3,-(SP)
2309
2310 021222 013703 002136      MOV      LONUM,R3
2311 021226 013701 002134      MOV      HINUM,R1
2312 021232 012702 177771      MOV      #-7,R2
2313 021236 006303      1$:      ASL      R3
2314 021240 006101      ROL      R1
2315 021242 005202      INC      R2
2316 021244 001374      BNE      1$
2317 021246 063703 002136      ADD      LONUM,R3
2318 021252 005501      ADC      R1
2319 021254 063701 002134      ADD      HINUM,R1
2320 021260 062703 001057      ADD      #1057,R3
2321 021264 005501      ADC      R1
2322 021266 062701 047401      ADD      #47401,R1
2323 021272 010337 002134      MOV      R3,HINUM
2324 021276 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 000205      RTS      R5
2329
2330      .SBTTL  ROUTINE TO WRITE PACKS INITIALLY
2331
2332      ;ROUTINE TO WRITE PACK WITH PATTERN, ALL TRACKS WILL BE
2333      ;WRITTEN (EXCEPT BAD SECTOR TRACK)
2334      ;FORMAT IS # OF WORDS (WORD 1), PATTERN ADDRESS (WORD 2)
2335      ;PATTERN (WORDS 3 - 128)
2336      ;WE WILL ATTEMPT TO WRITE MULTIPLE SECTORS AT A TIME
2337      ;(MINIMUM 10 SECTORS) IF AN ERROR OCCURS WE WILL THEN
2338      ;WRITE INDIVIDUAL SECTORS FOR THAT TRACK. WE DO WRITES,
2339      ;READS AND INCORE COMPARISONS TO VERIFY.
2340
2341      ;
2342      ;CALL JSR R5,WPACK
2343
2344 021312 010046      WPACK:  MOV      R0,-(SP)      ;SAVE REGISTERS
2345 021314 010146      MOV      R1,-(SP)
2346 021316 010246      MOV      R2,-(SP)
2347 021320 010346      MOV      R3,-(SP)
2348 021322 016446 000110      MOV      BBA(R4),-(SP)
2349 021326 005764 000120      TST      WRIPG(R4)      ;SEE IF WRITE IN PROGRESS WAS SET
2350 021332 001033      BNE      1$              ;JUMP IF SET - DON'T PRINT MESSAGE
2351 021334      PRINTF #FMT18,#MSWRPK
      (8) 021334 012746 004205      MOV      #MSWRPK,-(SP)
      (7) 021340 012746 006712      MOV      #FMT18,-(SP)
      (6) 021344 012746 000002      MOV      #2,-(SP)
      (3) 021350 010600      MOV      SP,R0
      (4) 021352 104017      EMT      C$PNTF
  
```

```

(4) 021354 062706 000006      ADD    #6,SP
2352 021360                    PRINTF #FMT17,#MRLCS,DCS(R4),#DRNM,<B,DRSEL+1(R4)>
(11) 021360 005046            CLR    -(SP)
(11) 021362 156416 000107      BISB  DRSEL+1(R4),(SP)
(10) 021366 012746 003634      MOV    #DRNM,-(SP)
(9) 021372 016446 000104      MOV    DCS(R4),-(SP)
(8) 021376 012746 002337      MOV    #MRLCS,-(SP)
(7) 021402 012746 005663      MOV    #FMT17,-(SP)
(6) 021406 012746 000005      MOV    #5,-(SP)
(3) 021412 010600            MOV    SP,0
(4) 021414 104017            EMT   C$PNTF
(4) 021416 062706 000014      ADD    #14,SP
2353 021422 004537 022500      1$:   JSR    R5,HDHOME          ;HEADS HOME
2354
2355
2356
2357
2358
2359
2360 021426 005037 002200      CLR    TEMP1                ;TEMP1=HEAD
2361 021432 005001            CLR    R1                    ;R1=CYL
2362 021434 022701 077600      CONWR: CMP   #77600,R1         ;CYL=255?
2363 021440 001014            BNE   STWRT                  ;NO GO WRITE TRACK
2364 021442 005737 002200      TST   TEMP1                  ;YES, CHECK IF HEAD = 1?
2365 021446 001411            BEQ   STWRT                  ;HEAD = 0 GO WRITE
2366 021450 004537 022500      ENDWR: JSR  R5,HDHOME        ;HEADS HOME
2367 021454 012664 000110      MOV    (SP)+,BBA(R4)
2368 021460 012603            MOV    (SP)+,R3
2369 021462 012602            MOV    (SP)+,R2
2370 021464 012601            MOV    (SP)+,R1
2371 021466 012600            MOV    (SP)+,R0
2372 021470 000205            RTS     R5                    ;END EXIT
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
021472 005002            STWRT: CLR   R2                ;INITIAL SECTOR 0
021474 012764 002266 000110    MOV    #BUF1,BBA(R4)         ;BUFFER START
021502 012764 175400 000042    MOV    #-1280,BMP(R4)       ;10 SECTORS
021510 004537 017230            JSR    R5,WRTBUF             ;WRITE BUFFER INTO MEMORY
021514 010164 000040 201$:   MOV    R1,BDA(R4)           ;SET UP SECTOR
021520 053764 002200 000040    BIS    TEMP1,BDA(R4)
021526 005764 000120            TST   WRIPG(R4)              ;WAS WRITE IN PROGRESS SET?
021532 001406 762$:   BEQ   762$                  ;JUMP IF NOT SET
021534 026464 000122 000040    CMP    PRPOS(R4),BDA(R4)     ;AT THE SAME ADDRESS WHEN DIED?
021542 001402 762$:   BEQ   762$                  ;JUMP IF ON CYLINDER
021544 000137 022154 762$:   JMP   952$                  ;ELSE, LOOK AT THE NEXT CYL ADDRESS
021550 050264 000040            BIS    R2,BDA(R4)
021554 012764 002266 000110    MOV    #BUF1,BBA(R4)         ;SET UP TO WRITE
021562 012764 000012 000044    MOV    #WRITE,FUNC(R4)       ;WRITE
021570 004537 014112            JSR    R5,LDFUNC
021574 004537 021052            JSR    R5,WTRDY              ;WAIT FOR READY
021600 005774 000104            TST   @DCS(R4)               ;ERROR
021604 100003 203$:   BPL   203$
021606 004537 021136 205$:   JSR    R5,ISDRST
021612 000421 2$:   BR    2$
  
```

```

2397
2398 021614 012764 000002 000044 203$: MOV #WRCHK, FUNC(R4)
2399 021622 004537 014112 JSR R5, LDFUNC
2400 021626 004537 021052 JSR R5, WTRDY
2401 021632 005774 000104 TST @DCS(R4) ;ERROR
2402 021636 100763 BMI 205$ ;YES GO DO SECTORS INDIVIDUALLY
2403
2404
2405 021640 062702 000012 ADD #10, R2 ;NEXT GROUP
2406 021644 022702 000050 CMP #40, R2 ;DONE?
2407 021650 001321 BNE 201$ ;NO, GO BACK
2408 021652 000137 022154 JMP 952$ ;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 2$: CLR R2 ;R2 = SECTOR
2414
2415 021660 012764 177600 000042 MOV #-128, BMP(R4) ;LOAD WORD COUNT
2416 021666 010164 000040 3$: MOV R1, BDA(R4) ;SETUP DISK ADDRESS
2417 021672 053764 002200 000040 BIS TEMP1, BDA(R4)
2418 021700 050264 000040 BIS R2, BDA(R4)
2419
2420 021704 012764 002266 000110 MOV #BUF1, BBA(R4)
2421 021712 004537 017230 JSR R5, WRBUF ;WRITE A BUFFER
2422 021716 005037 002124 91$: CLR RWCNT ;CLEAR RETRYS OUT
2423 021722 005037 002174 98$: CLR DECNT ;
2424 021726 012764 000012 000044 96$: MOV #WRITE, FUNC(R4) ;WRITE FUNCTION
2425 021734 004537 014112 JSR R5, LDFUNC
2426 021740 004537 021052 JSR R5, WTRDY ;WAIT FOR WRITE TO FINISH
2427
2428 021744 005774 000104 TST @DCS(R4) ;ERROR ON WRITE?
2429 021750 100023 BPL 85$ ;NO, CO READ
2430
2431 021752 004537 021136 JSR R5, ISDRST
2432 021756 016437 000040 002172 MOV BDA(R4), CHKSEC ;YES, CHECK IF SECTOR IS IN
2433 021764 004537 023744 JSR R5, CKBDSC ;BAD SECTOR FILE
2434 021770 005737 002170 TST HD^FND ;IF SET, IT WAS
2435 021774 001050 BNE 95$ ;YES GO TO NEXT SECTOR
2436
2437 021776 005237 002174 INC DECNT ;NO, GIVE IT ONE MORE TRY
2438 022002 023727 002174 000002 CMP DECNT, #2. ;IT MAY HAVE BEEN NOISE.
2439 022010 001346 BNE 96$
2440
2441
2442 022012 004537 022240 JSR R5, INBAD
2443 022016 000437 BR 95$
2444
2445
2446 022020 005037 002122 85$: CLR RECNT ;CLEAR RETRY COUNT
2447 022024 012764 000002 000044 80$: MOV #WRCHK, FUNC(R4) ;
2448 022032 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 BPL 81$ ;NO, GO COMPARE

```

```

2453 022050 004537 021136      JSR      R5,ISDRST
2454
2455 022054 016437 000040 002172      MOV      BDA(R4),CHKSEC ;CHECK IF SECTOR IS
2456 022062 004537 023744      JSR      R5,CKBDSC      ;A KNOWN BAD SECTOR
2457 022066 005737 002170      TST      HDRFND         ;IT WAS THEN
2458 022072 001011      BNE      95$            ;GO TO NEXT SECTOR
2459
2460 022074 005237 002122      INC      RECNT          ;GIVE IT ANOTHER CHANCE
2461 022100 023727 002122 000002      CMP      RECNT,#2.
2462 022106 001346      BNE      80$
2463
2464 022110 004537 022240      JSR      R5,INBAD
2465 022114 000400      BR
2466
2467 022116      81$:
2468
2469 022116 062702 000012      95$:  ADD      #10,R2        ;NEXT SECTOR (OFFSET BY 10)
2470 022122 020227 000047      CMP      R2,#39.       ;DONE WITH TRACK?
2471 022126 003002      BGT      951$          ;YES NEXT TRACK
2472 022130 000137 021666      JMP      3$            ;NO GO BACK FOR NEXT SECTOR
2473 022134      951$:
2474 022134 005202      INC      R2            ;NEXT SECTOR
2475 022136 162702 000050      SUB      #40,R2        ;DONE WITH TRACK?
2476 022142 020227 000012      CMP      R2,#10.
2477 022146 001402      BEQ      952$          ;YES
2478 022150 000137 021666      JMP      3$            ;NO
2479 022154      952$:
2480
2481 022154 005737 002200      TST      TEMP1         ;WHICH SURFACE?
2482 022160 001420      BEQ      5$            ;TOP (0), BRANCH
2483
2484 022162 005037 002200      CLR      TEMP1         ;BOTTOM, SWITCH TO TOP WITH
2485 022166 062701 000200      ADD      #200,R1
2486 022172 012764 000205 000040      MOV      #205,BDA(R4) ;SEEK, GO IN ALSO
2487 022200 012764 000006 000044 4$:  MOV      #SEEK,FUNC(R4) ;GO SEEK
2488 022206 004537 014112      JSR      R5,LDFUNC
2489 022212 004537 021052      JSR      R5,WTRDY
2490
2491 022216 000137 021434      JMP      CONWR
2492
2493 022222 012737 000100 002200 5$:  MOV      #HEAD,TEMP1   ;WAS TOP, MAKE BOTTOM.
2494 022230 012764 000021 000040      MOV      #21,BDA(R4)
2495 022236 000760      BR      4$
2496
2497
2498 022240 016337 000000 002250 INBAD: MOV      CS(R3),E.CS
2499 022246 016337 000002 002252      MOV      BA(R3),E.BA
2500 022254 016337 000004 002254      MOV      DA(R3),E.DA
2501 022262 016337 000006 002256      MOV      MP(R3),E.MP
2502 022270 016337 000006 002260      MOV      MP(R3),E.MP1
2503 022276 016337 000006 002262      MOV      MP(R3),E.MP2
2504 022304      ERRHRD 199,NWRTS,ERR13
(3) 022304 104463      TRAP    T$ERCODE
(5) 022306 000307      .WORD  199
(5) 022310 002541      .WORD  NWPTS
(5) 022312 005172      .WORD  ERR13
    
```

```

2505 022314 005264 000012      INC      ERRCNT(R4)
2506 022320 005737 007546      TST      T.DRP                ;ARE WE COUNTING ERRORS
2507 022324 001413              BEQ      2$                    ;NO
2508 022326 026437 000012 007506  CMP      ERRCNT(R4),ERLMT     ;PAST IT
2509 022334 103407              BLO      2$                    ;NO
2510 022336 012737 003125 002126  MOV      #ERLMTM,WHY
2511 022344 004537 020220      JSR      R5,DRDRV
2512 022350 012705 021450      MOV      #ENDWR,R5
2513
2514 022354 000205      2$:     RTS      R5
2515      .SBTTL  ROUTINE FOR SYSTEM CLOCK
2516
2517      ;ROUTINE TO READ SYSTEM CLOCK
2518      ;USES 'REGTIM' FROM DIAGNOSTIC SUPERVISOR
2519
2520 022356 005737 002264  GETSYS: TST      SYSCLK        ;DO WE HAVE A CLOCK
2521 022362 001002              BNE      4$                    ;YES, GO SERVICE IT
2522 022364              BREAK                       ;NO, CALL SUPER FOR ^C
2523 022366 000205              EMT      C$BRK
2524 022370              RTS      R5                    ;EXIT
2525      (3) 022370 104045      4$:     REQTIM  R0                ;GET PRESENT TIME
2526      (3) 022372 020037 002240      EMT      C$REQTIM
2527 022376 001437              1$:     CMP      R0,LSTTIM        ;HAS IT MOVED
2528 022400 013701 002240      BEQ      3$                    ;NO MOVEMENT SINCE LAST CALL
2529 022404 010037 002240      MOV      LSTTIM,R1            ;CALCULATE DIFFERENCE
2530 022412 060037 002242      MOV      R0,LSTTIM           ;AND FIX ACCORDINGLY
2531 022416 022737 000074 002242  SUB      R1,R0
2532 022424 003024              2$:     ADD      R0,SECOND        ;BUMP SECONDS
2533 022426 162737 000074 002242  CMP      #60.,SECOND         ;SECONDS OVERFLOW
2534 022434 005237 002236      BGT      3$
2535 022440 005237 002244      7$:     SUB      #60.,SECOND
2536 022444 022737 000074 002242  INC      INTERVAL            ;TIME BETWEEN REPORTS
2537 022452 002765              INC      MINUTE               ;BUMP MINUTES
2538 022454 022737 000074 002244  CMP      #60.,SECOND
2539 022462 003005              BGT      3$
2540 022464 005237 002246      INC      HOUR
2541 022470 162737 000074 002244  CMP      #60.,MINUTE
2542 022476 000205              3$:     BGT      3$
2543      RTS      R5
2544      .SBTTL  HEADS HOME ROUTINE
2545
2546      ;ROUTINE TO BRING HEADS OVER TRACK 0
2547
2548 022500 010046  HDHOME: MOV      RC,-(SP)        ;SAVE R0
2549 022502 012764 000010 000044  MOV      #RDHDR,FUNC(R4)     ;READ HEADER
2550 022510 004537 014112      JSR      R5,LDFUNC           ;GO DO IT.
2551 022514 004537 021052      JSR      R5,WTRDY
2552
2553 022520 016300 000006      MOV      MP(R3),R0           ;GET HEADER
2554 022524 042700 000177      BIC      #177,R0             ;ONLY CYLINDER
2555 022530 010064 000040      MOV      R0,BDA(R4)         ;MOVE IT TO BUFFERED DA
2556 022534 052764 000001 000040  BIS      #MK,BDA(R4)         ;SET MARKER
2557 022542 012764 000006 000044  MOV      #SEEK,FUNC(R4)     ;LOAD SEEK
2558 022550 004537 014112      JSR      R5,LDFUNC           ;SEEK.
  
```

```

2559 022554 004537 021052          JSR    R5,WTRDY          ;WAIT.
2560 022560 016464 000122 000050  MOV    PRPOS(R4),LSTHDR(R4)
2561 022566 005064 000122          CLR    PRPOS(R4)        ;SET BUFFER TO HOME
2562 022572 012600          MOV    (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 - BMP(R4) HAS WORD COUNT
2573          ;      - 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    99$             ;NO, CALCULATE ONE
2577 022606 013702 007540          MOV    T.MXS,R2       ;LOAD SECTOR
2578 022612 000421          BR     5$              ;GO GET WC
2579 022614 004537 021214 99$:   JSR    R5,RAND         ;GET RANDOM # FOR SECTOR
2580 022620 013702 002136          MOV    LONUM,R2
2581 022624 042702 177700 1$:   BIC    #177700,R2     ;0-77 ONLY
2582 022630 023702 007540          CMP    T.MXS,R2       ;R2 LOWER THAN MAX
2583 022634 103003          BHS    3$             ;BRANCH IF YES
2584 022636 006202          ASR    R2              ;HALF IT
2585 022640 005202          INC    R2              ;INC SO NOT 0
2586 022642 000770          BR     1$
2587 022644 020237 007542 3$:   CMP    R2,T.MNS       ;MIN OKAY
2588 022650 103002          BHS    5$
2589 022652 006102          ROL    R2
2590 022654 000763          BR     1$

```

```

2591
2592
2593          ;NOW GET WORD COUNT
2594

```

```

2595 022656 005737 007574 5$:   TST    T.STIP
2596 022662 001003          BNE    95$
2597 022664 013737 002272 007526  MOV    MAXWC,T.MXB
2598 022672 023737 002272 007526 95$:  CMP    MAXWC,T.MXB
2599 022700 103021          BHS    97$
2600
2601 022702          PRINTF #FMT13D,#OVER,T.MXB,MAXWC
(10) 022702 013746 002272  MOV    MAXWC,-(SP)
(9)  022706 013746 007526  MOV    T.MXB,-(SP)
(8)  022712 012746 003256  MOV    #OVER,-(SP)
(7)  022716 012746 006462  MOV    #FMT13D,-(SP)
(6)  022722 012746 000004  MOV    #4,-(SP)
(3)  022726 010600          MOV    SP,R0
(4)  022730 104017          EMT    C$PNTF
(4)  022732 062706 000012  ADD    #12,SP
2602 022736 013737 002272 007526  MOV    MAXWC,T.MXB
2603
2604 022744 023737 007526 007550 97$:  CMP    T.MXB,T.MNB     ;MIN MAX EQUAL
2605 022752 003006          BGT    6$
2606 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     9$
2610 022770 004537 021214          6$:   JSR     R5,RAND          ;GET RANDOM WORD COUNT
2611 022774 013703 002136          MOV     LONUM,R3
2612 023000 042703 160000          7$:   BIC     #160000,R3      ;MAX.!!!
2613 023004 023703 007526          CMP     T.MXB,R3
2614 023010 103003                    BHIS   8$
2615 023012 006203                    ASR    R3
2616 023014 005203                    INC    R3
2617 023016 000770                    BR     7$
2618 023020 020337 007550          8$:   CMP     R3,T.MNB
2619 023024 103002                    BHIS   9$
2620 023026 006103                    ROL    R3
2621 023030 000763                    BR     7$
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
2627 023032 012701 000050          9$:   MOV     #40.,R1          ;SETUP FOR FOURTY SECTORS
2628 023036 005403                    NEG    R3                  ;MAKE WORD COUNT NEGATIVE
2629 023040 010364 000042          MOV     R3,BMP(R4)        ;LOAD WORD COUNT
2630 023044 005301                    11$:  DEC     R1                  ;DOWN COUNT MINIMUM START SECT NEEDED
2631 023046 062703 000200          ADD     #128.,R3          ;ONE SECTOR'S WORTH
2632 023052 100774                    BMI    11$                 ;STILL NEED ANOTHER SECTOR
2633 023054 020201                    CMP    R2,R1              ;DID RANDOM SECTOR SUFFICE
2634 023056 101401                    BLOS   '2$                ;BRANCH IF SUFFICED
2635 023060 010102                    MOV    R1,R2              ;NO, THEN MAKE IT FIT
2636 023062 016464 000122 000040 12$:  MOV     PRPOS(R4),BDA(R4)
2637 023070 042764 000077 000040  BIC     #77,BDA(R4)
2638 023076 050264 000040          BIS    R2,BDA(R4)
2639 023102 000205                    RTS     R5
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655          .SBTTL  ROUTINE TO DUMP BUFFER ON DCK
2656
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          : CALCULATE THE STARTING BUS ADDRESS FOR THE COMPARE
2665          :
2666
2667 023110 012737 000200 002314      MOV    #128.,DWCNT1
2668 023116 016400 000040              MOV    BDA(R4),R0          ;GET STARTING BUS ADDRESS
2669
2670 023122 013701 002254              MOV    E.DA,R1           ;GET PRESENT DISK ADDRESS
2671 023126 042700 177700              BIC    #177700,R0        ;SAVE SECTOR BITS
2672 023132 042701 177700              BIC    #177700,R1
2673 023136 010002                      MOV    R0,R2            ;SAVE A COPY
2674 023140 010103                      MOV    R1,R3            ;SAVE ANOTHER
2675 023142 160203                      SUB    R2,R3            ;GET DIFF OF SECTORS
2676 023144 005002                      CLR    R2               ;CALCULATE WORD COUNT
2677 023146 062702 000200 93$:      ADD    #128.,R2         ;ONE SECTORS WORTH
2678 023152 005303                      DEC    R3               ;DONE
2679 023154 001374                      BNE    93$              ;NO
2680 023156 016403 000042              MOV    BMP(R4),R3       ;GET WORD COUNT
2681 023162 005403                      NEG    R3               ;MAKE IT POSITIVE
2682 023164 020203                      CMP    R2,R3            ;WORKING WITH FULL SECTOR
2683 023166 003005                      BGT    94$              ;NO, GO CALC PARTIAL SECTOR
2684 023170 013702 002252              MOV    E.BA,R2         ;PRESENT BUS ADDRESS
2685 023174 162702 000400              SUB    #400,R2          ;START OF COMPARE
2686 023200 000412                      BR     96$              ;GO COMPARE BUFFER
2687 023202 160302                      94$:  SUB    R3,R2         ;GET SECTOR DIFF
2688 023204 012700 000200              MOV    #128.,R0
2689 023210 160200                      SUB    R2,R0
2690 023212 010037 002314              MOV    R0,DWCNT1
2691 023216 006300                      ASL    R0
2692 023220 013702 002252              MOV    E.BA,R2
2693 023224 160002                      SUB    R0,R2
2694 023226 96$:  PRINTB #FMT13,#BUSAD,R2,#CRLDA,LHKSEC
(11) 023226 013746 002172              MOV    CHKSEC,-(SP)
(10) 023232 012746 002407              MOV    #CRLDA,-(SP)
(9)  023236 010246                      MOV    R2,-(SP)
(8)  023240 012746 003737              MOV    #BUSAD,-(SP)
(7)  023244 012746 006445              MOV    #FMT13,-(SP)
(6)  023250 012746 000005              MOV    #5,-(SP)
(3)  023254 010600                      MOV    SP,R0
(4)  023256 104014                      EMT    C$PNTB
(4)  023260 062706 000014              ADD    #14,SP
2695 023264 012700 024430              MOV    #PATLST,R0      ;CHECK PATTERN LIST
2696 023270 012701 000010              MOV    #8.,R1
2697 023274 022062 000002 1$:  CMP    (R0)+,2(R2)
2698 023300 001415                      BEQ    2$
2699 023302 005301                      DEC    R1
2700 023304 001373                      BNE    1$
2701
2702 023306 3$:  PRINTB #FMT14,#NOREV
(8)  023306 012746 003434              MOV    #NOREV,-(SP)
(7)  023312 012746 006507              MOV    #FMT14,-(SP)
(6)  023316 012746 000002              MOV    #2,-(SP)
(3)  023322 010600                      MOV    SP,R0
(4)  023324 104014                      EMT    C$PNTB
(4)  023326 062706 000006              ADD    #C,SP
2703 023332 000532                      BR     STDMP
  
```

```

2704
2705 023334 021227 000200      2$:  CMP      (R2),#128.
2706 023340 101362                BHI      3$
2707 023342 005037 002174      CLR      DECNT
2708 023346 013701 007570      MOV      T,CLT,R1
2709
2710 023352 012237 002176      MOV      (R2)+,TEMPO      ;NONZERO WORD COUNT
2711 023356 013737 002176 002312      MOV      TEMPO,DWCNT
2712 023364 005437 002312      NEG      DWCNT
2713 023370 012237 002200      MOV      (R2)+,TEMP1
2714 023374 162737 000002 002176      SUB      #2,TEMPO
2715 023402 012737 000002 002202      MOV      #2,TEMP2
2716 023410 013703 002200      MOV      TEMP1,R3      ;WORD
2717 023414 012737 000020 002210      MOV      #16,TEMP5      ;PATTERN ADDRESS
2718 023422 005737 002176      4$:  TST      TEMPO      ;16 ENTRIES
2719 023426 001417                BEQ      6$      ;ZERO OR PATTERN
2720 023430 005337 002176      DEC      TEMPO      ;ZERO BRANCH
2721 023434 005737 002210      TST      TEMP5
2722 023440 001005                BNE      5$      ;WITHIN LIST
2723 023442 012737 000020 002210      MOV      #16,TEMP5
2724 023450 013703 002200      MOV      TEMP1,R3
2725 023454 012337 002232      5$:  MOV      (R3)+,GDDAT
2726 023460 005337 002210      DEC      TEMP5
2727 023464 000402                BR       7$
2728 023466 005037 002232      6$:  CLR      GDDAT
2729 023472 005237 002312      7$:  INC      DWCNT
2730 023476 021237 002232      CMP      (R2),GDDAT
2731 023502 001422                BEQ      8$
2732
2733 023504 005237 002174      INC      DECNT
2734 023510 005701                TST      R1
2735 023512 001416                BEQ      8$
2736 023514 005301                DEC      R1
2737 023516      PRINTB #FMT14B,TEMP2,GDDAT,(R2)
(10) 023516 011246      MOV      (R2),-(SP)
(9) 023520 013746 002232      MOV      GDDAT,-(SP)
(8) 023524 013746 002202      MOV      TEMP2,-(SP)
(7) 023530 012746 006530      MOV      #FMT14B,-(SP)
(6) 023534 012746 000004      MOV      #4,-(SP)
(3) 023540 010600      MOV      SP,R0
(4) 023542 104014      EMT      C$PNTB
(4) 023544 062706 000012      ADD      #12,SP
2738
2739 023550 005237 002202      8$:  INC      TEMP2
2740 023554 005722                TST      (R2)+
2741 023556 023737 002202 002314      CMP      TEMP2,DWCNT1
2742 023564 003716                BLE      4$
2743 023566      PRINTB #FMT9A,DECNT,TEMP2
(9) 023566 013746 002202      MOV      TEMP2,-(SP)
(8) 023572 013746 002174      MOV      DECNT,-(SP)
(7) 023576 012746 006230      MOV      #FMT9A,-(SP)
(6) 023602 012746 000003      MOV      #3,-(SP)
(3) 023606 010600      MOV      SP,R0
(4) 023610 104014      EMT      C$PNTB
(4) 023612 062706 000010      ADD      #10,SP
2744
  
```

```

2745 023616 000205          RTS      R5
2746
2747 023620 013701 007570  STDMP:  MOV      T,CLT,R1
2748 023624 012703 000012          MOV      #10.,R3
2749 023630          1$:  PRINTB  #FMT14A,(R2)
      (8) 023630 011246          MOV      (R2),-(SP)
      (7) 023632 012746 006516          MOV      #FMT14A,-(SP)
      (6) 023636 012746 000002          MOV      #2,-(SP)
      (3) 023642 010600          MOV      SP,R0
      (4) 023644 104014          EMT      C$PNTB
      (4) 023646 062706 000006          ADD      #6,SP
2750 023652 005722          TST      (R2)+
2751 023654 005303          DEC      R3
2752 023656 001012          BNE      2$
2753 023660          PRINTB  #FMT14C
      (7) 023660 012746 006525          MOV      #FMT14C,-(SP)
      (6) 023664 012746 000001          MOV      #1,-(SP)
      (3) 023670 010600          MOV      SP,R0
      (4) 023672 104014          EMT      C$PNTB
      (4) 023674 062706 000004          ADD      #4,SP
2754 023700 012703 000012          MOV      #10.,R3
2755 023704 005337 002314          2$:  DEC      DWCNT1
2756 023710 001001          BNE      3$
2757 023712 000205          RTS      R5
2758 023714 005301          3$:  DEC      R1
2759 023716 001344          BNE      1$
2760 023720 000205          RTS      R5
2761
2762
2763
2764
2765          ;ROUTINE TO CLEAR ALL DRIVE INFO, USED ON START OR
2766          ;RESTART IF CALLED. CAN BE USED TO CLEAR INDIVIDUAL DRIVE
2767          ;INFO BY BITMAP FOLLOWING CALL
2768          ;CALL JSR R5,CLEAR
2769          ;
2770
2771
2772 023722 010446          CLEAR:  MOV      R4,-(SP)          ;SAVE R4
2773 023724 012704 025056          MOV      #DRBUF,R4          ;GET BUFFER STARTS
2774 023730 005024          2$:  CLR      (R4)+          ;CLEAR
2775 023732 020427 026316          CMP      R4,#ENDBUF          ;AT END OF BUFFERS
2776 023736 001374          BNE      2$          ;NO, GO TO 2$
2777 023740 012604          4$:  MOV      (SP)+,R4          ;RESTORE CURRENT BUFFER POINTER
2778 023742 000205          RTS      R5          ;EXIT
2779
2780          .SBTTL  ROUTINE TO CHECK FOR BAD SECTOR
2781
2782          ;ROUTINE TO MATCH BAD SECTOR.....BDA(R4) IS SECTOR WE ARE LOOKING
2783          ;FOR IN LIST POINTED TO BY R$ECP1(R4).....HDRFND IS SET IF WE FIND IT.
2784          ;
2785
2786
2787 023744 005037 002170          (KBDSC: CLR      HDRFND          ;CLEAR FLAG
2788 023750 010046          MOV      RC,-(SP)          ;SAVE RC
2789 023752 010146          MOV      R1,-(SP)          ;SAVE R1

```

```

2790 023754 010246      MOV     R2,-(SP)      ;SAVE R2
2791 023756 010346      MOV     R3,-(SP)      ;SAVE R3
2792 023760 012700 000020  MOV     #16,R0        ;16 ENTRIES
2793 023764 016402 000112  1$:    MOV     BSECT(R4),R2 ;GET WHERE WE'RE LOOKING
2794 023770 005712      2$:    TST     (R2)         ;END
2795 023772 100411      BMI     4$
2796 023774 023712 002172  CMP     CHKSEC,(R2)   ;HAVE WE GOT A MATCH
2797 024000 001404      BEQ     3$            ;THEN GO SET INDICATOR, ELSE
2798 024002 005722      TST     (R2)+
2799 024004 005300      DEC     R0
2800 024006 001370      BNE     2$
2801 024010 000402      BR      4$
2802
2803 024012 005237 002170  3$:    INC     HDRFND      ;SET FLAG FOUND
2804
2805 024016 012603  4$:    MOV     (SP)+,R3
2806 024020 012602      MOV     (SP)+,R2
2807 024022 012601      MOV     (SP)+,R1
2808 024024 012600      MOV     (SP)+,R0
2809 024026 000205      RTS     R5
2810
2811
2812      ;BUFFER TO STORE BAD SECTOR LISTS
2813
2814 024030 000020  BSEC0: .BLKW 16.
2815 024070 000020  BSEC1: .BLKW 16.
2816 024130 000020  BSEC2: .BLKW 16.
2817 024170 000020  BSEC3: .BLKW 16.
2818 024230 000020  BSEC4: .BLKW 16.
2819 024270 000020  BSEC5: .BLKW 16.
2820 024330 000020  BSEC6: .BLKW 16.
2821 024370 000020  BSEC7: .BLKW 16.
2822
2823      ;LIST OF PATTERNS USED IN WRITING
2824
2825 024430 024450  PATLST: PAT0
2826 024432 024510      PAT1
2827 024434 024550      PAT2
2828 024436 024610      PAT3
2829 024440 024650      PAT4
2830 024442 024710      PAT5
2831 024444 024750      PAT6
2832 024446 025010      PAT7
2833
2834 024450 000000  PAT0:  .WORD 0
2835 024452 000000      .WORD 0
2836 024454 000000      .WORD 0
2837 024456 000000      .WORD 0
2838 024460 000000      .WORD 0
2839 024462 000000      .WORD 0
2840 024464 000000      .WORD 0
2841 024466 000000      .WORD 0
2842 024470 000000      .WORD 0
2843 024472 000000      .WORD 0
2844 024474 000000      .WORD 0
2845 024476 000000      .WORD 0
  
```

2846	024500	000000	.WORD	0
2847	024502	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	052525	.WORD	052525
2863	024540	177252	.WORD	177252
2864	024542	177252	.WORD	177252
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	052525	.WORD	52525
2887	024614	052525	.WORD	52525
2888	024616	125252	.WORD	125252
2889	024620	125252	.WORD	125252
2890	024622	125252	.WORD	125252
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	155555		.WORD	155555
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	150442	.WORD	150442
2922	024714	064221	.WORD	64221
2923	024716	132110	.WORD	132110
2924	024720	055044	.WORD	55044
2925	024722	026422	.WORD	26422
2926	024724	013211	.WORD	13211
2927	024726	105504	.WORD	105504
2928	024730	042642	.WORD	42642
2929	024732	021321	.WORD	21321
2930	024734	110550	.WORD	110550
2931	024736	044264	.WORD	44264
2932	024740	022132	.WORD	22132
2933	024742	011055	.WORD	11055
2934	024744	104426	.WORD	104426
2935	024746	042213	.WORD	42213
2936				
2937	024750	177777	PAT6: .WORD	177777
2938	024752	177777	.WORD	177777
2939	024754	177777	.WORD	177777
2940	024756	177777	.WORD	177777
2941	024760	177777	.WORD	177777
2942	024762	177777	.WORD	177777
2943	024764	177777	.WORD	177777
2944	024766	177777	.WORD	177777
2945	024770	177777	.WORD	177777
2946	024772	177777	.WORD	177777
2947	024774	177777	.WORD	177777
2948	024776	177777	.WORD	177777
2949	025000	177777	.WORD	177777
2950	025002	177777	.WORD	177777
2951	025004	177777	.WORD	177777
2952	025006	177777	.WORD	177777
2953				
2954	025010	045513	PAT7: .WORD	45513
2955	025012	122645	.WORD	122645
2956	025014	151322	.WORD	151322
2957	025016	064551	.WORD	64551
2958	025020	132264	.WORD	132264
2959	025022	055132	.WORD	55132
2960	025024	026455	.WORD	26455
2961	025026	113226	.WORD	113226
2962	025030	045513	.WORD	45513
2963	025032	122645	.WORD	122645
2964	025034	151322	.WORD	151322
2965	025036	064551	.WORD	64551
2966	025040	132264	.WORD	132264
2967	025042	055132	.WORD	55132
2968	025044	026455	.WORD	26455
2969	025046	113226	.WORD	113226
2970				
2971				
2972				
2973	025050	000240	ENDOF PROGRAM:	NOP

```
2974 025052          ENDTST
(3) 025052          L10022:
(3) 025052 104001   EMT      C$ETST
2975 025054 000000   HALT
2976
2977          .SBTTL  DRIVE INFORMATION BUFFERS
2978
2979          ;DRIVE INFORMATION BUFF.LR
2980
2981
2982          .LIST  ME
2983
2984 025056          DRBUF:
3030
(1) 025056 000000   SKCNT          ;SEEK OPERATION COUNT
(1) 025060 000002   RXFR1          ;READ OPERATION COUNT (BITS) LOW ORDER
(1) 025062 000004   RXFR2          ;                "                "                HIGH ORDER
(1) 025064 000006   WXFR1          ;WRITE OPERATION COUNT (BITS) LOW ORDER
(1) 025066 000010   WXFR2          ;                "                "                HIGH ORDER
(1) 025070 000012   ERRCNT        ;ERROR COUNT - HARD
(1) 025072 000014   SFTCNT        ;ERROR COUNT - SOFT
(1) 025074 000016   SKECNT        ;SEEK ERROR COUNT
(1) 025076 000020   DERCNT        ;DRIVE ERROR COUNT
(1) 025100 000022   DRCRER        ;DATA CRC ERROR COUNT
(1) 025102 000024   HRCRER        ;HEADER CRC ERROR COUNT
(1) 025104 000026   DLTCNT        ;DATA LATE ERROR COUNT
(1) 025106 000030   OPICNT        ;OPERATION INCOMPLETE ERROR COUNT
(1) 025110 000032   HNFERR        ;HEADER NOT FOUND ERROR COUNT
(1) 025112 000034   NXMCNT        ;NON EXISTANT MEMORY ERROR COUNT
(1) 025114 000036   RETRY         ;PRESENT RETRY NUMBER
(1) 025116 000040   BDA          ;                DISK ADDRESS CONTENTS
(1) 025120 000042   BMP          ;PRESENT MULTIPURPOSE CONTENTS
(1) 025122 000044   FUNC         ;LAST FUNCTION LOADED
(1) 025124 000046   BCSADR        ;CSR IMAGE OF LAST COMMAND
(1) 025126 000050   LSTHDR        ;LAST POSITION ON DISK
(1) 025130 000052   RTYPE        ;ERROR ON WHICH RECOVERY IS IN PROGRESS
(1) 025132 000054   SKCNT1        ;SEEK COUNT LOW ORDER
(1) 025134 000056   PRFLGS        ;PROGRAM INTERNAL FLAGS
(1) 025136 000060   RXFR3         ;READ COUNT THIRD
(1) 025140 000062   WXFR3         ;WRITE COUNT THIRD
(1) 025142 000064   LSTDA        ;DISK ADDRESS OF SOFT ERROR
(1) 025144 000066   DIFWD        ;LAST DIFFERENCE WORD OF SEEK
(1) 025146 000070   DPHOUR        ;TIME DRIVE WAS DROPPED
(1) 025150 000072   TRERR        ;TRACKING ERROR COUNT
(1) 025152 000074   DATCER        ;
(1) 025154 000076   DOWCK        ;WRITE CHECK NECESSARY
(1) 025156 000100   SERNM1        ;SERIAL NUMBER OF CARTRIDGE
(1) 025160 000102   SERNM2        ;SERIAL NUMBER OF CARTRIDGE
(1) 025162 000104   DCS          ;CSR ADDRESS
(1) 025164 000106   DRSEL        ;DRIVE SELECT BITS(8,9,10)
(1) 025166 000110   BBA          ;PRESENT BUS ADDRESS CONTENTS
(1) 025170 000112   BSECP        ;POINTER TO BAD SECTOR FILE
(1) 025172 000114   RSEEK        ;
(1) 025174 000116   SOFTCS        ;CSR AT TIME OF SOFT ERROR
(1) 025176 000120   WRIPG        ;WRITE IN PROGRESS DURING PWR FAIL
(1) 025200 000122   PRPOS        ;PRESENT POSITION ON DISK
```


(1)				
(1)				
(1)	025202	000000	SKCNT	:SEEK OPERATION COUNT
(1)	025204	000002	RXFR1	:READ OPERATION COUNT (BITS) LOW ORDER
(1)	025206	000004	RXFR2	:HIGH ORDER
(1)	025210	000006	WXFR1	:WRITE OPERATION COUNT (BITS) LOW ORDER
(1)	025212	000010	WXFR2	:HIGH ORDER
(1)	025214	000012	ERRCNT	:ERROR COUNT - HARD
(1)	025216	000014	SFTCNT	:ERROR COUNT - SOFT
(1)	025220	000016	SKECNT	:SEEK ERROR COUNT
(1)	025222	000020	DERCNT	:DRIVE ERROR COUNT
(1)	025224	000022	DCRCER	:DATA CRC ERROR COUNT
(1)	025226	000024	HRCER	:HEADER CRC ERROR COUNT
(1)	025230	000026	DLTCNT	:DATA LATE ERROR COUNT
(1)	025232	000030	OPICNT	:OPERATION INCOMPLETE ERROR COUNT
(1)	025234	000032	HNFERR	:HEADER NOT FOUND ERROR COUNT
(1)	025236	000034	NXMCNT	:NON EXISTANT MEMORY ERROR COUNT
(1)	025240	000036	RETRY	:PRESENT RETRY NUMBER
(1)	025242	000040	BDA	:DISK ADDRESS CONTENTS
(1)	025244	000042	BMP	:PRESENT MULTIPURPOSE CONTENTS
(1)	025246	000044	FUNC	:LAST FUNCTION LOADED
(1)	025250	000046	BCSADR	:CSR IMAGE OF LAST COMMAND
(1)	025252	000050	LSTHDR	:LAST POSITION ON DISK
(1)	025254	000052	RTYPE	:ERROR ON WHICH RECOVERY IS IN PROGRESS
(1)	025256	000054	SKCNT1	:SEEK COUNT LOW ORDER
(1)	025260	000056	PRFLGS	:PROGRAM INTERNAL FLAGS
(1)	025262	000060	RXFR3	:READ COUNT THIRD
(1)	025264	000062	WXFR3	:WRITE COUNT THIRD
(1)	025266	000064	LSTDA	:DISK ADDRESS OF SOFT ERROR
(1)	025270	000066	DIFWD	:LAST DIFFERENCE WORD OF SEEK
(1)	025272	000070	DPHOUR	:TIME DRIVE WAS DROPPED
(1)	025274	000072	TRERR	:TRACKING ERROR COUNT
(1)	025276	000074	DATCER	
(1)	025300	000076	DOWCK	:WRITE CHECK NECESSARY
(1)	025302	000100	SERNM1	:SERIAL NUMBER OF CARTRIDGE
(1)	025304	000102	SERNM2	:SERIAL NUMBER OF CARTRIDGE
(1)	025306	000104	DCS	:CSR ADDRESS
(1)	025310	000106	DRSEL	:DRIVE SELECT BITS(8,9,10)
(1)	025312	000110	BBA	:PRESENT BUS ADDRESS CONTENTS
(1)	025314	000112	BSECTP	:POINTER TO BAD SECTOR FILE
(1)	025316	000114	RSEEK	
(1)	025320	000116	SOFTCS	:CSR AT TIME OF SOFT ERROR
(1)	025322	000120	WRIPG	:WRITE IN PROGRESS DURING PWR FAIL
(1)	025324	000122	PRPOS	:PRESENT POSITION ON DISK
(1)				
(1)				
(1)	025326	000000	SKCNT	:SEEK OPERATION COUNT
(1)	025330	000002	RXFR1	:READ OPERATION COUNT (BITS) LOW ORDER
(1)	025332	000004	RXFR2	:HIGH ORDER
(1)	025334	000006	WXFR1	:WRITE OPERATION COUNT (BITS) LOW ORDER
(1)	025336	000010	WXFR2	:HIGH ORDER
(1)	025340	000012	ERRCNT	:ERROR COUNT - HARD
(1)	025342	000014	SFTCNT	:ERROR COUNT - SOFT
(1)	025344	000016	SKECNT	:SEEK ERROR COUNT
(1)	025346	000020	DERCNT	:DRIVE ERROR COUNT
(1)	025350	000022	DCRCER	:DATA CRC ERROR COUNT

(1)	025352	000024	HRCRER	:HEADER CRC ERROR COUNT
(1)	025354	000026	DLTCNT	:DATA LATE ERROR COUNT
(1)	025356	000030	OPICNT	:OPERATION INCOMPLETE ERROR COUNT
(1)	025360	000032	HNFERR	:HEADER NOT FOUND ERROR COUNT
(1)	025362	000034	NXMCNT	:NON EXISTANT MEMORY ERROR COUNT
(1)	025364	000036	RETRY	:PRESENT RETRY NUMBER
(1)	025366	000040	BDA	:DISK ADDRESS CONTENTS
(1)	025370	000042	BMP	:PRESENT MULTIPURPOSE CONTENTS
(1)	025372	000044	FUNC	:LAST FUNCTION LOADED
(1)	025374	000046	BCSADR	:CSR IMAGE OF LAST COMMAND
(1)	025376	000050	LSTHDR	:LAST POSITION ON DISK
(1)	025400	000052	RTYPE	:ERROR ON WHICH RECOVERY IS IN PROGRESS
(1)	025402	000054	SKCNT1	:SEEK COUNT LOW ORDER
(1)	025404	000056	PRFLGS	:PROGRAM INTERNAL FLAGS
(1)	025406	000060	RXFR3	:READ COUNT THIRD
(1)	025410	000062	WXFR3	:WRITE COUNT THIRD
(1)	025412	000064	LSTDA	:DISK ADDRESS OF SOFT ERROR
(1)	025414	000066	DIFWD	:LAST DIFFERENCE WORD OF SEEK
(1)	025416	000070	DPHOUR	:TIME DRIVE WAS DROPPED
(1)	025420	000072	TRERR	:TRACKING ERROR COUNT
(1)	025422	000074	DATCER	
(1)	025424	000076	DOWCK	:WRITE CHECK NECESSARY
(1)	025426	000100	SERNM1	:SERIAL NUMBER OF CARTRIDGE
(1)	025430	000102	SERNM2	:SERIAL NUMBER OF CARTRIDGE
(1)	025432	000104	DCS	:CSR ADDRESS
(1)	025434	000106	DRSEL	:DRIVE SELECT BITS(8,9,10)
(1)	025436	000110	BBA	:PRESENT BUS ADDRESS CONTENTS
(1)	025440	000112	BSECPT	:POINTER TO BAD SECTOR FILE
(1)	025442	000114	RSEEK	
(1)	025444	000116	SOFTCS	:CSR AT TIME OF SOFT ERROR
(1)	025446	000120	WRIPG	:WRITE IN PROGRESS DURING PWR FAIL
(1)	025450	000122	PRPOS	:PRESENT POSITION ON DISK
(1)				
(1)	025452	000000	SKCNT	:SEEK OPERATION COUNT
(1)	025454	000002	RXFR1	:READ OPERATION COUNT (BITS) LOW ORDER
(1)	025456	000004	RXFR2	:READ OPERATION COUNT (BITS) HIGH ORDER
(1)	025460	000006	WXFR1	:WRITE OPERATION COUNT (BITS) LOW ORDER
(1)	025462	000010	WXFR2	:WRITE OPERATION COUNT (BITS) HIGH ORDER
(1)	025464	000012	ERRCNT	:ERROR COUNT - HARD
(1)	025466	000014	SFTCNT	:ERROR COUNT - SOFT
(1)	025470	000016	SKECNT	:SEEK ERROR COUNT
(1)	025472	000020	DERCNT	:DRIVE ERROR COUNT
(1)	025474	000022	DCRCER	:DATA CRC ERROR COUNT
(1)	025476	000024	HRCRER	:HEADER CRC ERROR COUNT
(1)	025500	000026	DLTCNT	:DATA LATE ERROR COUNT
(1)	025502	000030	OPICNT	:OPERATION INCOMPLETE ERROR COUNT
(1)	025504	000032	HNFERR	:HEADER NOT FOUND ERROR COUNT
(1)	025506	000034	NXMCNT	:NON EXISTANT MEMORY ERROR COUNT
(1)	025510	000036	RETRY	:PRESENT RETRY NUMBER
(1)	025512	000040	BDA	:DISK ADDRESS CONTENTS
(1)	025514	000042	BMP	:PRESENT MULTIPURPOSE CONTENTS
(1)	025516	000044	FUNC	:LAST FUNCTION LOADED
(1)	025520	000046	BCSADR	:CSR IMAGE OF LAST COMMAND
(1)	025522	000050	LSTHDR	:LAST POSITION ON DISK
(1)	025524	000052	RTYPE	:ERROR ON WHICH RECOVERY IS IN PROGRESS

(1)	025526	000054	SKCNT1	:SEEK COUNT LOW ORDER
(1)	025530	000056	PRFLGS	:PROGRAM INTERNAL FLAGS
(1)	025532	000060	RXFR3	:READ COUNT THIRD
(1)	025534	000062	WXFR3	:WRITE COUNT THIRD
(1)	025536	000064	LSTDA	:DISK ADDRESS OF SOFT ERROR
(1)	025540	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	DOWCK	:WRITE CHECK NECESSARY
(1)	025552	000100	SERNM1	:SERIAL NUMBER OF CARTRIDGE
(1)	025554	000102	SERNM2	:SERIAL NUMBER OF CARTRIDGE
(1)	025556	000104	DCS	:CSR ADDRESS
(1)	025560	000106	DRSEL	:DRIVE SELECT BITS(8,9,10)
(1)	025562	000110	BBA	:PRESENT BUS ADDRESS CONTENTS
(1)	025564	000112	BSECTP	:POINTER TO BAD SECTOR FILE
(1)	025566	000114	RSECK	
(1)	025570	000116	SOFTCS	:CSR AT TIME OF SOFT ERROR
(1)	025572	000120	WRIPG	:WRITE IN PROGRESS DURING PWR FAIL
(1)	025574	000122	PRPOS	:PRESENT POSITION ON DISK
(1)				
(1)	025576	000000	SKCNT	:SEEK OPERATION COUNT
(1)	025600	000002	RXFR1	:READ OPERATION COUNT (BITS) LOW ORDER
(1)	025602	000004	RXFR2	:READ OPERATION COUNT (BITS) HIGH ORDER
(1)	025604	000006	WXFR1	:WRITE OPERATION COUNT (BITS) LOW ORDER
(1)	025606	000010	WXFR2	:WRITE OPERATION COUNT (BITS) HIGH ORDER
(1)	025610	000012	ERRCNT	:ERROR COUNT - HARD
(1)	025612	000014	SFTCNT	:ERROR COUNT - SOFT
(1)	025614	000016	SKECNT	:SEEK ERROR COUNT
(1)	025616	000020	DERCNT	:DRIVE ERROR COUNT
(1)	025620	000022	DCRCER	:DATA CRC ERROR COUNT
(1)	025622	000024	HRCRCR	:HEADER CRC ERROR COUNT
(1)	025624	000026	DLTCNT	: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	BDA	: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	RXFR3	:READ COUNT THIRD
(1)	025660	000062	WXFR3	: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)	025672	000074	DATCER	
(1)	025674	000076	DOWCK	:WRITE CHECK NECESSARY
(1)	025676	000100	SERNM1	:SERIAL NUMBER OF CARTRIDGE
(1)	025700	000102	SERNM2	:SERIAL NUMBER OF CARTRIDGE

(1)	025702	000104	DCS	:CSR ADDRESS
(1)	025704	000106	DRSEL	:DRIVE SELECT BITS(8,9,10)
(1)	025706	000110	BBA	:PRESENT BUS ADDRESS CONTENTS
(1)	025710	000112	BSECPT	:POINTER TO BAD SECTOR FILE
(1)	025712	000114	RSEEK	
(1)	025714	000116	SOFTCS	:CSR AT TIME OF SOFT ERROR
(1)	025716	000120	WRIPG	:WRITE IN PROGRESS DURING PWR FAIL
(1)	025720	000122	PRPOS	:PRESENT POSITION ON DISK
(1)				
(1)	025722	000000	SKCNT	:SEEK OPERATION COUNT
(1)	025724	000002	RXFR1	:READ OPERATION COUNT (BITS) LOW ORDER
(1)	025726	000004	RXFR2	:HIGH ORDER
(1)	025730	000006	WXFR1	:WRITE OPERATION COUNT (BITS) LOW ORDER
(1)	025732	000010	WXFR2	:HIGH ORDER
(1)	025734	000012	ERRCNT	:ERROR COUNT - HARD
(1)	025736	000014	SFTCNT	:ERROR COUNT - SOFT
(1)	025740	000016	SKECNT	:SEEK ERROR COUNT
(1)	025742	000020	DERCNT	:DRIVE ERROR COUNT
(1)	025744	000022	DRCRCR	:DATA CRC ERROR COUNT
(1)	025746	000024	HRCRCR	:HEADER CRC ERROR COUNT
(1)	025750	000026	DLTCNT	: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	BDA	: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	RXFR3	:READ COUNT THIRD
(1)	026004	000062	WXFR3	:WRITE COUNT THIRD
(1)	026006	000064	LSTDA	:DISK ADDRESS OF SOFT ERROR
(1)	026010	000066	DIFWD	:LAST DIFFERENCE WORD OF SEEK
(1)	026012	000070	DPHOUR	:TIME DRIVE WAS DROPPED
(1)	026014	000072	TRERR	:TRACKING ERROR COUNT
(1)	026016	000074	DATCER	
(1)	026020	000076	DOWCK	:WRITE CHECK NECESSARY
(1)	026022	000100	SERNM1	:SERIAL NUMBER OF CARTRIDGE
(1)	026024	000102	SERNM2	:SERIAL NUMBER OF CARTRIDGE
(1)	026026	000104	DCS	:CSR ADDRESS
(1)	026030	000106	DRSEL	:DRIVE SELECT BITS(8,9,10)
(1)	026032	000110	BBA	:PRESENT BUS ADDRESS CONTENTS
(1)	026034	000112	BSECPT	:POINTER TO BAD SECTOR FILE
(1)	026036	000114	RSEEK	
(1)	026040	000116	SOFTCS	:CSR AT TIME OF SOFT ERROR
(1)	026042	000120	WRIPG	:WRITE IN PROGRESS DURING PWR FAIL
(1)	026044	000122	PRPOS	:PRESENT POSITION ON DISK
(1)				
(1)	026046	000000	SKCNT	:SEEK OPERATION COUNT
(1)	026050	000002	RXFR1	:READ OPERATION COUNT (BITS) LOW ORDER

(1)	026052	000004	RXFR2	: " " " " HIGH ORDER
(1)	026054	000006	WXFR1	: WRITE OPERATION COUNT (BITS) LOW ORDER
(1)	026056	000010	WXFR2	: " " " " HIGH ORDER
(1)	026060	000012	ERRCNT	: ERROR COUNT - HARD
(1)	026062	000014	SFTCNT	: ERROR COUNT - SOFT
(1)	026064	000016	SKECNT	: SEEK ERROR COUNT
(1)	026066	000020	DERCNT	: DRIVE ERROR COUNT
(1)	026070	000022	DCRCER	: DATA CRC ERROR COUNT
(1)	026072	000024	HRCRCR	: HEADER CRC ERROR COUNT
(1)	026074	000026	DLTCNT	: DATA LATE ERROR COUNT
(1)	026076	000030	OPICNT	: OPERATION INCOMPLETE ERROR COUNT
(1)	026100	000032	HNFERR	: HEADER NOT FOUND ERROR COUNT
(1)	026102	000034	NXMCNT	: NON EXISTANT MEMORY ERROR COUNT
(1)	026104	000036	RETRY	: PRESENT RETRY NUMBER
(1)	026106	000040	BDA	: " " DISK ADDRESS CONTENTS
(1)	026110	000042	BMP	: PRESENT MULTIPURPOSE CONTENTS
(1)	026112	000044	FUNC	: LAST FUNCTION LOADED
(1)	026114	000046	BCSADR	: CSR IMAGE OF LAST COMMAND
(1)	026116	000050	LSTHDR	: LAST POSITION ON DISK
(1)	026120	000052	RTYPE	: ERROR ON WHICH RECOVERY IS IN PROGRESS
(1)	026122	000054	SKCNT1	: SEEK COUNT LOW ORDER
(1)	026124	000056	PRFLGS	: PROGRAM INTERNAL FLAGS
(1)	026126	000060	RXFR3	: READ COUNT THIRD
(1)	026130	000062	WXFR3	: WRITE COUNT THIRD
(1)	026132	000064	LSTDA	: DISK ADDRESS OF SOFT ERROR
(1)	026134	000066	DIFWD	: LAST DIFFERENCE WORD OF SEEK
(1)	026136	000070	DPHOUR	: TIME DRIVE WAS DROPPED
(1)	026140	000072	TRERR	: TRACKING ERROR COUNT
(1)	026142	000074	DATCER	
(1)	026144	000076	DOWCK	: WRITE CHECK NECESSARY
(1)	026146	000100	SERNM1	: SERIAL NUMBER OF CARTRIDGE
(1)	026150	000102	SERNM2	: SERIAL NUMBER OF CARTRIDGE
(1)	026152	000104	DCS	: CSR ADDRESS
(1)	026154	000106	DRSEL	: DRIVE SELECT BITS(8,9,10)
(1)	026156	000110	BBA	: PRESENT BUS ADDRESS CONTENTS
(1)	026160	000112	BSECPT	: POINTER TO BAD SECTOR FILE
(1)	026162	000114	RSEEK	
(1)	026164	000116	SOFTCS	: CSR AT TIME OF SOFT ERROR
(1)	026166	000120	WRIPG	: WRITE IN PROGRESS DURING PWR FAIL
(1)	026170	000122	PRPOS	: PRESENT POSITION ON DISK
(1)				
(1)	026172	000000	SKCNT	: SEEK OPERATION COUNT
(1)	026174	000002	RXFR1	: READ OPERATION COUNT (BITS) LOW ORDER
(1)	026176	000004	RXFR2	: " " " " HIGH ORDER
(1)	026200	000006	WXFR1	: WRITE OPERATION COUNT (BITS) LOW ORDER
(1)	026202	000010	WXFR2	: " " " " HIGH ORDER
(1)	026204	000012	ERRCNT	: ERROR COUNT - HARD
(1)	026206	000014	SFTCNT	: ERROR COUNT - SOFT
(1)	026210	000016	SKECNT	: SEEK ERROR COUNT
(1)	026212	000020	DERCNT	: DRIVE ERROR COUNT
(1)	026214	000022	DCRCER	: DATA CRC ERROR COUNT
(1)	026216	000024	HRCRCR	: HEADER CRC ERROR COUNT
(1)	026220	000026	DLTCNT	: DATA LATE ERROR COUNT
(1)	026222	000030	OPICNT	: OPERATION INCOMPLETE ERROR COUNT
(1)	026224	000032	HNFERR	: HEADER NOT FOUND ERROR COUNT

(1)	026226	000034	NXMCNT	:NON EXISTANT MEMORY ERROR COUNT
(1)	026230	000036	RETRY	:PRESENT RETRY NUMBER
(1)	026232	000040	BDA	:DISK ADDRESS CONTENTS
(1)	026234	000042	BMP	:PRESENT MULTIPURPOSE CONTENTS
(1)	026236	000044	FUNC	:LAST FUNCTION LOADED
(1)	026240	000046	BCSADR	:CSR IMAGE OF LAST COMMAND
(1)	026242	000050	LSTHDR	:LAST POSITION ON DISK
(1)	026244	000052	RTYPE	:ERROR ON WHICH RECOVERY IS IN PROGRESS
(1)	026246	000054	SKCNT1	:SEEK COUNT LOW ORDER
(1)	026250	000056	PRFLGS	:PROGRAM INTERNAL FLAGS
(1)	026252	000060	RXFR3	:READ COUNT THIRD
(1)	026254	000062	WXFR3	:WRITE COUNT THIRD
(1)	026256	000064	LSTDA	:DISK ADDRESS OF SOFT ERROR
(1)	026260	000066	DIFWD	:LAST DIFFERENCE WORD OF SEEK
(1)	026262	000070	DPHOUR	:TIME DRIVE WAS DROPPED
(1)	026264	000072	TRERR	:TRACKING ERROR COUNT
(1)	026266	000074	DATCER	
(1)	026270	000076	DOWCK	:WRITE CHECK NECESSARY
(1)	026272	000100	SERNM1	:SERIAL NUMBER OF CARTRIDGE
(1)	026274	000102	SERNM2	:SERIAL NUMBER OF CARTRIDGE
(1)	026276	000104	DCS	:CSR ADDRESS
(1)	026300	000106	DRSEL	:DRIVE SELECT BITS(8,9,10)
(1)	026302	000110	BBA	:PRESENT BUS ADDRESS CONTENTS
(1)	026304	000112	BSECPT	:POINTER TO BAD SECTOR FILE
(1)	026306	000114	RSEEK	
(1)	026310	000116	SOFTCS	:CSR AT TIME OF SOFT ERROR
(1)	026312	000120	WRIPG	:WRITE IN PROGRESS DURING PWR FAIL
(1)	026314	000122	PRPOS	:PRESENT POSITION ON DISK
(1)				
3031			.NLIST ME	
3032				
3033	026316	000000	ENDBUF: .WORD 0	
3034				
3035				
3036	026320		BGNMOD HRDPRM	
3037				
3038	026320		BGNHRD	
(3)	026320	000025	.WORD L10024-L\$HARD/2	
3039				
3040	026322		GPRML CNTYPE,CNT,1,YES	
(4)	026322	004130	.WORD T\$CODE	
(4)	026324	026374	.WORD CNTYPE	
(4)	026326	000001	.WORD 1	
3041	026330		GPRMA CSRMSG,CSR,0,160000,177776,YES	
(4)	026330	000031	.WORD T\$CODE	
(4)	026332	026401	.WORD CSRMSG	
(4)	026334	160000	.WORD T\$LOLIM	
(4)	026336	177776	.WORD T\$HILIM	
3042	026340		GPRMA VECMSG,VECT,0,0,776,YES	
(4)	026340	001031	.WORD T\$CODE	
(4)	026342	026426	.WORD VECMSG	
(4)	026344	000000	.WORD T\$LOLIM	
(4)	026346	000776	.WORD T\$HILIM	
3043	026350		GPRMD BRMSG,PRIOR,0,340,0,7,YES	
(4)	026350	002032	.WORD T\$CODE	
(4)	026352	026415	.WORD BRMSG	

(4)	026354	000340				.WORD	340
(4)	026356	000000				.WORD	T\$LOLIM
(4)	026360	000007				.WORD	T\$HILIM
3044	026362					GPRMD	DRMSG,DRBT,0,03400,0,7,YES
(4)	026362	003032				.WORD	T\$CODE
(4)	026364	026435				.WORD	DRMSG
(4)	026366	003400				.WORD	03400
(4)	026370	000000				.WORD	T\$LOLIM
(4)	026372	000007				.WORD	T\$HILIM
3045							
3046	026374					ENDHRD	
(2)						.EVEN	
(3)	026374				L'0024:		
3047							
3051							
3052	026374	046122	030461	000	CNTYPE:	.ASCIZ	/RL11/
3053	026401	102	051525	040440	CSRMSG:	.ASCIZ	/BUS ADDRESS/
3054	026415	102	020122	042514	BRMSG:	.ASCIZ	/BR LEVEL/
3055	026426	042526	052103	051117	VECMMSG:	.ASCIZ	/VECTOR/
3056	026435	104	044522	042526	DRMSG:	.ASCIZ	/DRIVE/
3057							
3061							
3062		026444				.EVEN	
3063							
3064	026444					ENDMOD	
3065							
3066	026444				BGNMOD	SFTPRM	
3067							
3068	026444				BGNSFT		
(3)	026444	000217			.WORD	L10025-L\$SOFT/2	
3069							
3070	026446				GPRMD	RTMSG,RLT,D,177777,0,177777,YES	
(4)	026446	000052			.WORD	T\$CODE	
(4)	026450	027311			.WORD	RTMSG	
(4)	026452	177777			.WORD	177777	
(4)	026454	000000			.WORD	T\$LOLIM	
(4)	026456	177777			.WORD	T\$HILIM	
3071	026460				GPRMD	SRTMSG,SRLT,D,177777,0,177777,YES	
(4)	026460	031052			.WORD	T\$CODE	
(4)	026462	027134			.WORD	SRTMSG	
(4)	026464	177777			.WORD	177777	
(4)	026466	000000			.WORD	T\$LOLIM	
(4)	026470	177777			.WORD	T\$HILIM	
3072	026472				GPRML	FDCHK,DCKFG,1,YES	
(4)	026472	020130			.WORD	T\$CODE	
(4)	026474	027616			.WORD	FDCHK	
(4)	026476	000001			.WORD	1	
3073	026500				XFERF	5\$	
(5)	026500	006044			.WORD	T\$CODE	
3074	026502				GPRMD	CHKLMT,CLMT,D,177777,0,128.,YES	
(4)	026502	032052			.WORD	T\$CODE	
(4)	026504	027153			.WORD	CHKLMT	
(4)	026506	177777			.WORD	177777	
(4)	026510	000000			.WORD	T\$LOLIM	
(4)	026512	000200			.WORD	T\$HILIM	
3075	026514				GPRMD	INMSG,TYT,D,177777,1,177777,YES	

(4)	026514	005052	.WORD	T\$CODE
(4)	026516	027421	.WORD	INMSG
(4)	026520	177777	.WORD	177777
(4)	026522	000001	.WORD	T\$LLOLIM
(4)	026524	177777	.WORD	T\$HILIM
3076	026526		GPRML	DRPMS,DRFLG,1,YES
(4)	026526	021130	.WORD	T\$CODE
(4)	026530	027677	.WORD	DRPMS
(4)	026532	000001	.WORD	1
3077	026534		XFERF	3\$
(5)	026534	032044	.WORD	T\$CODE
3078	026536		GPRMD	ERMSG,ELT,D,177777,0,177777,YES
(4)	026536	001052	.WORD	T\$CODE
(4)	026540	027225	.WORD	ERMSG
(4)	026542	177777	.WORD	177777
(4)	026544	000000	.WORD	T\$LLOLIM
(4)	026546	177777	.WORD	T\$HILIM
3079	026550		GPRMD	SFTMSG,SEL,D,177777,0,177777,YES
(4)	026550	023052	.WORD	T\$CODE
(4)	026552	027241	.WORD	SFTMSG
(4)	026554	177777	.WORD	177777
(4)	026556	000000	.WORD	T\$LLOLIM
(4)	026560	177777	.WORD	T\$HILIM
3080	026562		GPRMD	DERPMS,DCD,D,177777,0,177777,YES
(4)	026562	036052	.WORD	T\$CODE
(4)	026564	027742	.WORD	DERPMS
(4)	026566	177777	.WORD	177777
(4)	026570	000000	.WORD	T\$LLOLIM
(4)	026572	177777	.WORD	T\$HILIM
3081	026574		GPRMD	SEMSG,SET,D,177777,0,177777,YES
(4)	026574	002052	.WORD	T\$CODE
(4)	026576	027323	.WORD	SEMSG
(4)	026600	177777	.WORD	177777
(4)	026602	000000	.WORD	T\$LLOLIM
(4)	026604	177777	.WORD	T\$HILIM
3082	026606		GPRMD	DREMSG,DET,D,177777,0,177777,YES
(4)	026606	025052	.WORD	T\$CODE
(4)	026610	027336	.WORD	DREMSG
(4)	026612	177777	.WORD	177777
(4)	026614	000000	.WORD	T\$LLOLIM
(4)	026616	177777	.WORD	T\$HILIM
3083	026620		GPRML	STLMT,OPFLG,1,YES
(4)	026620	024130	.WORD	T\$CODE
(4)	026622	027642	.WORD	STLMT
(4)	026624	000001	.WORD	1
3084	026626		XFERF	2\$
(5)	026626	013044	.WORD	T\$CODE
3085	026630		GPRMD	DAMSG,DAT,D,177777,1,177776,YES
(4)	026630	003052	.WORD	T\$CODE
(4)	026632	027351	.WORD	DAMSG
(4)	026634	177777	.WORD	177777
(4)	026636	000001	.WORD	T\$LLOLIM
(4)	026640	177776	.WORD	T\$HILIM
3086	026642		GPRMD	SKMSG,SKT,D,177777,1,177776,YES
(4)	026642	004052	.WORD	T\$CODE
(4)	026644	027401	.WORD	SKMSG

3\$:

(4)	026646	177777	.WORD	177777
(4)	026650	000001	.WORD	T\$LOLIM
(4)	026652	177776	.WORD	T\$HILIM
3087	026654		GPRML	2\$: CHANGE,CHFLG,1,YES
(4)	026654	010130	.WORD	T\$CODE
(4)	026656	027451	.WORD	CHANGE
(4)	026660	000001	.WORD	1
3088	026662		XFERF	1\$
(5)	026662	106044	.WORD	T\$CODE
3089	026664		GPRML	STIPMS,STIP,1,YES
(4)	026664	034130	.WORD	T\$CODE
(4)	026666	027104	.WORD	STIPMS
(4)	026670	000001	.WORD	1
3090	026672		XFERF	6\$
(5)	026672	013044	.WORD	T\$CODE
3091	026674		GPRMD	MXBUF,MXB,D,177777,3,5120.,YES
(4)	026674	011052	.WORD	T\$CODE
(4)	026676	027505	.WORD	MXBUF
(4)	026700	177777	.WORD	177777
(4)	026702	000003	.WORD	T\$LOLIM
(4)	026704	012000	.WORD	T\$HILIM
3092	026706		GPRMD	MINBUF,MNB,D,177777,3.,5120.,YES
(4)	026706	022052	.WORD	T\$CODE
(4)	026710	027516	.WORD	MINBUF
(4)	026712	177777	.WORD	177777
(4)	026714	000003	.WORD	T\$LOLIM
(4)	026716	012000	.WORD	T\$HILIM
3093	026720		GPRML	6\$: RONLY,ROF,1,YES
(4)	026720	026130	.WORD	T\$CODE
(4)	026722	027173	.WORD	RONLY
(4)	026724	000001	.WORD	1
3094	026726		GPRML	RANPAT,RAN,1,YES
(4)	026726	027130	.WORD	T\$CODE
(4)	026730	027203	.WORD	RANPAT
(4)	026732	000001	.WORD	1
3095	026734		XFERT	4\$
(5)	026734	006024	.WORD	T\$CODE
3096	026736		GPRMD	ONLONE,PAT,0,17,0,7,YES
(4)	026736	030032	.WORD	T\$CODE
(4)	026740	027213	.WORD	ONLONE
(4)	026742	000017	.WORD	17
(4)	026744	000000	.WORD	T\$LOLIM
(4)	026746	000007	.WORD	T\$HILIM
3097	026750		GPRML	4\$: WCKMSG,WCK,1,YES
(4)	026750	035130	.WORD	T\$CODE
(4)	026752	027733	.WORD	WCKMSG
(4)	026754	000001	.WORD	1
3098	026756		GPRMD	7\$: CMMSG,RDT,D,177777,0,128.,YES
(4)	026756	006052	.WORD	T\$CODE
(4)	026760	027770	.WORD	CMMSG
(4)	026762	177777	.WORD	177777
(4)	026764	000000	.WORD	T\$LOLIM
(4)	026766	000200	.WORD	T\$HILIM
3099	026770		GPRMD	DEMSG,DDT,D,177777,0,175,YES
(4)	026770	007052	.WORD	T\$CODE
(4)	026772	027255	.WORD	DEMSG

(4)	026774	177777				.WORD	177777
(4)	026776	000000				.WORD	T\$LOLIM
(4)	027000	000175				.WORD	T\$HILIM
3100	027002					GPRMD	MXHD,MXH,D,100,0,1,YES
(4)	027002	012052				.WORD	T\$CODE
(4)	027004	027527				.WORD	MXHD
(4)	027006	000100				.WORD	100
(4)	027010	000000				.WORD	T\$LOLIM
(4)	027012	000001				.WORD	T\$HILIM
3101	027014					GPRMD	MINHD,MNH,D,100,0,1,YES
(4)	027014	013052				.WORD	T\$CODE
(4)	027016	027536				.WORD	MINHD
(4)	027020	000100				.WORD	100
(4)	027022	000000				.WORD	T\$LOLIM
(4)	027024	000001				.WORD	T\$HILIM
3102	027026					GPRMD	MXCYL,MXC,D,77600,0,255.,YES
(4)	027026	014052				.WORD	T\$CODE
(4)	027030	027545				.WORD	MXCYL
(4)	027032	077600				.WORD	77600
(4)	027034	000000				.WORD	T\$LOLIM
(4)	027036	000377				.WORD	T\$HILIM
3103	027040					GPRMD	MINCYL,MNC,D,77600,0,255.,YES
(4)	027040	015052				.WORD	T\$CODE
(4)	027042	027555				.WORD	MINCYL
(4)	027044	077600				.WORD	77600
(4)	027046	000000				.WORD	T\$LOLIM
(4)	027050	000377				.WORD	T\$HILIM
3104	027052					GPRMD	MXSEC,MXS,D,77,0,39.,YES
(4)	027052	016052				.WORD	T\$CODE
(4)	027054	027565				.WORD	MXSEC
(4)	027056	000077				.WORD	77
(4)	027060	000000				.WORD	T\$LOLIM
(4)	027062	000047				.WORD	T\$HILIM
3105	027064					GPRMD	MINSEC,MNS,D,77,0,39.,YES
(4)	027064	017052				.WORD	T\$CODE
(4)	027066	027575				.WORD	MINSEC
(4)	027070	000077				.WORD	77
(4)	027072	000000				.WORD	T\$LOLIM
(4)	027074	000047				.WORD	T\$HILIM
3106							
3107	027076				1\$:	GPRML	AUTOMS,AUTO,1,YES
(4)	027076	033130				.WORD	T\$CODE
(4)	027100	027605				.WORD	AUTOMS
(4)	027102	000001				.WORD	1
3108							
3109	027104					ENDSF T	
(2)						.EVEN	
(3)	027104				L10025:		
3110							
3111							
3115							
3116	027104	052123	050111	046125	STIPMS:	.ASCIZ	%STIPULATE R/W XFER SIZE%
3117	027134	042523	045505	051040	SRTMSG:	.ASCIZ	/SEEK RETRY LMT/
3118	027153	043	047440	020106	CHKLMT:	.ASCIZ	/# OF ERR DUMPED/
3119	027173	122	020104	047117	RDONLY:	.ASCIZ	/RD ONLY/
3120	027203	122	047101	050040	RANPAT:	.ASCIZ	/RAN PAT/

```

3121 027213 127 044510 044103 ONLONE: .ASCIZ /WHICH ONE/
3122 027225 110 042122 042440 ERMSG: .ASCIZ /HRD ERR LMT/
3123 027241 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 027323 123 020113 051105 SEMSG: .ASCIZ /SK ERR LMT/
3127 027336 051104 042440 051122 DREMSG: .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 020105 INMSG: .ASCIZ /TIME BETW REPORTS (MIN)/
3131 027451 103 040510 043516 CHANGE: .ASCIZ %CHANGE SEEK, R/W PARAMETERS%
3132 027505 115 054101 054040 MXBUF: .ASCIZ /MAX XFER/
3133 027516 044515 020116 043130 MINBUF: .ASCIZ /MIN XFER/
3134 027527 115 054101 044040 MXHD: .ASCIZ /MAX HD/
3135 027536 044515 020116 042110 MINHD: .ASCIZ /MIN HD/
3136 027545 115 054101 041440 MXCYL: .ASCIZ /MAX CYL/
3137 027555 115 047111 041440 MINCYL: .ASCIZ /MIN CYL/
3138 027565 115 054101 051440 MXSEC: .ASCIZ /MAX SEC/
3139 027575 115 047111 051440 MINSEC: .ASCIZ /MIN SEC/
3140 027605 103 045510 042040 AUTOMS: .ASCIZ /CHK DRDY/
3141 027616 040504 040524 042040 FDCHK: .ASCIZ /DATA DMP ON DCK ERR/
3142 027642 051104 050117 042040 STLMT: .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 046440 DERPMS: .ASCIZ /DATA MISCOMPARE LIMIT/
3146 027770 047527 042122 020123 CMMSG: .ASCIZ /WORDS PER SECTOR COMPARED ON READ/

```

3147
3148 .EVEN

3149
3153
3154
3155 030032 .ENDMOD

3156
3157
3158 . =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
 (2) .EVEN
 (3) 030132 L\$LAST::

3166
3167
3168

3170			.SBTTL	DIAGNOSTIC SUPERVISOR -- LOW CORE SET UP
14041	060726	000000	.WORD	0 ;SPACE FOR USER POOL POINTER
14042	060730	000000	.WORD	0 ;SIZE
14043	060732	000000	.WORD	0 ;CHECKSUM (NOT CURRENTLY USED)
14044	060734	000000	.WORD	0 ;SIZE OF H.W. PTAB. ALLOCATION
14045		060740	END.SUPV=.	+2
14046		000200	.END	200

ABOFLA 030456 G	BIT5 = 000040 G	CLKBFR 055622	CSDODU= 000053	DCS = 000104
ABOPAS 030374 G	BIT6 = 000100 G	CLKCNT 030370 G	CSDRPT= 000024	DDT = 000016
ABO.FM 032736	BIT7 = 000200 G	CLKJUM 056226 G	CSDU = 000055	DECMG 047422
ADDCOD 011236 G	BIT8 = 000400 G	CLKRES 057230 G	C\$EDIT= 000002	DECNT 002174
AFREAD 015662	BIT9 = 001000 G	CLKSER 057364 G	C\$ERDF= 000002	DELMT 007522
AFSI 030164 G	BLD.HW 035620	CLKSON 030430 G	C\$ERHR= 000003	DEMSG 027255
AFWRCK 015674	BLOCK 053232	CLK.SE 035200	C\$ERSF= 000001	DERCNT= 000020
AGSTAT 016116	BMP = 000042	CLMT = 000064	C\$ERSO= 000004	DERMSG 003235
ALLOC 051076	BPRIOR 002164	CLNCOD 011072 G	C\$ESCA= 000010	DERPMS 027742
APT.ER 032066	BRMSG 026415	CLRDAT 010164	C\$ESEG= 000005	DERR = 040000
ARDHOR 015754	BSECT= 000112	CLRWCK 011422	C\$ESUB= 000003	DET = 000052
ASEEK 015624	BSECO 024030	CLR.MA 035454	C\$ETST= 000001	DEV.CO 030144 G
ASSEMB= 000010	BSEC1 024070	CMMSG 027770	C\$EXIT= 000032	DIAGMC= 000000
AUTO = 000066	BSEC2 024130	CMRD 007520	C\$GMAN= 000043	DIAG.T 030464 G
AUTOMS 027605	BSEC3 024170	CNT = 000010	C\$GPHR= 000042	DIFMSG 002433
AWRITE 016222	BSEC4 024230	CNTFLG 002304	C\$GPRI= 000040	DIFWD = 000066
ASAAV 034734	BSEC5 024270	CNTRLR1 002150	C\$GTIM= 000052	DLT = 010000
ASAAW 034750	BSEC6 024330	CNTRLR2 002152	C\$INIT= 000011	DLTCNT= 000026
ASAAZ 034762	BSEC7 024370	CNTYPE 026374	C\$INLP= 000020	DMPBUF 023104
ASAAZ 034770	BUF1 002266	CNVT 053676	C\$KWF= 000035	DMPDCK 003070
ASAAZ 035004	BUF2 002270	COMMAN 030202 G	C\$KWON= 000034	DNRDY 002474
ASABA 035014	BUSAD 003737	COMMTA 053512	C\$LOOP= 000100	DOWCK = 000076
BA = 000002	BVEC 002162	CONTCL 057310 G	C\$MANI= 000051	DPDVD 060074 G
BA16 = 000020	B\$AAB 037222	CONWR 021434	C\$MSG = 000023	DPHOUR= 000070
BA17 = 000040	B\$AAF 037134	CRDY = 000200	C\$PNTB= 000014	DPMIN = 000071
BBA = 000110	CALLPC= 000022	CRLBA 002375	C\$PNTF= 000017	DPMUL 057762 G
BCSADR= 000046	CALLPS= 000024	CRLCS 002347	C\$PNTS= 000016	DRBT = 000006
BCSR 002160	CALLSP= 000026	CRLDA 002407	C\$PNTX= 000015	DRBUF 025056
BDA = 000040	CALLTC= 000030	CRLF 047610	C\$POIN= 000040	DRDRV 020220
BDRSEL 002166	CAL.CL 055620	CRLMP 002421	C\$QIO = 000377	DRDY = 000001
BGN.SU= 030132	CAL.TI 055656 G	CS = 000000	C\$RDBU= 000007	DREMSG 027336
BINMSG 047406	CART 002444	CSR = 000000	C\$REFG= 000050	DRFLG = 000042
BIT0 = 000001 G	CEND 021026	CSRMSG 026401	C\$REQT= 000045	DRLMT 007556
BIT00 = 000001 G	CHANGE 027451	CSTUFF 021150	C\$RESE= 000033	DRMSG 026435
BIT01 = 000002 G	CHFLG = 000020	CURR.S 030140 G	C\$REVI= 000002	DRNM 003634
BIT02 = 000004 G	CHKFNC 015532	CURR.T 030142 G	C\$RPT = 000025	DROP 004103
BIT03 = 000010 G	CHKLMT 027153	CYLSK 002140	C\$SEFG= 000047	DROPCO 011322 G
BIT04 = 000020 G	CHKLUP 037236	C\$AAD 042500	C\$SPRI= 000041	DRPMS 027677
BIT05 = 000040 G	CHKSEC 002172	C\$AAE 042512	C\$SVEC= 000037	DRPRS 002131
BIT06 = 000100 G	CHKSTR 051440	C\$AAK 043510	C\$TPRI= 000013	DRSEL = 000106
BIT07 = 000200 G	CHKTTY 047526	C\$AAL 043654	C\$UNBU= 000031	DRST = 000013
BIT08 = 000400 G	CHK.MA 035376	C\$ABRT= 000021	C\$WTM = 000026	DRUT 002130
BIT09 = 001000 G	CHK.PC 042526	C\$ADR = 000020	C\$WTU = 000027	DRVER 002664
BIT1 = 000002 G	CHK.SW 031566	C\$AU = 000054	DA = 000004	DSE = 000400
BIT10 = 002000 G	CHRCNT 050760	C\$BRK = 000022	DALMT 007512	DSPCOD 007602 G
BIT11 = 004000 G	CH.FLA 035104	C\$BSEG= 000004	DAMSG 027351	DUNIT. 030400 G
BIT12 = 010000 G	CH.PAS 035122	C\$BSUB= 000002	DAT = 000006	DVC.FT 043460
BIT13 = 020000 G	CKBDSC 023744	C\$BUFF= 000030	DATCER= 000074	DWCNT 002312
BIT14 = 040000 G	CKDATA 020500	C\$CEFG= 000046	DCD = 000074	DWCNT1 002314
BIT15 = 100000 G	CKDERR 016514	C\$CLEA= 000012	DCDMSG 003212	D\$AAG 044364
BIT2 = 000004 G	CLEAR 023722	C\$CLP1= 000006	DCKFG = 000040	D\$AAH 044402
BIT3 = 000010 G	CLEAR. 036520	C\$CVEC= 000036	DCRC = 004000	D\$AAI 047150
BIT4 = 000020 G	CLKACC 030372 G	C\$DCLN= 000044	DCRCER= 000022	D\$AAJ 047154

DSAAM	047172	ERR13	005172	G	FMT13D	006462	GLBDAT	002122	G	INBAD	022240		
DSAAL	047210	ERR2	004312	G	FMT14	006507	GLBEQA	002122	G	INCALL	002326		
DSAAM	047220	ERR3	004376	G	FMT14A	006516	GLBERR	004304	G	ININIT	030410	G	
EF.CON=	000036	G	ERR4	004514	G	FMT14B	006530	GLBSUB	011364	G	INITCO	007664	G
EF.NEW=	000035	G	ERR6	004566	G	FMT14C	006525	GLBTXT	002330	G	INITIA	047436	
EF.PWR=	000034	G	ERR7	004626	G	FMT15	006572	GOERRX	015522	INIT.M	035522		
EF.RES=	000037	G	ERR8	004664	G	FMT17	005663	GOFIN	015526	INIT.R	030224	G	
EF.STA=	000040	G	ERR9	005050	G	FMT18	006712	G\$BIT =	000003	INMSG	027421		
EF01 =	000001	G	ERT	003766	FMT2	005705	GSTAT =	000004	INPUTA	050364			
EF02 =	000002	G	ESC.PC	042524	FMT2A	005726	GSTFNC	013346	INTEN =	000100			
EF03 =	000003	G	EV.COU	030136	G	FMT3	005750	GWDA	022576	INTERV	002236		
EF04 =	000004	G	EXHAUS	002576	FMT4	006013	G\$EXCP=	000400	INTFOR	043662			
EF05 =	000005	G	EXIT	016312	FMT5	006044	G\$HILI=	000002	INTR1	014222	G		
EF06 =	000006	G	EXIT1	016344	FMT6	006060	G\$LOLI=	000001	INTR2	014232			
EF07 =	000007	G	EXP	004061	FMT7	006106	G\$NO =	000000	INVAL.	034470			
EF08 =	000010	G	E.BA	002252	FMT7A	006141	G\$OFFS=	000400	INVINT	043520			
EF09 =	000011	G	E.CS	002250	FMT8	006162	G\$OF SI=	000376	INV.SW	031522			
EF10 =	000012	G	E.DA	002254	FMT9	006172	G\$PRMA=	000001	IN.SUF	036472			
EF11 =	000013	G	E.MP	002256	FMT9A	006230	G\$PRMD=	000002	ISDRST	021136			
EF12 =	000014	G	E.MP1	002260	FORM.T	044026	G\$PRML=	000000	ISSUE	014052			
EF13 =	000015	G	E.MP2	002262	FREE	051334	G\$RADA=	000140	I\$AU =	000041			
EF14 =	000016	G	FASCI	002306	FRMT16	006650	G\$RADB=	000000	I\$CLN =	000041			
EF15 =	000017	G	FASPNT	002310	FUNC =	000044	G\$RADD=	000040	I\$DU =	000041			
EF16 =	000020	G	FDCHK	027616	F\$AU =	000015	G\$RADF=	000200	I\$HRD =	000041			
ELT =	000002	FILINF	011460	FILL	050256	F\$BGN =	000040	G\$RADL=	000120	I\$INIT=	000041		
EMT.TR	030462	G	FILL.C	000204	G	F\$CLEA=	000007	G\$RADO=	000020	I\$MOD =	000041		
END	010466	FINDBF	010742	F\$DU =	000016	F\$END =	000041	G\$RADI=	000100	I\$MSG =	000041		
ENDBUF	026316	FINERR	016452	F\$HARD=	000004	F\$HARD=	000004	G\$XFER=	000004	I\$PWR =	000041		
ENDOFF	025050	FLAGS	030176	G	F\$HW =	000013	G\$YES =	000010	ISRPT =	000041			
ENDWR	021450	FLAG1	030200	G	F\$INIT=	000006	HCE =	040000	I\$SEG =	000041			
END.OF	036506	FLAGTA	053430	F\$JMP =	000050	HCORE	034674	H\$SFT =	000041				
END.SU=	060740	FLAG.I	035164	F\$MOD =	000000	H\$COREQ	034604	H\$SRV =	000041				
ENVIRO	030204	G	FLA.SE	053376	F\$MSG =	000011	H\$CORET	030420	G	H\$SUB =	000041		
EOP.CH	057406	G	FLG.MA	035124	F\$PWR =	000017	H\$CRC =	004000	H\$STST =	000041			
EOP.FM	032752	FMTS1	006717	F\$RPT =	000012	H\$RCER=	000024	J\$JMP =	000167				
EOP.IN	035116	FMTS1A	006775	F\$SEG =	000003	HC.ADR	030170	G	KBPTR	030242	G		
EPS	003660	FMTS1B	007013	F\$SOFT=	000005	HC.DEF	030162	G	KBUF	030244	G		
ERLMT	007506	FMTS2	007044	F\$SRV =	000010	HC.DIA	030160	G	LDFUNC	014112			
ERLMTM	003125	FMTS2A	007057	F\$SUB =	000002	H\$HOME	022500	LIMIT	007504				
ERMSG	027225	FMTS2B	007144	F\$SW =	000014	H\$DRFND	002170	LINE.F	030460	G			
ERR =	100000	FMTS3	007210	F\$TEST=	000001	HEAD =	000100	LINE1	005230				
ERRCNT=	000012	FMTS3A	007311	GARBAG	050762	HERTZ.	034544	LINE2	005410				
ERREX	016350	FMTS4	007344	GDDAT	002232	HINUM	002134	LINE3	005474				
ERRFOR	043732	FMTS5	007430	GETCHR	047466	HNF =	010000	LIST	017470				
ERRHAN	042532	FMT1	005642	GETCMN	053052	HNFERR=	000032	LOAD.F	035120				
ERRVEC	002316	FMT1A	005676	GETDST	021122	HOLDSP=	000020	LOGMSG	047430				
ERR.HR	043470	FMT10	006302	GETPAR	044544	HOUR	002246	LONUM	002136				
ERR.NU	030134	G	FMT10A	006336	GETSWI	052046	HPTCOD	007466	G	LPBFR	030240	G	
ERR.SF	043474	FMT10B	006407	GETSYS	022356	HRDPRM	026320	G	LPCNTR	030236	G		
ERR1	004304	G	FMT11	006415	GET.TW	051616	HWSEC	003524	LPS	003645			
ERR1FO	044016	FMT12	006435	GHDR	017156	HW.ADR	030166	G	LPT.AD	034562			
ERR10	005114	G	FMT13	006445	GLBDAT	002122	H\$AAB	054224	LPT.RE	034556			
ERR12	005164	G			GLBEQA	002122	ILLEG	003574	LSI.RE	034552			

LSTDA = 000064	L\$UNIT 002012 G	MODR 057674 G	NUM.LA 044220	PRI07 = 000340 G
LSTDR1 002154	L.CLK. 034530	MP = 000006	NUM.NO 030174 G	PRNTST 050626
LSTDR2 002156	L10000 004310	MPT 003562	NUM.UN 030602	PRO.CM 035076
LSTHDR= 000050	L10001 004374	MRDR 003015	NUNITS 037210	PRPOS = 000122
LSTTIM 002240	L10002 004512	MRLCS 002337	NWRTS 002541	PTAB.S 030416 G
LUP 055524	L10003 004564	MRT 003755	NXM = 020000	PUTCHR 047442
LUP.AD 042530	L10004 004624	MSFER 002645	NXMCNT= 000034	PWR.FLG 002276
L\$APT 002036 G	L10005 004662	MSG.AD 030156 G	NXTFOR 053670	PWR.FA 060566 G
L\$AU 011236 G	L10006 005046	MSG.TY 030132 G	OCTMSG 047414	PWR.FL 030222 G
L\$AUT 002074 G	L10007 005112	MSKER 002634	ODRDRV 020214	PWR.MS 060714
L\$CCP 002106 G	L10010 005162	MST 004004	ONLONE 027213	PWR.SA 060710
L\$CLEA 011072 G	L10011 005170	MSTART 004165	OPCALL 002324	PWR.UP 060712
L\$CO 002032 G	L10012 005226	MST1 004021	OPFLG = 000050	P.CLK. 034536
L\$DEPO 002011 G	L10013 007502	MSWRPK 004205	OPI = 002000	RAN = 000056
L\$DESC 002102 G	L10014 007602	MTCR 004224	OPICNT= 000030	RAND 021214
L\$DEVP 002064 G	L10015 007662	MTDCRC 004134	OVER 003256	RANPAT 027203
L\$DISP 007604 G	L10016 011070	MTDLT 004141	OSAPTS= 000001	RCD 004072
L\$DR 002112 G	L10017 011234	MTDRV 004160	OSAU = 000001	RCNT 017450
L\$DRCT 002070 G	L10020 011320	MTEST 012354	OSBGNR= 000001	RDBDSC 017506
L\$DRS 002072 G	L10021 011362	MTGS 004234	OSBGNS= 000001	RDDFNC 014002
L\$DRST 002112 G	L10022 025052	MTHCRC 004126	OSDU = 000001	RDHDR = 000010
L\$DTP 002040 G	L10023 016450	MTHNF 004121	OSGNSW= 000001	RDHFNC 013712
L\$DU 011322 G	L10024 026374	MTNXM 004153	OSPOIN= 000001	RDONLY 027173
L\$DUT 002076 G	L10025 027104	MTOPI 004146	OSPWR = 000001	RDT = 000014
L\$DVTY 002114 G	MAIN 012614	MTRD 004274	PARSES 053124	READ = 000014
L\$EF 002056 G	MAJ.IN 030214 G	MTRH 004254	PAR.LA 047112	READ.P 055626 G
L\$EFLG 002034 G	MAJ.LO 055624	MTSK 004244	PASS.C 030146 G	RECNT 002122
L\$EXP1 002042 G	MAJ.US 030216 G	MTWR 004264	PAT = 000060	REGBAC 060316 G
L\$EXP2 002044 G	MAN.TI 001244	MUL 057630 G	PATLST 024430	REGSAV 060302 G
L\$EXP3 002046 G	MAP16 060332 G	MVCER 002735	PAT0 024450	REPORT 011670
L\$HARD 026322 G	MASK.B 037234	MVEC 003610	PAT1 024510	REQ 003313
L\$HPCP 002016 G	MASK.W 037232	MXB = 000022	PAT2 024550	REQN.P 030206 G
L\$HPTP 002022 G	MAXWC 002272	MXBUF 027505	PAT3 024610	REQN.T 035100
L\$HW 007470 G	MBDMSC 003473	MXC - 000030	PAT4 024650	RESTAR 010544
L\$ICP 002104 G	MDCER 003040	MXCYL 027545	PAT5 024710	RETRY = 000036
L\$INIT 007654 G	MDERS 002673	MXH = 000024	PAT6 024750	RE.SET 031670
L\$LADP 002026 G	MDHEDR 002000 G	MXHD 027527	PAT7 025010	RLT = 000000
L\$LAST 030132 G	MDSER 002721	MXS = 000034	PEROTH 012624	RNTEMP 002234
L\$MREV 002050 G	MEM.SI 034572	MXSEC 027565	POWER 011070	ROF = 000054
L\$NAME 002000 G	MFUNC 002361	NEWPRI 057354 G	PRFLGS= 000056	RPS 003673
L\$REPP 002066 G	MHDR 003055	NEXTAR 053614	PRGER 002535	RPTCOD 007606 G
L\$REV 002010 G	MINBUF 027516	NOCRDY 002464	PRINTC 050736	RSEEK = 000114
L\$RPT 007606 G	MINCYL 027555	NODRIV 003622	PRINTF 054244	RSTACK 057556 G
L\$SOFT 026446 G	MINHD 027536	NOLOAD 003376	PRIOR = 000004	RTMSG 027311
L\$SPC 002062 G	MINSEC 027575	NOPWR 003706	PRIOR1 002226	RTYPE = 000052
L\$SPCP 002020 G	MINUTE 002244	NORDY 002507	PRIOR2 002230	RT1 004036
L\$SPTP 002024 G	MIN.IN 030210 G	NGREV 003434	PRI00 = 000000 G	RWCNT 002124
L\$STA 002030 G	MIN.US 030212 G	NO.CLK 034520	PRI01 = 000040 G	RXFR1 = 000002
L\$SW 007504 G	MK - 000001	NO.FLA 053410	PRI02 = 000100 G	RXFR2 = 000004
L\$TIML 002014 G	MNB - 000044	NO.LPT 050726	PRI03 = 000140 G	RXFR3 = 000060
L\$TIMU 002054 G	MNC - 000032	NO.PTA 034724	PRI04 = 000200 G	SAVE 014240
L\$TIM1 002052 G	MNH - 000026	NR - 000000	PRI05 = 000240 G	SAVEDO= 032066
L\$TSTI 002100 G	MNS 000036	NUMBIN 044052	PRI06 - 000300 G	SEARCH 051564

SECMSK 002142	STIPMS 027104	TIM.CO 030220 G	TSSMSG= 010012	WGE = 002000
SECOND 002242	STLMT 027642	TIM.OP 044024	TSSRPT= 010015	WGEST 002762
SEEK = 000006	STRCHR 050316	TOO.MA 047360	TSSOF= 010025	WHY 002126
SEGSTA 030432 G	STRT.T 035102	TRACK 003110	TSSSRV= 010023	WIDTH 044420
SEL = 000046	STWRT 021472	TRERR = 000072	TSSSW = 010014	WL = 020000
SELMT 007510	ST.SET 031734	TRPFLG 002300	TSTES= 010022	WRBUF 017230
SEMSG 027323	ST1 002320	TRPHAN 011662	T.AUT 007572	WRCHK = 000002
SERLMT 003147	ST2 002322	TST.AB 037346	T.CLT 007570	WRINIT 002144
SERNM1= 000100	SUNIT. 035106	TST.TO 031550	T.DCD 007600	WRIPG = 000120 G
SERNM2= 000102	SUPERV 032770	TYINT 007516	T.DCK 007544	WRITE = 000012
SET = 000004	SUPFLA 030376 G	TYPEC 047754	T.DRP 007546	WRPACK 021312
SETWCK 011364	SUPV.T 030550 G	TYPEPC 043650	T.MNB 007550	WRPOS 002146
SET.MA 035310	SUP.PR 031506	TYPFLA 053272	T.MNC 007536	WRTFNC 013724
SEXHAU 003326	SVCGBL= 000000	TYPLIN 047652	T.MNH 007532	WTRDY 021052
SFEMSG 003170	SVC.HAN 037424	TYPNUM 047234	T.MNS 007542	WXFR1 = 000006
SFLMT 007552	SVCINS= 000000	TYPSTR 047672	T.MXB 007526	WXFR2 = 000010
SFTCNT= 000014	SVCSUB= 177777	TYP.ER 043500	T.MXC 007534	WXFR3 = 000062
SFTMSG 027241	SVCTAG= 000000	TYT = 000012	T.MXH 007530	XCHFLG 007524
SFTPRM 026444 G	SVCTST= 177777	TY.UNI 036512	T.MXS 007540	XEQDIA 057442 G
SHIFT 060414 G	SWCHAN 034716	T\$ARGC= 000001	T.PAT 007564	XEQSUB 057430 G
SIGN = 000004	SWITCH 053570	T\$CODE= 033130	T.RAN 007562	XEQ.CL 037152
SKCNT = 000000	SWSEC 003544	T\$ERCO= 000063	T.ROF 007560	XEQ.CM 034462
SKCNT1= 000054	SW.ADR 030172 G	T\$ERRN= 000307	T.SLT 007566	XEQ.IN 036634
SKDON = 000001	SW.PTA 034702	T\$EXCP= 000000	T.STA 007554	XEQ.LA 032724
SKECNT= 000016	SYSCLK 002264	T\$HILI= 000047	T.STIP 007574	XEQ.OP 036726
SKFNC 013366	SYSMSK 002132	T\$LOLI= 000000	T.WCK 007576	XEQ.PR 032126
SKHS = 000020	SYS.FT 043450	T\$LSYM= 010000	T1 012354 G	XEQ.TE 036772
SKLMT 007514	S\$LSYM= 010000	T\$NEST= 177777	UDERR 002612	XEXIT 015620
SKMSG 027401	TAGX 012766	T\$NSKO= 000000	UNIT.D 030150 G	XTIME 056314 G
SKRETR= 016034	TEMPO 002176	T\$NSK1= 000005	UNI.MA 035026	XTIMEN 057140
SKT = 000010	TEMP1 002200	T\$SAVL= 177777	UNLOAD 003352	XTIMST 056336
SKTO = 010000	TEMP2 002202	T\$SEGL= 177777	USER.P 030412 G	XXDP.D 034502
SMSG 002564	TEMP3 002204	T\$SUBN= 000000	USER.T 030414 G	X\$ALWA= 000000
SOFTCS= 000116	TEMP4 002206	T\$TAGL= 177777	UUT 002274	X\$FALS= 000040
SOPLMT 003414	TEMP5 002210	T\$TAGN= 010026	VALID. 030652	X\$OFFS= 000400
SPE = 004000	TEMP6 002212	T\$TEMP= 000000	VAL.LA 031472	X\$TRUE= 000020
SPEC.U 035024	TEMP7 002214	T\$TEST= 000001	VAL.SW 035136	\$BREG 035176
SPTCOD 007502 G	TEMP8 002216	T\$TSTM= 177777	VC = 001000	\$ENDAD 057414 G
SPV.SE 000400	TEMP9 002220	T\$TSTS= 000001	VECMG 026426	\$SAV2 060460 G
SRLT = 000062	TERMI 055614	T\$SAU = 010020	VECT = 000002	\$SAV3 060474 G
SRTMSG 027134	TERMLI 053416	T\$SCLE= 010017	VECT1 002222	\$SAV4 060512 G
STARTC 057304 G	TERMTA 047400	T\$SDU = 010021	VECT2 002224	\$SAV5 060532 G
STDMP 023620	TEST.M 035036	T\$SHAR= 010024	WCK = 000072	. = 060736
STFLG 002302	TIME 002330	T\$SHW = 010013	WCKMSG 027733	
STIP = 000070	TIMFLG 030366 G	T\$INI- 010016	WDE = 100000	

. ABS. 060736 000

ERRORS DETECTED: 0

DSKZ:CZRLEB,DSKZ:CZRLEB=CZRLEB/ML,CZRLEB.P11,CZRLEB.SUP
RUN-TIME: 40 38 1 SECONDS
RUN-TIME RATIO: 120/79=1.5
CORE USED: 16K (31 PAGES)

