

# RL11,RLV11

RL01,02 BSFT  
CZRLMA0

AH-F134A-MC

COPYRIGHT © 1979  
FICHE 1 OF 1

MAY 1979

**digital**  
MADE IN USA

IDENTIFICATION

SEQ 0001

PRODUCT CODE: AC-F135A-MC  
PRODUCT NAME: (ZRLMAD RLO1/02 BAD SECTOR FILE TOOL  
DATE CREATED: 5-JAN-79  
MAINTAINER: DIAGNOSTIC ENGINEERING  
AUTHOR: C. CAMPBELL

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

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

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

COPYRIGHT (C) 1979, DIGITAL EQUIPMENT CORPORATION

TABLE OF CONTENTS

1.0	GENERAL INFORMATION
1.1	PROGRAM ABSTRACT
1.1.1	STRUCTURE OF PROGRAM
1.1.2	DIAGNOSTIC INFORMATION
1.2	SYSTEM REQUIREMENTS
1.2.1	HARDWARE REQUIREMENTS
1.2.2	SOFTWARE REQUIREMENTS
1.3	RELATED DOCUMENTS AND STANDARDS
1.4	DIAGNOSTIC HIERARCHY PREREQUISITES
1.5	ASSUMPTIONS
2.0	OPERATING INSTRUCTIONS
2.1	HOW TO RUN THIS DIAGNOSTIC
2.1.1	THE SIX STEPS OF EXECUTION
2.1.2	SAMPLE RUN-THROUGH
2.2	DETAILS OF COMMANDS AND SYNTAX
2.2.1	TABLE OF COMMAND VALIDITY
2.2.2	COMMAND SYNTAX
2.3	HARDWARE PARAMETERS
2.4	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 IS TO BE RUN STANDALONE UNDER XXDP. 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  
-----

THERE IS NO SPECIFIC RUN TIME ASSOCIATED WITH THIS UTILITY PROGRAM. HOWEVER, TO WRITE THE WORST CASE DATA PATTERN ON THE DISK AND THEN VERIFY THE DATA BY READING SHOULD TAKE LESS THAN 1 MINUTE FOR AN RL01 AND LESS THAN 2 MINUTES FOR AN RL02.

1.2 SYSTEM REQUIREMENTS  
-----1.2.1 HARDWARE REQUIREMENTS  
-----

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

- KW11P, KW11L (OPTIONAL)
- LINE PRINTER (OPTIONAL)

1.2.2        SOFTWARE REQUIREMENTS

(ZRLMAO RLO1/02 BAD SECTOR FILE TOOL

1.3         RELATED DOCUMENTS AND STANDARDS

RLO1 USERS MANUAL (EK-RLO1-PG-PRE)  
XXDP USERS MANUAL

1.4         DIAGNOSTIC HIERARCHY PREREQUISITES

THE RLO1/02 SUBSYSTEM SHOULD HAVE SUCCESSFULLY RUN THE FOLLOWING PROGRAMS:

CVRLAAO	RLV11 RLO1/02 DISKLESS TEST (RLV11 ONLY)
CZRLGAO	RL11/RLV11 RLO1/02 CONTROLLER TEST (PART 1)
CZRLHAO	RL11/RLV11 RLO1/02 CONTROLLER TEST (PART 2)
CZRLIAO	RLO1/02 DRIVE TEST (PART 1)
CZRLJAO	RLO1/02 DRIVE TEST (PART 2)
CZRLKAO	RL11/RLV11 RLO1/02 PERFORMANCE EXERCISER

1.5         ASSUMPTIONS

THE HARDWARE OTHER THAN THE RLO1/02 SUBSYSTEM IS ASSUMED TO WORK PROPERLY. FALSE ERRORS MAY BE REPORTED IF THE PROCESSOR, ETC., DO NOT FUNCTION PROPERLY.

THIS UTILITY WILL COMFORM TO ALL INTERFACE SPECIFICATIONS FOR THE DIAGNOSTIC SUPERVISOR.

THE INTERNAL FORMAT OF THE BAD SECTOR FILE WILL BE THE SAME AS DESCRIBED BY THE DEC STD-144 DOCUMENT FOR REPORTING AND UPDATING THE INFORMATION CONTAINED IN THAT FILE.

NO SUPPORT WILL BE GIVEN FOR THE RLBA/RLO1 DISK CONTROLLER ON ANY PDP-8 SYSTEM...THIS IS A PDP-11 UTILITY ONLY!

2.0         OPERATING INSTRUCTIONS

2.1        HOW TO RUN THIS DIAGNOSTIC2.1.1     THE SIX STEPS OF EXECUTION

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

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

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

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

QUESTION	MEANING
-----	-----
L-CLK (L) N ?	IS THERE AN L-CLOCK?
P-CLK (L) N ?	IS THERE A 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 UTILITY WILL ISSUE THE PROMPT "DS-B>". FROM THIS POINT UNTIL THE TIME WHEN YOU RESTART XXDP, YOU WILL BE TALKING TO THE UTILITY, 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). HOWEVER, IN ORDER TO USE THE PROGRAM, ALL YOU NEED TO SAY IS SOMETHING LIKE THIS:

STA

THINGS TO NOTE HERE: ONLY THE FIRST THREE CHARACTERS OF THIS OR ANY COMMAND AT THE "DS-B>" LEVEL NEED TO BE TYPED. THE "FLAGS" SWITCH MAY SPECIFY ANY OF A NUMBER OF FLAGS, BUT THE MAIN USEFUL ONES ARE:

HOE	HALT ON ERROR
IER	INHIBIT ERROR PRINTOUT

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

WHEN YOU HAVE TYPED IN A "START" COMMAND, THE UTILITY WILL COME BACK WITH THE QUESTION "# UNITS?" TO WHICH YOU SHOULD RESPOND BY TYPING IN THE NUMBER OF DEVICES YOU WISH TO TEST (THE UTILITY USES ONLY 1 DRIVE).

\*\*\*\*\*  
\* STEP 4 \*  
\*\*\*\*\*

WHEN YOU HAVE TYPED IN THE NUMBER OF UNITS TO BE TESTED, THE UTILITY 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 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, AND THE ANSWERS WILL BE PUT INTO THE SOFTWARE P-TABLE IN THE PROGRAM.

\*\*\*\*\*  
\* STEP 6 \*  
\*\*\*\*\*

AFTER YOU HAVE ANSWERED THE SOFTWARE QUESTIONS, THE UTILITY WILL BEGIN TO EXECUTE. 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.

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 UTILITY WILL RETURN TO COMMAND MODE.

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

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



2.1.2 SAMPLE RUN-THROUGH - COMMON TO DIAGNOSTIC SUPERVISOR  
-----

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

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

1. ISSUE ANOTHER "START" COMMAND (THUS GOING THRU ALL OF STEPS 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

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:IFR:LOE:HOE=0

THIS WILL DO THE FOLLOWING:

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

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

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

1. START
2. RESTART
3. CONTINUE

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

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

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

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

1. START
2. RESTART

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

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

	BY WHOM ENTERED: -----
.R CZRLMA	O
CZRLM	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	D,O
# UNITS (D) ? 2	D,O
UNIT 1	D
BUS ADDRESS (O) 174400 ?	D,O
DRIVE (O) 0 ?	D,O
UNIT 2	D
BUS ADDRESS (O) 174400 ?	D,O
DRIVE (O) 0 ? 1	D,O
CHANGE SW (L) ? N	D,O
CZRLM HRD ERR 00004 TST 003 SUB 002 PC:004130 ERR HLT	
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
^C	

DS-B>RESTART/PASS:1 D.O

CHANGE SW (L) ? N D.O

-----  
-----  
-----  
-----

2.2 DETAILS OF COMMANDS AND SYNTAX

2.2.1 TABLE OF COMMAND VALIDITY

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

<u>HOW ENTERED</u>	<u>LEGAL COMMANDS</u>
1. OPERATOR ENTERED 'RUN DIAG'	START PRINT DISPLAY FLAGS ZFLAGS
2. DIAGNOSTIC HAS FINISHED ALL ITS REQUESTED PASSES	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

2.2.2      COMMAND SYNTAX - STANDARD TO THE DIAGNOSTIC SUPERVISOR  
-----

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

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

THE SWITCH ARGUMENTS ARE AS FOLLOWS:

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

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

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

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

IER INHIBIT ERROR REPORTING

IBE INHIBIT BASIC ERROR REPORTS

IXE INHIBIT EXTENDED ERROR REPORTS

PRI DIRECT ALL MESSAGES TO A LINE PRINTER

PNT PRINT NUMBER OF TEST BEING EXECUTED

BOE BELL ON ERROR

UAM RUN IN UNATTENDED MODE, BYPASSING MANUAL INTERVENTION TESTS  
ISR INHIBIT STATISTICAL REPORTS  
IDU INHIBIT DROPPING OF UNITS BY DIAGNOSTIC

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

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

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

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

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

THE SWITCH ARGUMENTS ARE AS IN THE START COMMAND EXCEPT:

1. "UNIT-LIST" IS A SEQUENCE OF LOGICAL UNIT NUMBERS RANGING  
FROM 1 THRU N (N = NUMBER OF UNITS BEING TESTED) SPECIFY-  
ING 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 SPECI-  
FIED MUST NOT HAVE BEEN DROPPED BY THE OPERATOR DROP COM-  
MAND. THE UNIT-LIST DEFAULTS TO "ALL THAT HAVE NOT BEEN  
DROPPED BY OPERATOR COMMAND". THE EFFECT OF THE UNIT-LIST  
LASTS UNTIL THE NEXT START (WHERE IT IS AUTOMATICALLY  
RESET TO "ALL") OR THE NEXT RESTART.
2. ALL UNSPECIFIED FLAG SETTINGS ARE UNCHANGED.

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

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

THE SWITCH ARGUMENTS ARE AS IN THE START COMMAND EXCEPT:

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

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

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

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

1. UNSPECIFIED FLAG SETTINGS ARE UNCHANGED

\*\*\*\*\*  
 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.

\*\*\*\*\*  
PRINT)  
\*\*\*\*\*

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

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

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

\*\*\*\*\*  
FLA(GS)  
\*\*\*\*\*

THE CURRENT SETTINGS OF ALL FLAGS ARE PRINTED.

\*\*\*\*\*  
ZFL(AGS)  
\*\*\*\*\*

ALL FLAGS ARE CLEARED.



2.3        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.

BUS ADDRESS (D) 174400?

ANSWER WITH THE BUS ADDRESS OF THE CONTROLLER.

DRIVE (O) 0?

ANSWER WITH THE DRIVE(S) CONNECTED TO THE CONTROLLER

2.4        SOFTWARE PARAMETERS

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

"CHANGE S.W. ?"

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

"SAWTOOTH WRITE CYCLE? (L) Y ?"

IF "Y" THEN THE WRITE PACK COMMAND (#5) WILL CAUSE THE PACK TO BE WRITTEN IN A FORWARD AND REVERSE DIRECTION UTILIZING A 'SAWTOOTH' SEEK PATTERN. THIS WILL ATTEMPT TO DETECT POSITIONER PROBLEMS. IF "N" FOR NO, THEN THE PACK WILL BE WRITTEN FORWARD AND REVERSE USING AN INCREMENTAL SEEK - THIS IS THE FASTEST BUT NOT NECESSARILY THE MOST DIFFICULT.

"WRITE CYCLES PER TRACK? (D) 2 ?"

THE DEFAULT NUMBER OF TIMES TO WRITE A SELECTED TRACK DURING THE WRITE PACK COMMAND (#5). IF A HIGHER NUMBER IS SELECTED, THEN IT MAY BE POSSIBLE TO DETECT A TRACK DRIFTING POSITIONER PROBLEM.

3.0        ERROR INFORMATION

ERROR INFORMATION IS COMPLETE IN GIVING ALL INFORMATION NECESSARY.

THE 'RLCS' AND DRIVE STATUS REGISTER ARE GIVEN AS WELL AS CYLINDER, TRACK, SECTOR AND DRIVE INVOLVED IN ERROR.

ANY DETECTED HARDWARE FAILURES WILL RESULT IN AN APPROPRIATE ERROR MESSAGE (THE PROPER DISK SUBSYSTEM DIAGNOSTIC(S) SHOULD BE PERFORMED).

UNEXPECTED "TRAPS" WILL RESULT IN A PROPER ERROR MESSAGE AND WILL CAUSE THE UTILITY TO RESTART.

A POWER FAILURE WILL CAUSE THE PROGRAM TO RESTART.

SOFTWARE DETECTED FAILURES - SUCH AS THE DETECTION OF A MISSING BAD SECTOR FILE OR A PARTIALLY DESTROYED BAD SECTOR FILE - WILL CAUSE THE UTILITY TO RESTART AFTER THE FAILURE IS DIAGNOSED AND A DIAGNOSTIC ERROR MESSAGE PRINTED.

### 3.1 ERROR REPORTING

ALL ERROR INFORMATION IS PRINTED ON THE CONSOLE DEVICE. ERROR REPORTS ARE AIMED AT BEING SELF EXPLANATORY.

REGISTER DESCRIPTIONS CAN BE FOUND IN SECTION 5.0.

#### ERROR MESSAGES:

"DRIVE IS NOT READY FOR USE"

THIS MESSAGE IS PRINTED WHEN THE SELECTED DRIVE IS NOT ABLE TO PERFORM A GIVEN TASK. THE DRIVE WILL BE ELIMINATED FROM THE TEST TABLE.

"SEEK ERR"

AN ERROR IS DETECTED AFTER A SEEK COMMAND WAS ISSUED.

"DR ERR WILL NOT RESET"

ISSUING A DRIVE RESET WOULD NOT CLEAR THE DRIVE ERROR CONDITION.

"DR WOULD NOT LOAD"

ON PROGRAM START, THE SELECTED DRIVE DID NOT HAVE 'HEADS OVER PACK' BIT SET.

"PACK IS WRITE LOCKED"

THIS IS JUST A WARNING MESSAGE. IF A WRITE COMMAND IS ISSUED, THEN THIS WOULD INDICATE AN ERROR.

"TIMEOUT - DR NOT RDY"

THE DRIVE WAS EXPECTED TO BE 'READY' AFTER A COMMAND WAS ISSUED AND IT NEVER FINISHED THE FUNCTION.

"NO DRIVES"

THE PROGRAM TRYED TO SELECT A DRIVE FOR USE BUT FAILED TO FIND ONE.

"UPDATING DENIED - INVALID PASSWORD"

NORMALLY, THIS PROGRAM WILL NEVER PRINT THIS MESSAGE! PROGRAM PASSWORD CHECKING IS NORMALLY INHIBITED. A USER MAY INVOKE THE PASSWORD CHECK IF THE WORD 'PASWD' AT ADDRESS 2274 IS CHANGED TO A NON-ZERO NUMBER...THIS 'NUMBER' THEN BECOMES THE PROGRAM PASSWORD AND MUST BE USED TO ENABLE ANY WRITING ON THE SELECTED PACK.

"CAN'T UPDATE THE BAD SECTOR FILE ON PACK"

THIS IS AN INDICATION THAT THE PACK IS WRITE PROTECTED OR THE FUNCTION 'WRITE' CANNOT BE COMPLETED.

"BAD READ OF BAD SECTOR FILE (20 SECTORS)"

AN ERROR WAS DETECTED WHILE TRYING TO READ 20 SECTORS OF DATA FROM THE 'FACTORY' OR 'FIELD' AREAS IN THE BAD SECTOR FILE.

"MORE THAN 25. BAD SPOTS FOUND ON THIS PACK"

THIS MESSAGE WARNS THE USER THAT THE SELECTED PACK ALREADY HAS MORE THAN 25. ENTRIES IN THE BAD SECTOR FILE. THE PACK SPECIFICATION ALLOWS ONLY 16. BAD SPOTS ON THE PACK BEFORE THE PACK IS CLASSIFIED AS 'BAD'.

"SOFT ERR ENCOUNTERED"

DURING A WRITE OR READ DATA FUNCTION, AN ERROR HAS BEEN DETECTED. THE ERROR WILL ALSO REPORT THE STARTING SECTOR NUMBER OF THE DATA TRANSFER AND THE CONTENTS OF THE DRIVE 'RLCS' REGISTER AND THE DRIVE STATUS.

"HARD ERROR"

THIS MESSAGE IS TO INFORM THE USER THAT THE 'SOFT' COULD NOT BE RECOVERED. THE STARTING SECTOR NUMBER OF THE DATA TRANSFER WILL BE RECORDED FOR LATER USE IN UPDATING THE BAD SECTOR FILE.

"RLOT MAX CYL = 255."

THE USER CANNOT ADD TO OR DELETE FROM THE BAD SECTOR FILE ANY INVALID DISK ADDRESS.

"ENTRY ALREADY EXISTS IN THE BAD SECTOR FILE"

A REDUNDANT ENTRY CANNOT BE ENTERED INTO THE BAD SECTOR FILE.

"NO SUCH ENTRY IN THE 'FIELD' FILE"

IF AN ENTRY DOES NOT EXIST IN THE 'FIELD' AREA OF THE BAD SECTOR FILE, THEN IT CANNOT BE REMOVED FROM THE FILE. THIS PROTECTS ENTRIES IN THE 'FACTORY' FILE FROM BEING DELETED.

"NO FACTORY FILE FOUND"

THE PROGRAM TRIED TO READ THE FIRST 20 SECTORS OF THE LAST TRACK TO IDENTIFY THE 'FACTORY' BAD SECTOR FILE...AND FAILED TO MAKE THAT IDENTIFICATION. EITHER THE 'FACTORY' FILE WAS DESTROYED OR THE DATA ON THIS TRACK DOES NOT COMFORM TO THE 'DEC STD-144' SPEC.

"NO FIELD FILE FOUND"

SAME AS FOR THE 'FACTORY' FILE MESSAGE ABOVE.

3.2 ERROR HALTS

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

4.0 PERFORMANCE AND PROGRESS REPORTS

4.1 PERFORMANCE REPORTS

THIS PROGRAM WILL NOT GIVE ANY PERFORMANCE REPORTS.

4.2        PROGRESS REPORTS

THIS PROGRAM WILL NOT GIVE ANY PROGRESS REPORTS.

5.0        DEVICE INFORMATION TABLES

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

RLCS - CONTROL AND STATUS REGISTER (XXXXX0)

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

BIT 0 - DRIVE READY

RLBA - BUS ADDRESS REGISTER (XXXXY2)

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

RLDA - DISK ADDRESS REGISTER (XXXXX4)

FOR READ/WRITE FUNCTIONS

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

FOR SEEK FUNCTION  
-----

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

FOR GET STATUS FUNCTION  
-----

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

RLMP - MULTIPURPOSE REGISTER  
-----

FOR READ/WRITE FUNCTION  
-----

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

FOR READ HEADER FUNCTION  
-----

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

FOR GET STATUS FUNCTION  
-----

HAS DRIVE STATUS

BIT 15 - WRITE DATA ERROR  
BIT 14 - CURRENT HEAD ERROR (CHE)  
BIT 13 - WRITE LOCK STATUS (WL)  
BIT 12 - SEEK TIME OUT (SKTO)  
BIT 11 - SPIN ERROR (SPE)  
BIT 10 - WRITE GATE ERROR (WGE)  
BIT 9 - VOLUME CHECK (VC)  
BIT 8 - DRIVE SELECT ERROR (DSE)  
BIT 7 - DRIVE TYPE IS RL02 IF SET

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

6.0      UTILITY - SUMMARY OF COMMANDS

THIS UTILITY HAS THE FOLLOWING COMMANDS:

<u>INPUT</u>	<u>ACTION</u>
1	REPORT THE CONTENTS OF THE BAD SECTOR FILE BOTH THE "FACTORY" AREA AND THE "FIELD" AREA
2	ADD A SECTOR TO THE BAD SECTOR FILE IN THE "FIELD" AREA OF THAT FILE
3	REMOVE A SECTOR FROM THE BAD SECTOR FILE - (ONLY IN THE "FIELD" AREA)
4	READ THE PACK TO FIND BAD SPOTS (READ ONLY)
5	WRITE THE PACK WITH THE WORST CASE DATA PATTERN THEN ISSUE THE 'VERIFY' COMMAND TO FIND BAD SPOTS.
6	ATTEMPT TO GENERATE THE BAD SECTOR FILE IF IT HAS BEEN DESTROYED ACCIDENTLY. ONLY THE "DUMMY" ENTRY WILL BE MADE FOR THE "FACTORY" AREA!
7	PRINT THE LIST OF AVAILABLE COMMANDS

1-	23	BIT AND OFFSET DEFINITIONS
1-	89	GLOBAL DATA AND CONSTANTS
1-	179	GLOBAL MESSAGES
1-	280	ERROR MESSAGES
1-	364	STATISTIC CODE
1-	371	INITIALIZATION CODE
2-	1	GLOBAL SUBROUTINES
4-	1	PROGRAM MAIN LOOP
23-	1	ROUTINE TO LOAD FUNCTION
23-	23	INTERRUPT SERVICE - VERY SHORT!
24-	1	BAD SECTOR FILE ROUTINE
25-	1	ROUTINE TO WAIT FOR CONTROLLER READY
25-	25	GET STATUS/DRIVE RESET ROUTINE
26-	1	ROUTINE TO WRITE PACKS INITIALLY
28-	1	ROUTINE FOR SYSTEM CLOCK
29-	1	HEADS HOME ROUTINE
30-	1	SEEK ROUTINE
30-	51	ROUTINE TO CHECK FOR BAD SECTOR
31-	1	DIAGNOSTIC SUPERVISOR -- LOW CORE SET UP



1		.TITLE CZRLMAO RL01/2 BSFT
2		.ENABLE AMA
3	000000	.ENABLE ABS
4		.NLIST ME,CND,MD
5	002000	.=2000
6		.MCALL SVC
7		
8	002000	SVC
9	000000	SVCINS=0
10	000000	SVCTAG=0
11		
12	002000	POINTER ALL
13		
14	002000	BGNMOD MDHEDR
15	002000	HEADER CZRLM,A,0,0,0,0,RL01,,TITLEX
	002000	.ASCII /C/
	002001	.ASCII /Z/
	002002	.ASCII /R/
	002003	.ASCII /L/
	002004	.ASCII /M/
	002005	.BYTE 0
	002006	.BYTE 0
	002007	.BYTE 0
	002010	.ASCII /A/
	002011	.ASCII /O/
	002012	.WORD 0
	002014	.WORD 0
	002016	.WORD L\$HARD
	002020	.WORD L\$SOFT
	002022	.WORD L\$HW
	002024	.WORD L\$SW
	002026	.WORD L\$LAST
	002030	.WORD 0
	002032	.WORD 0
	002034	.WORD 0
	002036	.WORD 0
	002040	.WORD L\$DISPATCH
	002042	.WORD 0
	002044	.WORD 0
	002046	.WORD 0
	002050	.BYTE C\$REVISION
	002051	.BYTE C\$EDIT
	002052	.WORD 0
	002054	.WORD 0
	002056	.WORD 0
	002060	.WORD 0
	002062	.WORD 0
	002064	.WORD L\$DVTYP
	002066	.WORD L\$RPT
	002070	.WORD L\$DR
	002072	.WORD L\$DRST
	002074	.WORD L\$AU
	002076	.WORD L\$DU
	002100	.WORD 14
	002102	.WORD TITLEX
	002104	.WORD L\$INIT
	002106	.WORD L\$CLEAN

```

16 002110                                ENDMOD
17
18 002110      122      114      060  TITLEX: .ASCIZ  ?RL01/02 BAD SEC FILE TOOL?
   002113      061      057      060
   002116      062      040      102
   002121      101      104      040
   002124      123      105      103
   002127      040      105      111
   002132      114      105      040
   002135      124      117      117
   002140      114      000

19
20 002142                                .EVEN
   002142 000000                       DEVREG
                                       .WORD 0
                                       .BLKW
   002146                                DEVTYP <RL01,RL02>
   002146      122      114      060  .ASCIZ /RL01,RL02/
   002151      061      054      122
   002154      114      060      062
   002157      000

                                       .EVEN
22
23      .SBTTL BIT AND OFFSET DEFINITIONS
24
25      ;DEFINITIONS
26
27 002160      BGNMOD  GLBEQAT
28
29 002160      EQUALS
30
31      000000      CS=0                ;CONTROL AND STATUS OFFSET
32      000002      BA=2                ;BUSADDRESS OFFSET
33      000004      DA=4                ;DISK ADDRESS OFFSET
34      000006      MP=6                ;MULTI PURPOSE OFFSET
35
36      ;CSR REGISTER
37
38      000001      SKDON=BIT0
39      000001      DRDY=BIT0           ;DRIVE READY
40      000100      INTEN=BIT6          ;INTERRUPT ENABLE
41      100000      ERR=BIT15          ;COMPOSITE ERROR
42      040000      DERR=BIT14         ;DRIVE ERROR
43      020000      NXM=BIT13          ;NON-EXISTANT MEMORY ERROR
44      010000      DLT=BIT12          ;DATA LATE
45      004000      DCRC=BIT11         ;DATA CRC ERROR
46      004000      HCRC=BIT11         ;HEADER CRC ERROR
47      010000      MNF=BIT12          ;HEADER NOT FOUND ERROR
48      002000      OPI=BIT10          ;OPERATION INCOMPLETE ERROR
49      000200      CRDY=BIT7          ;CONTROLLER READY
50      000040      BA17=BIT5          ;EXTENDED BUS ADDRESS BIT 17
51      000020      BA16=BIT4          ;EXTENDED BUS ADDRESS BIT 16
52
53      ;GET STATUS BITS
54
55      100000      WDE=BIT15           ;WRITE DATA ERROR
56      040000      HCE=BIT14         ;HEAD CURRENT ERROR
57      020000      WL=BIT13           ;WRITE LOCK

```

```

58      010000      SKTO=BIT12      ;SEEK TIMEOUT ERROR
59      004000      SPE=BIT11      ;SPINDLE TIMEOUT/UNDER/OVER SPEED
60      002000      WGE=BIT10      ;WRITE GATE ERROR
61      001000      VC=BIT9       ;VOLUME CHECK
62      000400      DSE=BIT8      ;DRIVE SELECT ERROR
63      000040      COP=BIT5      ;TOP COVER OPEN
64      000020      HOP=BIT4      ;HEADS OVER PACK
65      000010      BRHM=BIT3     ;BRUSHES HOME
66
67      ;COMMANDS
68
69      000002      WRCHK=BIT1     ;WRITE CHECK FUNCTION CODE
70      000004      GSTAT=BIT2     ;GET DRIVE STATUS FUNCTION CODE
71      000006      SEEK=BIT11:BIT2 ;SEEK FUNCTION CODE
72      000010      RDHDR=BIT3     ;READ HEADER FUNCTION CODE
73      000012      WRITE=BIT3:BIT1 ;WRITE FUNCTION CODE
74      000014      READ=BIT3:BIT2 ;READ FUNCTION CODE
75      000013      DRST=BIT3:BIT1:BIT0 ;DRIVE RESET COMMAND CODE FOR DRIVE COMMAND WORD
76      000003      GSBIT=BIT1:BIT0 ;GET STATUS COMMAND CODE FOR DRIVE COMMAND WORD
77      000001      MK=BIT0        ;MARKER BIT FOR DRIVE COMMAND WORD(SEEK,GET STATUS)
78      000004      SIGN=BIT2     ;DIRECTION FOR SEEK(0=AWAY FROM SPINDLE)
79      000020      SKHS=BIT4     ;HEAD SELECT FOR SEEK
80      000100      HEAD=BIT6     ;HEAD SELECT FOR READ,WRITE,GET STATUS
81
82      ;OFFSET FOR HARDWARE P-TABLE
83
84      000000      CSR=0
85      000002      DRBT=2
86
87 002160      ENDMOD
88
89      .SBTTL GLOBAL DATA AND CONSTANTS
90
91 002160      BGNMOD GLBDAT
92
93 002160 000000      ERRCNT: .WORD 0      ;ERROR COUNT - HARD
94 002162 000000      SFTCNT: .WORD 0      ;ERROR COUNT - SOFT
95 002164 000000      SKECNT: .WORD 0      ;SEEK ERROR COUNT
96 002166 000000      DERCNT: .WORD 0      ;DRIVE ERROR COUNT
97 002170 000000      WRTCNT: .WORD 0      ;WRITE PASS COUNT PER TRACK
98 002172 000000      RETRY: .WORD 0      ;PRESENT RETRY NUMBER
99 002174 000000      BDA: .WORD 0      ;DISK ADDRESS CONTENTS
100 002176 000000      BMP: .WORD 0      ;PRESENT MULTIPURPOSE CONTENTS
101 002200 000000      DCS: .WORD 0      ;CSR ADDRESS
102 002202 000000      E.DCS: .WORD 0      ;CONTENTS OF RLCS AT ERROR
103 002204 000000      E.STAT: .WORD 0      ;STATUS AT FAILURE TIME
104 002206 000000      BBA: .WORD 0      ;PRESENT BUS ADDRESS CONTENTS
105 002210 000000      FUNC: .WORD 0      ;LAST FUNCTION LOADED
106 002212 000000      BCSADR: .WORD 0      ;CSR IMAGE OF LAST COMMAND
107 002214 000000      LSTHDR: .WORD 0      ;LAST POSITION ON DISK
108 002216 000000      PRFLGS: .WORD 0      ;INTERNAL FLAGS
109 002220 000000      LSTDA: .WORD 0      ;DISK ADDRESS AT SOFT ERROR
110 002222 000000      DIFWD: .WORD 0      ;LAST DIFFERENCE WORD OF SEEK
111 002224 000000      SERNM1: .WORD 0      ;SERIAL NUMBER OF CARTRIDGE
112 002226 000000      SERNM2: .WORD 0      ;SERIAL NUMBER OF CARTRIDGE
113 002230 000000      NEWFAC: .WORD 0      ;FLAG TO BUILD A DUMMY FACTORY FILE
114 002232 000000      DRSEL: .WORD 0      ;DRIVE SELECT BITS(8,9,10)

```

```

115 002234 000000 BSECT: .WORD 0 ; POINTER TO BAD SECTOR FILE DATA STORAGE
116 002236 000000 RSEEK: .WORD 0 ; SEEK IN PROCESS OF RECOVERY
117 002240 000000 SOFTCS: .WORD 0 ; CSR OF SOFT ERROR
118 002242 000000 FWDFLG: .WORD 0 ; SAWTOOTH WRITE CONTROL FLAG
119 002244 000000 CVFLG: .WORD 0 ; 'CALL' FLAG FOR VERIFY ROUTINE
120 002246 000000 TDR: .WORD 0 ; TYPE OF DRIVE... RL01=1 RL02=2
121 002250 000000 WRIPG: .WORD 0 ; WRITE IN PROGRESS FLAG
122 002252 000000 PRPOS: .WORD 0 ; PRESENT POSITION ON DISK
123 002254 000000 NEWPOS: .WORD 0 ; NEW DESIRED CYLINDER ADDRESS
124 002256 000000 RECNT: .WORD 0 ; READ ERROR COUNT
125 002260 000000 NXTUNI: .WORD 0 ; POINTER OF UNIT SELECT SLOT IN 'SELTBL'
126 002262 000000 SYMSK: .WORD 0 ; MASK FOR 0-7 DRIVES
127 002264 100177 CYLSK: .WORD 100177 ; MASK FOR CYLINDER ONLY (RL01)
128 002266 100077 SECMSK: .WORD 100077 ; MASK OUT SECTOR BITS (RL01)
129 002270 000177 CMSK: .WORD 000177 ; CYL MASK FOR RL02
130 002272 000077 SMSK: .WORD 000077 ; SECT MASK FOR RL02
131 002274 000000 PASWD: .WORD 000000 ; PASSWORD (if=0 THEN NO CHECKING)
132 002276 000000 WRINIT: .WORD 0 ; WRITE INIT FLAG
133 002300 000160 BVEC: .WORD 160 ; VECTOR " "
134 002302 000240 BPPRIOR: .WORD 240
135 ;
136 ; THE FOLLOWING LOCATIONS ARE CLEARED AS A GROUP (DOWN TO 'STFLG')
137 ; THEREFORE DON'T INSERT ANY CONSTANTS
138 ;
139 002304 000000 LSTDR1: .WORD 0 ; BUFFER POINTER OF DRIVE
140 002306 000000 BCSR: .WORD 0 ; CSR FROM P-TABLE
141 002310 000000 BDRSEL: .WORD 0 ; DRIVE UNIT NUMBER FROM P-TABLE
142 002312 000000 HDRFND: .WORD 0 ; FLAG TO INDICATE HDR IN BAD LIST
143 002314 000000 CHKSEC: .WORD 0 ; SECTOR OF ERROR - USED BY BAD SECTOR LOCATION
144 002316 000000 DECNT: .WORD 0 ; DATA ERROR COUNT
145 002320 000000 TEMPO: .WORD 0 ; TEMP LOCATION
146 002322 000000 TEMP1: .WORD 0 ; TEMP LOCATION
147 002324 000000 TEMP2: .WORD 0 ; TEMP LOCATION
148 002326 000000 TEMP3: .WORD 0 ; TEMP LOCATION
149 002330 000000 LSTTIM: .WORD 0 ; LAST TIME ON SYSTEM CLOCK
150 002332 000000 SECOND: .WORD 0 ; SECONDS OF SYSTEM CLOCK
151 002334 000000 MINUTE: .WORD 0 ; MINUTES OF SYSTEM CLOCK
152 002336 000000 HOUR: .WORD 0 ; HOURS OF SYSTEM CLOCK
153 002340 000000 E.CS: .WORD 0 ; IMAGES OF REGISTERS
154 002342 000000 E.BA: .WORD 0 ; ON INTERRUPT
155 002344 000000 E.DA: .WORD 0 ;
156 002346 000000 E.MP: .WORD 0 ;
157 002350 000000 E.MP1: .WORD 0 ;
158 002352 000000 E.MP2: .WORD 0 ;
159 002354 000000 SYSCLK: .WORD 0 ; FLAG INDICATING PRESENCE OF SYSTEM CLOCK
160 002356 000000 BUF1: .WORD 0 ; BUFFER FOR FIRST CONTROLLER
161 002360 000000 MAXWC: .WORD 0 ; MAX WORD COUNT DETERMINED BY CORE
162 002362 000000 UUT: .WORD 0 ; NUMBER OF UNITS ON SYSTEM
163 002364 000000 SN1: .WORD 0 ; TYPED SERIAL # - LOW
164 002366 000000 SN2: .WORD 0 ; " " HIGH
165 002370 000000 WRTLOK: .WORD 0 ; WRITE LOCK FLAG
166 002372 000000 ACCESS: .WORD 0 ; ACCESS PRIV FOR UPDATING
167 002374 000000 PWRFIG: .WORD 0 ; POWER FAIL INDICATOR
168 002376 000000 TRPFLG: .WORD 0 ; TRAP OCCURANCE FLAG
169 002400 000000 CNTFLG: .WORD 0 ; CONTINUE FLAG
170 002402 000000 STFLG: .WORD 0 ; START FLAG
171 ;
    
```

```

172 ;END OF MASS CLEAR
173
174 002404 000004 ;ERRVEC: .WORD 4 ;ERROR VECTOR
175
176 002406 ENDMOD
177
178
179 .SBTTL GLOBAL MESSAGES
180
181 002406 BGNMOD GLBXT
182
183 ;GLOBAL TEXT
184
185
189
190 002406 122 114 103 MRLCS: .ASCIZ ''RLCS: ''
191 002415 040 050 122 CRLCS: .ASCIZ '' (RLCS): ''
192 002427 120 101 103 CART: .ASCIZ /PACK SERIAL #: /
193 002447 103 131 114 CMSG: .ASCIZ /CYLINDER: /
194 002462 040 110 105 HMSG: .ASCIZ / HEAD: /
195 002472 052 052 052 STARMMSG: .ASCIZ /*****/
196 002540 055 040 055 HYPHEN: .ASCIZ /- - - - -/
197 002606 124 117 124 TFMSG: .ASCIZ /TOTAL 'FIELD' ENTRYS = /
198 002636 124 117 124 TMMSG: .ASCIZ /TOTAL 'FACTORY' ENTRYS = /
199 002670 124 117 124 TSOFT: .ASCIZ /TOTAL 'SOFT' ERRORS FOUND = /
200 002725 124 117 124 THARD: .ASCIZ /TOTAL 'HARD' ERRORS FOUND = /
201 002762 123 117 106 MSREC: .ASCIZ /SOFT ERROR RECOVERED.../
202 003012 102 125 111 MBLD: .ASCIZ /BUILD A DUMMY BAD SECTOR FILE/
203 003050 102 125 111 BUILD: .ASCIZ /BUILDING BAD SECTOR FILE/
204 003101 040 123 105 SMSG: .ASCIZ / SECTOR: /
205 003113 101 124 040 BSEND: .ASCIZ /AT END OF FILE /
206 003133 123 105 105 MSKER: .ASCIZ /SEEK ERR/
207 003144 123 117 106 MSFER: .ASCIZ /SOFT ERR ENCOUNTERED/
208 003171 104 122 040 MDERS: .ASCIZ /DR ERR WILL NOT RESET/
209 003217 104 122 040 MRDER: .ASCIZ /DR ERR - RECOVERED/
210 003242 110 101 122 MHDER: .ASCIZ /HARD ERROR/
211 003255 104 122 040 NOLOAD: .ASCIZ /DR WOULD NOT LOAD/
212 003277 120 101 103 WRTLCK: .ASCIZ /PACK IS WRITE LOCKED/
213 003324 120 101 103 NEWLD: .ASCIZ /PACK WAS JUST LOADED/
214 003351 116 117 040 HWSEC: .ASCIZ /NO FACTORY FILE ENTRYS/
215 003400 116 117 040 NHWSEC: .ASCIZ /NO FACTORY FILE FOUND/
216 003426 116 117 040 SWSEC: .ASCIZ /NO FIELD FILE ENTRYS/
217 003453 116 117 040 NSWSEC: .ASCIZ /NO FIELD FILE FOUND/
218 003477 116 117 040 NOFLDE: .ASCIZ /NO SUCH ENTRY IN THE 'FIELD' FILE/
219 003541 103 117 115 MDONE: .ASCIZ /COMPLETED.../
220 003556 120 122 117 PRGER: .ASCIZ /PROGRAM BUG - DR NOT RDY/
221 003607 124 111 115 NOCRDY: .ASCIZ /TIMEOUT - NO 'CRDY'/
222 003633 116 117 040 NODRIV: .ASCIZ /NO DRIVES/
223 003645 040 104 122 DRNM: .ASCIZ / DRIVE: /
224 003656 105 116 124 PASSWORD: .ASCIZ /ENTER CORRECT PASSWORD TO ENABLE BAD SECTOR FILE UPDATES/
225 003747 125 120 104 DENIED: .ASCIZ /UPDATING DENIED - INVALID PASSWORD!/
226 004013 105 116 124 EXISTS: .ASCIZ /ENTRY ALREADY EXISTS IN BAD SECTOR FILE/
227 004063 122 105 101 VERIFY: .ASCIZ /READING PACK/
228 004100 127 122 111 MWRITE: .ASCIZ /WRITING PACK WITH WORST CASE DATA PATTERN/
229 004152 040 061 040 CMD1: .ASCIZ / 1 REPORT CONTENTS OF BAD SECTOR FILE/
230 004222 040 062 040 CMD2: .ASCIZ / 2 ADD A SECTOR TO THE 'FIELD' BAD SECTOR FILE/
231 004303 040 063 040 CMD3: .ASCIZ / 3 DELETE A SECTOR FROM THE 'FIELD' BAD SECTOR FILE/
    
```

```

232 004371      040      064      040  CMD4:  .ASCIZ  / 4  VERIFY PACK - READ ONLY/
233 004426      040      065      040  CMD5:  .ASCIZ  / 5  WRITE PACK WITH WORST CASE DATA PATTERN & VERIFY/
234 004514      040      066      040  CMD6:  .ASCIZ  / 6  MAKE A BAD SECTOR FILE/
235 004550      040      067      040  CMD7:  .ASCIZ  / 7  PRINT THIS HELP MESSAGE AGAIN/
236 004613      103      115      104  CMDDO: .ASCIZ  /CMD (7 = HELP) - /
237 004635      103      117      116  BSRM:  .ASCIZ  /CONTENTS OF THE 'FACTORY' BAD SEC FILE:/
238 004705      103      117      116  BSRF:  .ASCIZ  /CONTENTS OF THE 'FIELD' BAD SEC FILE:/
239 004753      102      101      104  BADBSF:.ASCIZ  /BAD READ OF BAD SECTOR FILE (20 SECTORS)/
240 005024      101      104      104  ABSMSG:.ASCIZ  /ADD CYLINDER, SECTOR, & HEAD TO 'FIELD' BAD SECTOR FILE/
241 005114
242 005114      103      131      114  ABSCYL:.ASCIZ  /CYLINDER # (0 TO 511.) - /
243 005146
244 005146      123      105      103  ABSSEC:.ASCIZ  /SECTOR # (0 TO 39.) - /
245 005175
246 005175      110      105      101  ABSHD: .ASCIZ  /HEAD # (0 OR 1) - /
247 005220      103      101      122  ABSSER:.ASCIZ  /CARTRIDGE HAS NO SERIAL NUMBER - ADD ONE?/
248 005272      111      116      120  ABSSNL:.ASCIZ  /INPUT THE LOW 5 OCTAL # OF SN /
249 005331      111      116      120  ABSSNH:.ASCIZ  /INPUT THE HIGH 5 OCTAL # OF SN /
250 005371      120      122      117  DOWRT: .ASCIZ  /PROCEED TO WRITE THE UPDATED BAD SECTOR FILE?/
251 005447      103      101      116  BADWRT:.ASCIZ  /CAN'T UPDATE THE BAD SECTOR FILE ON PACK/
252 005520      104      105      114  DELMSG:.ASCIZ  /DELETE A 'FIELD' BAD SECTOR FILE ENTRY/
253 005567      116      117      040  NOENTRY:.ASCIZ /NO SUCH ENTRY EXISTS TO DELETE!/
254 005627      122      114      060  RL1CLM:.ASCIZ  /RLO1 MAX CYLINDER = 255./
255 005660      111      123      040  VALSN: .ASCIZ  /IS THIS SERIAL NUMBER VALID/
256 005714      115      117      122  TBLFUL:.ASCIZ  /MORE THAN 25. BAD SPOTS FOUND ON THIS PACK!/
257 005770      103      117      116  TILLEND:.ASCIZ /CONTINUE TO END OF FILE/
258 006020      104      117      040  MSTWRT:.ASCIZ  /DO YOU REALLY WANT TO WRITE ON ALL SELECTED PACKS?/
259 006103      116      105      127  NEWENT:.ASCIZ  /NEW ENTRY.../
260 006120      130      106      105  ERRAT: .ASCIZ  /XFER ERROR STARTING AT /
261 006150      122      114      060  OVRMAX:.ASCIZ  /RLO1-RLO2 CARTRIDGE SPEC ALLOWS A MAX OF 16. BAD SECTORS /
262 006242      106      117      125  CK:    .ASCIZ  /FOUND/
263 006250      123      105      103  INBSF: .ASCIZ  /SECTOR IS IN BAD SECTOR FILE/
264 006305      103      110      105  CKFACT:.ASCIZ  /CHECKING FOR 'FACTORY' FILE.../
265 006344      103      110      105  CKFLD: .ASCIZ  /CHECKING FOR 'FIELD' FILE.../
266 006401      104      122      111  NOTRDY:.ASCIZ  /DRIVE NOT READY FOR USE/
267 006431      104      122      111  MDRTYP:.ASCIZ  /DRIVE TYPE = RLO/
268 006452      055      040      123  SAWFWD:.ASCIZ  /- SAWTOOTH FROM CYL 0/
269 006500      055      040      123  SAWREV:.ASCIZ  /- SAWTOOTH FROM LAST CYL/
270 006531      127      122      124  WRPKF: .ASCIZ  /WRT PACK FORWARD /
271 006553      127      122      124  WRPKR: .ASCIZ  /WRT PACK REVERSE /
272 006575      125      123      105  THISDRV:.ASCIZ /USE THIS SELECTED UNIT?/
276
277
278 006626
279
280
281
282 006626
283
284 006626
285 006626      010146
286
287
288 006630      004537      010572
289 006634      013737      002314      013462
290 006642      042737      177700      013462
291 006650      005037      013464

```

```

      .EVEN
      ENDMOD
      .SBTTL  ERROR MESSAGES
      BGNMOD  GLBERR
      BGNMSG  ERR1
      MOV     R1,-(SP)          ;SAVE R1
      ;ROUTINE TO REPORT THE POSITION OF CYLINDER, SECTOR & HEAD
      JSR     R5,PTIME         ;PRINT RUN TIME
      MOV     CHKSEC,BSFSEC    ;GET THE SECTOR IN ERROR
      BIC     #177700,BSFSEC   ;CLEAR THE JUNK BITS
      CLR     BSFMD           ;CLEAR THE HEAD #

```

```

292 006654 052737 000100 002314      BIT      #100,CHKSEC      ;HEAD 1??
293 006662 001402                      BEQ      1$            ;NO
294 006664 005237 013464                      INC      BSFHD         ;YES - SET IT TO 1
295 006670 013737 002314 013460 1$:      MOV      CHKSEC,BSFCYL ;GET ADDR AGAIN FOR THE CYLINDER
296 006676 042737 000177 013460      BIC      #177,BSFC     ;CLEAR THE HEAD & SECTOR #
297 006704 000337 013460      SWAB    BSFCYL
298 006710 000241                      CLC
299 006712 006137 013460      ROL     BSFCYL        ;CLEAR THE 'C' BIT
300 006716 103002                      BCC     2$            ;POSITION
301 006720 005237 013460      INC     BSFCYL        ;BR IF DON'T NEED OTHER BIT
302 006724                      2$:      PRINTB  #FMT16,#ERRAT,#CMMSG,BSFCYL,#SMSG,BSFSEC,#HMSG,BSFHD
      006724 013746 013464      MOV     BSFHD,-(SP)
      006730 012746 002462      MOV     #HMSG,-(SP)
      006734 013746 013462      MOV     BSFSEC,-(SP)
      006740 012746 003101      MOV     #SMSG,-(SP)
      006744 013746 013460      MOV     BSFCYL,-(SP)
      006750 012746 002447      MOV     #CMMSG,-(SP)
      006754 012746 006120      MOV     #ERRAT,-(SP)
      006760 012746 007232      MOV     #FMT16,-(SP)
      006764 012746 000010      MOV     #10,-(SP)
      006770 010600      MCV     SP,RO
      006772 104014      EMT     C$PNTB
      006774 062706 000022      ADD     #22,SP
303 007000 004537 023342      JSR     R5,GETDST     ;GET THE DRIVE STATUS
304 007004 010137 002204      MOV     R1,E.STAT
305 007010      PRINTB  #FMT17A,#CRLCS,E.DCS,E.STAT
      007010 013746 002204      MOV     E.STAT,-(SP)
      007014 013746 002202      MOV     E.DCS,-(SP)
      007020 012746 002415      MOV     #CRLCS,-(SP)
      007024 012746 007303      MOV     #FMT17A,-(SP)
      007030 012746 000004      MOV     #4,-(SP)
      007034 010600      MOV     SP,RO
      007036 104014      EMT     C$PNTB
      007040 062706 000012      ADD     #12,SP
306 007044      PRINTB  #FMT18,#HYPHEN
      007044 012746 002540      MOV     #HYPHEN,-(SP)
      007050 012746 007335      MOV     #FMT18,-(SP)
      007054 012746 000002      MOV     #2,-(SP)
      007060 010600      MOV     SP,RO
      007062 104014      EMT     C$PNTB
      007064 062706 000006      ADD     #6,SP
307 007070      PRINTB  #MCRLF
      007070 012746 007552      MOV     #MCRLF,-(SP)
      007074 012746 000001      MOV     #1,-(SP)
      007100 010600      MOV     SP,RO
      007102 104014      EMT     C$PNTB
      007104 062706 000004      ADD     #4,SP
308 007110 012601      MOV     (SP)+,R1     ;RESET R1
309 007112      ENDMSG
      007112                      L10000:      EMT     C$MSG
      007112 104023
310
311 007114      BGNMSG  ERR2
312 007114 010146      MOV     R1,-(SP)     ;SAVE R1
313 007116 004537 023342      JSR     R5,GETDST     ;GET THE DRIVE STATUS
314 007122 010137 002204      MOV     R1,E.STAT     ;SAVE STATUS FOR PRINTING
315 007126      PRINTB  #FMT17A,#CRLCS,E.DCS,E.STAT
  
```

```
007126 013746 002204      MOV      E. STAT, -(SP)
007132 013746 002202      MOV      E. DCS, -(SP)
007136 012746 002415      MOV      #CRLCS, -(SP)
007142 012746 007303      MOV      #FMT17A, -(SP)
007146 012746 000004      MOV      #4, -(SP)
007152 010600      MOV      SP, R0
007154 104014      EMT      C$PNTB
007156 062706 000012      ADD      #12, SP
316 007162      PRINTB  #FMT18, HYPHEN
007162 013746 002540      MOV      HYPHEN, -(SP)
007166 012746 007335      MOV      #FMT18, -(SP)
007172 012746 000002      MOV      #2, -(SP)
007176 010600      MOV      SP, R0
007200 104014      EMT      C$PNTB
007202 062706 000006      ADD      #6, SP
317 007206      PRINTB  #MCRLF
007206 012746 007552      MOV      #MCRLF, -(SP)
007212 012746 000001      MOV      #1, -(SP)
007216 010600      MOV      SP, R0
007220 104014      EMT      C$PNTB
007222 062706 000004      ADD      #4, SP
318 007226 012601      MOV      (SP)+, R1      ; RESET R1
319 007230      ENDMSG
007230      L10001:
007230 104023      EMT      C$MSG
323 007232 045 116 045 FMT16: .ASCIZ /%N%T%I%Z3%A.%T%Z2%A.%T%D1%N/
324 007266 045 116 045 FMT17: .ASCIZ /%N%T%O6%T%O1/
325 007303 045 124 045 FMT17A: .ASCIZ /%T%O6%A STATUS WAS: %O6%N/
326 007335 045 116 045 FMT18: .ASCIZ /%N%T/
327 007342 045 116 045 FMT19: .ASCIZ /%N%N%T/
328 007351 045 116 045 FMT20: .ASCIZ /%N%ABAD SECTOR FILE HAS %Z3%A. ENTRIES/
329 007420 045 116 045 FMTSN: .ASCIZ /%N%T%O5%O5%N/
330 007435 045 116 045 FMTTB: .ASCIZ /%N%T%Z3%A./
331 007450 045 116 045 FMTCSH: .ASCIZ /%N%Z2%A.%T%Z3%A.%T%Z2%A.%T%D1/
332 007510 045 116 045 FMTMS: .ASCIZ /%N%N%ACOMMANDS AVAILABLE ARE:%N%T/
333 007552 045 116 000 MCRLF: .ASCIZ /%N/
334 007555 045 124 000 MSG: .ASCIZ /%T/
335 007560 045 116 045 TIME: .ASCIZ /%N%ATIME: %Z2%A:%Z2%A:%Z2%A /
336 007616 045 116 045 FDTYP: .ASCIZ /%N%T%O1%N/
337
341
342      .EVEN
343 007630      ENDMOD
344
345 007630      BGNMOD  HPTCODE
346 007630      BGNHW
007630 000002      .WORD  L10002-L$HW/2
347 007632 174400      .WORD  174400      ;(SR BASE ADDRESS DEFAULT
348 007634 000000      .WORD  0          ;DRIVE UNIT NUMBER DEFAULT
349 007636      ENDNHW
007636      L10002:
350 007636      ENDMOD
351 007636      BGNMOD  SPTCODE
352 007636      BGNSW
007636 000002      .WORD  L10003-L$SW/2
353 007640 000001      WRTSAW: .WORD  1          ;DEFAULT TO SAWTOOTH WRITE CYCLE
354 007642 000002      WRTLIM: .WORD  2          ;DEFAULT TO 2 WRITE PASSES PER TRACK
```



```

355 007644      ENDSW
      007644      L10003:
356 007644      ENDMOD
357
358 007644      BGNMOD  DSPCODE
359
360 007644      DISPATCH  1
      007644      .WORD    1
      007646      .WORD    11
361
362 007650      ENDMOD
363
364      .SBTTL  STATISTIC CODE
365
366 007650      BGNMOD  RPTCODE
367 007650      BGNRPT
368 007650      ENDRPT
      007650      L10004:
      007650      104025  EMT      CSRPT
369 007652      ENDMOD
370
371      .SBTTL  INITIALIZATION CODE
372
373 007652      BGNMOD  INITCODE      ;START OF INITIALIZE CODE
374
375 007652      BGNINIT
376
377 007652      SETPRI  #340      ;PRIORITY TO SEVEN
      007652      012700  000340  MOV      #340,R0
      007656      104041  EMT      C$SPRI
378
379 007660      BRESET      ;FOR LSI-11 CPU'S
      007660      104033  EMT      C$RESET
380 007662      005037  002402  CLR      STFLG
381 007666      005037  002400  CLR      CNTFLG      ;CLEAR CONT
382 007672      005037  002374  CLR      PWRFLG      ;CLEAR THE POWER FAIL FLAG
383 007676      READEF  #EF.PWR
      007676      012700  000034  MOV      #EF.PWR,R0
      007702      104050  EMT      C$REFG
384 007704      BNCOMPLETE  3$
      007704      103064  BCC      3$
385 007706      005237  002374  !NC     PWRFLG      ;INDICATE POWER FAIL
386 007712      013702  002362  MOV      UUT,R2      ;GET NUMBER OF UNITS SELECTED
387 007716      005302  DEC      R2
388 007720      006302  ASL     R2
389 007722      006302  ASL     R2
390 007724      062702  010330  ADD     #SELTBL,R2      ;POINT TO THE CORRECT SLOT
391 007730      012237  002200  11$:  MOV     (R2)+,DCS      ;GET THE DCS ADDRESS
392 007734      011237  002232  MOV     (R2),DRSEL      ;AND GET THE DRIVE BITS
393 007740      052737  000200  002232  BIS     #200,DRSEL      ;ADD IN THE CRDY BIT
394 007746      013777  002232  172224  MOV     DRSEL,@DCS      ;SELECT THE DRIVE
395 007754      012701  000074  MOV     #60,,R1      ;SETUP A TIMER
396 007760      032777  000001  172212  12$:  BIT     #1,@DCS      ;DRIVE READY UP?
397 007766      001015  BNE     14$      ;YUP - RESET DRIVE & HEADS HOME
398
399 007770      WAITMS  #10.      ;WAIT A WHILE
      007770      012700  000012  MOV     #10.,R0

```

```

007774 104026
400 007776 005301
401 010000 001367
402
403
404 010002 005742
405 010004 005022
406
407 010006 162702 000004
408 010012 022702 010330
409 010016 003744
410 010020 000404
411
412 010022 004537 023356
413 010026 004537 024700
414
415 010032 005737 002354
416 010036 001405
417 010040
    010040 012700 000001
    010044 104034
418 010046 004537 024544
419 010052 000137 010326
420
421 010056
    010056 012700 000036
    010062 104050
422 010064
    010064 103004
423
424 010066 005237 002400
425 010072 000137 010172
426
427 010076 004537 025230
428
429 010102 012700 002304
430 010106 005020
431 010110 020027 002404
432 010114 001374
433
434 010116 012700 010330
435 010122 012720 177777
436 010126 020027 010370
437 010132 001373
438
439 010134 013703 002012
440 010140 010337 002362
441 010144 012704 010330
442 010150 005001
443 010152
    010152 010100
    010154 104042
444 010156
    010156 103002
445 010160 012024
446 010162 011024
447 010164 005201

EMT      C$WTM
DEC      R1
BNE      12$
          :UPDATE THE TIMER
          :IF MORE TIME, THEN TRY AGAIN

          ;DRIVE NOT READY IN TIME - KILL THE ENTRY
TST      -(R2)
CLR      (R2)+
          :CORRECT THE POINTER
          :KILL THE ENTRY WORD FOR DCS

13$:     SUB      #4,R2
          :POINT TO THE NEXT ENTRY IN LIST
CMP      #SELTBL,R2
          :DONE?
BLE      11$
BR       15$
          :NO - DO THIS UNIT ALSO
          :YES - PROCEED

14$:     JSR      R5,ISDRST
          :RESET THE DRIVE SELECTED
          :AND BRING THE HEADS HOME
JSR      R5,HDHOME

15$:     TST      SYSCLK
          :CLOCK TICK?
BEQ      4$
          :BR IF NO
          :YES - SET FOR 1 SEC INTERVALS
CLKON    #1
MOV      #1,R0
EMT      C$KWON
JSR      R5,GETSYS
          :THEN GET THE TIME
JMP      POWER

4$:      JMP      POWER

3$:      REDEF   #EF.CONTINUE
MOV      #EF.CONTINUE,R0
EMT      C$REFG
          :CONTINUE FROM CONSOLE?
BNCOMPLETE 1$
BCC      1$
          :NO, CONTINUE W/ INIT CODE

INC      CNTFLG
JMP      END
          :YES SET CONT FLAG, GO TO END OF INIT

1$:      JSR      R5,CLEAR
          :CLEAR ALL DRIVE STORAGE BUFFERS

2$:      MOV      #LSTDR1,R0
          :CLEAR FLAGS
CLRDAT:  CLR      (R0)+
          :
CMP      R0,#STFLG+2
          :MASS CLEAR
BNE      CLRDAT
          :DO TILL TABLE IS ZEROED

CLRSTB:  MOV      #SELTBL,R0
          :INIT THE SELECT TABLE
MOV      #-1,(R0)+
          :END OF THE TABLE?
CMP      R0,#STBLE
          :NO CLEAR THE NEXT
BNE      CLRSTB

MOV      L$UNIT,R3
          :GET NUMBER OF UNITS
MOV      R3,UUT
          :SAVE L$UNIT
MOV      #SELTBL,R4
          :INIT SELECT TABLE POINTER
CLR      R1
          :INIT P-TABLE
1$:      GPHARD  R1,R0
          :GET A P-TABLE
MOV      R1,R0
EMT      C$GPHRD
BNCOMPLETE 2$
BCC      2$
          :
MOV      (R0)+,(R4)+
          :GET CSR INTO SELECT TABLE STORAGE
MOV      (R0),(R4)+
          :GET DRIVE INTO TABLE
2$:      INC      R1
          :POINT TO NEXT
```

```

448 010166 005303          DEC      R3          ;DOWN COUNT
449 010170 001370          BNE      1$          ;DO WHILE
450
451 010172          END:
452 010172 013704 002362    MOV      UUT,R4
453 010176 006304          ASL      R4
454 010200 006304          ASL      R4
455 010202 062704 010330    ADD      #SELTBL,R4          ;POINT TO THE SELECT TABLE
456 010206 012704 177777    MOV      #-1,R4          ;FORCE A TERMINATE IN THE TABLE
457 010212          2$: READEF #EF.START          ;START COMMAND
      010212 012700 000040    MOV      #EF.START,RO
      010216 104050          EMT      C$REFG
458 010220          BNCOMPLETE RESTART          ;NO, CHK RESTART
      010220 103002          BCC      RESTART
459 010222 005237 002402    INC      STFLG          ;SET START INDICATOR
460
461 010226          RESTART:
462 010226          SETVEC BVEC,#INTR1,BPRIOR          ;SET CONTROLLER VECTOR
      010226 013746 002302    MOV      BPRIOR,-(SP)
      010232 012746 021700    MOV      #INTR1,-(SP)
      010236 013746 002300    MOV      BVEC,-(SP)
      010242 012746 000003    MOV      #3,-(SP)
      010246 104037          EMT      C$SVEC
      010250 062706 000010    ADD      #10,SP
463
464 010254 012737 026442 002356 FINDBF: MOV      #BSFILE,BUF1          ;ALL XFERS TO BSFILE STORAGE
465 010262 012737 005000 002360    MOV      #5120./2,MAXWC          ;MAX XFER SIZE = 1/2 TRACK
466 010270 012737 002356 002206    MOV      #BUF1,BBA          ;POINT TO THE DATA STORAGE AREA
467 010276 012737 025330 002234    MOV      #BSECO,BSECT          ;POINT TO THE BAD SECTOR FILE DATA
468 010304 005737 002400          TST      CNTFLG          ;HERE FROM 'CON' CMD?
469 010310 001006          BNE      POWER          ;BR IF TRUE
470
471 010312          1$: CLKON #1          ;TURN CLOCK ON?
      010312 012700 000001    MOV      #1,RO
      010316 104034          EMT      C$KWON
472 010320          BNCOMPLETE POWER          ;WAS THERE A CLOCK?
      010320 103002          BCC      POWER
473 010322 005237 002354          INC      SYSCLK          ;YES, SET FLAG FOR ONE!
474 010326          POWER:
475 010326          L10005: ENDINIT
      010326 104011          EMT      C$INIT
476
477 010330          SELTBL: .BLKW 16.
478 010370 177777          STBLE: .WORD -1
479 010372          ENDMOD
480
481 010372          BGNMOD CLNCODE
482 010372          BGNCLN
483
484 010372          SETVEC ERRVEC,#TRPHAN,#340
      010372 012746 000340    MOV      #340,-(SP)
      010376 012746 010564    MOV      #TRPHAN,-(SP)
      010402 013746 002404    MOV      ERRVEC,-(SP)
      010406 012746 000003    MOV      #3,-(SP)
      010412 104037          EMT      C$SVEC
      010414 062706 000010    ADD      #10,SP

```

485	010420			SETPRI	#PRI00				
	010420	012700	000000	MOV	#PRI00,RO				:PRIORITY TO ZERO
	010424	104041		EMT	C\$SPRI				
486	010426			CLRVEC	BVEC				:RELEASE VECTOR OF FIRST CONTROLLER
	010426	013700	002300	MOV	BVEC,RO				
	010432	104036		EMT	C\$CVEC				
487									
488	010434			3\$: CLRVEC	ERRVEC				
	010434	013700	002404	MOV	ERRVEC,RO				
	010440	104036		EMT	C\$CVEC				
489	010442	005737	00234	TST	SYSCLK				
490	010446	001400		BEQ	4\$				
491	010450			4\$: BRESET					:TAKE CARE OF LSI-11
	010450	104033		EMT	C\$RESET				
492	010452			ENDCLN					
	010452			L10006:					
	010452	104012		EMT	C\$CLEAN				
493									
494	010454			ENDMOD					
495									
496	010454			BGNMOD	ADDCODE				
497	010454			BGNAU					
498	010454			ENDAU					
	010454			L10007:					
	010454	104054		EMT	C\$AU				
499	010456			ENDMOD					
500									
501	010456			BGNMOD	DROPCODE				
502	010456			BGNDU					
503	010456			ENDDU					
	010456			L10010:					
	010456	104055		EMT	C\$DU				
504	010460			ENDMOD					

```

1          .SBTTL  GLOBAL SUBROUTINES
2
3 010460   BGNMOD  GLBSUB
4 010460   STARS
           ;*****
5          ;FIRST & SELDRV -- DRIVE SELECT ROUTINE
6 010460   STARS
           ;*****
7
8 010460   012704 010330   FIRST:  MOV    #SELTBL,R4           ;POINT TO THE SELECT TABLE
9 010464   010437 002260           MOV    R4,NXTUNI
10
11 010470   013704 002260   SELDRV: MOV    NXTUNI,R4           ;SETUP THE POINTER
12 010474   005714           10$:  TST    (R4)           ;CHECK FOR A VALID ENTRY
13 010476   100402           BMI    1$           ;OK TO GO ON
14 010500   022424           CMP    (R4)+,(R4)+   ;POINT TO THE NEXT ENTRY SLOT
15 010502   000774           BR     10$          ;AND TRY AGAIN
16
17 010504   012437 002200   1$:  MOV    (R4)+,DCS           ;GET THE CSR ADDR FROM TABLE
18 010510   022737 177777 002200   CMP    #-1,DCS           ;END OF THE TABLE?
19 010516   001001           BNE   2$           ;NO - CONTINUE
20 010520   000416           BR     4$           ;EXIT +1
21
22 010522   012437 002232   2$:  MOV    (R4)+,DRSEL          ;GET THE DRIVE SELECT BITS
23 010526   004537 023342           JSR   R5,GETDST        ;GET THE DRIVE STATUS
24 010532   012737 000001 002246   MOV    #1,TDR           ;DEFAULT TO RL01 TYPE
25 010540   032701 000200           BIT   #BIT7,R1         ;IS IT AN RL02?
26 010544   001403           BEQ   3$           ;NO
27 010546   012737 000002 002246   MOV    #2,TDR           ;YES - SET FOR AN RL02
28
29 010554   022525           3$:  CMP    (R5)+,(R5)+   ;RETURN +2 - NORMAL EXIT
30 010556   010437 002260   4$:  MOV    R4,NXTUNI        ;SAVE THE 'NEXT' SLOT POINTER
31 010562   000205           RTS   R5             ;EXIT
32
33 010564   005237 010564   TRPHAN: INC    TRPHAN
34 010570   000002           RTI
35
36 010572   STARS
           ;*****
37          ;PTIME -- ROUTINE TO PRINT THE SYSTEM RUNTIME IF A CLOCK IS PRESENT
38 010572   STARS
           ;*****
39
40 010572   005737 002354   PTIME:  TST    SYSCLK           ;CLOCK PRESENT?
41 010576   001420           BEQ   1$           ;NO
42 010600   004537 024544           JSR   R5,GETSYS        ;YES - GET THE RUN TIME FROM SUPERVISOR
43 010604           PRINTB #TIME,HOUR,MINUTE,SECOND
           MOV    SECOND,-(SP)
           MOV    MINUTE,-(SP)
           MOV    HOUR,-(SP)
           MOV    #TIME,-(SP)
           MOV    #4,-(SP)
           MOV    SP,R0
           EMT    C$PNTB
           ADD    #12,SP
44 010640   000205   1$:  RTS   R5             ;EXIT

```

```
1 010642 STARS
2          :*****
3 010642 :DRVID -- ROUTINE TO PRINT THE SELECTED UNIT IDENTIFICATION
          STARS
          :*****
4
5 010642 DRVID: PRINTF #FMT17,#MRLCS,DCS,#DRNM,<B,DRSEL+1>
010642 CLR -(SP)
010644 005046 153716 002233 BISB DRSEL+1,(SP)
010650 012746 003645 MOV #DRNM,-(SP)
010654 013746 002200 MOV DCS,-(SP)
010660 012746 002406 MOV #MRLCS,-(SP)
010664 012746 007266 MOV #FMT17,-(SP)
010670 012746 000005 MOV #5,-(SP)
010674 010600 MOV SP,R0
010676 104017 EMT C$PNTF
010700 062706 000014 ADD #14,SP
6 010704 PRINTF #FDTYP,#MDRTYP,TDR
010704 013746 002246 MOV TDR,-(SP)
010710 012746 006431 MOV #MDRTYP,-(SP)
010714 012746 007616 MOV #FDTYP,-(SP)
010720 012746 000003 MOV #3,-(SP)
010724 010600 MOV SP,R0
010726 104017 EMT C$PNTF
010730 062706 000010 ADD #10,SP
7 010734 000205 RTS R5
8
9 010736 STARS
10         :*****
11 010736 :DRNRDY -- ROUTINE TO PRINT THE DRIVE SELECTED ISN'T READY
          STARS
          :*****
12
13 010736 004537 010642 DRNRDY: JSR R5,DRVID
14 010742 PRINTF #MSG,#NOTRDY
010742 012746 006401 MOV #NO'RDY,-(SP)
010746 012746 007555 MOV #MSG,-(SP)
010752 012746 000002 MOV #2,-(SP)
010756 010600 MOV SP,R0
010760 104017 EMT C$PNTF
010762 062706 000006 ADD #6,SP
15 010766 004537 010774 JSR R5,DRDRV ;DROP THE DRIVE SELECTED
16 010772 000205 RTS R5
17
18 010774 STARS
19         :*****
20         :DRDRV -- ROUTINE TO KILL A UNIT ENTRY INTO THE SELTBL AREA IF THE
          : PGM DETERMINES A UNIT IS NOT ABLE TO BE USED
21 010774 STARS
          :*****
22
23 010774 013704 002260 DRDRV: MOV NXTUNI,R4 ;POINT TO THE 'NEXT' UNIT SLOT
24 011000 162704 000004 SUB #4,R4 ;POINT TO THE CURRENT UNIT
25 011004 005024 CLR (R4)+ ;KILL THE ENTRY
26 011006 005024 CLR (R4)+ ;KILL DRSEL ENTRY
27 011010 000205 RTS R5 ;EXIT
28 011012 ENDMOD
```

```

1          .SBTTL PROGRAM MAIN LOOP
2
3 011012   BGNTST
4 011012   STARS
           ;*****
5          ;THIS IS WHERE CONTROL IS PASSED AFTER THE INITIAL QUESTIONS HAVE
6          ;BEEN ANSWERED FOR THE P-TABLE STORAGE.
7 011012   STARS
           ;*****
8
9 011012   MTEST:
10 011012  004537 010460   JSR      R5,FIRST           ;SELECT THE 1ST DRIVE
11 011016  000137 011310   JMP      WHATCMD          ;NO - UNITS
12 011022  000404                BR      2$
13 011024  004537 010470   1$: JSR      R5,SELDRV       ;SELECT ANOTHER UNIT
14 011030  000137 011310   JMP      WHATCMD          ;NO MORE TO SELECT
15 011034  012777 000200 171136 2$: MOV      #200,@DCS        ;CHECK IF DRIVE THERE
16 011042  053777 002232 171130  BIS      DRSEL,@DCS
17 011050  012700 000000   MOV      #0.,R0          ;STALL
18 011054  005300                13$: DEC      R0
19 011056  001376                BNE      13$
20 011060  004537 023342   JSR      R5,GETDST        ;GET THE CURRENT DRIVE STATUS
21 011064  010137 002320   MOV      R1,TEMPO        ;SAVE THE STATUS
22 011070                PRINTF   #MCRLF
    011070  012746 007552   MOV      #MCRLF,-(SP)
    011074  012746 000001   MOV      #1,-(SP)
    011100  010600                MOV      SP,R0
    011102  104017                EMT      C$PNTF
    011104  062706 000004   ADD      #4,SP
23 011110  004537 010642   JSR      R5,DRVID        ;TELL OPR THE UNIT SELECTED
24
25 011114  032737 000020 002320 130$: BIT      #4OP,TEMPO        ;ARE THE HEADS LOADED?
26 011122  001015                BNE      131$            ;BR IF OK
27 011124                PRINTF   #FMT18,#NOLOAD
    011124  012746 003255   MOV      #NOLOAD,-(SP)
    011130  012746 007335   MOV      #FMT18,-(SP)
    011134  012746 000002   MOV      #2,-(SP)
    011140  010600                MOV      SP,R0
    011142  104017                EMT      C$PNTF
    011144  062706 000004   ADD      #6,SP
28 011150  004537 010774   JSR      R5,DRDRV        ;DROP THIS DRIVE
29 011154  000451                BR      15$
30 011156  032737 020000 002320 131$: BIT      #WL,TEMPO        ;IS THE PACK WRITE LOCKED?
31 011164  001414                BEQ      132$            ;BR IF NOT WRT LOCKED
32 011166                PRINTF   #FMT18,#WRTLCK
    011166  012746 003277   MOV      #WRTLCK,-(SP)
    011172  012746 007335   MOV      #FMT18,-(SP)
    011176  012746 000002   MOV      #2,-(SP)
    011202  010600                MOV      SP,R0
    011204  104017                EMT      C$PNTF
    011206  062706 000006   ADD      #6,SP
33 011212  005237 002370                INC      WRTLCK          ;SET THE WRITE LOCK FLAG
34 011216  032737 001000 002320 132$: BIT      #VC,TEMPO        ;PACK JUST LOADED?
35 011224  001412                BEQ      133$            ;JUMP IF NOT
36 011226                PRINTF   #FMT18,#NEWLD
    011226  012746 003324   MOV      #NEWLD,-(SP)
    011232  012746 007335   MOV      #FMT18,-(SP)

```

```
011236 012746 000002      MOV    #2,-(SP)
011242 010600      MOV    SP,R0
011244 104017      EMT    C$PNTF
011246 062706 000006      ADD    #6,SP
37 011252 004537 023356      133$: JSR    R5,ISDRST      ;RESET THE DRIVE
38 011256 004537 023342      JSR    R5,GETDST     ;GET THE DRIVE STATUS AGAIN
39 011262 032777 100000 170710  BIT    #ERR,@DCS     ;COMPOSITE ERROR STILL SET?
40 011270 001403      BEQ    15$          ;NOPE - SKIP OVER
41 011272      ERRDF 170.,MDERS
011272 104422      TRAP  T$ERCODE
011274 000252      .WORD 170
011276 003171      .WORD MDERS
42
43 011300      15$: SETPRI #0          ;PRIORITY TO ZERO
011300 012700 000000      MOV    #0,R0
011304 104041      EMT    C$SPRI
44 011306 000646      BR     1$          ;SELECT THE NEXT UNIT
```



```
1 011310 STARS
2
3 011310 :*****
:HERE IS THE 'CMD>' QUARY LOOP FOR THINGS TO DO
STARS
:*****

4
5 011310 005737 002402 WHATCMD: TST STFLG ;JUST STARTING?
6 011314 001561 BEQ NXTCMD ;NOPE - SKIP OVER THE BLURB
7 011316 005737 002274 TST PASWD ;DO THE PASSWORD STUFF?
8 011322 001440 BEQ HLPMSG ;NO - PRINT THE HELP BLURB
9 011324 PRINTF #MCRLF
011324 012746 007552 MOV #MCRLF,-(SP)
011330 012746 000001 MOV #1,-(SP)
011334 010600 MOV SP,RO
011336 104017 EMT C$PNTF
011340 062706 000004 ADD #4,SP
10 011344 GMANID PASWD,TEMPO,0,177777,1,177777,NO ;GET THE PASSWORD
011344 104043 EMT C$GMAN
011346 000406 BR 10000$
011350 002320 .WORD TEMPO
011352 000022 .WORD T$CODE
011354 003656 .WORD PASWORD
011356 177777 .WORD 177777
011360 000001 .WORD T$LOLIM
011362 177777 .WORD T$HILIM
011364
11 011364 023737 002274 002320 10000$: CMP PASWD,TEMPO ;CORRECT PASSWORD?
12 011372 001414 BEQ HLPMSG ;YUP
13 011374 005237 002372 INC ACCESS ;SET THE DENIED FLAG
14 011400 PRINTF #FMT18,#DENIED ;& TELL OPR
011400 012746 003747 MOV #DENIED,-(SP)
011404 012746 007335 MOV #FMT18,-(SP)
011410 012746 000002 MOV #2,-(SP)
011414 010600 MOV SP,RO
011416 104017 EMT C$PNTF
011420 062706 000006 ADD #6,SP
15
16 011424 HLPMSG: PRINTF #FMTMS,#CMD1 ;PRINT THE HELP BLURB
011424 012746 004152 MOV #CMD1,-(SP)
011430 012746 007510 MOV #FMTMS,-(SP)
011434 012746 000002 MOV #2,-(SP)
011440 010600 MOV SP,RO
011442 104017 EMT C$PNTF
011444 062706 000006 ADD #6,SP
17 011450 PRINTF #FMT18,#CMD2
011450 012746 004222 MOV #CMD2,-(SP)
011454 012746 007335 MOV #FMT18,-(SP)
011460 012746 000002 MOV #2,-(SP)
011464 010600 MOV SP,RO
011466 104017 EMT C$PNTF
011470 062706 000006 ADD #6,SP
18 011474 PRINTF #FMT18,#CMD3
011474 012746 004303 MOV #CMD3,-(SP)
011500 012746 007335 MOV #FMT18,-(SP)
011504 012746 000002 MOV #2,-(SP)
011510 010600 MOV SP,RO
011512 104017 EMT C$PNTF
```

19	011514	062706	000006	ADD	#6,SP
	011520			PRINTF	#FMT18,#CMD4
	011520	012746	004371	MOV	#CMD4,-(SP)
	011524	012746	007335	MOV	#FMT18,-(SP)
	011530	012746	000002	MOV	#2,-(SP)
	011534	010600		MOV	SP,RO
	011536	104017		EMT	C\$PNTF
20	011540	062706	000006	ADD	#6,SP
	011544			PRINTF	#FMT18,#CMD5
	011544	012746	004426	MOV	#CMD5,-(SP)
	011550	012746	007335	MOV	#FMT18,-(SP)
	011554	012746	000002	MOV	#2,-(SP)
	011560	010600		MOV	SP,RO
	011562	104017		EMT	C\$PNTF
21	011564	062706	000006	ADD	#6,SP
	011570			PRINTF	#FMT18,#CMD6
	011570	012746	004514	MOV	#CMD6,-(SP)
	011574	012746	007335	MOV	#FMT18,-(SP)
	011600	012746	000002	MOV	#2,-(SP)
	011604	010600		MOV	SP,RO
	011606	104017		EMT	C\$PNTF
22	011610	062706	000006	ADD	#6,SP
	011614			PRINTF	#FMT18,#CMD7
	011614	012746	004550	MOV	#CMD7,-(SP)
	011620	012746	007335	MOV	#FMT18,-(SP)
	011624	012746	000002	MOV	#2,-(SP)
	011630	010600		MOV	SP,RO
	011632	104017		EMT	C\$PNTF
23	011634	062706	000006	ADD	#6,SP
	011640			PRINTF	#MCRLF
	011640	012746	007552	MOV	#MCRLF,-(SP)
	011644	012746	000001	MOV	#1,-(SP)
	011650	010600		MOV	SP,RO
	011652	104017		EMT	C\$PNTF
	011654	062706	000004	ADD	#4,SP

```

1 011660 005037 013456      NXTCMD: CLR      FACNUM      ;CLEAR ENTRY COUNTER
2 011664 005037 013454      CLR      FLDNUM
3 011670 004537 024544      JSR      R5,GETSYS      ;UPDATE THE RUN TIME
4
5 011674      PRINTF #MCRLF
  011674 012746 007552      MOV      #MCRLF,-(SP)
  011700 012746 000001      MOV      #1,-(SP)
  011704 010600      MOV      SP,R0
  011706 104017      EMT      C$PNTF
  011710 062706 000004      ADD      #4,SP
6 011714      GMANID CMDDO,INPUT,D,7,1,7,NO
  011714 104043      EMT      C$GMAN
  011716 000406      BR       10001$
  011720 011766      .WORD   INPUT
  011722 000042      .WORD   T$CODE
  011724 004613      .WORD   CMDDO
  011726 000007      .WORD   7
  011730 000001      .WORD   T$LOLIM
  011732 000007      .WORD   T$HILIM
  10001$:
7
8 011734 013700 011766      MOV      INPUT,R0      ;GET THE CMD REQUEST TYPED
9 011740 006300      ASL      R0              ;SHIFT FOR PROPER INDEX INTO LIST
10 011742 000170 011746     JMP      @LIST(R0)      ;DO THE FUNCTION REQUESTED
11
12 011746 000000      LIST: .WORD   0          ;NOTHING FOR FUNCTION '0'
13 011750 011770      BSRPT                    ; 1 REPORT CONTENTS
14 011752 014102      BSADD                    ; 2 ADD AN ENTRY INTO 'FIELD' FILE
15 011754 015454      BSDEL                    ; 3 DELETE AN ENTRY FROM 'FIELD' FILE
16 011756 016752      BSVERIFY                 ; 4 READ THE PACK
17 011760 016300      BSWRITE                  ; 5 WRITE THE PACK
18 011762 021054      BSMAKE                   ; 6 MAKE A BAD SECTOR FILE
19 011764 011424      HI PMSG                  ; 7 PRINT THE COMMANDS AVAILABLE AGAIN
20
21 011766 000000      INPUT: .WORD   0        ;STORAGE FOR TYPED COMMAND

```

```

1 011770          STARS
2                ;*****
3                ;THIS IS THE ROUTINE TO REPORT THE CONTENTS OF THE BAD SECTOR FILE
4                ;FOR THE DRIVE SELECTED. 'BSFILE' CONTAINS THE ENTIRE LAST TRACK OF
5                ;THE CARTRIDGE (BAD SECTOR FILE). 1ST REPORT THE CARTRIDGE SERIAL
6                ;NUMBER FOLLOWED BY THE CONTENTS OF THE 'FACTORY' BAD SECTOR FILE
7 011770          STARS
8                ;*****
9 011770 004537 010460 BSRPT: JSR    R5,FIRST      ;SELECT A DRIVE
10 011774 000137 011660   JMP    NXTCMD      ;NONE AVAIL!
11 012000 000404         BR      BSRPTL
12 012002 004537 010470 BSRPTS: JSR    R5,SELDRV   ;SELECT THE NEXT UNIT
13 012006 000137 011660   JMP    NXTCMD      ;ALL DONE
14 012012 004537 023222 BSRPTL: JSR    R5,LOADED  ;SEE IF DRIVE READY FOR OPR
15 012016 005737 002320   TST    TEMPO      ;READY?
16 012022 001404         BEQ    IS          ;YUP
17 012024 004537 010736   JSR    R5,DRNRDY
18 012030 000137 012002   JMP    BSRPTS     ;SELECT THE NEXT UNIT
19
20 012034          IS:   PRINTF  #FMT19,#STARMSG
    012034 012746 002472   MOV    #STARMSG,-(SP)
    012040 012746 007342   MOV    #FMT19,-(SP)
    012044 012746 000002   MOV    #2,-(SP)
    012050 010600         MOV    SP,RO
    012052 104017         EMT    C$PNTF
    012054 062706 000006   ADD    #6,SP
21 012060 004537 021710   JSR    R5,RDBDSC   ;READ THE BAD SECTOR FILE
22 012064          PRINTF  #MCRLF
    012064 012746 007552   MOV    #MCRLF,-(SP)
    012070 012746 000001   MOV    #1,-(SP)
    012074 010600         MOV    SP,RO
    012076 104017         EMT    C$PNTF
    012100 062706 000004   ADD    #4,SP
23 012104 004537 010642   JSR    R5,DRVID
24 012110          PRINTF  #MCRLF
    012110 012746 007552   MOV    #MCRLF,-(SP)
    012114 012746 000001   MOV    #1,-(SP)
    012120 010600         MOV    SP,RO
    012122 104017         EMT    C$PNTF
    012124 062706 000004   ADD    #4,SP
25 012130 005037 013476   CLR    PSNFG      ;CLEAR THE PRINT FLAG FOR SER # MSG
26
27                ;HERE TO REPORT CONTENTS OF THE 'FACTORY' FILE
28 012134 004537 023026 BSRFAC: JSR    R5,RDFACT ;READ THE FACTORY FILE FROM BD SEC FILE
29 012140          PRINTF  #FMT18,#BSRM
    012140 012746 004635   MOV    #BSRM,-(SP)
    012144 012746 007335   MOV    #FMT18,-(SP)
    012150 012746 000002   MOV    #2,-(SP)
    012154 010600         MOV    SP,RO
    012156 104017         EMT    C$PNTF
    012160 062706 000006   ADD    #6,SP
30 012164 005037 013456   CLR    FACNUM     ;INIT THE FACTORY ENTRY COUNTER
31 012170 012737 000020   MOV    #16,,SECMAX ;LAST SECTOR PAIR IN FACTORY FILE
32 012176 005037 013470   CLR    SECNUM     ;POINT TO THE 1ST PAIR OF SECTORS
33 012202 005002         CLR    R2        ;CLEAR THE INDEX INTO THE BSFILE STORAGE
    
```

013466

34	012204	004537	013750	JSR	R5,BSFOK	:FIND A SECTOR TO USE IN FACTORS AREA	
35	012210	005737	013474	TST	BSFOKF	:SEE IF ERROR DETECTED	
36	012214	001441		BEQ	10\$	:JUMP IF OK	
37	012216			PRINTF	#FMT18,#NHWSEC		
	012216	012746	003400	MOV	#NHWSEC,-(SP)		
	012222	012746	007335	MOV	#FMT18,-(SP)		
	012226	012746	000002	MOV	#2,-(SP)		
	012232	010600		MCV	SP,RO		
	012234	104017		EMT	C\$PNTF		
	012236	062706	000006	ADD	#6,SP		
38	012242	004537	020614	JSR	R5,NEWBSF	:BUILD A NEW FILE	
39	012246	005737	002320	TST	TEMPO	:DID I?	
40	012252	001407		BEQ	1\$	:NO	
41	012254	013737	002320	MOV	TEMPO,NEWFAC	:SET THE FLAG	
42	012262	004537	020536	JSR	R5,COPY	:YES - COPY TO WHOLE FILE	
43	012266	004537	015122	JSR	R5,WRTBSF	:ASK IF TIME TO WRITE PACK	
44	012272			PRINTF	#FMT18,#HYPHEN		
	012272	012746	002540	MOV	#HYPHEN,-(SP)		
	012276	012746	007335	MOV	#FMT18,-(SP)		
	012302	012746	000002	MOV	#2,-(SP)		
	012306	010600		MCV	SP,RO		
	012310	104017		EMT	C\$PNTF		
	012312	062706	000006	ADD	#6,SP		
45	012316	000544		BR	BSRFLD	:DO THE FIELD REPORT	
46							
47							
48	012320					:START PROCESSING THE ENTRYS	
49	012320			10\$:			
	012320	013746	002224	PRINTF	#FMTSN,#CART,SERNM2,SERNM1		
	012324	013746	002226	MOV	SERNM1,-(SP)		
	012330	012746	002427	MOV	SERNM2,-(SP)		
	012334	012746	007420	MOV	#CART,-(SP)		
	012340	012746	000004	MOV	#FMTSN,-(SP)		
	012344	010600		MOV	#4,-(SP)		
	012346	104017		MOV	SP,RO		
	012350	062706	000012	EMT	C\$PNTF		
	012354	005237	013476	ADD	#12,SP		
50	012360	005037	002322	INC	PSNFG	:SET THE FLAG	
51	012364	016203	026442	CLR	TEMP1		
52	012370	005703		11\$:	MOV	BSFILE(R2),R3	:GET THE CYLINDER # FROM ENTRY
53	012372	100002		TST	R3	:SEE IF ITS OK TO USE	
54	012374	000137	013500	BPL	2\$	:OK	
55	012400	005237	013456	JMP	NOFACT	:WHOOOPS...ERROR	
56	012404	022737	000176	2\$:	INC	FACNUM	:COUNT THIS ENTRY
57	012412	001506		CMP	#126.,FACNUM	:END OF FILE LIMIT?	
58	012414	022737	000062	BEQ	BSRFLD	:YUP	
59	012422	001040		CMP	#50.,FACNUM	:TIME TO QUIT PRINTING?	
60	012424	005737	002322	BNE	21\$	:NO	
61	012430	001035		TST	TEMP1	:PRINTED ERROR MESSAGE YET?	
62	012432			BNE	21\$	:YUP	
63	012432	012746	006150	PRINTF	#FMT19,#OVRMAX	:TELL OPR OVER LIMIT	
	012436	012746	007342	MOV	#OVRMAX,-(SP)		
	012442	012746	000002	MOV	#FMT19,-(SP)		
	012446	010600		MOV	#2,-(SP)		
	012450	104017		MOV	SP,RO		
	012452	062706	000006	EMT	C\$PNTF		
64	012456			ADD	#6,SP		
				PRINTF	#MCRLF		

```
012456 012746 007552      MOV      #MCRLF,-(SP)
012462 012746 000001      MOV      #1,-(SP)
012466 010600      MOV      SP,R0
012470 104017      EMT      C$PNTF
012472 062706 000004      ADD      #4,SP
65 012476      GMANIL   TILLEND,TEMPO,177777,NO
012476 104043      EMT      C$GMAN
012500 000404      BR       10002$
012502 002320      .WORD   TEMPO
012504 000120      .WORD   T$CODE
012506 005770      .WORD   TILLEND
012510 177777      .WORD   177777
012512      10002$:
66 012512 005737 002320      TST      TEMPO          ;NO?
67 012516 001444      BEQ      BSRFLD         ;QUIT PRINTING ENTRIES
68 012520 005237 002322      INC      TEMPI         ;SET THE PRINT ERROR FLAG
69 012524 010337 013460      21$:  MOV     R3,BSFCYL     ;SAVE THE CYLINDER NUMBER
70 012530 005722      TST      (R2)+         ;POINT TO HEAD & SECTOR ENTRY
71 012532 016203 026442      MOV     BSFILE(R2),R3  ;GET IT
72 012536 110337 013462      MOV     R3,BSFSEC      ;SAVE THE SECTOR NUMBER
73 012542 000303      SWAB    R3             ;PUT THE HEAD # IN LOW BYTE
74 012544 110337 013464      MOV     R3,BSFHD      ;SAVE THE HEAD NUMBER
75 012550 005722      TST      (R2)+         ;POINT TO THE NEXT ENTRY
76 012552      PRINTF  #FMTCSH,FACNUM,#CMMSG,BSFCYL,#SMSG,BSFSEC,#HMSG,BSFHD
012552 013746 013464      MOV     BSFHD,-(SP)
012556 012746 002462      MOV     #HMSG,-(SP)
012562 013746 013462      MOV     BSFSEC,-(SP)
012566 012746 003101      MOV     #SMSG,-(SP)
012572 013746 013460      MOV     BSFCYL,-(SP)
012576 012746 002447      MOV     #CMMSG,-(SP)
012602 013746 013456      MOV     FACNUM,-(SP)
012606 012746 007450      MOV     #FMTCSH,-(SP)
012612 012746 000010      MOV     #10,-(SP)
012616 010600      MOV     SP,R0
012620 104017      EMT      C$PNTF
012622 062706 000022      ADD     #22,SP
77 012626 000656      BR       11$          ;PROCESS THE NEXT ENTRY
78
79      ;HERE TO REPORT THE CONTENTS OF THE 'FIELD' FILE
80 012630 004537 023066      BSRFLD: JSR     R5,RDFIELD ;GET THE FIELD BD SEC FILE
81 012634 005002      CLR     R2             ;POINT TO THE 1ST SECTOR OF THE 'FIELD' FILE
82 012636 012737 000020 013466      MOV     #16.,SECMAX    ;SET THE LAST USABLE SECTOR NUMBER
83 012644 005037 013470      CLR     SECNUM        ;POINT TO THE 1ST SECTOR IN FIELD FILE
84 012650 005037 013454      CLR     FLDNUM        ;CLEAR THE FIELD ENTRY COUNTER
85 012654 005037 013476      CLR     PSNFG         ;CLEAR THE PRINT FLAG FOR SERIAL #
86 012660      PRINTF  #FMT18,#BSRF
012660 012746 004705      MOV     #BSRF,-(SP)
012664 012746 007335      MOV     #FMT18,-(SP)
012670 012746 000002      MOV     #2,-(SP)
012674 010600      MOV     SP,R0
012676 104017      EMT      C$PNTF
012700 062706 000006      ADD     #6,SP
87 012704 004537 013750      JSR     R5,BSFOK       ;FIND A SECTOR TO USE IN THE FIELD AREA
88 012710 005737 013474      TST     BSFOK         ;ANY ERROR DETECTED?
89 012714 001436      BEQ     10$          ;JUMP IF OK
90 012716      PRINTF  #FMT18,#NSWSEC
012716 012746 003453      MOV     #NSWSEC,-(SP)
```

```
012722 012746 007335      MOV      #FMT18,-(SP)
012726 012746 000002      MOV      #2,-(SP)
012732 010600      MOV      SP,R0
012734 104017      EMT      C$PNTF
012736 062706 000006      ADD      #6,SP
91 012742 004537 020614      JSR      R5,NEWBSF      ;BUILD A NEW FILE
92 012746 005737 002320      TST      TEMPO          ;DID I?
93 012752 001404      BEQ      1$            ;NO
94 012754 004537 020536      JSR      R5,COPY        ;YES - COPY TO WHOLE FILE
95 012760 004537 015122      JSR      R5,WRIBSF      ;ASK IF TIME TO WRITE FILE ON PACK
96 012764      1$: PRINTF #FMT18,#HYPHEN
012764 012746 002540      MOV      #HYPHEN,-(SP)
012770 012746 007335      MOV      #FMT18,-(SP)
012774 012746 000002      MOV      #2,-(SP)
013000 010600      MOV      SP,R0
013002 104017      EMT      C$PNTF
013004 062706 000006      ADD      #6,SP
97 013010 000544      BR       BSRTOT        ;PRINT THE TOTALS FOUND
98
99
100 013012      ;HERE TO PROCESS ENTRYS FROM THE FIELD FILE
101 013012      10$: PRINTF #FMTSN,#CART,SERNM2,SERNM1
013012 013746 002224      MOV      SERNM1,-(SP)
013016 013746 002226      MOV      SERNM2,-(SP)
013022 012746 002427      MOV      #CART,-(SP)
013026 012746 007420      MOV      #FMTSN,-(SP)
013032 012746 000004      MOV      #4,-(SP)
013036 010600      MOV      SP,R0
013040 104017      EMT      C$PNTF
013042 062706 000012      ADD      #12,SP
102 013046 005237 013476      INC      PSNFG          ;SET THE PRINT FLAG
103 013052 005037 002322      CLR      TEMP1
104 013056 016203 026442      11$: MOV      BSFILE(R2),R3      ;GET THE CYLINDER # FROM ENTRY
105 013062 005703      TST      R3            ;SEE IF ITS OK TO USE
106 013064 100002      BPL      2$            ;OK
107 013066 000137 013566      JMP      NOFIELD       ;ERROR.
108 013072 005237 013454      2$: INC      FLDNUM      ;COUNT THIS ENTRY
109 013076 022737 000176 013454      CMP      #126.,FLDNUM   ;END OF FIELD ENTRY LIMIT?
110 013104 001506      BEQ      BSRTOT        ;YUP
111 013106 022737 000062 013454      CMP      #50.,FLDNUM   ;TIME TO QUIT PRINTING?
112 013114 001040      BNE      21$          ;NO
113 013116 005737 002322      TST      TEMP1        ;PRINT THE ERROR MESSAGE?
114 013122 001035      BNE      21$          ;NO
115 013124      PRINTF #FMT19,#OVRMAX  ;YES - TELL OPR
013124 012746 006150      MOV      #OVRMAX,-(SP)
013130 012746 007342      MOV      #FMT19,-(SP)
013134 012746 000002      MOV      #2,-(SP)
013140 010600      MOV      SP,R0
013142 104017      EMT      C$PNTF
013144 062706 000006      ADD      #6,SP
116 013150      PRINTF #MCRLF
013150 012746 007552      MOV      #MCRLF,-(SP)
013154 012746 000001      MOV      #1,-(SP)
013160 010600      MOV      SP,R0
013162 104017      EMT      C$PNTF
013164 062706 000004      ADD      #4,SP
117 013170      GMANIL TILLEND,TEMPO,177777,NO
```

```

013170 104043      EMT      CS&GMAN
013172 000404      BR       10003$
013174 002320      .WORD   TEMPO
013176 000120      .WORD   T&CODE
013200 005770      .WORD   TILLEND
013202 177777      .WORD   177777
013204
118 013204 005737 002320      10003$: TST      TEMPO          :QUIT?
119 013210 001444      BEQ      BSRTOT        :YUP
120 013212 005237 002322      INC      TEMP1         :SET THE PRINT FLAG
121 013216 010337 013460      21$:  MOV     R3,BSFCYL  :SAVE THE CYLINDER NUMBER
122 013222 005722      TST      (R2)+         :POINT TO HEAD & SECTOR ENTRY
123 013224 016203 026442      MOV     BSFILE(R2),R3  :GET IT
124 013230 110337 013462      MOV     R3,BSFSEC     :SAVE THE SECTOR NUMBER
125 013234 000303      SWAB    R3            :PUT THE HEAD # IN LOW BYTE
126 013236 110337 013464      MOV     R3,BSFHD     :SAVE THE HEAD NUMBER
127 013242 005722      TST      (R2)+         :POINT TO THE NEXT ENTRY
128 013244      PRINTF  #FMTC&SH,FLDNUM,#CMSG,BSFCYL,#SMSG,BSFSEC,#HMSG,BSFHD
013244 013746 013464      MOV     BSFHD,-(SP)
013250 012746 002462      MOV     #HMSG,-(SP)
013254 013746 013462      MOV     BSFSEC,-(SP)
013260 012746 003101      MOV     #SMSG,-(SP)
013264 013746 013460      MOV     BSFCYL,-(SP)
013270 012746 002447      MOV     #CMSG,-(SP)
013274 013746 013454      MOV     FLDNUM,-(SP)
013300 012746 007450      MOV     #FMTC&SH,-(SP)
013304 012746 000010      MOV     #10,-(SP)
013310 010600      MOV     SP,RO
013312 104017      EMT      C&PNTF
013314 062706 000022      ADD     #22,SP
129 013320 000656      BR       11$          :PROCESS THE NEXT ENTRY
130
131
132
133
133 013322      BSRTOT: PRINTF #FM&TB,#TMMSG,FACNUM
013322 013746 013456      MOV     FACNUM,-(SP)
013326 012746 002636      MOV     #TMMSG,-(SP)
013332 012746 007435      MOV     #FM&TB,-(SP)
013336 012746 000003      MOV     #3,-(SP)
013342 010600      MOV     SP,RO
013344 104017      EMT      C&PNTF
013346 062706 000010      ADD     #10,SP
134 013352      PPINTF #FM&TB,#TFMSG,FLDNUM
013352 013746 013454      MOV     FLDNUM,-(SP)
013356 012746 002606      MOV     #TFMSG,-(SP)
013362 012746 007435      MOV     #FM&TB,-(SP)
013366 012746 000003      MOV     #3,-(SP)
013372 010600      MOV     SP,RO
013374 104017      EMT      C&PNTF
013376 062706 000010      ADD     #10,SP
135 013402      PRINTF #FM&TB,#STARMSG
013402 012746 002472      MOV     #STARMSG,-(SP)
013406 012746 007335      MOV     #FM&TB,-(SP)
013412 012746 000002      MOV     #2,-(SP)
013416 010600      MOV     SP,RO
013420 104017      EMT      C&PNTF
013422 062706 000006      ADD     #6,SP

```



```

136 013426          PRINTF  #MCRLF
      013426 012746 007552  MOV    #MCRLF,-(SP)
      013432 012746 000601  MOV    #1,-(SP)
      013436 010600          MOV    SP,R0
      013440 104017          EMT    C$PNTF
      013442 062706 000004  ADD    #4,SP
137 013446 000137 012002  JMP    BSRPTS          ;SELECT NEXT UNIT
138
139          ;HERE IS THE STORAGE FOR THIS ROUTINE
140
141 013452 000000  NOSNUM: .WORD 0          ;HAVE SERIAL # FLAG
142 013454 000000  FLDNUM: .WORD 0          ;NUMBER OF CURRENT FIELD ENTRY
143 013456 000000  FACNUM: .WORD 0          ;NUMBER OF THE CURRENT FACTORY ENTRY
144 013460 000000  BSFCYL: .WORD 0          ;CURRENT CYLINDER FROM ENTRY IN PROCESS
145 013462 000000  BSFSEC: .WORD 0          ;CURRENT SECTOR FROM ENTRY
146 013464 000000  BSFHD:  .WORD 0          ;CURRENT SURFACE (HEAD) FROM ENTRY IN PROCESS
147 013466 000000  SECMAX: .WORD 0          ;LAST USUABLE SECTOR NUMBER IN SELECTED SECTION
148 013470 000000  SECNUM: .WORD 0          ;CURRENT SECTOR BEING USED TO EXTRACT ENTRIES
149 013472 000000  SECOLD: .WORD 0          ;START ADDR OF THE 'FOUND' SECTOR IN RAD SEC FILE
150 013474 000000  BSFOKF: .WORD 0          ;ERROR DETECT FLAG
151 013476 000000  PSNFG:  .WORD 0          ;PRINT FLAG FOR SERIAL #
152
153          ;HERE IF AT END OF THE FACTORY FILE
154
155 013500 005737 013456  NOFACT: TST    FACNUM          ;WAS ANY ENTRY DETECTED?
156 013504 001014          BNE    1$                    ;YUP
157 013506 005037 013456  CLR    FACNUM          ;CLEAR THE ENTRY COUNTER FOR FACTORY SECTORS
158 013512          PRINTF  #FMT18,#HWSEC
      013512 012746 003351  MOV    #HWSEC,-(SP)
      013516 012746 007335  MOV    #FMT18,-(SP)
      013522 012746 000002  MOV    #2,-(SP)
      013526 010600          MOV    SP,R0
      013530 104017          EMT    C$PNTF
      013532 062706 000006  ADD    #6,SP
159 013536          1$: PRINTF  #FMT18,#HYPHEN
      013536 012746 002540  MOV    #HYPHEN,-(SP)
      013542 012746 007335  MOV    #FMT18,-(SP)
      013546 012746 000002  MOV    #2,-(SP)
      013552 010600          MOV    SP,R0
      013554 104017          EMT    C$PNTF
      013556 062706 000006  ADD    #6,SP
160 013562 000137 012630  JMP    BSRFLD          ;DO THE FIELD SECTION
161
162          ;HERE IF AT THE END OF THE FIELD FILE
163
164 013566 005737 013454  NOFIELD: TST   FLDNUM          ;ANY FIELD ENTRIES?
165 013572 001014          BNE    1$                    ;YUP
166 013574 005037 013454  CLR    FLDNUM          ;NO - CLEAR THE ENTRY COUNTER
167 013600          PRINTF  #FMT18,#SWSEC
      013600 012746 003426  MOV    #SWSEC,-(SP)
      013604 012746 007335  MOV    #FMT18,-(SP)
      013610 012746 000002  MOV    #2,-(SP)
      013614 010600          MOV    SP,R0
      013616 104017          EMT    C$PNTF
      013620 062706 000006  ADD    #6,SP
168 013624          1$: PRINTF  #FMT18,#HYPHEN
      013624 012746 002540  MOV    #HYPHEN,-(SP)

```

```

013630 012746 007335      MOV    #FMT18,-(SP)
013634 012746 000002      MOV    #2,-(SP)
013640 010600      MOV    SP,R0
013642 104017      EMT    C$PNTF
013644 062706 000006      ADD    #6,SP
169 013650 000137 013322      JMP    BSRTOT          ;DO THE TOTALS
170
171      ;HERE IF NO SERIAL NUMBER OR 2 0'S NOT DETECTED IN 1ST 4 WORDS OF SECTOR
172
173 013654 062737 000004 013470  NOMSEC: ADD    #4,SECNUM      ;UPDATE THE SECTOR NUMBER
174 013662 023737 013470 013466      CMP    SECNUM,SECMAX  ;AT THE END OF SECTION?
175 013670 101416      BLOS   1$             ;BRANCH IF OK
176 013672      PRINTF #FMT18,#BSEND
      MOV    #BSEND,-(SP)
      MOV    #FMT18,-(SP)
      MOV    #2,-(SP)
      MOV    SP,R0
      EMT    C$PNTF
      ADD    #6,SP
177 013716 052737 177777 013474      BIS    #177777,BSFOKF ;SET THE ERROR FLAG
178 013724 000465      BR     BSFOKX        ;EXIT THE SECTOR FIND ROUTINE WITH ERROR SET
179 013726 012702 000400      1$:  MOV    #128.*2,R2
180 013732 013703 013470      MOV    SECNUM,R3
181 013736 062702 000400      2$:  ADD    #128.*2,R2   ;UP THE OFFSET BY 1 SECTOR
182 013742 005303      DEC    R3            ;DECREMENT THE COUNTER - MULTIPLY
183 013744 001374      BNE   2$            ;BACK FOR ANOTHER IF NOT DONE
184 013746 000400      BR     BSFOK        ;SEE IF THIS SECTOR IS OK

```

```

1          ;HERE IS THE SECTOR FIND ROUTINE FOR THE BAD SECTOR FILE
2
3 013750 005037 013474 BSFOK: CLR BSFOKF ;CLEAR THE ERROR FLAG
4 013754 012737 026442 013472 MOV #BSFILE,SECOLD ;GET BASIC ADDRESS
5 013762 060237 013472 ADD R?,SECOLD ;ADD IN THE OFFSET
6 013766 005762 026442 TST BSFILE(R2) ;SEE IF ANY SERIAL NUMBER
7 013772 100730 BMI NOMSEC ;IF -1 JUMP TO ERROR SERVICE
8 013774 001006 BNE 1$ ;IF >0 THEN HAVE A NUMBER
9 013776 005762 026444 TST BSFILE+2(R2) ;ANY SER. NUM IN WORD ??
10 014002 001003 BNF 1$ ;BR IF OK
11 014004 052737 177777 013452 BIS #177777,NOSNUM ;SET THE NO SERIAL NUMBER FLAG
12 014012 022762 177777 027040 1$: CMP #-1,BSFILE+254.(R2) ;END OF SECTOR OK ALSO?
13 014020 001315 BNE NOMSEC ;NO - NO MATCH HERE!
14 014022 022762 177777 027042 CMP #-1,BSFILE+256.(R2) ;END OK HERE TOO?
15 014030 001311 BNE NOMSEC ;NO - DON'T HAVE A BAD SECT FILE YET!
16 014032 016237 026442 002224 MOV BSFILE(R2),SERNM1 ;SAVE LOW ORDER OF SERIAL #
17 014040 005722 TST (R2)+ ;POINT TO HIGH ORDER OF SERIAL #
18 014042 005762 026442 TST BSFILE(R2) ;SEE IF LEGAL
19 014046 100702 BMI NOMSEC ;NO - ERROR
20 014050 016237 026442 002226 MOV BSFILE(R2),SERNM2 ;SAVE HIGH ORDER OF SERIAL #
21 014056 005722 TST (R2)+ ;POINT TO A BLANK ENTRY
22 014060 005762 026442 TST BSFILE(R2) ;SEE IF LEGAL (=0)
23 014064 001273 BNE NOMSEC ;NO - ERROR
24 014066 005722 TST (R2)+ ;POINT TO LAST CHECK ENTRY SPOT
25 014070 005762 026442 TST BSFILE(R2) ;SEE IF LEGAL (=0)
26 014074 001267 BNE NOMSEC ;NO - ERROR
27 014076 005722 TST (R2)+ ;POINT TO 1ST VALID ENTRY IN SECTOR
28 014100 000205 BSFOKX: RTS R5 ;EXIT - R2 POINTS TO THE OFFSET VALUE
29 ;AND SECNUM = CURRENT SECTOR IN FILE

```

```

1 014102          STARS
2                ;:*****
3 014102          ;HEREF IS THE ROUTINE TO ADD AN ENTRY INTO THE 'FIELD' BAD SECTOR FILE
4                STARS
5                ;:*****
5 014102          BSADD: PRINTF  #FMT19,#ABSMSG
014102 012746 005024      MOV      #ABSMSG,-(SP)
014106 012746 007342      MOV      #FMT19,-(SP)
014112 012746 000002      MOV      #2,-(SP)
014116 010600      MOV      SP,RO
014120 104017      EMT      C$PNTF
014122 062706 000006      ADD      #6,SP
6 014126 004537 010460      JSR      R5,FIRST      ;SELECT 1ST UNIT
7 014132 000137 011660      JMP      NXTCMD      ;NONE AVAIL.!
8 014136 000404      BR
9 014140 004537 010470      BSADDS: JSR      R5,SELDRV  ;SELECT THE NEXT UNIT
10 014144 000137 011660      JMP      NXTCMD      ;ALL DONE
11 014150 004537 023222      BSADDL: JSR      R5,LOADED ;DRV READY?
12 014154 005737 002320      TST     TEMPO        ;WELL?
13 014160 001403      BEQ     1$           ;YES
14 014162 004537 010736      JSR      R5,DRNRDY
15 014166 000764      BR      BSADDS      ;SELECT THE NEXT UNIT
16
17 014170 004537 010642      1$:     JSR      R5,DRVID  ;TELL OPR WHAT DRIVE
18 014174 004537 021710      JSR      R5,RDBDSC    ;GET A FRESH COPY OF THE BAD SECTOR FILE
19 014200 005737 002372      TST     ACCESS       ;ALLOWED TO DO IT?
20 014204 001414      BEQ     BSATD        ;YUP
21 014206          PRINTF  #FMT18,#DENIED
014206 012746 003747      MOV      #DENIED,-(SP)
014212 012746 007335      MOV      #FMT18,-(SP)
014216 012746 000002      MOV      #2,-(SP)
014222 010600      MOV      SP,RO
014224 104017      EMT      C$PNTF
014226 062706 000006      ADD      #6,SP
22 014232 000137 011660      JMP      NXTCMD      ;TELL OPR NOT ALLOWED & EXIT
23
24 014236          BSATD: GMANIL  THISDRV,TEMPO,1,NO
014236 104043      EMT      C$GMAN
014240 000404      BR      10004$
014242 002320      .WORD   TEMPO
014244 000120      .WORD   T$CODE
014246 006575      .WORD   THISDRV
014250 000001      .WORD   1
014252
25 014252 005737 002320      10004$: TST     TEMPO
26 014256 001730      BEQ     BSADDS      ;BRANCH IF NOT TO USE THIS DRIVE
27
28 014260 005737 013452      BSASN:  TST     NOSNUM  ;SEE IF NEED A SFRIAL NUMBER
29 014264 001470      BEQ     GETCYL      ;JUMP IF HAVE A NUMBER
30 014266          PRINTF  #MCRLF
014266 012746 007552      MOV      #MCRLF,-(SP)
014272 012746 000001      MOV      #1,-(SP)
014276 010600      MOV      SP,RO
014300 104017      EMT      C$PNTF
014302 062706 000004      ADD      #4,SP
31 014306          GETSN:  CMANIL  ABSER,TEMPO,1,NO

```

```

014306 104043      EMT      C$GMAN
014310 000404      BR       10005$
014312 002320      .WORD   TEMPO
014314 000120      .WORD   T$CODE
014316 005220      .WORD   ABSSER
014320 000001      .WORD   1
014322
32 014322 005737 002320 10005$: TST      TEMPO          ;SEE IF YES (=1)
33 014326 001447      BEQ      GETCYL        ;BRANCH IF NO
34 014330      GMANID  ABSSNL,SN1,0,77777,1,77777,NO
014330 104043      EMT      C$GMAN
014332 000406      BR       10006$
014334 002364      .WORD   SN1
014336 000022      .WORD   T$CODE
014340 005272      .WORD   ABSSNL
014342 077777      .WORD   77777
014344 000001      .WORD   T$LOLIM
014346 077777      .WORD   T$HILIM
014350
35 014350      10006$: GMANID  ABSSNH,SN2,0,77777,0,77777,NO
014350 104043      EMT      C$GMAN
014352 000406      BR       10007$
014354 002366      .WORD   SN2
014356 000022      .WORD   T$CODE
014360 005331      .WORD   ABSSNH
014362 077777      .WORD   77777
014364 000000      .WORD   T$LOLIM
014366 077777      .WORD   T$HILIM
014370
36 014370      10007$: PRINTF  #FMTSN,#CART,SN2,SN1
014370 013746 002364      MOV     SN1,-(SP)
014374 013746 002366      MOV     SN2,-(SP)
014400 012746 002427      MOV     #CART,-(SP)
014404 012746 007420      MOV     #FMTSN,-(SP)
014410 012746 000004      MOV     #4,-(SP)
014414 010600      MOV     SP,RO
014416 104017      EMT      C$PNTF
014420 062706 000012      ADD     #12,SP
37 014424      GMANID  VALSN,TEMPO,177777,NO
014424 104043      EMT      C$GMAN
014426 000404      BR       10010$
014430 002320      .WORD   TEMPO
014432 000120      .WORD   T$CODE
014434 005660      .WORD   VALSN
014436 177777      .WORD   177777
014440
38 014440 005737 002320 10010$: TST      TEMPO          ;JMP IF NO
39 014444 001720      BEQ      GETSN         ;NO VALID SERIAL NUMBER - ASK AGAIN
40
41 014446      GETCYL: GMANID  ABSCYL,BSFCYL,D,777,0,511.,NO
014446 104043      EMT      C$GMAN
014450 000406      BR       10011$
014452 013460      .WORD   BSFCYL
014454 000042      .WORD   T$CODE
014456 005114      .WORD   ABSCYL
014460 000777      .WORD   777
014462 000000      .WORD   T$LOLIM

```

```

014464 000777          .WORD  T$HILIM
014465
42
43
44 014466 022737 000001 002246      CMP      #1,TDR          :RL01=1
45 014474 001017          BNE      GETSEC         :SKIP CHECK IF RL02
46 014476 022737 000377 013460      CMP      #255.,BSFCYL   :VALID RL01 CYLINDER?
47 014504 103013          BHIS     GETSEC         :YUP
48 014506          PRINTF  #FMT19,#RL1CLM   :NO - TELL OPR
014506 012746 005627      MOV      #RL1CLM,-(SP)
014512 012746 007342      MOV      #FMT19,-(SP)
014516 012746 000002      MOV      #2,-(SP)
014522 010600          MOV      SP,RO
014524 104017          EMT      C$PNTF
014526 062706 000006      ADD      #6,SP
49 014532 000745          BR       GETCYL         :GET CYL AGAIN
50 014534          GETSEC: GMANID  ABSSEC,BSFSEC,D,77,0,39.,NO
014534 104043          EMT      C$GMAN
014536 000406          BR       10012$
014540 013462          .WORD   BSFSEC
014542 000042          .WORD   T$CODE
014544 005146          .WORD   ABSSEC
014546 000077          .WORD   77
014550 000000          .WORD   T$L0LIM
014552 000047          .WORD   T$HILIM
014554
51 014554          10012$: GMANID  ABSHD,BSFHD,D,3,0,1,NO
014554 104043          EMT      C$GMAN
014556 000406          BR       10013$
014560 013464          .WORD   BSFHD
014562 000042          .WORD   T$CODE
014564 005175          .WORD   ABSHD
014566 000003          .WORD   3
014570 000000          .WORD   T$L0LIM
014572 000001          .WORD   T$HILIM
014574
52
53          10013$:
54          :CHECK TO SEE IF THE NEW ENTRY ALREADY EXISTS IN THE FILE
55 014574 013700 013460      MOV      BSFCYL,RO      :GET THE CYL TYPED
56 014600 000300          SWAB     RO
57 014602 000241          CLC
58 014604 006000          ROR      RO             :CLEAR THE 'C' BIT
59 014606 103002          BCC      1$            :BR IF DON'T NEED THE EXTRA BIT
60 014610 052700 100000      BIS      #BIT15,RO      :ADD HIGH ORDER BIT IN CYL #
61 014614 053700 013462      1$:  BIS      BSFSEC,RO    :ADD IN THE SECTOR NUMBER
62 014620 005737 013464      TST      BSFHD         :ON HEAD 0??
63 014624 001402          BEQ      ACKENT        :BR IF HEAD 0
64 014626 052700 000100      BIS      #100,RO       :MAKE IT HEAD #1
65 014632 010037 002314      ACKENT: MOV      RO,CHKSEC :SAVE FOR THE CHECK
66 014636 004537 025252      JSR      R5,CKBDSC     :CHECK TO SEE IF ALREADY IN BAD SECT FILE
67 014642 005737 002312      TST      HDRFND        :HEADER IN FILE?
68 014646 001414          BEQ      1$            :BR IF NOT IN FILE
69 014650          PRINTF  #FMT18,#EXISTS :TELL OPR ENTRY IN FILE NOW
014650 012746 004013      MOV      #EXISTS,-(SP)
014654 012746 007335      MOV      #FMT18,-(SP)
014660 012746 000002      MOV      #2,-(SP)

```

```

014664 010600          MOV     SP,R0
014666 104017          EMT     C$PNTF
014670 062706 000006   ADD     #6,SP
70 014674 000137 014140  JMP     BSADDS          ;SELECT THE NEXT UNIT
71
72
73 ;WE NOW HAVE THE NEW ENTRY DATA NEEDED TO GENERATE A BAD SECTOR FILE
74 ;ENTRY...FIND A FREE SPOT IN THE BAD SECTOR FILE 'FIELD' AREA FOR THE
75 ;ADDITION THEN UPDATE THE BAD SECTOR FILE ITSELF (MEDIA).
76 014700          1$: PRINTF #FMT18,#NEWENT          ;TELL OPR ITS A NEW ENTRY
014700 012746 006103   MOV     #NEWENT,-(SP)
014704 012746 007335   MOV     #FMT18,-(SP)
014710 012746 000002   MOV     #2,-(SP)
014714 010600          MOV     SP,R0
014716 104017          EMT     C$PNTF
014720 062706 000006   ADD     #6,SP
77 014724 005037 013470   CLR     SECNUM          ;START THE SEARCH AT SECTOR 20.
78 014730 005002          CLR     R2              ;POINT TO THE STARTING AREA IN BSFILE
79 014732 012737 000020 013466   MOV     #16.,SECMAX     ;THIS IS THE LAST AVAIL. SECTOR PAIR
80 014740 005037 013454          CLR     FLDNUM          ;START AT ENTRY #1
81 014744 004537 013750          JSR     R5,BSFOK        ;FIND A 'FIELD' SECTOR AREA
82 014750 005737 013474          TST     BSFOKF          ;ON A SECTOR?
83 014754 001421          BEQ     2$              ;YUP
84 014756          PRINTF #FMT18,#NSWSEC          ;NO - TELL OPR
014756 012746 003453   MOV     #NSWSEC,-(SP)
014762 012746 007335   MOV     #FMT18,-(SP)
014766 012746 000002   MOV     #2,-(SP)
014772 010600          MOV     SP,R0
014774 104017          EMT     C$PNTF
014776 062706 000006   ADD     #6,SP
85 015002 004537 020614          JSR     R5,NEWBSF      ;ASK OPR IF TIME TO MAKE A 'FIELD' BSF
86 015006 005737 002320          TST     TEMPO          ;WAS A FILE BUILT?
87 015012 001002          BNE     2$              ;YES - CONTINUE
88 015014 000137 014140  JMP     BSADDS          ;SELECT THE NEXT UNIT
89
90 015020 005737 013452          2$: TST     NOSNUM          ;PACK HAVE A SERIAL # ??
91 015024 001406          BEQ     21$             ;YUP
92 015026 013777 002364 176436   MOV     SN1,@SECOLD     ;NO - SAVE LOW 5 #
93 015034 013777 002366 176432   MOV     SN2,@SECOLD+2   ;SAVE HIGH 5 #
94 015042 005237 013454          21$: INC     FLDNUM          ;COUNT THIS ENTRY TO BE TESTED
95 015046 005762 026442          TST     BSFILE(R2)     ;SEE IF A FREE SLOT
96 015052 100403          BMI     3$              ;I FOUND IT...
97 015054 062702 000004          ADD     #4,R2          ;POINT TO THE NEXT ENTRY
98 015060 000757          BR      2$              ;AND TRY THE NEXT SLOT
99
100 015062 013762 013460 026442  3$: MOV     BSFCYL,BSFILE(R2) ;INSERT THE CYLINDER NUMBER
101 015070 013703 013464          MOV     BSFHD,R3       ;GET THE SELETED HEAD
102 015074 000303          SWAB    R3              ;SWAP BYTES TO POSITION THE HD BIT
103 015076 063703 013462          ADD     BSFSEC,R3      ;R3 NOW HAS COMPLETE HD & SEC ENTRY
104 015102 010362 026444          MOV     R3,BSFILE+2(R2) ;INSERT 2ND HALF OF ENTRY
105
106 ;INSERT THE ENTRY INTO REST OF THE 'FIELD' FILE
107
108 015106 004537 020536          4$: JSR     R5,COPY        ;UPDATE THE REST OF THE FILE
109 015112 004537 015122          JSR     R5,WRTBSF      ;ASK OPR IF TIME TO UPDATE PACK
110 015116 000137 014140          JMP     BSADDS          ;SELECT THE NEXT UNIT

```

```

1 015122          STARS
2                ;:*****
3                ;HERE IS WHERE THE 'FIELD' FILE IS WRITTEN ON THE PACK
4                ;THE OPERATOR IS ASKED IF ITS TIME TO UPDATE THE PACK...IF NOT, THEN
5 015122          ;THIS CCDE IS ABORTED.
6                STARS
7                ;:*****
7 015122 010146   WRTBSF: MOV     R1,-(SP)           ;SAVE R1
8 015124          PRINTF  #MCRLF
015124 012746 007552  MOV     #MCRLF,-(SP)
015130 012746 000001  MOV     #1,-(SP)
015134 010600      MOV     SP,R0
015136 104017      EMT     C$PNTF
015140 062706 000004  ADD     #4,SP
9 015144          GMANIL  DOWRT,TEMPO,177777,NO
015144 104043      EMT     C$GMAN
015146 000404      BR      10014$
015150 002320      .WORD  TEMPO
015152 000120      .WORD  T$CODE
015154 005371      .WORD  DOWRT
015156 177777      .WORD  177777
015160          10014$:
10 015160 005737 002320  TST     TEMPO           ;YES? (=1)
11 015164 001531      BEQ     3$             ;EXIT IF 'NO'
12 015166 004537 023222  JSR     R5,LOADED      ;READY?
13 015172 005737 002320  TST     TEMPO           ;WELL?
14 015176 001413      BEQ     11$           ;YES
15 015200          PRINTF  #FMT18,#NOTRDY      ;NO
015200 012746 006401  MOV     #NOTRDY,-(SP)
015204 012746 007335  MOV     #FMT18,-(SP)
015210 012746 000002  MOV     #2,-(SP)
015214 010600      MOV     SP,R0
015216 104017      EMT     C$PNTF
015220 062706 000006  ADD     #6,SP
16 015224 000511      BR      3$
17 015226 004537 023342  JSR     R5,GETDST
18 015232 032701 020000  BIT     #WL,R1         ;DRIVE WRITE LOCKED?
19 015236 001413      BEQ     12$           ;NO
20 015240          PRINTF  #FMT18,#WRTLCK      ;YES
015240 012746 003277  MOV     #WRTLCK,-(SP)
015244 012746 007335  MOV     #FMT18,-(SP)
015250 012746 000002  MOV     #2,-(SP)
015254 010600      MOV     SP,R0
015256 104017      EMT     C$PNTF
015260 062706 000006  ADD     #6,SP
21 015264 000471      BR      3$
22 015266 012737 173000 002176 12$:  MOV     #-20,*128.,BMP  ;SETUP THE WORDCOUNT
23 015274 012737 077724 002174      MOV     #77724,BDA    ;START THE WRITE AT SECTOR 20. (RL01)
24 015302 022737 000001 002246      CMP     #1,TDR        ;RL02?
25 015310 001403      BEQ     1$           ;JUMP IF RL01
26 015312 012737 177724 002174      MOV     #177724,BDA   ;START AT SECTOR 20. FOR RL02
27 015320 012737 000012 002210 1$:  MOV     #WRITE,FUNC   ;LOAD THE FUNCTION
28 015326 005737 002230      TST     NEWFAC        ;MAKING A DUMMY 'FACTORY' FILE?
29 015332 001405      BEQ     13$           ;NO
30 015334 042737 000077 002174      BIC     #77,BDA       ;YUP - START AT SECTOR 00
31 015342 005037 002230      CLR     NEWFAC       ;CLEAR THE FLAG ALSO

```



```
32 015346 005237 002250      13$: INC WRIPG ;SET THE WRITE IN PROGRESS FLAG
33 015352 004537 021570      JSR R5,LDFUNC ;DO THE WRITE OF THE UPDATED 'FIELD' FILE
34 015356 004537 023264      JSR R5,WTRDY ;WAIT FOR READY
35 015362 005777 164612      TST @DCS ;WAS THE XFER GOOD?
36 015366 100014      BPL Z$ ;JUMP IF OK
37
38 ;HERE IF AN ERROR DETECTED WHILE UPDATING THE MEDIA
39
40 015370      PRINTF #FMT18,#BADWRT
   015370 012746 005447      MOV #BADWRT,-(SP)
   015374 012746 007335      MOV #FMT18,-(SP)
   015400 012746 000002      MOV #2,-(SP)
   015404 010600      MOV SP,R0
   015406 104017      EMT C$PNTF
   015410 062706 000006      ADD #6,SP
41 015414 000137 011660      JMP NXTCMD ;BACK TO THE QUARY LOOP
42 015420 004537 010572      2$: JSR R5,PTIME ;PRINT THE SYS RUN TIME
43 015424      PRINTF #FMT18,#MDONE ;TELL OPR - DONE
   015424 012746 003541      MOV #MDONE,-(SP)
   015430 012746 007335      MOV #FMT18,-(SP)
   015434 012746 000002      MOV #2,-(SP)
   015440 010600      MOV SP,R0
   015442 104017      EMT C$PNTF
   015444 062706 000006      ADD #6,SP
44 015450 012601      3$: MOV (SP)+,R1 ;RESET R1
45 015452 000205      RTS R5 ;EXIT ROUTINE
```

```

1 015454          STARS
2                ;*****
3                ;HERE IS THE CODE TO SERVICE REMOVING AN ENTRY FROM THE 'FIELD' BAD
4 015454          ;SECTOR FILE
5                STARS
6                ;*****
6 015454          BSDEL: PRINTF #FMT19,#DELMSG           ;TELL OPR ABOUT TO DELETE...
015454 012746 005520      MOV #DELMSG,-(SP)
015460 012746 007342      MOV #FMT19,-(SP)
015464 012746 000002      MOV #2,-(SP)
015470 010600      MOV SP,RO
015472 104017      EMT C$PNTF
015474 062706 000006      ADD #6,SP
7 015500 004537 010460      JSR R5,FIRST           ;SELECT THE 1ST UNIT
8 015504 000137 011660      JMP NXTCMD           ;NONE AVAIL.!
9 015510 000404          BR BSDELL
10 015512 004537 01047C     BSDELS: JSR R5,SELDRV       ;SELECT NEXT UNIT
11 015516 000137 011660      JMP NXTCMD           ;ALL DONE
12 015522 004537 023222     BSDELL: JSR R5,LOADED     ;READY?
13 015526 005737 002320      TST TEMPO
14 015532 001403          BEQ 1$               ;YES
15 015534 004537 010736      JSR R5,DRNRDY
16 015540 000764          BR BSDELS           ;SELECT THE NEXT UNIT
17
18 015542 004537 010642     1$: JSR R5,DRVID         ;TELL OPR WHAT DRIVE SELECTED
19 015546 004537 021710      JSR R5,RDBDSC       ;GET A FRESH COPY OF THE BAD SEC FILE
20 015552 005737 002372      TST ACCESS          ;ALLOWED TO PROCEED?
21 015556 001414          BEQ BSDELTD         ;YES
22 015560          PRINTF #FMT18,#DENIED           ;NO - TELL OPR
015560 012746 003747      MOV #DENIED,-(SP)
015564 012746 007335      MOV #FMT18,-(SP)
015570 012746 000002      MOV #2,-(SP)
015574 010600      MOV SP,RO
015576 104017      EMT C$PNTF
015600 062706 000006      ADD #6,SP
23 015604 000137 011660      JMP NXTCMD           ;BACK TO THE QUARY LOOP
24
25 015610          BSDELTD: GMANIL THISDRV,TEMPO,1,NO
015610 104043          EMT C$GMAN
015612 000404          BR 10015$
015614 002320          .WORD TEMPO
015616 000120          .WORD T$CODE
015620 006575          .WORD THISDRV
015622 000001          .WORD 1
10015$:
26 015624 005737 002320      TST TEMPO
27 015630 001730          BEQ BSDELS           ;RE-SELECT IF NOT THIS DRIVE
28
29 015632          BSDEL1: PRINTF #MCRLF
015632 012746 007552      MOV #MCRLF,-(SP)
015636 012746 000001      MOV #1,-(SP)
015642 010600      MOV SP,RO
015644 104017      EMT C$PNTF
015646 062706 000004      ADD #4,SP
30 015652          GMANID DELCYL,BSFCYL,D,777,0,511.,NO
015652 104043          EMT C$GMAN

```

015654	000406		BR	10016\$	
015656	013460		.WORD	BSFCYL	
015660	000042		.WORD	T\$CODE	
015662	005114		.WORD	DEL CYL	
015664	000777		.WORD	777	
015666	000000		.WORD	T\$LOLIM	
015670	000777		.WORD	T\$HILIM	
31 015672			10016\$:		
015672	104043		GMANID	DELSEC,BSFSEC,D,77,0,39.,NO	
015674	000406		EMT	C\$GMAN	
015676	013462		BR	10017\$	
015700	000042		.WORD	BSFSEC	
015702	005146		.WORD	T\$CODE	
015704	000077		.WORD	DELSEC	
015706	000000		.WORD	77	
015710	000047		.WORD	T\$LOLIM	
015712			.WORD	T\$HILIM	
32 015712			10017\$:		
015712	104043		GMANID	DELHD,BSFHD,D,3,0,1,NO	
015714	000406		EMT	C\$GMAN	
015716	013464		BR	10020\$	
015720	000042		.WORD	BSFHD	
015722	005175		.WORD	T\$CODE	
015724	000003		.WORD	DELHD	
015726	000000		.WORD	3	
015730	000001		.WORD	T\$LOLIM	
015732			.WORD	T\$HILIM	
33 015732	013700	013460	10020\$:		
34 015736	000300		MOV	BSFCYL,RO	:COPY THE CYL TO REMOVE
35 015740	000241		SWAB	RO	:PUT IT IN HIGH BYTE
36 015742	006000		CLC		
37 015744	103002		ROR	RO	
38 015746	052700	100000	BCC	1\$	:BR IF DON'T WANT ANOTHER BIT
39 015752	053700	013462	BIS	#BIT15,RO	:ADD IN HIGH ORDER CYL BIT
40 015756	005737	013464	1\$:	BIS BSFSEC,RO	:ADD IN THE SECTOR NUMBER
41 015762	001402		TST	BSFHD	:ON HEAD 0??
42 015764	052700	000100	BEQ	2\$	:YUP
43 015770	010037	002314	2\$:	BIS #100,RO	:NO - POINT TO HEAD 1
44 015774	004537	025252	MOV	RO,CHKSEC	:SAVE THE COMPACTED DISK ADDRESS
45 016000	005737	002312	JSR	R5,CKBDC	:CHECK TO SEE IF ENTRY EXISTS
46 016004	001014		TST	HDRFND	:FOUND?
47 016006			BNE	10\$	:YUP
016006	012746	005567	PPINTF	#FMT18,#NOENTRY	:NOPE
016012	012746	007335	MOV	#NOENTRY,-(SP)	
016016	012746	000002	MOV	#FMT18,-(SP)	
016022	010600		MOV	#2,-(SP)	
016024	104017		MOV	SP,RO	
016026	062706	000006	EMT	C\$PNTF	
48 016032	000137	015512	ADD	#6,SP	
49			JMP	BSDELS	:SELECT THE NEXT UNIT
50 016036	004537	023066	10\$:	JSR R5,RDFIELD	:GET THE FIELD BAD SEC FILE
51 016042	005002		CLR	R2	
52 016044	005037	013470	CLR	SECNUM	
53 016050	005037	013454	CLR	FLDNUM	
54 016054	012737	000020	MOV	#16.,SECMAX	
55 016062	004537	013750	JSR	R5,BSFOK	:POINT TO A WORK AREA

```

56 016066 005737 013474      TST      BSFOKF          ;POINTING TO A VALID AREA?
57 016072 001421              BEQ      11$            ;YUP - PROCEED
58 016074              PRINTF  #FMT18,#NSWSEC  ;TELL OPR THAT ERROR EXISTS
    016074 012746 003453      MOV      #NSWSEC,-(SP)
    016100 012746 007335      MOV      #FMT18,-(SP)
    016104 012746 000002      MOV      #2,-(SP)
    016110 010600              MOV      SP,R0
    016112 104017              EMT      C$PNTF
    016114 062706 000006      ADD      #6,SP
59 016120 004537 020614      JSR      R5,NEWSBF     ;SEE IF OPR WANTS TO MAKE A FILE
60 016124 005737 002320      TST      TEMPO         ;PROCEED IF A 'FIELD' FILE BUILT
61 016130 001002              BNE      11$            ;BR - FILE WAS BUILT
62 016132 000137 015512      JMP      BSDELS        ;SELECT THE NEXT UNIT
63
64 016136 023762 013460 026442 11$:  CMP      BSFCYL,BSFILE(R2) ;AT CORRECT ENTRY?
65 016144 001027              BNE      20$            ;NOPE! UPDATE POINTER
66 016146 013737 013464 002322      MOV      BSFHD,TEMP1   ;GET THE HEAD SELECTED
67 016154 000337 002322      SWAB    TEMP1         ;PUT IT IN HIGH BYTE
68 016160 053737 013462 002322      BIS      BSFSEC,TEMP1  ;ADD IN THE SECTOR BITS
69 016166 023762 002322 026444      CMP      TEMP1,BSFILE+2(R2) ;CORRECT SECTOR TOO?
70 016174 001013              BNE      20$            ;NO - UPDATE POINTER
71
72              ;HAVE THE ENTRY SLOT NOW ... KILL THE ENTRY & MOVE ALL OTHERS UP 1
73
74 016176 016262 026446 026442 12$:  MOV      BSFILE+4(R2),BSFILE(R2) ;MOVE NEXT CYL ENTRY UP
75 016204 016262 026450 026444      MOV      BSFILE+6(R2),BSFILE+2(R2) ;MOVE NEXT SECT ENTRY UP
76 016212 005762 026446      TST      BSFILE+4(R2)  ;END OF ENTRIES YET?
77 016216 100422              BMI      3$            ;YUP - EXIT
78 016220 022222              CMP      (R2)+,(R2)+   ;POINT TO THE NEXT SLOT OF ENTRIES
79 016222 000765              BR       12$           ;AND DO AGAIN
80
81              ;UPDATE THE ENTRY SLOT POINTER
82
83 016224 022222              20$:  CMP      (R2)+,(R2)+   ;UPDATE POINTER BY 2 LOCATIONS
84 016226 005762 026442      TST      BSFILE(R2)   ;END OF ENTRIES?
85 016232 100341              BPL      11$            ;NO - LOOK AT THIS SLOT
86
87              ;HERE IF NO 'FIELD' ENTRY DETECTED ON THE PACK
88
89 016234              PRINTF  #FMT18,#NOFLDE  ;TELL OPR NO 'FIELD' ENTRY HERE
    016234 012746 003477      MOV      #NOFLDE,-(SP)
    016240 012746 007335      MOV      #FMT18,-(SP)
    016244 012746 000002      MOV      #2,-(SP)
    016250 010600              MOV      SP,R0
    016252 104017              EMT      C$PNTF
    016254 062706 000006      ADD      #6,SP
90 016260 000137 015512      JMP      BSDELS        ;SELECT THE NEXT UNIT
91
92              ;HERE TO CLEAR THIS ENTRY FROM REST OF FIELD BAD SECTOR FILE
93              ;WILL COPY THIS MODIFIED SECTOR PAIR INTO THE ENTIRE 'FIELD' BAD SEC FILE
94
95 016264 004537 020536      3$:  JSR      R5,COPY       ;COPY UPDATE TO WHOLE FILE
96 016270 004537 015122      JSR      R5,WRTBSF    ;ASK OPR IF TIME TO UPDATE
97 016274 000137 015512      JMP      BSDELS        ;SELECT THE NEXT UNIT

```

```
1 016300 STARS
2 ;*****
3 ;BSWRITE -- ROUTINE TO WRITE THE WHOLE PACK WITH THE WORST CASE DATA PATTERN
4 ; FOR THE RL01/2 THEN ISSUE THE 'VERIFY' (READ PACK) COMMAND. THIS
5 ; WILL CHECK THE PACK FOR BAD SPOTS AND COMPARE THE FOUND ENTRIES
6 ; WITH THE EXISTING BAD SECTOR FILE.
6 016300 STARS
7 ;*****
8 016300 BSWRITE: PRINTF #FMT19,#MWRITE ;TELL OPR WHATS HAPPENING
016300 012746 004100 MOV #MWRITE,-(SP)
016304 012746 007342 MOV #FMT19,-(SP)
016310 012746 000002 MOV #2,-(SP)
016314 010600 MOV SP,R0
016316 104017 EMT C$PNTF
016320 062706 000006 ADD #6,SP
9 016324 GMANIL MSTWRT,TEMPO,177777,NO
016324 104043 EMT C$GMAN
016326 000404 BR 10021$
016330 002320 .WORD TEMPO
016332 000120 .WORD T$CODE
016334 006020 .WORD MSTWRT
016336 177777 .WORD 177777
016340 10021$:
10 016340 005737 002320 TST TEMPO
11 016344 001402 BEQ 1$ ;QUIT IF CAN'T WRITE ON ALL PACKS
12 016346 004537 010460 JSR R5,FIRST ;SELECT THE 1ST UNIT
13 016352 000137 011660 1$: JMP NXTCMD ;NONE AVAIL.!
14 016356 000404 BR BSWRTL
15 016360 004537 010470 BSWRTS: JSR R5,SELDRV ;SELECT THE NEXT UNIT
16 016364 000137 011660 JMP NXTCMD ;ALL DONE
17
18 016370 004537 010642 BSWRTL: JSR R5,DRVID ;TELL OPR WHAT DRIVE SELECTED
19 016374 004537 023222 JSR R5,LOADED
20 016400 005737 002320 TST TEMPO
21 016404 001403 BEQ 1$ ;DRV READY
22 016406 004537 010736 JSR R5,DRNRDY ;TELL OPR NOT READY
23 016412 000762 BR BSWRTS ;SELECT THE NEXT UNIT
24
25 016414 004537 010572 1$: JSR R5,PTIME ;PRINT THE RUN TIME
26 016420 004537 023342 JSR R5,GETDST ;GET STATUS OF DRV
27 016424 032701 020000 BIT #WL,R1 ;WRITE LOCKED?
28 016430 001416 BEQ 2$ ;NO
29 016432 004537 010642 JSR R5,DRVID ;TELL THE DRIVE ID
30 016436 PRINTF #MSG,#WRTLCK ;YES
016436 012746 003277 MOV #WRTLCK,-(SP)
016442 012746 007555 MOV #MSG,-(SP)
016446 012746 000002 MOV #2,-(SP)
016452 010600 MOV SP,R0
016454 104017 EMT C$PNTF
016456 062706 000006 ADD #6,SP
31 016462 000137 016360 11$: JMP BSWRTS ;SELECT THE NEXT UNIT
32
33 016466 2$: PRINTF #MCRLF
016466 012746 007552 MOV #MCRLF,-(SP)
016472 012746 000001 MOV #1,-(SP)
016476 010600 MOV SP,R0
```

	016500	104017		EMT	CSPNTF	
	016502	062706	000004	ADD	#4,SP	
34	016506	004537	016720	JSR	R5,CLRBSN	;CLEAR THE TEMP STORAGE FOR HARD ERRORS
35	016512	005037	002162	CLR	SFTCNT	;CLEAR THE SOFT ERROR COUNTER
36	016516	005037	002160	CLR	ERRCNT	;CLEAR THE HARD ERROR COUNTER
37						
38	016522			PRINTF	#FMT18,#WRPKF	;PRINT WRITE PACK FWD
	016522	012746	006531	MOV	#WRPKF,-(SP)	
	016526	012746	007335	MOV	#FMT18,-(SP)	
	016532	012746	000002	MOV	#2,-(SP)	
	016536	010600		MOV	SP,R0	
	016540	104017		EMT	CSPNTF	
	016542	062706	000006	ADD	#6,SP	
39	016546	005737	007640	TST	WRISAW	;SAWTOOTH WRT?
40	016552	001412		BEQ	38	;NO
41	016554			PRINTF	#MSG,#SAWFWD	;YES - TELL OPR
	016554	012746	006452	MOV	#SAWFWD,-(SP)	
	016560	012746	007555	MOV	#MSG,-(SP)	
	016564	012746	000002	MOV	#2,-(SP)	
	016570	010600		MOV	SP,R0	
	016572	104017		EMT	CSPNTF	
	016574	062706	000006	ADD	#6,SP	
42	016600	005037	002242	38:	CLR	FWDFLG ;SET CONTROL FOR FWD SAWTOOTH WRITE
43	016604	004537	023434	JSR	R5,WRPACK	;WRITE THE PACK
44	016610	004537	016744	JSR	R5,CVERIFY	;CALL THE VERIFY ROUTINE
45						
46	016614	004537	010572	JSR	R5,PTIME	;TELL THE HALF TIME
47	016620			PRINTF	#FMT18,#WRPKR	;TELL OPR WRT PACK REVERSE
	016620	012746	006553	MOV	#WRPKR,-(SP)	
	016624	012746	007335	MOV	#FMT18,-(SP)	
	016630	012746	000002	MOV	#2,-(SP)	
	016634	010600		MOV	SP,R0	
	016636	104017		EMT	CSPNTF	
	016640	062706	000006	ADD	#6,SP	
48	016644	005737	007640	TST	WRISAW	;SAWTOOTH WRT?
49	016650	001412		BEQ	48	;NO
50	016652			PRINTF	#MSG,#SAWREV	;YES
	016652	012746	006500	MOV	#SAWREV,-(SP)	
	016656	012746	007555	MOV	#MSG,-(SP)	
	016662	012746	000002	MOV	#2,-(SP)	
	016666	010600		MOV	SP,R0	
	016670	104017		EMT	CSPNTF	
	016672	062706	000006	ADD	#6,SP	
51	016676	005237	002242	48:	INC	FWDFLG ;SET CONTROL FOR REVERSE SAWTOOTH WRT
52	016702	004537	023434	JSR	R5,WRPACK	;WRITE THE PACK
53	016706	004537	016744	JSR	R5,CVERIFY	;CALL THE VERIFY ROUTINE
54	016712	004537	017216	JSR	R5,ENDRD1	;PRINT THE TOTALS OF ERRORS DETECTED
55	016716	000620		BR	BSWRTS	;SELECT THE NEXT DRIVE

```
1  
2  
3  
4 016720 010146  
5 016722 012701 025726  
6 016726 012721 177777  
7 016732 022701 026322  
8 016736 001373  
9 016740 012601  
10 016742 000205
```

;HERE TO CLEAR THE TEMP BAD SECTOR FILE STORAGE OF 'HARD' ERROR SPOTS  
;ON THE PACK

```
CIRBSN: MOV R1,-(SP) ;SAVE R1  
MOV #BSECN,R1 ;POINT TO THE 1ST LOCATION IN THE TABLE  
18: MOV #-1,(R1)+ ;INIT THIS ADDR OF TABLE  
CMP #BSECNF,R1 ;DONE?  
BNE 18 ;NO - DO THIS ADDR ALSO  
MOV (SP)+,R1 ;RESET R1  
RTS R5
```

```

1 016744          STARS
2                ;*****
3                ;BSVERIFY -- ROUTINE TO READ THE PACK TO FIND BAD SPOTS. SPOTS THAT
4                ;      ARE 'BAD' AFTER 16 RETRYS TO RECOVER THE DATA WILL BE ENTERED
5                ;      INTO A TEMPORY AREA FOR LATER INSERTION INTO THE REAL BAD
6 016744          STARS
7                ;*****
8 016744 005237 002244  (VERIFY: INC   CVFLG           ;SET THE 'CALLED' FLAG
9 016750 000402          BR      COMVER         ;GOTO THE COMMON VERIFY CODE
10
11 016752 005037 002244  BSVERIFY: CLR   CVFLG           ;CLEAR THE 'CALLED' FLAG
12
13 016756          COMVER: PRINTF #FMT19,#VERIFY
14 016756 012746 004063  MOV      #VERIFY,-(SP)
15 016762 012746 007342  MOV      #FMT19,-(SP)
16 016766 012746 000002  MOV      #2,-(SP)
17 016772 010600          MOV      SP,R0
18 016774 104017          EMT      C$PNTF
19 016776 062706 000006  ADD      #6,SP
20 017002 005737 002244  TST      CVFLG           ;'CALLED'?
21 017006 001011          BNE     BSVERL          ;YES - SKIP SELECT CODE
22 017010 004537 010460  JSR      R5,FIRST      ;NO - SELECT THE 1ST UNIT
23 017014 000137 011660  JMP      NXTCMD        ;NONE AVAIL.!
24 017020 000404          BR      BSVERL
25 017022 004537 010470  BSVERS: JSR     R5,SELDV ;SELECT THE NEXT UNIT
26 017026 000137 011660  JMP      NXTCMD        ;ALL DONE
27 017032 004537 023222  BSVERL: JSR     R5,LOADED ;DRV RDY?
28 017036 005737 002320  TST      TEMPO
29 017042 001410          BEQ     1$             ;YES
30 017044 004537 010736  JSR      R5,DRNRDY
31 017050 005737 002244  TST      CVFLG           ;'CALLED'?
32 017054 001002          BNE     10$           ;YES
33 017056 000137 017022  JMP      BSVERS        ;SELECT THE NEXT UNIT
34 017062 000205          10$:   RTS      R5     ;NO - EXIT NOW
35
36 017064          1$:   PRINTF #MCRLF
37 017064 012746 007552  MOV      #MCRLF,-(SP)
38 017070 012746 000001  MOV      #1,-(SP)
39 017074 010600          MOV      SP,R0
40 017076 104017          EMT      C$PNTF
41 017100 062706 000004  ADD      #4,SP
42 017104 022737 000005 011766  CMP      #5,INPUT      ;HERE FROM 'WRITE' COMMAND?
43 017112 001406          BEQ     11$           ;YUP
44 017114 004537 016720  JSR      R5,CLRBSN     ;NO - INIT HARD ERROR STORAGE AREA
45 017120 005037 002162  CLR      SFTCNT        ;CLEAR THE SOFT ERROR COUNTER
46 017124 005037 002160  CLR      ERRCNT        ;CLEAR THE HARD ERROR COUNT
47 017130 004537 021710  11$:   JSR      R5,R0BDSC  ;GET A FRESH COPY OF THE BAD SECTOR FILE
48 017134 004537 024700  JSR      R5,MDHOME     ;PUT THE HEADS OVER CYLINDER 0
49 017140 012737 002356 002206  MOV      #BUF1,BBA     ;POINT TO THE BUFFER FOR READ/WRITE
50 017146 012737 173000 002176  MOV      #-2560.,BMP   ;SAVE THE WC FOR 20 SECTORS
51 017154 005037 002322  CLR      TEMPI        ;START AT HEAD 0
52 017160 005001          CLR      R1           ;START AT CYLINDER 0
53 017162 022737 000001 002246  CONREAD: CMP     #1,DR  ;DRIVE - RL01?
54 017170 001101          BNE     CRD2         ;NO - MUST BE AN RL02

```



```

45 017172 022701 077600      CMP      #077600,R1      ;AT RLO1 LAST CYL?
46 017176 001101      BNE      STREAD        ;NO - READ THIS TRACK
47
48 017200 005737 002322      CRD1:   TST      TEMP1      ;ON LAST CYL ... IS IT LAST TRACK?
49 017204 001476      BEQ      STREAD        ;NO - DO THE READ
50
51 017206 005737 002244      ENDRD:  TST      CVFLG      ;'CALLED'?
52 017212 001401      BEQ      ENDRD1       ;NO - PROCEED WITH THE TOTALS PRINTOUT
53 017214 000205      RTS      R5            ;YES - EXIT NOW
54 017216 004537 010572      ENDRD1: JSR      R5,PTIME    ;PRINT THE DONE READING TIME
55 017222      PRINTF  #MSG,#MDONE      ;TELL OPR ALL DONE
    017222 012746 003541      MOV      #MDONE,-(SP)
    017226 012746 007555      MOV      #MSG,-(SP)
    017232 012746 000002      MOV      #2,-(SP)
    017236 010600      MOV      SP,R0
    017240 104017      EMT      C$PNTF
    017242 062706 000006      ADD      #6,SP
56 017246 004537 010642      JSR      R5,DRVID      ;TELL OPR WHICH DRIVE
57 017252      PRINTF  #FMT1B,#TSOFT,SFTCNT ;PRINT TOTAL 'SOFT' ERRORS
    017252 013746 002162      MOV      SFTCNT,-(SP)
    017256 012746 002670      MCV     #TSOFT,-(SP)
    017262 012746 007435      MOV      #FMT1B,-(SP)
    017266 012746 000003      MOV      #3,-(SP)
    017272 010600      MOV      SP,R0
    017274 104017      EMT      C$PNTF
    017276 062706 000010      ADD      #10,SP
58 017302      PRINTF  #FMT1B,#THARD,ERRCNT ;PRINT TOTAL 'HARD' ERRORS
    017302 013746 002160      MCV     ERRCNT,-(SP)
    017306 012746 002725      MOV      #THARD,-(SP)
    017312 012746 007435      MOV      #FMT1B,-(SP)
    017316 012746 000003      MOV      #3,-(SP)
    017322 010600      MOV      SP,R0
    017324 104017      EMT      C$PNTF
    017326 062706 000010      ADD      #10,SP
59 017332      PRINTF  #FMT1B,#HYPHEN
    017332 012746 002540      MOV      #HYPHEN,-(SP)
    017336 012746 007335      MOV      #FMT1B,-(SP)
    017342 012746 000002      MOV      #2,-(SP)
    017346 010600      MOV      SP,R0
    017350 104017      EMT      C$PNTF
    017352 062706 000006      ADD      #6,SP
60 017356 004537 020142      JSR      R5,ADDEND      ;SEE IF OPR WANTS TO UPDATE BAD SEC FILE
61 017362 005737 002244      TST      CVFLG      ;'CALLED'?
62 017366 001001      BNE      1$          ;YES
63 017370 000614      BR      BSVERS      ;SELECT THE NEXT UNIT
64 017372 000205      1$:     RTS      R5      ;NO - EXIT NOW
65
66      ;HERE TO CHECK THE END OF AN RLO2
67 017374 022701 177600      CRD2:   CMP      #177600,R1 ;RLO2 LAST CYL?
68 017400 001677      BEQ      CRD1        ;YES - CHECK TO SEE IF LAST TRACK TOO

```

```

1          ;HERE TO READ THE TRACK SELECTED...WILL TRY 20 SECTORS AT A WHACK
2          ;IF AN ERROR IS DETECTED, I WILL THEN TRY TO RECOVER BY READING ONE
3          ;SECTOR AT A TIME. A SECTOR IS DEEMED 'HARD ERROR' AFTER 16 RETRYS.
4          ;ALL BAD SPOTS WILL ENTER A TEMP BAD SEC FILE STORAGE AREA...TO BE ADDED
5          ;TO THE REAL BAD SECTOR FILE AFTER WHOLE PACK HAS BEEN READ.
6
7 017402 0050G2          STREAD: CLR      R2          ;START AT SECTOR 0 ON THIS TRACK
8 017404 010137 002174  SRD1:  MOV     R1,BDA       ;INSERT THE CYL # INTO DISK ADDR
9 017410 053737 002322 002174  BIS     TEMP1,BDA    ;ADD THE HEAD NUMBER (0 OR 1)
10 017415 050237 002174          BIS     R2,BDA       ;ADD THE SECTOR NUMBER
11 017422 012737 000014 002210  MOV     #READ,FUNC   ;GET A READ FUNCTION
12 017430 004537 021570          JSR     R5,LDFUNC    ;ISSUE THE READ CMD
13 017434 004537 023264          JSR     R5,WTRDY    ;WAIT FOR READ TO FINISH
14
15 017440 005777 162534          TST     @DCS        ;ANY ERROR ON THE READ?
16 017444 100030          BPL     NXTSEC      ;BR IF OK
17
18          ;HERE IF AN ERROR DETECTED ON THE READ ... READ ONE SECTOR AT A TIME
19          ;TILL WHOLE TRACK HAS BEEN READ
20
21 017446          PRINTF  =,TCRLF
   017446 012746 007552  MOV     #MCRLF,-(SP)
   017452 012746 000001  MOV     #1,-(SP)
   017456 010600          MOV     SP,R0
   017460 104017          EMT     C$PNTF
   017462 062706 000004  ADD     #4,SP
22 017466 017737 162506 002202  MOV     @DCS,E.DCS   ;GET THE ERROR DETECTED
23 017474 013737 002174 002314  MOV     BDA,CHKSEC
24 017502          ERRSOFT 400,MSFER,ERR1 ;TELL OPR ABOUT THE ERROR DETECTED
   017502 104464          TRAP   T$ERCODE
   017504 000620          .WORD 400
   017506 003144          .WORD MSFER
   017510 006626          .WORD ERR1
25 017512 004537 023356          JSR     R5,ISDRST   ;ISSUE A DRIVE RESET TO CLEAR THE ERROR
26 017516 012737 000001 002316  MOV     #1,DECNT     ;START RECOVERY WITH 1 SOFT ERROR DETECTED
27 017524 000446          BR      ONESEC      ;RECOVER THE TRACK DATA...SLOWLY!

```

```

1          ;HERE TO SELECT THE NEXT SECTOR ADDR TO READ FROM ON THIS TRACK
2          ;HERE IF NO ERPOR DETECTED ON PREV. READ CMD
3
4 017526 062702 000024      NXTSEC: ADD    #20.,R2      ;POINT TO THE NEXT SPOT ON THE TRACK
5 017532 022707 000050      CMP      #40.,R2      ;END OF THE TRACK?
6 017536 001322            BNE      SRD1        ;NO - DO THE READ
7
8          ;HERE TO SELECT THE NEXT TRACK TO READ ... WILL DO A SEEK TO NEXT HEAD
9          ;OR TO THE NEXT CYLINDER.
10
11 017540 005737 002322      NXTTRK: TST    TEMP1      ;ON HEAD #1 NOW?
12 017544 001427            BEQ      5$          ;NO - SEEK TO NEXT TRACK SAME CYL
13 017546 005037 002322      CLR      TEMP1      ;SET FOR NEXT CYL HEAD 0
14 017552 062701 000200      ADD      #200,R1      ;POINT TO THE NEXT CYLINDER
15 017556 042701 000177      BIC      #177,R1      ;CLEAR UNEXPECTED JUNK BITS
16 017562 012737 000200 002174  MOV      #200,BDA
17
18 017570 052737 000005 002174 4$:  BIS      #SIGN!MK,BDA  ;SET FOR A SEEK CMD
19 017576 012737 000006 002210  MOV      #SEEK,FUNC   ;GET THE SEEK CMD
20 017604 004537 021570      JSR      R5,LDFUNC   ;ISSUE THE SEEK
21 017610 004537 023264      JSR      R5,WTRDY    ;WAIT TILL READY
22 017614 010137 002252      MOV      R1,PRPOS    ;SAVE THE PRESENT POSITION ON DISK
23 017620 000137 017162      JMP      CONREAD     ;CONTINUE READING THE PACK
24
25 017624 012737 000100 002322 5$:  MOV      #HEAD,TEMP1  ;SAVE HEAD SELECT STATUS
26 017632 012737 000020 002174  MOV      #SKHS,BDA    ;SET FOR SEEK TO NEXT TRACK SAME CYL
27 017640 000753            BR       4$          ;ISSUE THE SEEK

```

```

1 017642          STARS
2                ;*****
3                ;HERE TO TRY AND RECOVER THE DATA ON SELECTED TRACK BY READING 1 SECTOR
4 017642          ;AT A TIME. SECTOR WILL BE MARKED 'BAD' AFTER 16 RETRYs AND NO RECOVERY.
5                STARS
6                ;*****
6 017642 005002    ONESEC: CLR      R2                ;START AT SECTOR 0 ON THIS TRACK
7 017644 012737 177600 002176    MOV      #-128.,BMP      ;SET THE WC AT 1 SECTORS WORTH
8 017652 013737 002252 002174    1$:  MOV      PRPOS,BDA    ;GET THE CYL # TO START AT
9 017660 050237 002174          BIS      R2,BDA        ;ADD IN THE SECTOR NUMBER
10 017664 053737 002322 002174   BIS      TEMP1,BDA     ;AND THE TRACK (HEAD 0 OR 1)
11
12                ;READ A SECTOR
13 017672 012737 000014 002210    2$:  MOV      #READ,FUNC   ;GET A READ FUNCTION
14 017700 004537 021570          JSR      R5,LDFUNC     ;ISSUE THE READ
15 017704 004537 023264          JSR      R5,WTRDY     ;WAIT FOR READY
16
17 017710 005777 162264          TST      @DCS         ;THIS SECTOR READ OK?
18 017714 100053          BPL      3$          ;BE IF OK - SELECT NEXT SECTOR
19
20                ;ERROR IN THIS SECTOR - TRY A MAX OF 16 TIMES TO RECOVER
21
22 017716 017737 162256 002202    MOV      @DCS,E.DCS   ;SAVE THE DETECTED ERROR
23 017724 005237 002316          INC      DECNT        ;COUNT THIS RETRY
24 017730 013737 002174 002314    MOV      BDA,CHKSEC   ;SEE IF THIS SECTOR IS ALREADY IN
25 017736 004537 025252          JSR      R5,CKBDSC    ;THE BAD SECTOR FILE
26 017742 005737 002312          TST      HDRFND      ;IN THE FILE NOW?
27 017746 001423          BEQ      21$         ;BR IF ERROR
28 017750          PRINTF #FMT18,#INBSF ;TELL OPR SECT IS IN BSF ALREADY
   017750 012746 006250          MOV      #INBSF,-(SP)
   017754 012746 007335          MOV      #FMT18,-(SP)
   017760 012746 000002          MOV      #2,-(SP)
   017764 010600          MOV      SP,R0
   017766 104017          EMT      C$PNTF
   017770 062706 000006          ADD      #6,SP
29 017774          PRINTF #MCRLF
   017774 012746 007552          MOV      #MCRLF,-(SP)
   020000 012746 000001          MOV      #1,-(SP)
   020004 010600          MOV      SP,R0
   020006 104017          EMT      C$PNTF
   020010 062706 000004          ADD      #4,SP
30 020014 000443          BR       30$         ;DO THE NEXT SECTOR ON THIS TRACK
31
32 020016 022737 000020 002316    21$:  CMP      #16.,DECNT   ;TIME TO MARK IT AS A BAD SPOT?
33 020024 001403          BEQ      22$         ;YES
34 020026 004537 023356          JSR      R5,ISDRST   ;NO - ISSUE A DRIVE RESET
35 020032 000717          BR       2$         ;AND CONTINUE
36 020034 004537 024400          22$:  JSR      R5,INBAD   ;YES - MAKE A TEMP BAD SPOT ENTRY
37 020040 004537 023356          23$:  JSR      R5,ISDRST ;RESET THE DRIVE FOR THE NEXT FUNCTION
38
39                ;HERE TO SELECT THE NEXT SECTOR TO RECOVER IN THIS TRACK
40
41 020044 005737 002316          3$:  TST      DECNT        ;ANY DETECTED?
42 020050 001427          BEQ      31$         ;BR IF NONE THIS SECTOR
43 020052 005237 002162          INC      SFTCNT      ;UPDATE THE SOFT ERROR COUNT
44 020056          PRINTF #FMT18,#MSREC ;TELL OPR 'RECOVERED'

```

```
020056 012746 002762      MOV      #MSREC,-(SP)
020062 012746 007335      MOV      #FMT18,-(SP)
020066 012746 000602      MOV      #2,-(SP)
020072 010600      MOV      SP,R0
020074 104017      EMT      C$PNTF
020076 062706 000006      ADD      #6,SP
45 020102      PRINTF  #MCRLF
020102 012746 007552      MOV      #MCRLF,-(SP)
020106 012746 000001      MOV      #1,-(SP)
020112 010600      MOV      SP,R0
020114 104017      EMT      C$PNTF
020116 062706 000004      ADD      #4,SP
46 020122 000402      BR       31$
47 020124 004537 023356      30$: JSR      R5,ISDRST      ;RESET THE DRIVE
48
49 020130 005202      31$: INC      R2          ;POINT TO THE NEXT SECTOR
50 020132 022702 000050      CMP      #40.,R2        ;END OF THIS TRACK?
51 020136 002200      BGE      NXTTRK        ;YES - BACK TO NORMAL 20 SEC READS
52 020140 000644      BR       1$            ;NO - READ THIS SECTOR
53
54 020142      STARS
;*****
55      ADDFND -- ROUTINE TO ASK OPR IF THE NEW BAD SPOTS FOUND BY THE
56      'WRITE' COMMAND OR THE 'VERIFY' COMMAND IS TO BE ADDED TO THE
57      BAD SECTOR FILE ON THE PACK.
58 020142      STARS
;*****
59
60 020142 005737 002372      ADDFND: TST      ACCESS      ;ALLOWED TO UPDATE THE PACK?
61 020146 001172      BNE      ADDFEX          ;NO - EXIT NOW
62 020150 004537 021710      JSR      R5,RDBDSC      ;GET A FRESH COPY OF THE BAD SECTOR FILE
63 020154 004537 023066      JSR      R5,RDFIELD     ;GET A CORE COPY OF THE 'FIELD' FILE
64 020160 012701 025726      MOV      #BSECN,R1      ;POINT TO THE NEW ENTRY TABLE
65 020164 005737 002160      TST      ERRCNT        ;SEE IF ANY NEW BAD SPOTS
66 020170 001561      BEQ      ADDFEX          ;EXIT IF NONE DETECTED
67 020172 005002      CLR      R2            ;CLEAR AN INDEX INTO BAD SECTOR FILE
68 020174 005037 013470      CLR      SECNUM        ;START AT THE 1ST SECTOR IN 'FIELD'
69 020200 012737 000020 013466      MOV      #16.,SECMAX    ;SETUP THE LIMIT
70 020206 004537 013750      JSR      R5,BSFOK       ;SET R2 TO POINT INTO THE FILE
71 020212 005737 013474      TST      BSFOKF        ;SEE IF FIELD FILE EXISTS
72 020216 001420      BEQ      1$            ;BR IF OK
73 020220      PRINTF  #FMT18,#NSWSEC ;TELL OPR THAT NO 'FIELD' FILE EXISTS
020220 012746 003453      MOV      #NSWSEC,-(SP)
020224 012746 007335      MOV      #FMT18,-(SP)
020230 012746 000002      MOV      #2,-(SP)
020234 010600      MOV      SP,R0
020236 104017      EMT      C$PNTF
020240 062706 000006      ADD      #6,SP
74 020244 004537 020614      JSR      R5,NEWBSF      ;ASK IF TIME TO MAKE A 'FIELD' FILE
75 020250 005737 002320      TST      TEMPO          ;WAS A 'FIELD' FILE BUILT?
76 020254 001001      BNE      1$            ;BR IF YES
77 020256 000526      BR       ADDFEX          ;NO - EXIT
78 020260 011137 002314      1$: MOV      (R1),CHKSEC   ;GET AN ENTRY
79 020264 005737 002314      TST      CHKSEC        ;DONE?
80 020270 100515      BMI      4$            ;YUP - UPDATE REST OF 'FIELD' FILE
81
82      ;HERE TO SEE IF ENTRY ALREADY EXISTS...SHOULDN'T
```

```
83
84 020272 004537 025252      JSR    R5,CKBDC      ;WELL...
85 020276 005737 002312      TST    HDRFND        ;FIND IN LIST ALREADY?
86 020302 001106              BNE    3$            ;YES - LOOK AT THE NEXT ENTRY
87
88                               ;HERE TO ASK OPR IF THIS ENTRY TO BE ADDED TO BAD SEC FILE
89
90 020304 011137 013460      MOV    (R1),BSFCYL   ;GET DA FOR CYL #
91 020310 042737 000177 013460 BIC    #177,BSFCYL   ;CLEAR HEAD & SECTOR #
92 020316 000337 013460      SWAB   BSFCYL
93 020322 000241              CLC
94 020324 006137 013460      ROL    BSFCYL
95 020330 103002              BCC    11$
96 020332 005237 013460      INC    BSFCYL        ;ADD IN LOW ORDER CYL #
97 020336 011137 013462      MOV    (R1),BSFSEC   ;GET DA FOR SEC VALUE
98 020342 042737 177700 013462 BIC    #177700,BSFSEC ;CLEAR CYLINDER # & HEAD
99 020350 005037 013464      CLR    BSFHD        ;START AT HEAD 0
100 020354 032711 000100      BIT    #100,(R1)     ;HEAD 1?
101 020360 001402              BEQ    2$            ;NO - ITS HEAD 0
102 020362 005237 013464      INC    BSFHD        ;POINT TO HEAD 1
103 020366              2$: PRINTF #FMT16,#NEWENT,#CMSG,BSFCYL,#SMSG,BSFSEC,#HMSG,BSFHD
    020366 013746 013464      MOV    BSFHD,-(SP)
    020372 012746 002462      MOV    #HMSG,-(SP)
    020376 013746 013462      MOV    BSFSEC,-(SP)
    020402 012746 003101      MOV    #SMSG,-(SP)
    020406 013746 013460      MOV    BSFCYL,-(SP)
    020412 012746 002447      MOV    #CMSG,-(SP)
    020416 012746 006103      MOV    #NEWENT,-(SP)
    020422 012746 007232      MOV    #FMT16,-(SP)
    020426 012746 000010      MOV    #10,-(SP)
    020432 010600      MOV    SP,R0
    020434 104017      EMT    C$PNTF
    020436 062706 000022      ADD    #22,SP
104 020442      GMANIL ABSMSG,TEMPO,177777,NO ;ASK OPR IF OK TO ENTER
    020442 104043      EMT    C$GMAN
    020444 000404      BR    10022$
    020446 002320      .WORD TEMPO
    020450 000120      .WORD T$CODE
    020452 005024      .WORD ABSMSG
    020454 177777      .WORD 177777
    020456              10022$:
105 020456 005737 002320      TST    TEMPO        ;BR IF NO
106 020462 001416              BEQ    3$            ;NO - GET THE NEXT ENTRY
```

```
1 ;HERE TO INSERT THIS SPOT IN THE BAD SECTOR FILE
2
3 020464 005762 026442 21$: TST BSFILE(R2) ;SEE IF A FREE SPOT
4 020470 100403 BMI 22$ ;BR IF FOUND A FREE ENTRY
5 020472 062702 000004 ADD #4,R2 ;POINT TO THE NEXT ENTRY SLOT
6 020476 000772 BR 21$ ;AND TRY AGAIN
7
8 020500 013762 013460 026442 22$: MOV BSFCYL,BSFILE(R2) ;INSERT THE CYL # INTO FILE
9 020506 011162 026444 MOV (R1),BSFILE+2(R2) ;ADD THE SECTOR NUM & HEAD
10 020512 042762 177600 026444 BIC #177600,BSFILE+2(R2) ;CLEAR CYL # BITS
11
12 ;HERE TO UPDATE THE POINTER INTO THE TEMP BAD SEC TABLE
13 020520 005721 3$: TST (R1)+ ;UPDATE THE POINTER
14 020522 000656 BR 1$ ;PROCESS THIS ENTRY
15
16 020524 004537 020536 4$: JSR R5,COPY ;COPY UPDATES TO WHOLE FILE
17 020530 004537 015122 JSR R5,WRTBSF ;ASK OPR IF TIME TO UPDATE PACK
18
19 ;HERE TO EXIT THIS PHASE
20 020534 000205 ADDFEX: RTS R5 ;EXIT
```

```
1 020536          STARS
2                :*****
3                :HERE TO UPDATE THE REST OF THE 'FIELD' IMAGE FILE
4                :THIS ROUTINE USES CONTENTS OF 'SECOLD' AS THE POINTER TO THE START OF THE
5 020536          :UPDATE AREA IN 'BSFILE'
6                :*****
7 020536 010246    COPY:  MOV   R2,-(SP)      ;SAVE R2
8 020540 010346    MOV   R3,-(SP)      ;SAVE R3
9 020542 005002    CLR    R2           ;POINT TO THE 1ST ADDR IN BSFILE STORAGE AREA
10 020544 013703 013472 1$:  MOV   SECOLD,R3    ;GET THE ADDR OF THE WORD SPACE
11 020550 012737 002000 002320  MOV   #4*256.,TEMPO ;GET # WORDS PER SECTOR PAIR
12 020556 060337 002320    ADD   R3,TEMPO    ;TEMPO = 1ST ADDR OF THE NEXT SECTOR PAIR
13
14 020562 022702 012000    2$:  CMP   #2560.*2,R2    ;END OF BSFILE?
15 020566 001407    BEQ   4$           ;YES
16 020570 011362 026442    3$:  MOV   (R3),BSFILE(R2) ;COPY WORK SPACE TO WHOLE 'FIELD' FILE
17 020574 022322    CMP   (R3)+,(R2)+    ;UPDATE POINTERS
18 020576 020337 002320    CMP   R3,TEMPO      ;DONE?
19 020602 001760    BEQ   1$           ;YES - RESET WORK SPACE POINTER
20 020604 000766    BR    2$           ;AND CONTINUE
21
22 020606 012603    4$:  MOV   (SP)+,R3      ;RESET R3
23 020610 012602    MOV   (SP)+,R2      ;RESET R2
24 020612 000205    RTS    R5           ;EXIT
```



```

1 020614          STARS
2                ;*****
3                ;NEWSBF -- ROUTINE TO ASK OPR IF TIME TO CREATE A BAD SECTOR
4 020614          ;      FILE IF THE AREA CAN'T BE RECOGNIZED AS A BAD SECTOR FILE.
5                ;*****
6 020614 010146   NEWSBF: MOV     R1,-(SP)      ;SAVE R1
7 020616          GMANIL  MBLD,TEMPO,177777,NO
8 020616 104043   EMT     CSGMAN
9 020620 000404   BR      10023$
10 020622 002320  .WORD   TEMPO
11 020624 000120  .WORD   T$CODE
12 020626 003012  .WORD   MBLD
13 020630 177777  .WORD   177777
14 020632          10023$:
15 020632 005737 002320 TST     TEMPO      ;BR IF NO
16 020636 001502 BEQ     2$        ;EXIT
17                ;HERE TO INIT THE BSFILE STORAGE FOR BUILDING A FILE
18 020640 012701 026442 1$:  MOV     #BSFILE,R1    ;SETUP A POINTER
19 020644 010137 013472  MOV     R1,SECOLD    ;POINT TO THE START OF THE 'UPDATE' AREA
20 020650 012721 177777 11$: MOV     #-1,(R1)+    ;INIT A LOCATION
21 020654 022701 030442  CMP     #BSFILE+1024.,R1 ;DONE??
22 020660 001373  BNE     11$      ;NO - PROCEED TO INIT
23 020662 012701 026442  MOV     #BSFILE,R1    ;GET START AGAIN
24 020666 005021  CLR     (R1)+        ;CLEAR
25 020670 005021  CLR     (R1)+
26 020672 005021  CLR     (R1)+
27 020674 005011  CLR     (R1)
28 020676 005737 002364  TST     SN1          ;ALREADY HAVE A SERIAL NUMBER?
29 020702 001020  BNE     13$          ;YUP - TELL OPR WHAT IT IS
30 020704          12$:  GMANID  ABSSNL,SN1,0,77777,1,77777,NO ;GET SERIAL # LOW 5
31 020704 104043   EMT     CSGMAN
32 020706 000406   BR      10024$
33 020710 002364   .WORD   SN1
34 020712 000022   .WORD   T$CODE
35 020714 005272   .WORD   ABSSNL
36 020716 077777   .WORD   77777
37 020720 000001   .WORD   T$LOLIM
38 020722 077777   .WORD   T$HILIM
39 020724          10024$:
40 020724 0104043  GMANID  ABSSNH,SN2,0,77777,0,77777,NO ;GET SERIAL # HIGH 5
41 020724 104043   EMT     CSGMAN
42 020726 000406   BR      10025$
43 020730 002366   .WORD   SN2
44 020732 000022   .WORD   T$CODE
45 020734 005331   .WORD   ABSSNH
46 020736 077777   .WORD   77777
47 020740 000000   .WORD   T$LOLIM
48 020742 077777   .WORD   T$HILIM
49 020744          10025$:
50 020744          13$:  PRINTF  #FMTSN,#(ART,SN2,SN1
51 020744 013746 002364  MOV     SN1,-(SP)
52 020750 013746 002366  MOV     SN2,-(SP)

```

```
020754 012746 002427      MOV    #CART,-(SP)
020760 012746 007420      MOV    #FMTSN,-(SP)
020764 012746 000604      MOV    #4,-(SP)
020770 010600      MOV    SP,RO
020772 104017      EMT    C$PNTF
020774 062706 000012      ADD    #12,SP
29 021000      GMANIL VALSN,TEMPO,177777,NO
021000 104043      EMT    C$GMAN
021002 000404      BR     10026$
021004 002320      .WORD TEMPO
021006 000120      .WORD T$CODE
021010 005660      .WORD VALSN
021012 177777      .WORD 177777
021014
30 021014 005737 002320      10026$: TST    TEMPO          ;SEE IF TYPED IN SERIAL NUMBER IS OK
31 021020 001731      BEQ    12$           ;NO - GET A NEW SERIAL NUMBER
32 021022 013737 002364 026442      MOV    SN1,BSFILE    ;SAVE THE SERIAL NUMBER LOW 5
33 021030 013737 002366 026444      MOV    SN2,BSFILE+2 ;AND SEREIAL NUMBER HIGH 5
34 021036 005237 002320      INC    TEMPO        ;INDICATE FILE BUILT - 1 SECTOR
35 021042 000205      RTS    R5
36
37 021044 005037 002320      2$:   CLR    TEMPO          ;INDICATE NO FILE BUILT
38 021050 012601      MOV    (SP)+,R1
39 021052 000205      RTS    R5
```

```
1 021054          STARS
2                :*****
3                ;BSMAKE -- ROUTINE TO CREATE A 'FACTORY' OR 'FIELD' BAD SECTOR FILE.
4                ;      THIS ROUTINE ABORTS IF 'UPDATE' ACCESS TO THE PACK IS DENIED.
5                ;      THE 'FACTORY' FILE WILL CONTAIN ONLY THE DUMMY HEADERS...NO
6 021054          STARS
7                :*****
8 021054          BSMMAKE: PRINTF  #FMT19,#BUILD  ;TELL OPR WHATS HAPPENING
021054 012746 003050      MOV      #BUILD,-(SP)
021060 012746 007342      MOV      #FMT19,-(SP)
021064 012746 000002      MOV      #2,-(SP)
021070 010600      MOV      SP,RO
021072 104017      EMT      C$PNTF
021074 062706 000006      ADD      #6,SP
9 021100 004537 010460      JSR      R5,FIRST      ;SELECT THE 1ST UNIT
10 021104 000137 011660     JMP      NXTCMD      ;NONE AVAIL.!
11 021110 000404      BR      BSMKL
12 021112 004537 010470     BSMKS: JSR      R5,SELDRV      ;SELECT THE NEXT UNIT
13 021116 000137 011660     JMP      NXTCMD      ;ALL DONE
14 021122 004537 023222     BSMKL: JSR      R5,LOADED      ;DRV RDY?
15 021126 005737 002320     TST      TEMPO
16 021132 001403      BEQ      1$,          ;YES
17 021134 004537 010736     JSR      R5,DRNRDY
18 021140 000764      BR      BSMKS      ;SELECT THE NEXT UNIT
19
20 021142 004537 010642     1$:      JSR      R5,DRVID      ;TELL OPR WHAT DRIVE SELECTED
21 021146 005737 002372     TST      ACCESS      ;ALLOWED TO UPDATE PACK?
22 021152 001414      BEQ      10$,        ;YES - PROCEED
23 021154          PRINTF  #FMT18,#DENIED      ;NO - TELL OPR
021154 012746 003747      MOV      #DENIED,-(SP)
021160 012746 007335      MOV      #FMT18,-(SP)
021164 012746 000002      MOV      #2,-(SP)
021170 010600      MOV      SP,RO
021172 104017      EMT      C$PNTF
021174 062706 000006      ADD      #6,SP
24 021200 000137 011660     JMP      NXTCMD      ;QUIT NOW
25
26 021204          10$:   GMANIL  THISDRV,TEMPO,1,NO
021204 104043      EMT      C$GMAN
021206 000404      BR      10027$
021210 002320      .WORD  TEMPO
021212 000120      .WORD  T$CODE
021214 006575      .WORD  THISDRV
021216 000001      .WORD  1
27 021220 005737 002320     10027$: TST      TEMPO
28 021224 001732      BEQ      BSMKS      ;RE-SELECT IF NOT THIS DRIVE
29
30 021226 004537 021710     11$:   JSR      R5,RDBDSC      ;GET A FRESH COPY OF THE 'BAD SEC FILE'
31 021232 004537 023026     JSR      R5,RDFACT      ;THEN A CORE IMAGE OF THE FACTORY FILE
32 021236          PRINTF  #FMT19,#CKFACT      ;TELL OPR CHECKING FACT FILE
021236 012746 006305      MOV      #CKFACT,-(SP)
021242 012746 007342      MOV      #FMT19,-(SP)
021246 012746 000002      MOV      #2,-(SP)
021252 010600      MOV      SP,RO
```

```
021254 104017          EMT      C$PNTF
021256 062706 000006   ADD      #6,SP
33 021262 005002          CLR      R2          ;POINT TO 1ST WORD IN CORE IMAGE
34 021264 005037 013470   CLR      SECNUM     ;START AT 1ST SECTOR PAIR IN FILE
35 021270 012737 000020 013466   MOV      #16.,SECMAX ;STOP AT THIS SECT PAIR
36 021276 004537 013750   JSR      R5,BSFOK   ;SEE IF ANY RECOGNIZED 'FACTORY' FILE
37 021302 005737 013474   TST      BSFOKF     ;WELL???
38 021306 001013          BNE      12$        ;NO - ASK IF TIME TO MAKE ONE
39 021310          PRINTF  #MSG,#OK   ;MSG TO OPR 'FOUND'
021310 012746 006242   MOV      #OK,-(SP)
021314 012746 007555   MOV      #MSG,-(SP)
021320 012746 000002   MOV      #2,-(SP)
021324 010600          MOV      SP,R0
021326 104017          EMT      C$PNTF
021330 062706 000006   ADD      #6,SP
40 021334 000426          BR       2$          ;JUMP OVER BUILD CODE
41
42          ;HERE TO BUILD A DUMMY 'FACTORY' FILE SO AT LEAST ONE EXISTS...WILL NOT
43          ;CONTAIN ANY ENTRIES!
44
45 021336          12$: PRINTF  #MSG,#NHWSEC ;TELL OPR THAT NO 'FACTORY' EXISTS
021336 012746 003400   MOV      #NHWSEC,-(SP)
021342 012746 007555   MOV      #MSG,-(SP)
021346 012746 000002   MOV      #2,-(SP)
021352 010600          MOV      SP,R0
021354 104017          EMT      C$PNTF
021356 062706 000006   ADD      #6,SP
46 021362 004537 020614   JSR      R5,NEWBSF  ;ASK IF TIME TO BUILD ONE
47 021366 005737 002320   TST      TEMPO     ;DID I MAKE A DUMMY FILE?
48 021372 001407          BEQ      2$        ;NO - CHECK ON THE 'FIELD' FILE
49 021374 013737 002320 002230   MOV      TEMPO,NEWFAC ;SET FACTORY FLAG
50 021402 004537 020536   JSR      R5,COPY    ;MAKE THE WHOLE 'FACTORY' FILE
51 021406 004537 015122   JSR      R5,WRTBSF  ;ASK OPR IF TIME TO UPDATE PACK
```

```

1                                     ;HERE TO SEE IF A 'FIELD' FILE HAS TO BE BUILT
2
3 021412 004537 023066                2$:   JSR    R5,RDFIELD      ;GET A CORE IMAGE OF THE 'FIELD' FILE
4 021416                                PRINTF #FMT19,#CKFLD    ;TELL OPR CHECKING FOR FIELD FILE
   021416 012746 006344                MOV    #CKFLD,-(SP)
   021422 012746 007342                MOV    #FMT19,-(SP)
   021426 012746 000002                MOV    #2,-(SP)
   021432 010600                MOV    SP,R0
   021434 104017                EMT    C$PNTF
   021436 062706 000006                ADD    #6,SP
5 021442 005002                CLR    R2                ;START AT 1ST WORD IN BUFFER
6 021444 005037 013470                CLR    SECNUM           ;AND 1ST SECTOR PAIR OF FILE
7 021450 012737 000020 013466        MOV    #16.,SECMAX     ;SETUP THE LIMIT FOR SEARCH
8 021456 004537 013750                JSR    R5,BSFOK        ;POINT TO A VALID AREA
9 021462 005737 013474                TST    BSFOK           ;FIND THE 'FIELD' AREA?
10 021466 001013                BNE    21$             ;NO - ASK IF TIME TO MAKE ONE
11 021470                                PRINTF #MSG,#OK        ;TELL OPR 'FOUND' FILE
   021470 012746 006242                MOV    #OK,-(SP)
   021474 012746 007555                MOV    #MSG,-(SP)
   021500 012746 000002                MOV    #2,-(SP)
   021504 010600                MCV    SP,R0
   021506 104017                EMT    C$PNTF
   021510 062706 000006                ADD    #6,SP
12 021514 000423                BR     4$              ;PROCEED
13
14 021516                                21$:  PRINTF #MSG,#NSWSEC ;TELL OPR NO 'FIELD' FILE
   021516 012746 003453                MOV    #NSWSEC,-(SP)
   021522 012746 007555                MOV    #MSG,-(SP)
   021526 012746 000002                MOV    #2,-(SP)
   021532 010600                MOV    SP,R0
   021534 104017                EMT    C$PNTF
   021536 062706 000006                ADD    #6,SP
15 021542 004537 020614                JSR    R5,NEWBSF      ;ASK OPR IF TIME TO BUILD A FILE
16 021546 005737 002320                TST    TEMPO           ;BUILT A FILE?
17 021552 001404                BEQ    4$              ;BR IF NO
18 021554 004537 020536                3$:   JSR    R5,COPY     ;COPY 'FIELD' TO ENTIRE FILE
19 021560 004537 015122                JSR    R5,WRTBSF     ;ASK OPR IF TIME TO UPDATE PACK
20
21 021564 000137 021112                4$:   JMP    BSMKS      ;SELECT THE NEXT UNIT

```

```

ROUTINE TO LOAD FUNCTION

1          .SBTTL  ROUTINE TO LOAD FUNCTION
2          :CALL   JSR    R5,LDFUNC
3          :ALL INFORMATION MUST BE SET UP IN DRIVE BUFFER
4          :R4 HAS POINTER TO BUFFER
5
6 021570 013703 002200      LDFUNC: MOV    DCS,R3          ;GET CSR FOR DRIVE
7 021574 032713 000200      BIT    #BIT7,(R3)       ;CAN WE ISSUE COMMAND?
8 021600 001003              BNE    1$                ;YES, GO ISSUE COMMAND
9
10 021602              ERRSF  200.,PRGER          ;THIS ERROR SHOULD NEVER PRINT
    021602 104421          TRAP  T$ERCODE
    021604 000310          .WORD 200
    021606 003556          .WORD PRGER
11
12 021610 017763 160372 000002 1$:  MOV    @BBA,BA(R3)      ;LOAD BUS ADDRESS REGISTER
13 021616 013763 002174 000004      MOV    BDA,DA(R3)      ;LOAD DISK ADDRESS REGISTER
14 021624 013763 002176 000006      MOV    BMP,MP(R3)     ;LOAD MULTI-PURPOSE REGISTER
15 021632 013737 002210 002212      MOV    FUNC,BCSADR    ;GET FUNCTION
16 021640 053737 002232 002212      BIS    DRSEL,BCSADR   ;SET DRIVE SELECT BITS
17 021646 052737 000201 002212      BIS    #CRDY!DRDY,BCSADR ;SET CRDY & DRDY IN IMAGE
18 021654 042737 002000 002212      BIC    #OPI,BCSADR    ;WE'RE CLEAR BIT 10 FOR DRIVE 7-4 (OKAY?)
19 021662 013763 002212 000000      MOV    BCSADR,CS(R3) ;LOAD CSR
20 021670 042763 000200 000000      BIC    #CRDY,CS(R3)  ;ISSUE FUNCTION
21 021676 000205              RTS    R5                ;EXIT
22
23          .SBTTL  INTERRUPT SERVICE - VERY SHORT!
24 021700      BGNSRV  INTR1
25
26 021700 042777 000100 160272  INTR1: BIC    #INTEN,@DCS
27 021706      ENDSRV
    021706      L10012:
    021706 000002              RTI

```

```

1          .SBTTL  BAD SECTOR FILE ROUTINE
2
3 021710   STARS
4          :*****
5          :ROUTINE TO RECOVER BAD SECTOR FILE AND SAVE IT FOR
6          :COMPARISON UPON ERROR ON READS/Writes.  WE WILL ONLY
7          :RESERVE SPACE FOR 16 BAD SECTORS PER DRIVE.
8          :WE WILL ISSUE A DRIVE RESET FIRST, READ HEADER, POSITION
9          :TO LAST TRACK (CYLINDER 255, SURFACE 1) AND READ IN
10         :THE FIRST SECTOR FOR FACTORY BAD, AND THE 20TH FOR
11         :FIELD BAD SECTORS.  R4 WILL CONTAIN THE BUFFER POINTER
12         :TO THE DRIVE WE WANT TO READ
13         :
14 021710   :CALL  JSR    R5,RDBDSC
15         STARS
16         :*****
16 021710   RDBDSC: MOV    R0,-(SP)          ;SAVE REGISTERS
17 021712   MOV    R1,-(SP)          ;
18 021714   MOV    R2,-(SP)          ;
19 021716   MOV    R3,-(SP)          ;
20 021720   004537 023356 21$: JSR    R5,ISDRST        ;RESET THE DRIVE
21 021724   012737 000010 002210 MOV    #RDHDR,FUNC      ;READ HEADER TO FIND POSITION
22 021732   004537 021570 JSR    R5,LDFUNC        ;ON DISK
23 021736   004537 023264 JSR    R5,WTRDY
24 021742   005777 160232 TST    @DCS            ;ERROR DETECTED?
25 021746   100011 BP_    22$              ;NO
26 021750   017737 160224 002202 MOV    @DCS,E.DCS      ;YES - SAVE THE RLCS STATUS
27 021756   ERRHRD 500.,MHDR,ERR2
   021756   104463 TRAP   T$ERCODE
   021760   000764 .WORD  500
   021762   003242 .WORD  MHDR
   021764   007114 .WORD  ERR2
28 021766   000137 023014 JMP    9$              ;FORCED EXIT
29
30 021772   016300 000006 22$: MOV    MP(R3),R0      ;GET HEADER AND CALCULATE
31 021776   022737 000001 002246 CMP    #1,TDR          ;RL02 TYPE DRIVE?
32 022004   001005 BNE    23$            ;JUMP IF RL02
33 022006   043700 002264 BIC    CYLSK,R0       ;HERE FOR RL01
34 022012   012701 077600 MOV    #77600,R1
35 022016   000404 BR     25$
36 022020   043700 002270 23$: BIC    CMSK,R0          ;HERE FOR RL02
37 022024   012701 177600 MOV    #177600,R1
38 022030   160001 25$: SUB    R0,R1
39 022032   010137 002174 MOV    R1,BDA
40 022036   052737 000025 002174 BIS    #SKHS!SIGN!MK,BDA
41 022044   012737 000006 002210 MOV    #SEEK,FUNC
42 022052   004537 021570 JSR    R5,LDFUNC        ;SEEK TO THE BAD SEC FILE CYLINDER
43 022056   004537 023264 JSR    R5,WTRDY        ;WAIT FOR DONE
44 022062   005777 160112 TST    @DCS            ;ERROR DETECTED?
45 022066   100011 BPL    26$            ;NO
46 022070   017737 160104 002202 MOV    @DCS,E.DCS      ;YES - SAVE THE RLCS STATUS
47 022076   ERRHRD 510.,MHDR,ERR2
   022076   104463 TRAP   T$ERCODE
   022100   000776 .WORD  510
   022102   003242 .WORD  MHDR
   022104   007114 .WORD  ERR2

```

```

48 022106 000137 023014          JMP      9$          :FORCED EXIT
49
50 022112 012737 000010 002210 26$: MOV      #RDHDR, FUNC
51 022120 004537 021570          JSR      R5, LDFUNC  :READ THE HEADER ON THIS CYLINDER
52 022124 004537 023264          JSR      R5, WTRDY
53 022130 005777 160044          TST     @DCS        :ERROR DETECTED?
54 022134 100011          BPL     27$        :NO
55 022136 017737 160036 002202  MOV     @DCS, E.DCS  :YES - SAVE THE ERROR STATUS IN RLCS
56 022144          ERRHRD  520., MHDR, ERR2
    022144 104463          TRAP   T$ERCODE
    022146 001010          .WORD  520
    022150 003242          .WORD  MHDR
    022152 007114          .WORD  ERR2
57 022154 000137 023014          JMP      9$          :FORCED EXIT
58
59 022160 016300 000006          27$:  MOV     MP(R3), R0  :NO ERROR - GET THE HEADER WORD
60 022164 042700 000077          BIC     #77, R0     :CLEAR THE SECTOR BITS FROM HEADER WORD
61 022170 022737 000001 002246  CMP     #1, TDR     :RL01?
62 022176 001007          BNE     300$       :NO - JMP FOR RL02
63 022200 022700 077700          CMP     #77700, R0  :HERE FOR RL01 - LAST TRACK?
64 022204 001245          BNE     21$        :NO - TRY AGAIN
65 022206 012737 077700 002174  MOV     #77700, BDA  :YES - SET FOR A READ CMD
66 022214 000406          BR      555$
67 022216 022700 177700          300$:  CMP     #177700, R0  :HERE FOR RL02 - LAST TRACK (511.)?
68 022222 001236          BNE     21$        :NO - TRY AGAIN
69 022224 012737 177700 002174  MOV     #177700, BDA
70 022232 012737 177400 002176  555$:  MOV     #-256., BMP  :DO A 1 SECTOR READ
71 022240 012737 000014 002210  MOV     #READ, FUNC  :READ DATA FUNCTION
72
73 022246 005037 002326          CLR     TEMP3      :MANUFACTURING/FIELD FILE SWITCH
74 022252 005037 002320          CLR     TEMPO
75 022256 005037 002316          CLR     DECNT      :CLEAR THE COUNT OF ENTRIES DETECTED
76 022262 013702 002234          MOV     BSECPT, R2  :INITIALIZE LIST TO ALL 1'S
77 022266 012700 000176          MOV     #126., R0   :126 ENTRIES
78 022272 012722 177777          11$:  MOV     #-1, (R2)+  :INIT ENTRY TO -1
79 022276 005300          DEC     R0
80 022300 001374          BNE     11$        :EXIT IF STORAGE INITED
81
82 022302 013702 002234          MOV     BSECPT, R2  :GET LIST TO STORE
83 022306 012700 000031          MOV     #25., R0    :25 ENTRIES
84 022312 004537 021570          4$:  JSR     R5, LDFUNC  :ISSUE THE READ CMD
85 022316 004537 023264          JSR     R5, WTRDY   :WAIT TILL SECTOR READ
86
87 022322 005777 157652          TST     @DCS        :WAS THE READ GOOD?
88 022326 100065          BPL     3$         :YES
89
90 022330 004537 023356          JSR     R5, ISDRST  :NO - RESET THE DRIVE
91 022334 062737 000004 002174  ADD     #4, BDA     :NEXT SECTOR
92 022342 005737 002326          TST     TEMP3      :MANUFACTURING OR FIELD BAD
93 022346 001424          BEQ     5$         :MANUFACTURING
94 022350 022737 000001 002246  CMP     #1, TDR     :RL01=1
95 022356 001024          BNE     400$       :MUST BE AN RL02
96 022360 022737 077750 002174  CMP     #77750, BDA  :END OF FACTORY FILE?
97 022366 001351          BNE     4$         :NO - READ NEXT SECTOR
98 022370          41$:  PRINTF  #FMT18, #SWSEC  :TELL OPR NO 'FIELD' FILE ON PACK
    022370 012746 003426          MOV     #SWSEC, -(SP)
    022374 012746 007335          MOV     #FMT18, -(SP)

```



```

022400 012746 000002      MOV    #2,-(SP)
022404 010600      MOV    SP,RO
022406 104017      EMT    C$PNTF
99 022410 062706 000006      ADD    #6,SP
022414 000137 022762      JMP    7$                ;EXIT
100
101 022420 023727 002174 077724 5$:    CMP    BDA,#77724        ;AT END OF MANUFACTURING BAD
102 022426 000410      BR     55$
103 022430 022737 177750 002174 400$:    CMP    #177750,BDA      ;AT END OF FIELD BAD FOR RLO2
104 022436 001325      BNE    4$                ;NO GO BACK FOR NEXT
105 022440 000753      BR     41$              ;PRINT 'FIELD' ERROR
106 022442 023727 002174 177724      CMP    BDA,#177724     ;AT END OF MANUFACTURING BAD
107 022450 001320      BNE    4$                ;NO, GET THE NEXT SECTOR PAIR
108 022452          55$:    PRINTF #FMT18,#HWSEC    ;TELL OPR NO 'FACTORY' FILE ON PACK
          56$:    MOV    #HWSEC,-(SP)
          022452 012746 003351      MOV    #FMT18,-(SP)
          022456 012746 007335      MOV    #2,-(SP)
          022462 012746 000002      MOV    SP,RO
          022466 010600      EMT    C$PNTF
          022470 104017      ADD    #6,SP
109 022472 062706 000006      JMP    7$                ;EXIT & HEADS HOME
          022476 000137 022762
110
111 022502 017701 157500          3$:    MOV    @BBA,R1          ;START OF LIST
112 022506 005037 013452      CLR    NOSNUM          ;CLEAR THE FOUND SERIAL NUMBER FLAG
113 022512 005721          TST    (R1)+           ;SEE IF A SERIAL NUMBER PRESENT
114 022514 001005      BNE    31$             ;YUP - SN WORD 0 >0
115 022516 005721          TST    (R1)+           ;NO ... SEE IF SN WORD 1 =0
116 022520 001005      BNE    32$             ;OK - SOME SERIAL NUM PRESENT
117 022522 005237 013452      INC    NOSNUM          ;NO - SET THE 'NO SERIAL NUMBER' FLAG
118 022526 000401      BR     32$
119 022530 005721          31$:    TST    (R1)+           ;SKIP OVER THE 2ND SERIAL NUM WORD
120 022532 022121          32$:    CMP    (R1)+,(R1)+     ;SKIP PAST THE 'BLANK' WORDS
121 022534 012137 002322          1$:    MOV    (R1)+,TEMP1     ;GET CYLINDER ENTRY WORD
122 022540 100446      BMI    2$                ;IF MINUS - END OF BAD SECTORS
123 022542 005237 002316      INC    DECNT           ;COUNT THIS ENTRY IN THE FILE
124 022546 012137 002324      MOV    (R1)+,TEMP2     ;GET TRACK AND CYLINDER
125 022552 000337 002322      SWAB   TEMP1           ;PUT CYLINDER IN HIGH BYTE
126 022556 000241      CLC
127 022560 006037 002322      ROR    TEMP1           ;ALIGN THE BITS
128 022564 103003      BCC    111$            ;NEED ANOTHER BIT?
129 022566 052737 100000 002322      BIS    #BIT15,TEMP1    ;YUP
130 022574 013712 002322          111$:    MOV    TEMP1,(R2)      ;STORE OFF CYLINDER PART
131 022600 013737 002324 002322      MOV    TEMP2,TEMP1     ;GET SECTOR
132 022606 042737 177700 002322      BIC    #177700,TEMP1   ;LEAVE ONLY SECTOR
133 022614 053712 002322      BIS    TEMP1,(R2)      ;SET IN SECTOR BITS
134 022620 042737 177377 002324      BIC    #177377,TEMP2   ;CLEAR ALL EXCEPT HEAD BIT
135 022626 006237 002324      ASR    TEMP2
136 022632 006237 002324      ASR    TEMP2
137 022636 053722 002324      BIS    TEMP2,(R2)+     ;SET IN HEAD
138 022642 005300      DEC    RO
139 022644 001333      BNE    1$
140 022646 005737 002320      TST    TEMPO           ;PRINT A MESSAGE?
141 022652 001330      BNE    1$                ;NO
142 022654 000423      BR     6$
143
144 022656 005737 002326          2$:    TST    TEMP3
145 022662 001037      BNE    7$                ;SWITCH TO FIELD BAD OR QUIT
                                ;QUIT, 7$

```

```

146 022664 022737 000001 002246      CMP      #1,TDR           ;RL01=1
147 022672 001004                    BNE      350$           ;MUST BE RL02
148 022674 012737 077724 002174      MOV      #77724,BDA     ;START AT FIELD SECTOR
149 022702 000403                    BR       36$
150 022704 012737 177724 002174 350$: MOV      #177724,BDA     ;START OF FIELD AREA FOR RL02
151 022712 012737 000001 002326 36$: MOV      #1,TEMP3
152 022720 000137 022312                    JMP      4$
153 022724                    6$: PRINTF #FMT18,#TBLFUL
    022724 012746 005714      MOV      #TBLFUL,-(SP)
    022730 012746 007335      MOV      #FMT18,-(SP)
    022734 012746 000002      MOV      #2,-(SP)
    022740 010600      MOV      SP,R0
    022742 104017      EMT      C$PNTF
    022744 062706 000006      ADD      #6,SP
154 022750 005237 002320      INC      TEMPO         ;SET THE PRINT FLAG
155 022754 012700 000170      MOV      #120.,R0      ;RESET THE COUNTER
156 022760 000665      BR       1$           ;AND CONTINUE
157
158 022762 005737 002320      7$: TST      TEMPO         ;OVER 25. ENTRIES?
159 022766 001412                    BEQ      9$           ;NO
160 022770                    PRINTF #FMT20,DECNT     ;PRINT # ENTRIES IN FILE
    022770 013746 002316      MOV      DECNT,-(SP)
    022774 012746 007351      MOV      #FMT20,-(SP)
    023000 012746 000002      MOV      #2,-(SP)
    023004 010600      MOV      SP,R0
    023006 104017      EMT      C$PNTF
    023010 062706 000006      ADD      #6,SP
161 023014 012603      9$: MOV      (SP)+,R3
162 023016 012602      MOV      (SP)+,R2
163 023020 012601      MOV      (SP)+,R1
164 023022 012600      MOV      (SP)+,R0
165 023024 000205      RTS      R5
166
167 ;ROUTINE TO READ THE 'FACTORY' FILE FROM THE BAD SECTOR FILE
168
169 023026 004537 023176      RDFACT: JSR     R5,CLRBSF ;CLEAR THE BSFILE STORAGE AREA
170 023032 012737 173000 002176      MOV      #-2560.,BMP    ;SAVE THE WORD COUNT
171 023040 012737 077700 002174      MOV      #77700,BDA     ;AND THE DISK ADDR FOR FACTORY FILE
172 023046 022737 000001 002246      CMP      #1,TDR ;IS IT AN RL02?
173 023054 001423                    BEQ      RDBSFILE      ;NO - READ THE FILE
174 023056 012737 177700 002174      MOV      #177700,BDA   ;HERE FOR RL02
175 023064 000417      BR       RDBSFILE     ;THEN READ THEE FILE
176
177 023066 004537 023176      RDFIELD: JSR    R5,CLRBSF ;CLEAR THE BSFILE STORAGE AREA
178 023072 012737 173000 002176      MOV      #-2560.,BMP    ;SAVE THE WORD COUNT
179 023100 012737 077724 002174      MOV      #77724,BDA     ;AND THE DISK ADDR FOR 'FIELD' FILE
180 023106 022737 000001 002246      CMP      #1,TDR ;IS DRIVE A RL02?
181 023114 001403                    BEQ      RDBSFILE      ;NO - READ THE FILE
182 023116 012737 177724 002174      MOV      #177724,BDA   ;HERE FOR RL02
183
184 023124 012737 000014 002210      RDBSFILE: MOV    #READ,FUNC ;SAVE THE COMMAND
185 023132 004537 021570      JSR      R5,LDFUNC     ;AND ISSUE IT
186 023136 004537 023264      JSR      R5,WTRDY      ;THEN WAIT FOR READY
187 023142 005777 157032      TST      @DCS ;WAS THERE ANY ERROR?
188 023146 100012      BPL      RDBSEX       ;NO - EXIT
189
190 023150      PRINTF #FMT19,#BADBSF ;TELL THE OPR AN ERROR HAPPENED

```

```

023150 012746 004753      MOV      #BADBSF,-(SP)
023154 012746 007342      MOV      #FMT19,-(SP)
023160 012746 000602      MOV      #2,-(SP)
023164 010600      MOV      SP,R0
023166 104017      EMT      C$PNTF
023170 062706 000006      ADD      #6,SP

191
192 023174 000205      RDBSEX: RTS      R5      ;EXIT
193
194 023176 010146      CLRBSF: MOV      R1,-(SP)      ;SAVE R1
195 023200 012701 026442      MOV      #BSFILE,R1      ;SETUP A POINTER
196 023204 012721 177777      1$:      MOV      #-1,(R1)+      ;SET BUFFER & POINT TO NEXT
197 023210 022701 033442      CMP      #BSFILE+2560.,R1      ;DONE?
198 023214 001373      BNE      1$      ;NO - INIT THE NEXT ADDR
199 023216 012601      MOV      (SP)+,R1      ;RESET R1
200 023220 000205      RTS      R5      ;EXIT
201
202 023222      STARS
      ;:*****
      ;LOADED -- CHECK FOR DRV READY
      STARS
      ;:*****

205
206 023222 010146      LOADED: MOV      R1,-(SP)      ;SAVE R1
207 023224 004537 023342      JSR      R5,GETDST      ;GET DRV STATUS
208 023230 005037 002320      CLR      TEMPO      ;CLEAR THE FLAG
209 023234 032701 000020      BIT      #HOP,R1      ;HEADS OVER PACK?
210 023240 001002      BNE      1$      ;YES
211 023242 005237 002320      INC      TEMPO      ;NO
212 023246 032701 000010      1$:      BIT      #BRHM,R1      ;BRUSHES HOME?
213 023252 001002      BNE      2$      ;YES
214 023254 005237 002320      INC      TEMPO
215 023260 012601      2$:      MOV      (SP)+,R1
216 023262 000205      RTS      R5      ;EXIT

```

```

1          .SBTTL ROUTINE TO WAIT FOR CONTROLLER READY
2
3          ;
4          ;ROUTINE TO WAIT FOR CONTROLLER READY UNDER FLAG
5          ;MODE. USED IN INITIALIZE PORTION OF PROGRAM I.E.
6          ;GETTING BAD SECTOR FILE, WRITING PACK INITIALLY
7
8 023264 010046      WTRDY: MOV      R0,-(SP)          ;SAVE REGISTERS
9 023266 010146      MOV      R1,-(SP)
10 023270 012701 001750  MOV      #1000.,R1          ;WAIT A WHILE
11 023274           1$:  WAITUS  #2.
    023274 012700 000002  MOV      #2.,R0
    023300 104027      EMT      C$WTU
12 023302 032777 000200 156670  BIT      #CRDY,@DCS          ;READY SET?
13 023310 001011      BNE      2$          ;YES, EXIT
14 023312 005301      DEC      R1          ;TIMED OUT?
15 023314 001367      BNE      1$          ;NO GO BACK
16
17 023316 017737 156656 002202  MOV      @DCS,E.DCS          ;SAVE THE STATUS FOR ERROR REPORT
18 023324           ERRDF  110.,NOCRDY,ERR2
    023324 104462      TRAP    T$ERCODE
    023326 000156      .WORD   110
    023330 003607      .WORD   NOCRDY
    023332 007114      .WORD   ERR2
19
20 023334 012601      2$:  MOV      (SP)+,R1          ;RESTORE REGISTERS
21 023336 012600      MOV      (SP)+,R0
22 023340 000205      RTS      R5
23
24
25          .SBTTL GET STATUS/DRIVE RESET ROUTINE
26
27          ;ROUTINE TO ISSUE DRIVE RESET
28          ;ALSO GET STATUS, R1 HAS STATUS IF GS
29          ;USES R3, DOES NOT SAVE IT
30
31 023342 013703 002200      GETDST: MOV      DCS,R3
32 023346 012763 000003 000004  MOV      #GSBIT,DA(R3)
33 023354 000405      BR      C$STUFF
34 023356 013703 002200      ISDRST: MOV      DCS,R3
35 023362 012763 000013 000004  MOV      #DRST,DA(R3)
36 023370 012763 000204 000000  CSTUFF: MOV      #CRDY!GSTAT,CS(R3)
37 023376 053763 002232 000000  BIS      DRSEL,CS(R3)
38 023404 042763 000200 000000  BIC      #CRDY,CS(R3)
39 023412 004537 023264      JSR      R5,WTRDY
40 023416 022763 000013 000004  CMP      #DRST,DA(P3)
41 023424 001402      BEQ      1$
42 023426 016301 000006      MOV      MP(R3),R1
43 023432 000205      1$:  RTS      R5
  
```

```

1      .SBTTL  ROUTINE TO WRITE PACKS INITIALLY
2
3      ;ROUTINE TO WRITE PACK WITH PATTERN, ALL TRACKS WILL BE
4      ;WRITTEN (EXCEPT BAD SECTOR TRACK)
5      ;FORMAT IS # OF WORDS (WORD 1), PATTERN ADDRESS (WORD 2)
6      ;PATTERN (WORDS 3 - 128)
7      ;WE WILL ATTEMPT TO WRITE MULTIPLE SECTORS AT A TIME
8      ;(MINIMUM 10 SECTORS) IF AN ERROR OCCURS WE WILL THEN
9      ;WRITE INDIVIDUAL SECTORS FOR THAT TRACK.  WE DO WRITES,
10     ;READS AND INCORE COMPARISONS TO VERIFY.
11
12
13     ;
14     ;CALL JSR R5,WRPACK
15
16     WRPACK:  MOV     R0,-(SP)           ;SAVE REGISTERS
17             MOV     R1,-(SP)
18             MOV     R2,-(SP)
19             MOV     R3,-(SP)
20             MOV     R4,-(SP)
21             MOV     BBA,-(SP)
22     1$:     JSR     R5,HDHOME         ;HEADS HOME
23             MOV     #BUF1,BBA
24             MOV     #-2560.,BMP      ;SET TO WRITE 20 SECTORS
25             JSR     R5,WRBUF         ;GENERATE THE WC DATA PATTERN
26
27     ;NOW ACTUALLY WRITE DATA OUT ON PACK, WILL NOT WRITE LAST
28     ;TRACK
29
30             CLR     R1               ;R1=CYL 000
31             CLR     R4               ;START AT 1ST CYLINDER
32             TST     FWDFLG          ;FORWARD DIRECTION?
33             BEQ     2$              ;YES
34             MOV     #510.,R4        ;SET FOR THE LAST CYL (RL02)
35             CMP     #1,TDR          ;DRIVE = RL01?
36             BNE     2$              ;NO - DA IS OK
37             BIC     #177400,R4      ;YES - MAX CYL IS 255.
38     2$:     JMP     SKWRT           ;SEEK TO THE START CYLINDER
39
40     CONWR:  CMP     #1,TDR          ;RL01=1
41             BNE     13$             ;MUST BE AN RL02
42             CMP     #077600,R1      ;RL01 LAST CYLINDER?
43             BNE     STWRT          ;NO - PROCEED TO WRITE TRACK
44     12$:   TST     TEMP1           ;ON HEAD 1 LAST TRACK?
45             BEQ     STWRT          ;NO - WRITE HEAD 0 LAST TRACK
46             BR     ENDWR
47     13$:   CMP     #177600,R1      ;LAST CYL FOR RL02?
48             BNE     STWRT          ;NO - GO WRITE TRACK
49             BR     12$             ;YES - TEST FOR LAST TRACK ON LAST CYL
50
51     ;HERE WHEN ALL DONE WRITING THE PACK
52
53     ENDWR:  MOV     (SP)+,BBA
54             MOV     (SP)+,R4
55             MOV     (SP)+,R3
56             MOV     (SP)+,R2
57             MOV     (SP)+,R1
58             MOV     (SP)+,R0

```

```

58 023610 000205          RTS      R5          ;END EXIT
59
60                          ;THIS PORTION WILL WRITE THE PACK USING MULTIPLE SECTORS IF A
61                          ;ERROR OCCURS WE WILL GO TO 2$ AND INDIVIDUAL SECTORS.
62                          ;IF AFTER 3 RETRYS ON A SECTOR NO RECOVERY CAN BE MADE, THEN THE SECTOR WILL
63                          ;BE MARKED 'BAD' IN THE TEMP BAD SEC FILE STORAGE AREA
64
65 023612 005002          STWRT:  CLR      R2          ;INITIAL SECTOR 0
66 023614 010137 002174  SWRT1:  MOV      R1,BDA       ;SET UP CYLINDER
67 023620 053737 002322 002174  BIS      TEMP1,BDA      ;INSERT THE HEAD NUMBER (0 OR 1)
68 023626 050237 002174          BIS      R2,BDA       ;ADD IN THE SECTOR NUMBER
69 023632 012737 000012 002210  MOV      #WRITE,FUNC   ;WRITE CMD
70 023640 004537 021570          JSR      R5,LDFUNC     ;ISSUE THE WRITE
71 023644 004537 023264          JSR      R5,WTRDY     ;WAIT FOR READY
72
73 023650 005777 156324          TST      @DCS         ;ERROR DETECTED?
74 023654 100030          BPL      WNXSEC       ;BR IF NO ERROR - GET NEXT SECTOR
75
76                          ;HERE IF AN ERROR WAS DETECTED - GOING TO WRITE THE TRACK ONE SECTOR
77                          ;AT A TIME ... >3 RETRYS = 'BAD' SECTOR
78
79 023656          PRINTF  #MCRLF
    023656 012746 007552  MOV      #MCRLF,-(SP)
    023662 012746 000001  MOV      #1,-(SP)
    023666 010600          MOV      SP,R0
    023670 104017          EMT      C$PNTF
    023672 062706 000004  ADD      #4,SP
80 023676 017737 156276 002202  MOV      @DCS,E.DCS   ;SAVE THE ERRJR DETECTED
81 023704 013737 002174 002314  MOV      BDA,CHKSEC
82 023712          ERRSOFT 410.,MSFER,ERR1
    023712 104464          TRAP    T$ERCODE
    023714 000632          .WORD  410
    023716 003144          .WORD  MSFER
    023720 006626          .WORD  ERR1
83 023722 004537 023356          JSR      R5,ISDRST   ;RESET THE DRIVE
84 023726 012737 000001 002316  MOV      #1,DECNT
85 023734 000471          BR       W1SEC       ;WRITE 1 SECTOR AT A TIME
86
87                          ;HERE TO SELECT THE NEXT SECTOR ON THIS TRACK
88
89 023736 062702 000024  WNXSEC: ADD     #20.,R2   ;NEXT GRGUP
90 023742 022702 000050          CMP     #40.,R2     ;DONE?
91 023746 001322          BNE     SWRT1       ;NO, GO BACK
92 023750 005237 002170          INC     WRTCNT      ;COUNT THIS WRITE PASS ON SELECTED TRK
93 023754 023737 002170 007642  CMP     WRTCNT,WRTLIM ;AT LIMIT FOR THIS TRACK?
94 023762 001313          BNE     STWRT       ;NO - DO THIS TRACK AGAIN
95 023764 005037 002170          CLR     WRTCNT      ;YES - CLEAR THE PASS COUNTER
96
97                          ;HERE TO SELECT THE NEXT TRACK WITH A SEEK CMD
98
99 023770 005737 007640  WNXTRK: TST     WRTSAW   ;DOING A SAWTOOTH WRITE CYCLE?
100 023774 001410          BEQ     3$          ;NO - DO INCREMENTAL
101 023776 005737 002242          TST     FWDFLG     ;SAWTOOTH FWD WRT?
102 024002 001003          BNE     2$          ;NO - DOING REVERSE WRT
103
104 024004 004537 024700          1$:  JSR      R5,HDHOME ;YES - SET THE HEADS OVER CYL #000
105 024010 000402          BR      3$
    
```

```

106
107 024012 004537 024776      2$:   JSR      R5,MDLAST      ;SET THE HEADS OVER THE LAST CYL
108
109 024016 005737 002322      3$:   TST      TEMP1          ;DOING HEAD 0 ??
110 024022 001432                BEQ      5$                  ;YES - SET FOR HEAD #1
111 024024 005737 002242                TST      FWDFLG            ;FWD WRITE?
112 024030 001404                BEQ      31$                ;YES - R4 IS AN UPCOUNTER
113 024032 005304                DEC      R4                 ;NO - DOWNCOUNT R4 (CYL COUNTER)
114 024034 002003                BGE     32$                ;PROCEED IF STILL HAVE SOME TO DO
115 024036 000137 023572                JMP      ENDWR             ;JUST COMPLETED THE PACK
116
117 024042 005204                31$:   INC      R4           ;POINT TO THE NEXT CYLINDER (FWD DIRECTION)
118
119                024044      SKWRT=.
120 024044 005037 002322      32$:   CLR      TEMP1         ;SET POINTER BACK TO HEAD #0
121
122 024050 010401                4$:   MOV      R4,R1          ;GET THE CYLINDER #
123 024052 000301                SWAB    R1                 ;POSITION THE BITS FOR DIRECT LOADING
124 024054 000241                CLC                                ;INTO THE DA REGISTER
125 024056 006001                ROR     R1                 ;FOR THE SEEK TO THE PROPER
126 024060 103002                BCC     41$                ;CYLINDER
127 024062 052701 100000                BIS     #BIT15,R1
128 024066 010137 002254                41$:   MOV     R1,NEWPOS      ;SET THE DESIRED DISK ADDRESS
129 024072 053737 002322 002254                BIS     TEMP1,NEWPI"      ;ADD IN THE SELECTED HEAD BIT
130 024100 004537 025062                JSR     R5,SKFNC          ;ISSUE THE SEEK TO THE DESIRED CYLINDER/HEAD
131 024104 000137 023534                JMP     CONWR             ;AND CONTINUE WRITING THE PACK
132
133 024110 012737 000100 002322 5$:   MOV     #HEAD,TEMP1      ;POINT TO HEAD #1
134 024116 000754                BR     4$                 ;AND SEEK THERE
135
136                ;IF AN ERROR OCCURS THEN WE COME HERE AND DO THE TRACK SECTOR
137                ;BY SECTOR.
138
139 024120 005002                WISEC: CLR     R2          ;R2 = SECTOR
140 024122 012737 177600 002176                MOV     #-128.,BMP       ;LOAD WORD COUNT
141 024130 013737 002252 002174 1$:   MOV     PRPOS,BDA        ;SETUP DISK ADDRESS
142 024136 053737 002322 002174                BIS     TEMP1,BDA        ;ADD IN THE HEAD NUMBER (0 OR 1)
143 024144 050237 002174                BIS     R2,BDA          ;ADD IN THE SECTOR NUMBER
144
145                ;HERE TO WRITE A SECTOR
146
147 024150 012737 000012 002210 2$:   MOV     #WRITE,FUNC      ;WRITE FUNCTION
148 024156 004537 021570                JSR     R5,LDFUNC        ;ISSUE THE WRITE
149 024162 004537 023264                JSR     R5,WTRDY        ;WAIT FOR WRITE TO FINISH
150
151 024166 005777 156006                TST     @DCS             ;ERROR ON WRITE?
152 024172 100055                BPL     3$               ;NO - SETUP FOR NEXT SECTOR
153
154                ;HERE IF ERROR ON 1 SECTOR WRITE
155
156 024174 017737 156000 002202                MOV     @DCS,E.DCS      ;SAVE THE DETECTED ERROR
157 024202 005237 002316                INC     DECNT            ;NO, GIVE IT ONE MORE TRY
158 024206 013737 002174 002314                MOV     BDA,CHKSEC      ;CHECK IF SECTOR IS IN
159 024214 004537 025252                JSR     R5,CKBDSC       ;BAD SECTOR FILE
160 024220 005737 002312                TST     HDRFND          ;IF SET, IT WAS
161 024224 001425                BEQ     21$              ;NO MATCH
162 024226                PRINTF #FMT18,#INBSF    ;TELL OPR SECT IN FILE ALREADY

```

```

024226 012746 006250      MOV      #INBSF,-(SP)
024232 012746 007335      MOV      #FMT18,-(SP)
024236 012746 000002      MOV      #2,-(SP)
024242 010600      MOV      SP,RO
024244 104017      EMT      C$PNTF
024246 062706 000006      ADD      #6,SP
163 024252      PRINTF  #M(CRLF
024252 012746 007552      MOV      #M(CRLF,-(SP)
024256 012746 000001      MOV      #1,-(SP)
024262 010600      MOV      SP,RO
024264 104017      EMT      C$PNTF
024266 062706 000004      ADD      #4,SP
164 024272 004537 023356      JSR      R5,ISDRST      ;RESET THE DRIVE
165 024276 000432      BR       31$           ;WORK ON NEXT SECTOR
166
167 024300 022737 000004 002316 21$:  CMP      #4,DECNT      ;IT MAY HAVE BEEN NOISE.
168 024306 001403      BEQ      22$           ;HARD ERROR?
169 024310 004537 023356      JSR      R5,ISDRST      ;NO - ISSUE A DRIVE RESET
170 024314 000715      BR       2$           ;AND TRY AGAIN
171
172 024316 004537 024400      22$:  JSR      R5,INBAD      ;TELL OPR & PUT IT IN TEMP STORAGE
173 024322 004537 023356      JSR      R5,ISDRST      ;RESET THE DRIVE
174 024326 005737 002316      3$:  TST      DECNT        ;ANY RECOVERY HERE?
175 024332 001414      BEQ      31$           ;NO
176 024334      PRINTF  #FMT18,#MSREC ;YES - TELL OPR 'RECOVERED'
024334 012746 002762      MOV      #MSREC,-(SP)
024340 012746 007335      MOV      #FMT18,-(SP)
024344 012746 000002      MOV      #2,-(SP)
024350 010600      MOV      SP,RO
024352 104017      EMT      C$PNTF
024354 062706 000006      ADD      #6,SP
177 024360 005237 002162      INC      SFTCNT        ;ADD 1 TO THE SOFT ERROR COUNT
178
179      ;SELECT THE NEXT SECTOR
180
181 024364 005202      31$:  INC      R2            ;POINT TO THE NEXT SECTOR
182 024366 020227 000050      CMP      R2,#40.       ;END OF THE TRACK?
183 024372 003656      BLE     1$            ;NO - DO THIS SECTOR
184 024374 000137 023770      JMP      WNXTRK        ;YES - DO NEXT TRACK

```



```

1 024400          STARS
2                ;*****
3                ;INBAD -- ROUTINE TO INSERT THE BAD SECTOR FOUND INTO A TEMP
4                ;          BAD SECTOR FILE AND TELL OPR THAT A BAD SECTOR (HARD ERR)
5 024400          ;          WAS DETECTED.
6                ;*****
7 024400 010146    INBAD:  MOV     R1,-(SP)          ;SAVE R1
8 024402 016337 000000 002340    MOV     CS(R3),E.CS
9 024410 016337 000002 002342    MOV     BA(R3),E.BA
10 024416 016337 000004 002344    MOV     DA(R3),E.DA
11 024424 016337 000006 002346    MOV     MP(R3),E.MP
12 024432 016337 000006 002350    MOV     MP(R3),E.MP1
13 024440 016337 000006 002352    MOV     MP(R3),E.MP2
14 024446 005037 002316    CLR     DECNT          ;CLEAR CURRENT SOFT ERROR COUNT
15 024452          ERRHRD 300.,MHDER,ERR1
    024452 104463    TRAP   T$ERCODE
    024454 000454    .WORD  300
    024456 003242    .WORD  MHDER
    024460 006626    .WORD  ERR1
16 024462 005237 002160    INC     ERRCNT          ;UPDATE THE HARD ERROR COUNT
17 024466 012701 025726    MOV     #BSECN,R1      ;POINT TO THE BAD SECTOR TEMP STORAGE
18 024472 005711          IBDN:  TST     (R1)      ;LOOK FOR A SPOT TO INSERT ENTRY
19 024474 100417          BMI     IBDN1          ;BR IF FOUND ONE
20 024476 005721          TST     (R1)+         ;POINT TO NEXT ENTRY ADDR
21 024500 022701 026010    CMP     #BSECN+50.,R1  ;END OF TABLE?
22 024504 001372          BNE     IBDN          ;NO - TRY THIS ENTRY SLOT
23 024506          PRINTF #FMT18,#TBLFUL ;YES - TELL OPR END OF TABLE (25. ENTRYS FOUND)
    024506 012746 005714    MOV     #TBLFUL,-(SP)
    024512 012746 007335    MOV     #FMT18,-(SP)
    024516 012746 000002    MOV     #2,-(SP)
    024522 010600          MOV     SP,R0
    024524 104017          EMT    C$PNTF
    024526 062706 000006    ADD     #6,SP
24 024532 000402          BR     IBDN2          ;EXIT
25 024534 013711 002314    IBDN1: MOV     CHKSEC,(R1) ;SAVE THE ENTRY IN TABLE
26 024540 012601          IBDN2: MOV     (SP)+,R1   ;RESET R1
27 024542 000205          RTS     R5
  
```

```

1          .SBTTL ROUTINE FOR SYSTEM CLOCK
2
3          ;ROUTINE TO READ SYSTEM CLOCK
4          ;USES 'REGTIM' FROM DIAGNOSTIC SUPERVISOR
5
6 024544 010046 GETSYS: MOV     R0,-(SP)      ;SAVE R0
7 024546 010146      MOV     R1,-(SP)      ;SAVE R1
8 024550 005737 002354  TST     SYSCLK      ;DO WE HAVE A CLOCK
9 024554 001002      BNE     4$           ;YES, GO SERVICE IT
10 024556      BREAK      ;NO, CALL SUPER FOR ^C
   024556 104022      EMT     C$BRK      ;
11 024560 000444      BR      3$           ;EXIT
12
13 024562      4$:  REQTIM R0              ;GET PRESENT TIME
   024562 104045      EMT     C$REQTIM
14 024564 020037 002330 1$:  CMP     R0,LSTTIM      ;HAS IT MOVED
15 024570 001440      BEQ     3$           ;NO MOVEMENT SINCE LAST CALL
16 024572 003002      BGT     5$           ;BR IF CURRENT TIME > LAST TIME
17 024574 005037 002330  CLR     LSTTIM      ;WHOOPS - NEVER LET CURRENT BE < LAST TIME
18 024600 013701 002330 5$:  MOV     LSTTIM,R1      ;CALCULATE DIFFERENCE
19 024604 010037 002330  MOV     R0,LSTTIM      ;AND FIX ACCORDINGLY
20 024610 160100      SUB     R1,R0
21 024612 060037 002332 2$:  ADD     R0,SECOND      ;BUMP SECONDS
22 024616 022737 000074 002332  CMP     #60.,SECOND      ;SECONDS OVERFLOW
23 024624 003022      BGT     3$
24 024626 162737 000074 002332 7$:  SUB     #60.,SECOND
25 024634 005237 002334      INC     MINUTE      ;BUMP MINUTES
26 024640 022737 000074 002332  CMP     #60.,SECOND
27 024646 003767      BLE     7$
28 024650 022737 000074 002334  CMP     #60.,MINUTE
29 024656 003005      BGT     3$
30 024660 005237 002336      INC     HOUR
31 024664 162737 000074 002334  SUB     #60.,MINUTE
32 024672 012601 3$:  MOV     (SP)+,R1      ;RESET R1
33 024674 012600      MOV     (SP)+,R0      ;RESET R0
34 024676 000205      RTS     R5

```

```

1          .SBTTL  HEADS HOME ROUTINE
2
3          ;ROUTINE TO BRING HEADS OVER TRACK 0
4
5 024700 010046          HDHOME: MOV    R0,-(SP)          ;SAVE R0
6 024702 012737 000010 002210  MOV    #RDHDR,FUNC        ;READ HEADER
7 024710 004537 021570          JSR    R5,LDFUNC         ;GO DO IT.
8 024714 004537 023264          JSR    R5,WTRDY
9
10 024720 016300          MOV    MP(R3),R0         ;GET HEADER
11 024724 042700          BIC    #177,R0          ;ONLY CYLINDER
12 024730 010037          MOV    R0,BDA           ;MOVE IT TO BUFFERED DA
13 024734 052737 000001 002174  BIS    #MK,BDA         ;SET MARKER
14 024742 012737 000006 002210  MOV    #SEEK,FUNC      ;LOAD SEEK
15 024750 004537 021570          JSR    R5,LDFUNC         ;SEEK!
16 024754 004537 023264          JSR    R5,WTRDY         ;WAIT.
17 024760 013737 002252 002214  MOV    PRPOS,LSTHDR
18 024766 005037 002252          CLR    PRPOS           ;SET BUFFER TO HOME
19 024772 012600          MOV    (SP)+,R0
20 024774 000205          RTS    R5
21
22          ;ROUTINE TO SET THE HEADS OVER THE LAST CYLINDER
23
24 024776 012737 000010 002210  HDLAST: MOV    #RDHDR,FUNC ;SET TO READ THE CURRENT POSITION
25 025004 004537 021570          JSR    R5,LDFUNC         ;READ HEADERS
26 025010 004537 023264          JSR    R5,WTRDY         ;WAIT TILL DONE
27
28 025014 016337 000006 002252  MOV    MP(R3),PRPOS    ;GET THE CURRENT POSITION
29 025022 042737 000177 002252  BIC    #177,PRPOS      ;SAVE ONLY THE CYL BITS
30 025030 012737 177600 002254  MOV    #177600,NEWPOS  ;SET LAST CYL FOR RL02
31 025036 022737 000001 002246  CMP    #1,TDR          ;DRIVE = RL01?
32 025044 001003          BNE    1$             ;NO - MUST BE RL02
33 025046 012737 077600 002254  MOV    #77600,NEWPOS   ;YES - SET RL01 LAST TRACK ADDRESS
34 025054 004537 025062          1$: JSR    R5,SKFNC       ;SEEK TO THE LAST TRACK
35 025060 000205          RTS    R5

```

```

1      .SBTTL  SEEK ROUTINE
2      ;ROUTINE TO SEEK TO A CYLINDER POINTED TO BY 'NEWPOS' FROM A CYLINDER
3      ;POINTED TO BY 'PRPOS'
4      ;EXITS WITH PRPOS CONTAINING THE NEW CYLINDER ADDRESS
5
6 025062 010146      SKFNC:  MOV    R1,-(SP)      ;SAVE R1
7 025064 010246      MOV    R2,-(SP)      ;SAVE R2
8 025066 013702 002254  MOV    NEWPOS,R2    ;SET THE DESIRED CYL
9 025072 013701 002252  MOV    PRPOS,R1     ;GET THE CURRENT POSITION
10 025076 042701 000177 BIC    #177,R1      ;CLEAR THE HEAD/SECTOR BITS
11 025102 042702 000177 BIC    #177,R2
12 025106 160102      SUB    R1,R2        ;CALC THE DIFFERENCE
13 025110 103002      BCC    1$           ;MAKE DIFFERENCE A POSITIVE NUMBER
14 025112 005402      NEG    R2
15 025114 000402      BR     2$
16 025116 052702 000004 1$:  BIS    #4,R2        ;SET THE DIRECTION BIT
17 025122 052702 000001 2$:  BIS    #MK,R2      ;SET THE SEEK MARKER BIT
18 025126 032737 000100 002254 BIT    #HEAD,NEWPOS ;GO TO HEAD #1?
19 025134 001402      BEQ    3$           ;NO
20 025136 052702 000020 3$:  BIS    #SKHS,R2    ;YES - SELECT THE HEAD BIT
21 025142 010237 002174  MOV    R2,BDA      ;SAVE THE DA
22 025146 010237 002222  MOV    R2,DIFWD    ;ALSO AS DIFFERENCE WORD
23 025152 012737 000006 002210  MOV    #SEEK,FUNC  ;SET TO DO A SEEK FUNCTION
24 025160 004537 021570  JSR    R5,LDFUNC   ;ISSUE THE SEEK
25 025164 004537 023264  JSR    R5,WTRDY    ;WAIT TILL READY SET
26 025170 005777 155004  TST    @DCS        ;SEEK ERROR DETECTED?
27 025174 100007      BPL    31$         ;NO
28 025176 017737 154776 002202  MOV    @DCS,E.DCS  ;YES - SAVE THE ERROR STATUS
29 025204      ERRHRD 530.,MSKER,ERR2
   025204 104463      TRAP  T$ERCODE
   025206 001022      .WORD 530
   025210 003133      .WORD MSKER
   025212 007114      .WORD ERR2
30
31 025214 013737 002254 002252 31$:  MOV    NEWPOS,PRPOS ;UPDATE THE CURRENT POSITION WORD
32 025222 012602      4$:  MOV    (SP)+,R2    ;RESET R2
33 025224 012601      MOV    (SP)+,R1    ;RESET R1
34 025226 000205      RTS    R5          ;EXIT
35
36      ;ROUTINE TO CLEAR ALL DRIVE INFO, USED ON START OR
37      ;RESTART IF CALLED. CAN BE USED TO CLEAR INDIVIDUAL DRIVE
38      ;INFO BY BITMAP FOLLOWING CALL
39      ;CALL JSR R5,CLEAR
40      ;
41
42
43 025230 010446      CLEAR: MOV    R4,-(SP)      ;SAVE R4
44 025232 012704 002160  MOV    #ERRCNT,R4  ;POINT TO THE 1ST TO CLEAR
45 025236 005024      2$:  CLR    (R4)+      ;CLEAR
46 025240 020427 002256  CMP    R4,#RECNT   ;AT END OF BUFFER
47 025244 001374      BNE    2$         ;NO, GO TO 2$
48 025246 012604      4$:  MOV    (SP)+,R4    ;RESTORE CURRENT BUFFER POINTER
49 025250 000205      RTS    R5          ;EXIT
50
51      .SBTTL  ROUTINE TO CHECK FOR BAD SECTOR
52
53      ;ROUTINE TO MATCH BAD SECTOR.....BDA IS SECTOR WE ARE LOOKING

```

```

54                                     ;FOR IN LIST POINTED TO BY BSECT.....HDRFND IS SET IF WE FIND IT.
55                                     ;
56
57 025252 005037 002312      CKBDSC: CLR      HDRFND      ;CLEAR FLAG
58 025256 010046              MOV      RO,-(SP)    ;SAVE RO
59 025260 010146              MOV      R1,-(SP)    ;SAVE R1
60 025262 0127C0 000177      MOV      #127.,RO    ;127 ENTRIES
61 025266 013701 002234      1$: MOV      BSECT,R1 ;GET WHERE WE'RE LOOKING
62 025272 022711 177777      2$: CMP      #-1,(R1) ;END OF ENTRYS?
63 025276 001411              BEQ      4$          ;BRANCH IF AT END
64 025300 023711 002314      CMP      (CHKSEC,(R1) ;HAVE WE GOT A MATCH
65 025304 001404              BEQ      3$          ;THEN GO SET INDICATOR, ELSE
66 025306 005721              TST      (R1)+
67 025310 005300              DEC      RO
68 025312 001367              BNE      2$
69 025314 000402              BR       4$
70
71 025316 005237 002312      3$: INC      HDRFND      ;SET FLAG FOUND
72
73 025322 012601              4$: MOV      (SP)+,R1
74 025324 012600              MCV     (SP)+,RO
75 025326 000205              RTS      R5
76
77 025330
78 025330
79
80
81 025330
82 025724 177777
83
84 025726
85 026322 177777
86 026324
87 026324
88
89 026324
90
91
92 026324
93
94 026324 010146
95 026326 010246
96 026330 010346
97
98 026332 013701 002176
99 026336 013702 002356
100 026342 012703 026374
101 026346 012322
102 026350 005201
103 026352 001404
104 026354 022703 026434

STARS
:*****
STARS
:*****
;BUFFER TO STORE BAD SECTOR LISTS
BSECO: .BLKW 126. ;STORAGE FOR BAD SPOTS IN BAD SECTOR FILE
       .WORD -1 ;FORCED TERMINATOR
BSECN: .BLKW 126. ;STORAGE FOR 'FOUND' BAD SPOTS
BSECNE: .WORD -1 ;FORCED TERMINATOR
STARS
:*****
STARS
:*****
STARS
:*****
;SUBROUTINE TO LOAD A MEMORY BUFFER WITH THE WORST CASE DATA PATTERN
;TO WRITE ON THE PACK.
STARS
:*****
WRBUF: MOV      R1,-(SP) ;SAVE R1
       MOV      R2,-(SP) ;SAVE R2
       MOV      R3,-(SP) ;AND R3
       MOV      BMP,R1 ;GET THE WORD COUNT FOR THE WRITE CMD
       MOV      BUF1,R2 ;GET THE BUFFER ADDRESS
1$: MOV      #WCPAT,R3 ;GET THE STARTING ADDRESS OF THE DATA PATTERN
2$: MOV      (R3)+,(R2)+ ;PUT THE DATA IN MEMORY BUFFER
       INC      R1 ;DOWNCOUNT THE WC (MINUS WC TO START WITH)
       BEQ      3$ ;EXIT IF ALL DONE BUILDING THE BUFFER
       CMP      #WCPAT+32.,R3 ;AT THE END OF THE DATA PATTERN TABLE?
  
```

```

105 026360 001372          BNE      28          ;NO - STORE THE NEXT FROM DATA TABLE
106 026362 000767          BR       18          ;YES - RESET THE DATA TABLE POINTER
107 026364 012603          38:     MOV      (SP)+,R3      ;RESET R3
108 026366 012602          MOV      (SP)+,R2
109 026370 012601          MOV      (SP)+,R1
110 026372 000205          RTS      R5          ;EXIT
111
112
113                          ;WORST CASE PATTERN USED IN WRITING
114
115 026374 155555          WCPAT:  .WORD    155555
116 026376 066666          .WORD    066666
117 026400 133333          .WORD    133333
118 026402 155555          .WORD    155555
119 026404 066666          .WORD    066666
120 026406 133333          .WORD    133333
121 026410 155555          .WORD    155555
122 026412 066666          .WORD    066666
123 026414 133333          .WORD    133333
124 026416 155555          .WORD    155555
125 026420 066666          .WORD    066666
126 026422 133333          .WORD    133333
127 026424 155555          .WORD    155555
128 026426 066666          .WORD    066666
129 026430 133333          .WORD    133333
130 026432 155555          .WORD    155555
131
132 026434 000240          ENDOFPROGRAM:  NOP
133 026436 026436          ENDTST
      026436 104001          L10011:      EMT      CSETST
134 026440 000000          HALT
135
136 026442          STARS
      ;:*****
137 026442          STARS
      ;:*****
138 026442          BSFILE:  .BLKW   2560.          ;STORAGE FOR BAD SECTOR FILE DATA
139 040442 177777          .WORD    -1          ;END OF STORAGE
140
141 040444          BSFPAT:  .BLKW   20.          ;EXTRA FOR PATCHING & MISTAKES ETC.
142 040514          STARS
      ;:*****
143 040514          STARS
      ;:*****
144
145 040514          BGNMOD  HRDPRM
146 040514          BGNHRD
      040514 000011          .WORD    L10013-L$HARD/2
147 040516          GPRMA   CSRMSG,CSR,0,160000,177776,YES
      040516 000031          .WORD    T$CODE
      040520 040540          .WORD    CSRMSG
      040522 160000          .WORD    T$LOLIM
      040524 177776          .WORD    T$HILIM
148 040526          GPRMD   DRMSG,DRBT,0,03400,0,7,YES
      040526 001032          .WORD    T$CODE
      040530 040554          .WORD    DRMSG
  
```

```

040532 003400 .WORD 03400
040534 000000 .WORD T$LOLIM
040536 000007 .WORD T$HILIM
149 040540 ENDHRD
      .EVEN
      040540 L10013:
150
154
155 040540 102 125 123 CSRMSG: .ASCIZ /BUS ADDRESS/
156 040554 104 122 111 DRMSG: .ASCIZ /DRIVE/
157
161
162 .EVEN
163
164 040562 ENDMOD
165
166 040562 BGNMOD SFTPRM
167 040562 BGNSFT
      040562 000010 .WORD L10014-L$SOFT/2
168 040564 GPRML DSWRT,0,1,YES
      040564 000130 .WORD T$CODE
      040566 040604 .WORD DSWRT
      040570 000001 .WORD 1
169 040572 GPRMD DWCNT,2,D,177777,1,177777,YES
      040572 001052 .WORD T$CODE
      040574 040632 .WORD DWCNT
      040576 177777 .WORD 177777
      040600 000001 .WORD T$LOLIM
      040602 177777 .WORD T$HILIM
170 040604 ENDSFT
      .EVEN
      040604 L10014:
174 .EVEN
175 040604 123 101 127 DSWRT: .ASCIZ /SAWTOOTH WRITE CYCLE?/
176 040632 127 122 111 DWCNT: .ASCIZ /WRITE CYCLES PER TRACK?/
177 .EVEN
181 040652 ENDMOD
182 041114 .=41114
183 041114 LASTAD
      .EVEN
      041114 L$LAST::
```

1

.SBTTL DIAGNOSTIC SUPERVISOR -- LOW CORE SET UP



```

35 071710 000000      .WORD 0      ;SPACE FOR USER POOL POINTER
36 071712 000000      .WORD 0      ;SIZE
37 071714 000000      .WORD 0      ;CHECKSUM (NOT CURRENTLY USED)
38 071716 000000      .WORD 0      ;SIZE OF H.W. PTAB. ALLOCATION
39          071722
40          000200      .END 200
END.SUPV-.+2

```

ASSEMBLY ROUTINES  
SYMBOL TABLE

ABOFLA	041440	G	BIT4	=	000020	G	CALLPS=	000024	CRD2	017374	CSQIO	=	000377	
ABOPAS	041356	G	BIT5	=	000040	G	CALLSP=	000026	CR LCS	002415	CSRDBU=	000007		
ABO.FM	043720		BIT6	=	000100	G	CALLTC=	000030	CRLF	060572	CSREFG=	000050		
ABSCYL	005114		BIT7	=	000200	G	CAL.CL	066602	CS	=	000000	CSREQT=	000045	
ABSHD	005175		BIT8	=	000400	G	CAL.TI	066640	CSR	=	000000	CSRESE=	000033	
ABSMG	005024		BIT9	=	001000	G	CART	002427	CSRMSG	040540	CSREVI=	000002		
ABSSEC	005146		BLD.HW	046602		CHKLUP	050220	CSTUFF	023370	CURR.S	041122	CSRPT	=	000025
ABSSER	005220		BLOCK	064214		CHKSEC	002314	CURR.T	041124	G	CSSEFG=	000047		
ABSSNH	005331		BMP	002176		CHKSTR	062422	CVERIF	016744		CSSPRI=	000041		
ABSSNL	005272		BPRIOR	002302		CHKTY	060510	CVFLG	002244		CSVSEC=	000037		
ACCESS	002372		BRHM	=	000010	CHK.MA	046360	CYLSK	002264		CSTPRI=	000013		
ACKENT	014632		BSADD	014102		CHK.PC	053510	CSAAD	053462		CSUNBU=	000031		
ADDCOD	010454	G	BSADDL	014150		CHK.SW	042550	CSAAE	053474		CSWTM	=	000026	
ADDFEX	020534		BSADDS	014140		CHRCNT	061742	CSAAK	054472		CSWTU	=	000027	
ADDFND	020142		BSASN	014260		CH.FLA	046066	CSAAL	054636		DA	=	000004	
AFSI	041146	G	BSATD	014236		CH.PAS	046104	CSABRT=	000021		DCRC	=	004000	
ALLOC	062060		BSDEL	015454		CKBDSC	025252	CSADR	=	000020	DCS		002200	
APT.ER	043050		BSDELL	015522		CKFACT	006305	CSAU	=	000054	DECMSG	060404		
ASSEMB=	000011		BSDELS	015512		CKFLD	006344	CSBRK	=	000022	DECNT	002316		
ASAAV	045716		BSDELT	015610		CLEAR	025230	CSBSEG=	000004		DEL CYL	005114		
ASAAW	045732		BSDEL1	015632		CLEAR.	047502	CSB SUB=	000002		DELHD	005175		
ASAAZ	045744		BSECN	025726		CLKACC	041354	G	CSBUFF=	000030	DELMSG	005520		
ASAAZ	045752		BSECNE	026322		CLKBFR	066604		CSCEFG=	000046	DELSEC	005146		
ASAAZ	045766		BSECPT	002234		CLKCNT	041352	G	CSCLEA=	000012	DENIED	003747		
ASABA	045776		BSECO	025330		CLKJUM	067210	G	CSCLP1=	000006	DERCNT	002166		
BA	=	000002	BSEND	003113		CLKRES	070212	G	CSVVEC=	000036	DERR	=	040000	
BADBSF	004753		BSFCYL	013460		CLKSER	070346	G	CSDCLN=	000044	DEV.CO	041126	G	
BADWRT	005447		BSFHD	013464		CLKSON	041412	G	CSDODU=	000053	DIAGMC=	000000		
BA16	=	000020	BSFILE	026442		CLK.SE	046162		CSDRPT=	000024	DIAG.T	041446	G	
BA17	=	000040	BSFOK	013750		CLNCOD	010372	G	CSDU	=	000055	DIFWD	002222	
BBA	002206		BSFOKF	013474		CLRBSF	023176		CSERDF=	000002	DLT	=	010000	
BCSADR	002212		BSFOKX	014100		CLRBSN	016720		CSERHR=	000003	DOWRT	005371		
BCSR	002306		BSFPAT	040444		CLRDAT	010106		CSERSF=	000001	DPDVD	071056	G	
BDA	002174		BSFSEC	013462		CLRSTB	010122		CSERSO=	000004	DPMUL	070744	G	
BDRSEL	002310		BSMAKE	021054		CLR.MA	046436		CSSECA=	000010	DRBT	=	000002	
BGN.SU=	041114		BSMKL	021122		CMDDO	004613		CSSESEG=	000005	DRDRV	010774		
BINMSG	060370		BSMKS	021112		CMD1	004152		CSSETST=	000001	DRDY	=	000001	
BIT0	=	000001	G	BSRF	004705	CMD2	004222		CSSETST=	000001	DRMSG	040554		
BIT00	=	000001	G	BSRFAC	012134	CMD3	004303		CSSETST=	000001	DRNM	003645		
BIT01	=	000002	G	BSRFLD	012630	CMD4	004371		CSSETST=	000001	DRNRDY	010736		
BIT02	=	000004	G	BSRM	004635	CMD5	004426		CSGPHR=	000042	DROPCO	010456	G	
BIT03	=	000010	G	BSRPT	011770	CMD6	004514		CSGPRI=	000040	DRSEL	002232		
BIT04	=	000020	G	BSRPTL	012012	CMD7	004550		CSGTIM=	000052	DRST	=	000013	
BIT05	=	000040	G	BSRPTS	012002	CMJG	002447		CSINIT=	000011	DRVID	010642		
BIT06	=	000100	G	BSRTOT	013322	CMSK	002270		CSINLP=	000020	DSE	=	000400	
BIT07	=	000200	G	BSVERI	016752	CNTFLG	002400		CSKWOFF=	000035	DSPCOD	007644	G	
BIT08	=	000400	G	BSVERL	017032	CNVT	064660		CSKWOFF=	000035	DSWRT	040604		
BIT09	=	001000	G	BSVERS	017022	COMMAN	041164	G	CSKWOFF=	000035	DUNIT.	041362	G	
BIT1	=	000002	G	BSWRT	016300	COMMTA	064474		CSKWOFF=	000035	DVC.FI	054442		
BIT10	=	002000	G	BSWRTL	016370	COMVER	016756		CSKWOFF=	000035	DWCNT	040632		
BIT11	=	004000	G	BSWRTS	016360	CONREA	017162		CSKWOFF=	000035	DSAAG	055346		
BIT12	=	010000	G	BUF1	002356	CONRCL	070272	G	CSKWOFF=	000035	DSAAM	055364		
BIT13	=	020000	G	BUILD	003050	CONWR	023534		CSKWOFF=	000035	DSAAJ	060132		
BIT14	=	040000	G	BVEC	002300	COP	=	000040	CSKWOFF=	000035	DSAAJ	060136		
BIT15	=	100000	G	B\$AAB	050204	COPY	020536		CSKWOFF=	000035	DSAAK	060154		
BIT2	=	000004	G	B\$AAF	050116	CRDY	=	000200	CSKWOFF=	000035	DSAAK	060172		
BIT3	=	000010	G	CALLPC=	000022	CRD1	017200		CSKWOFF=	000035	DSAAK	060172		

ASSEMBLY ROUTINES  
SYMBOL TABLE

EF.COM=	000036	G	FDTYP	007616	GLBEQA	002160	G	INIT.M	046504	L\$DRCT	002070	G
EF.NEW=	000035	G	FILL	061240	GLBERR	006626	G	INIT.R	041206	L\$DRS	002072	G
EF.PWR=	000034	G	FILL.C	000204	GLBSUB	010460	G	INPUT	011766	L\$DRST	002144	G
EF.RES=	000037	G	FINDBF	010254	GLBXTX	002406	G	INPUTA	061346	L\$DTP	002040	G
EF.STA=	000040	G	FIRST	010460	G\$BIT =	000003		INTEN =	000100	L\$DU	010456	G
EF01 =	000001	G	FLAGS	041160	G\$STAT =	000004		INTFOR	054644	L\$DUT	002076	G
EF02 =	000002	G	FLAG\$1	041162	G\$EXCP=	000400		INTR1	021700	L\$DVTY	002146	G
EF03 =	000003	G	FLAGTA	064412	G\$HILI=	000002		INVAL.	045452	L\$EF	002056	G
EF04 =	000004	G	FLAG.I	046146	G\$LOLI=	000001		INVINT	054502	L\$EFLG	002034	G
EF05 =	000005	G	FLA.SE	064360	G\$NO =	000000		INV.SW	042504	L\$EXP1	002042	G
EF06 =	000006	G	FLDNUM	013454	G\$OFFS=	000400		IN.SUF	047454	L\$EXP2	002044	G
EF07 =	000007	G	FLG.MA	046106	G\$OF\$1=	000376		ISDRST	023356	L\$EXP3	002046	G
EF08 =	000010	G	FMTCSH	007450	G\$PRMA=	000001		ISAU =	000041	L\$HARD	040516	G
EF09 =	000011	G	FMTMS	007510	G\$PRMD=	000002		ISCLN =	000041	L\$HPCP	002016	G
EF10 =	000012	G	FMTSN	007420	G\$PRML=	000000		ISDU =	000041	L\$HFTP	002022	G
EF11 =	000013	G	FMTTB	007435	G\$RADA=	000140		ISHRD =	000041	L\$HW	007632	G
EF12 =	000014	G	FMT16	007232	G\$RADB=	000000		ISINIT=	000041	L\$ICP	002104	G
EF13 =	000015	G	FMT17	007266	G\$RADD=	000040		ISMOD =	000041	L\$INIT	007652	G
EF14 =	000016	G	FMT17A	007303	G\$RADF=	000200		ISMSG =	000041	L\$LADP	002026	G
EF15 =	000017	G	FMT18	007335	G\$RADL=	000120		IS\$PWR =	000041	L\$LAST	041114	G
EF16 =	000020	G	FMT19	007342	G\$RADO=	000020		ISRPT =	000041	L\$MREV	002050	G
EMT.TR	041444	G	FMT20	007351	G\$RADT=	000100		ISSEG =	000041	L\$NAME	002000	G
END	010172		FORM.T	055010	G\$XFER=	000004		ISSFT =	000041	L\$REPP	002066	G
ENDOFP	026434		FREE	062316	G\$YES =	000010		ISSRV =	000041	L\$REV	002010	G
ENDRD	017206		FUNC	002210	HCE =	040000		ISSUB =	000041	L\$RPT	007650	G
ENDRD1	017216		FWDFLG	002242	H\$CORED	045656		ISTST =	000041	L\$SOFT	040564	G
ENDWR	023572		F\$AU =	000015	H\$COREQ	045566		J\$JMP =	000167	L\$SPC	002062	G
END.OF	047470		F\$BGN =	000040	H\$CORET	041402	G	KBPTR	041224	L\$SPCP	002020	G
END.SU=	071722		F\$CLEA=	000007	HCRC =	004000		KBUF	041226	L\$SPTP	002024	G
ENVIRO	041166	G	F\$DU =	000016	HC.ADR	041152	G	LDFUNC	021570	L\$STA	002030	G
EOP.CH	070370	G	F\$END =	000041	HC.DEF	041144	G	LINE.F	041442	L\$SW	007640	G
EOP.FM	043734		F\$HARD=	000004	HC.DIA	041142	G	LIST	011746	L\$TIML	002014	G
EOP.IN	046100		F\$HW =	000013	HDHOME	024700		LOADED	023222	L\$TIMU	002054	G
ERR =	100000		F\$INIT=	000006	HDLAS1	024776		LOAD.F	046102	L\$TIM1	002052	G
ERRAT	006120		F\$JMP =	000050	HDRFND	002312		LUGMSG	060412	L\$TST1	002100	G
ERRCNT	002160		F\$MOD =	000000	HEAD =	000100		LPBFR	041222	L\$UNIT	002012	G
ERRFOR	054714		F\$MSG =	000011	HERTZ.	045526		LPCNTR	041220	L.CLK.	045512	
ERRHAN	053514		F\$PWR =	000017	HLPMSG	011424		LPT.AD	045544	L10000	007112	
ERRVEC	002404		F\$RPT =	000012	HMSG	002462		LPT.RE	045540	L10001	007230	
ERR.HR	054452		F\$SEG =	000003	HNF =	010000		LSI.RE	045534	L10002	007636	
ERR.NU	041116	G	F\$SOFT=	000005	HOLDSP=	000020		LSTDA	002220	L10003	007644	
ERR.SF	054456		F\$SRV =	000010	HOP =	000020		LSTDR1	002304	L10004	007650	
ERR1	006626	G	F\$SUB =	000002	HOUR	002336		LSTHDR	002214	L10005	010326	
ERR1FO	055000		F\$SW =	000014	HPTCOD	007630	G	LSTTIM	002330	L10006	010452	
ERR2	007114	G	F\$TEST=	000001	HRDPRM	040514	G	LUP	066506	L10007	010454	
ESC.PC	053506		GARBAG	061744	HWSEC	003351		LUP.AD	053512	L10010	010456	
EV.COU	041120	G	GETCHR	060450	HW.ADR	041150	G	L\$APT	002036	L10011	026436	
EXISTS	004013		GETCMN	064034	HYPHEN	002540		L\$AU	010454	L10012	021706	
E.BA	002342		GETCYL	014446	H\$AAB	065206		L\$AUT	002074	L10013	040540	
E.CS	002340		GETDST	023342	IBDN	024472		L\$CCP	002106	L10014	040604	
E.DA	002344		GETPAR	055526	IBDN1	024534		L\$CIEA	010372	MAJ.IN	041176	G
E.DCS	002202		GETSEC	014534	IBDN2	024540		L\$CO	002032	MAJ.LO	066606	
E.MP	002346		GETSN	014306	INBAD	024400		L\$DEPO	002011	MAJ.US	041200	G
E.MP1	002350		GETSW1	063030	INBSF	006250		L\$DESC	002102	MAN.TI	001244	
E.MP2	002352		GETSYS	024544	ININIT	041372	G	L\$DEVP	002064	MAP16	071314	C
E.STAT	002204		GET.TW	062600	INITCO	007652	G	L\$DISP	007646	MASK.B	050216	
FACNUM	013456		GLBDAT	002160	INITIA	060420		L\$DR	002144	MASK.W	050214	

ASSEMBLY ROUTINES  
SYMBOL TABLE

MAXWC	002360	NXTCMD	011660	READ.P	066610 G	ST.SET	042716	TSCODE=	001052
MBLD	003012	NXTFOR	064652	RECNT	002256	SUNIT.	046070	TSERCO=	000063
MCRLF	007552	NXTSEC	017526	REGBAC	071300 G	SUPERV	043752	TSEERN=	001022
MDERS	003171	NXTTRK	017540	REGSAV	071264 G	SUPFLA	041360 G	TSEXCP=	000000
MDHEDR	002000 G	NXTUNI	002260	REQN.P	041170 G	SUPV.T	041532 G	TSHILI=	177777
MDONE	003541	OCTMSG	060376	REQN.T	046062	SUP.PR	042470	T\$LOLI=	000001
MDRTYP	006431	OK	006242	RESTAR	010226	SVCGBL=	000000	T\$LSYM=	010000
MEM.SI	045554	ONESEC	017642	RETRY	002172	SVCHAN	050406	T\$NEST=	177777
MHDER	003242	OPI =	002000	RE.SET	042652	SVCINS=	000000	T\$NSKO=	000000
MINUTE	002334	OVRMAX	006150	RLICLM	005627	SVCSUB=	177777	T\$NSK1=	000005
MIN.IN	041172 G	OSAPTS=	000001	RPTCOD	007650 G	SVCTAG=	000000	T\$SAVL=	177777
MIN.US	041174 G	OSAU =	000001	RSEEK	002236	SVCTST=	177777	T\$SEGL=	177777
MK =	000001	OSBGNR=	000001	RSTACK	070540 G	SWCHAN	045700	T\$SUBN=	000000
MODR	070656 G	OSBGNS=	000001	SAVEDO=	043050	SWITCH	064552	T\$TAGL=	177777
MF =	000006	OSDU =	000001	SAWFD	006452	SWRT1	023614	T\$TAGN=	010015
MRDER	003217	OSGNSW=	000001	SAWREV	006500	SWSEC	003426	T\$TEMP=	000000
MRLCS	002406	OSPOIN=	000001	SEARCH	062546	SW.ADR	041154 G	T\$TEST=	000001
MSFER	003144	OSPWR =	000001	SECMAX	013466	SW.PTA	045664	T\$TSTM=	177777
MSG	007555	PARSES	064106	SECMSK	002266	SYSCLK	002354	T\$TSTS=	000001
MSG.AD	041140 G	PAR.LA	060074	SECNUM	013470	SYSMSK	002262	T\$SAU =	010007
MSG.TY	041114 G	PASS.C	041130 G	SECOLD	013472	SYS.FT	054432	T\$SCLE=	010006
MSKER	003133	PASWD	002274	SECOND	002332	S\$LSYM=	010000	T\$SDU =	010010
MSREC	002762	PASWOR	003656	SEEK =	000006	TBLFUL	005714	T\$SHAR=	010013
MSTWRT	006020	POWER	010326	SEGSTA	041414 G	TDR	002246	T\$SHW =	010002
MTEST	011012	PRFLGS	002216	SELDIV	010470	TEMPO	002320	T\$SINI=	010005
MUL	070612 G	PRGER	003556	SELTBL	010330	TEMP1	002322	T\$SMMSG=	010001
MWRITE	004100	PRINTC	061720	SERNM1	002224	TEMP2	002324	T\$SRPT=	010004
NEWBSF	020614	PRINTF	065226	SERNM2	002227	TEMP3	002326	T\$SSOF=	010014
NEWENT	006103	PRI00 =	000000 G	SET.MA	0462 2	TERMI	066576	T\$SSRV=	010012
NEWFAC	002230	PRI01 =	000040 G	SFTCNT	002162	TERMLI	064400	T\$SSW =	010003
NEWLD	003324	PRI02 =	000100 G	SFTPRM	040562 G	TERMTA	060362	T\$STES=	010011
NEWPOS	002254	PRI03 =	000140 G	SHIFT	071376 G	TEST.M	046020	T1	011012 G
NEWPRI	070336 G	PRI04 =	000200 G	SIGN =	000004	TFMSG	002606	UNIT.D	041132 G
NEXTAR	064576	PRI05 =	000240 G	SKDON =	000001	THARD	002725	UNI.MA	046010
NHWSEC	003400	PRI06 =	000300 G	SKECNT	002164	THISDR	006575	USER.P	041374 G
NOCRDY	003607	PRI07 =	000340 G	SKFNC	025062	TILLEN	005770	USER.T	041376 G
NODRIV	003633	PRNTST	061610	SKHS =	000020	TIME	007560	UUT	002362
NOENTR	005567	PRO.CM	046060	SKTO =	010000	TIMFLG	041350 G	VALID.	041634
NOFACT	013500	PRPOS	002252	SKWRT =	024044	TIM.CO	041202 G	VALSN	005660
NOFIEL	013566	PSNFG	013476	SMSG	003101	TIM.OP	055006	VAL.LA	042454
NOFLDE	003477	PTAB.S	041400 G	SMSK	002272	TITLEX	002110	VAL.SW	046120
NOLOAD	003255	PTIME	010572	SN1	002364	TMMSG	002636	VC =	001000
NOMSEC	013654	PUTCHR	060424	SN2	002366	TOO.MA	060342	VERIFY	004063
NOSNUM	013452	PWRFLG	002374	SOFTCS	002240	TRPFLG	002376	WCPAT	026374
NOTRDY	006401	PWR.FA	071550 G	SPE =	004000	TRPHAN	010564	WDE =	100000
NO.CLK	045502	PWR.FL	041204 G	SPEC.U	046006	T\$OFT	002670	WGE =	002000
NO.FLA	064372	PWR.MS	071676	SPTCOD	007636 G	TST.AB	050330	WHATCM	011310
NO.LPT	061710	PWR.SA	071672	SPV.SE	000400	TST.TO	042532	WIDTH	055402
NO.PTA	045706	PWR.UP	071674	SRD1	017404	TYPEC	060736	WL =	020000
NR =	000000	P.CLK.	045520	STARMS	002477	TYPEPC	054632	WNXSEC	023736
NSWSEC	003453	RDBDSC	021710	STARTC	070266 G	TYPFLA	064254	WNXTRK	023770
NUMBIN	055034	RDBSEX	023174	STBLE	010370	TYPIN	060634	WRBUF	026324
NUM.LA	055202	RDBSFI	023124	STFLG	002402	TYPNUM	060216	WPCHK =	000002
NUM.NO	041156 G	RDFACT	023026	STRCHR	061300	TYPSTR	060654	WRINIT	002276
NUM.UN	041564	RDFIEL	023066	STREAD	017402	TYP.ER	054462	WRIPG	002250
NUMITS	050172	RDHDR =	000010	STRT.T	046064	TY.UNI	047474	WRITE =	000012
NXM =	020000	READ =	000014	STWRT	023612	T\$ARGC=	000002	WRPACK	023434

ASSEMBLY ROUTINES  
SYMBOL TABLE

MACRO V03.01 9-FEB-79 19:46:54 PAGE 180-4

J 8

SEQ 0100

WRPKF	006531	WRTSAW	007640	XEQ.IN	047616	XTIMST	067320	\$BREG	046160
WRPKR	006553	WTRDY	023264	XEQ.LA	043705	XXDP.D	045464	\$ENDAD	070376 G
WRTBSF	015122	W1SEC	024120	XEQ.OP	047710	X\$ALWA=	000000	\$SAV2	071442 G
WRTCNT	002170	XEQDIA	070424 G	XEQ.FR	043110	X\$FALS=	000040	\$SAV3	071456 G
WRTLCK	003277	XEQSUB	070412 G	XEQ.TE	047754	X\$OFFS=	000400	\$SAV4	071474 G
WRTLIM	007642	XEQ.CL	050134	XTIMF	067276 G	X\$TRUE=	000020	\$SAV5	071514 G
WRTLOK	002370	XEQ.CM	045444	XTIMEN	070122				

. ABS. 071720 000  
000000 001

ERRORS DETECTED: C

VIRTUAL MEMORY USED: 18517 WORDS ( 73 PAGES)  
DYNAMIC MEMORY AVAILABLE FOR 70 PAGES  
CZRLMA.BIN,CZRLMA-#SVCRT/M,CZRLMA,DOCTOR/E:5BL