

RL11,RLV11

BD SEC FIL TL  
CZRLMBO

AH-F134B-MC  
FICHE 1 OF 1

MAR 980  
COPYRIGHT © 98  
MADE IN USA



A microfiche grid containing approximately 12 columns and 15 rows of data. Each cell contains a small, high-contrast image of a document page, likely containing text and tables. The text is too small to be legible.





IDENTIFICATION

PRODUCT CODE: AC-F135B-MC  
PRODUCT NAME: CZRLMBO RLO<sup>1</sup>/02 BAD SECTOR FILE TOOL  
DATE CREATED: 5-JAN-79  
REVISED: 7-DEC-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 PROGRAMS
1.1.2	DIAGNOSTIC INFORMATION
1.2	SYSTEM REQUIREMENTS
1.2.1	HARDWARE REQUIREMENTS
1.2.2	SOFTWARE REQUIREMENTS
1.3	RELATED DOCUMENTS AND STANDARDS
1.4	DIAGNOSTIC HIERARCHY PREREQUISITES
1.5	ASSUMPTIONS
2.0	OPERATING INSTRUCTIONS
2.1	HOW TO RUN THIS DIAGNOSTIC
2.1.1	THE FIVE STEPS OF EXECUTION
2.1.2	SAMPLE RUN-THROUGH
2.2	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
3.1	ERROR REPORTING
3.2	ERROR HALTS
4.0	PERFORMANCE AND PROGRESS REPORTS
4.1	PERFORMANCE REPORTS
4.2	PROGRESS REPORTS
5.0	DEVICE INFORMATION TABLES
6.0	UTILITY-SUMMARY OF COMMANDS

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

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

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

THE DIAGNOSTIC PROGRAM IS LOADED IN THE LOWER 8K OF MEMORY. THE DIAGNOSTIC SUPERVISOR CODING OCCUPIES 0.25K OF THE UPPER PART OF MEMORY JUST BELOW THE XXDP+ MONITOR WHICH RESIDES IN THE UPPERMOST 1.5K OF MEMORY SPACE.

1.1.2      DIAGNOSTIC INFORMATION

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 REQUIREMENTS1.2.1      HARDWARE REQUIREMENTS

- \* PDP-11/LSI-11 PROCESSOR WITH 16K OR MORE OF MEMORY
- \* CONSOLE DEVICE (LA30,LA36,VT50,ETC.)
- \* 1 OR 2 RL11/RLV11 CONTROLLER(S) WITH:

- 1 - 8 RLO1 DRIVES WITH RLO1K CARTRIDGES CONTAINING A 'BAD SECTOR FILE'
- 1 - 8 RLO2 DRIVES WITH RLO2K CARTRIDGES CONTAINING A 'BAD SECTOR FILE'

- \* KW11-L OR KW11-P CLOCK
- \* LINE PRINTER (OPTIONAL)

#### 1.2.2 SOFTWARE REQUIREMENTS

-----

CZRLMBO RLO1/02 BAD SECTOR FILE TOOL

#### 1.3 RELATED DOCUMENTS AND STANDARDS

-----

RLO1 DISK SUBSYSTEM USER'S GUIDE (EK-RLO1-UG-002)  
XXDP+/SUPERVISOR USER'S MANUAL

#### 1.4 DIAGNOSTIC HIERARCHY PREREQUISITES

-----

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

CVRLABO	RLV11 RLO1/02 DISKLESS TEST (RLV11 ONLY)
CZRLGBO	RL11/RLV11 RLO1/02 CONTROLLER TEST (PART 1)
CZRLHBO	RL11/RLV11 RLO1/02 CONTROLLER TEST (PART 2)
CZRLIBO	RLO1/02 DRIVE TEST (PART 1)
CZRLJBO	RLO1/02 DRIVE TEST (PART 2)
CZRLKBO	RL11/RLV11 RLO1/02 PERFORMANCE EXERCISER
CZRLNAO	RLO1/02 DRIVE TEST (PART 3)

#### 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 CONFORM 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 RL8A/RL01 DISK CONTROLLER ON ANY PDP-8 SYSTEM...THIS IS A PDP-11 UTILITY ONLY!

2.0        OPERATING INSTRUCTIONS

2.1        HOW TO RUN THIS DIAGNOSTIC

2.1.1     THE FIVE STEPS OF EXECUTION

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

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

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

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

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

WHEN THIS UTILITY PROGRAM IS STARTED, THE FOLLOWING 5 STEPS WILL OCCUR:

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

THE UTILITY WILL ISSUE THE PROMPT 'DR>'. 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/PASS:1/FLAGS:HOE
```

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

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

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

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 3 \*  
\*\*\*\*\*

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

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

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 OCCURRED.

#### 2.1.2 SAMPLE RUN-THROUGH

-----

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

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

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



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

PRO/FLAGS:IER:LOE:HOE=0

THIS WILL DO THE FOLLOWING:

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

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

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

1. START
2. RESTART
3. CONTINUE

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

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

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

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

1. START
2. RESTART

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

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

	BY WHOM ENTERED: -----
.R CZRLMB	O
DRS LOADED	D
DIAG. RUN-TIME SERVICES REV. D APR-79	D
CZRLM-B-0	D
CZRLM IS A UTILITY PROGRAM FOR FORMATTING BAD SECTOR FILES	D
UNIT IS RL01, RL02	D
DR>STA/PASS:1/FLAGS:HOE	D,O
CHANGE HW (L) ? Y	D,O
# UNITS (D) ? 2	D,O
UNIT 0	D
BUS ADDRESS (O) 17440C ?	D,O
DRIVE (O) 0 ?	D,O
UNIT 1	D
BUS ADDRESS (O) 174400?	D,O
DRIVE (O) 0 ? 1	D,O
CHANGE SW (L) ? Y	D,O
SAWTOOTH WRITE CYCLE ? (L) Y ?	D,O
WRITE CYCLES PER TRACK ? (D) 2 ?	D,O
 CZRLM HRD ERR 00004 TST 003 SUB 002 PC:004130 ERR HLT	
DR>PRO/FLAGS:IER:LOE:HOE=0	D,O
 ***** AT THIS POINT THE DIAGNOSTIC IS LOOPING ON THE ERROR WITHOUT PRINTING ANYTHING. YOU CAN SCOPE THE ERROR UNTIL YOU HAVE LOCATED IT, THEN ^C OUT. *****	
^C	O
DR>CON/FLAGS:HOE:IER:LOE=0	D,O
CHANGE SW (L) ? N	D,O
^C	

DR>RESTART/PASS:1 D.0

CHANGE SW (L) ? N D.0

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

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:

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

## 2.2.2

COMMAND SYNTAX

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

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

THE SWITCH ARGUMENTS ARE AS FOLLOWS:

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

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

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

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

IER INHIBIT ERROR REPORTING

IBE INHIBIT BASIC ERROR REPORTS

IXE INHIBIT EXTENDED ERROR REPORTS

PRI DIRECT ALL MESSAGES TO A LINE PRINTER

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

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

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

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

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

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

THE SWITCH ARGUMENTS ARE AS IN THE START COMMAND EXCEPT:

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



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

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

THE SWITCH ARGUMENTS ARE AS IN THE START COMMAND EXCEPT:

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

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

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

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

1. UNSPECIFIED FLAG SETTINGS ARE UNCHANGED

\*\*\*\*  
EXIT  
\*\*\*\*

RETURN TO XXDP+ PROMPT MODE.

\*\*\*\*\*  
DRO(P)/UNITS:UNIT-LIST  
\*\*\*\*\*

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

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

\*\*\*\*\*  
ADD/UNITS:UNIT-LIST  
\*\*\*\*\*

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

\*\*\*\*\*  
PRI(NT)  
\*\*\*\*\*

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

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

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

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

THE CURRENT SETTINGS OF ALL FLAGS ARE PRINTED.

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

ALL FLAGS ARE CLEARED.

### 2.3        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 (0) 174400?

ANSWER WITH THE BUS ADDRESS OF THE CONTROLLER.

DRIVE (0) 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 FLEXIBILITY IN THE WAY THE PROGRAM BEHAVES. THE SOFTWARE PARAMETERS GIVE THE PROGRAM FLEXIBILITY IN THE WAY IT RUNS. THE PARAMETERS CAN BE MODIFIED ON A START, RESTART, OR CONTINUE BY ANSWERING (Y)ES TO THE FOLLOWING QUESTION:

'CHANGE S.W. ?'

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

'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 NCT 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)''

AN ERROR WAS DETECTED WHILE TRYING TO READ 10 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 10. 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.

'RL01 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 10 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 CONFORM 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 EXISTENT MEMORY ERROR  
BIT 12 - HEADER NOT FOUND (WITH BIT 10 SET)  
          - DATA LATE (WITH BIT 10 CLEAR)  
BIT 11 - HEADER CRC (WITH BIT 10 SET)  
          - DATA CRC (WITH BIT 10 CLEAR)

BIT 10 - OPERATION INCOMPLETE  
BIT 9/8 - DRIVE SELECT (0-3)  
BIT 7 - CONTROLLER READY  
BIT 6 - INTERRUPT ENABLE  
BIT 5 - EXTENDED BUS ADDRESS (BIT 17)  
BIT 4 - EXTENDED BUS ADDRESS (BIT 16)  
BIT 3-1 - FUNCTION CODE  
    0 - NOP (PDP-11) MAINT (LSI-11)  
    1 - WRITE CHECK  
    2 - GET DRIVE STATUS  
    3 - SEEK  
    4 - READ HEADER  
    5 - WRITE DATA  
    6 - READ DATA  
    7 - READ WITHOUT HEADER COMPARE

BIT 0 - DRIVE READY

RLBA - BUS ADDRESS REGISTER (XXXXX2)  
-----

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

6.0 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!

a

23	BIT AND OFFSET DEFINITIONS
90	MACRO DEFINITIONS
116	GLOBAL DATA AND CONSTANTS
219	GLOBAL MESSAGES
325	ERROR MESSAGES
408	STATISTIC CODE
416	LOAD PROTECTION TABLE
426	INITIALIZATION CODE
568	AUTO DROP SECTION
654	CLOCK INTERRUPT SERVICE ROUTINES
715	GLOBAL SUBROUTINES
841	PROGRAM MAIN LOOP
888	COMMAND QUERY LOOP
934	GLOBAL SUBROUTINES
1880	ROUTINE TO LOAD FUNCTION
1902	INTERRUPT SERVICE ROUTINE
1908	BAD SECTOR FILE ROUTINE
2135	ROUTINE TO WAIT FOR CONTROLLER READY
2157	GET STATUS/DRIVE RESET ROUTINE
2178	ROUTINE TO WRITE PACKS INITIALLY
2409	HEADS HOME ROUTINE
2445	SEEK ROUTINE
2497	ROUTINE TO CHECK FOR BAD SECTOR



1			.TITLE	CZRLMBO RL01/02 BD SEC FIL TL
2			.ENABLE	AMA
3			.ENABLE	ABS
4		002000	.=2000	
5			.MCALL	SVC
6				
7	002000		SVC	
8		000000	SVCINS=0	
9		000000	SVCTAG=0	
10				
11	002000		POINTER	BGNRPT,BGNSW,BGNSFT,BGNAU,BGNDU
12				
13				
14	002000		BGNMOD	MDHEDR
15	002000		HEADER	CZRLM,B,0,0,1
(4)	002000	103	.ASCII	/C/
(4)	002001	132	.ASCII	/Z/
(4)	002002	122	.ASCII	/R/
(4)	002003	114	.ASCII	/L/
(4)	002004	115	.ASCII	/M/
(6)	002005	000	.BYTE	0
(6)	002006	000	.BYTE	0
(5)	002007	000	.BYTE	0
(4)	002010	102	.ASCII	/B/
(4)	002011	060	.ASCII	/O/
(4)	002012	000000	.WORD	0
(4)	002014	000000	.WORD	0
(4)	002016	035534	.WORD	L\$HARD
(4)	002020	035602	.WORD	L\$SOFT
(4)	002022	010122	.WORD	L\$HW
(4)	002024	010130	.WORD	L\$SW
(4)	002026	035704	.WORD	L\$LAST
(4)	002030	000000	.WORD	0
(4)	002032	000000	.WORD	0
(4)	002034	000001	.WORD	1
(4)	002036	000000	.WORD	0
(4)	002040	010136	.WORD	L\$DISPATCH
(4)	002042	000000	.WORD	0
(4)	002044	000000	.WORD	0
(4)	002046	000000	.WORD	0
(4)	002050	003	.BYTE	C\$REVISION
(3)	002051	003	.BYTE	C\$EDIT
(4)	002052	000000	.WORD	0
(5)	002054	000000	.WORD	0
(4)	002056	000000	.WORD	0
(4)	002060	002216	.WORD	L\$DVTYP
(4)	002062	010140	.WORD	L\$RPT
(4)	002064	000000	.WORD	0
(4)	002066	000000	.WORD	0
(4)	002070	011726	.WORD	L\$AU
(4)	002072	011730	.WORD	L\$DU
(4)	002074	000000	.WORD	0
(4)	002076	002122	.WORD	L\$DESC
(4)	002100	104035	EMT	E\$LOAD
(4)	002102	000000	.WORD	0
(4)	002104	010150	.WORD	L\$INIT

```

(4) 002106 011644 .WORD L$CLEAN
(4) 002110 011224 .WORD L$AUTO
(4) 002112 010142 .WORD L$PROT
(4) 002114 000000 .WORD 0
(4) 002116 000000 .WORD 0
(4) 002120 000000 .WORD 0
16 002122 ENDMOD
17
18

```

```

19 002122 DESCRIPT <CZRLM IS A UTILITY PROGRAM FOR FORMATTING BAD SECTOR FILES>
(3) 002122 055103 046122 020115 .ASCIZ /CZRLM IS A UTILITY PROGRAM FOR FORMATTING BAD SECTOR FILES/
(3) 002130 051511 040440 052440
(3) 002136 044524 044514 054524
(3) 002144 050040 047522 051107
(3) 002152 046501 043040 051117
(3) 002160 043040 051117 040515
(3) 002166 052124 047111 020107
(3) 002174 040502 020104 042523
(3) 002202 052103 051117 043040
(3) 002210 046111 051505 000

```

```

(2) 002216 .EVEN
20
21 002216 DEVTYP <RL01,RL02>
(3) 002216 046122 030460 051054 .ASCIZ /RL01,RL02/
(3) 002224 030114 000062
(2) .EVEN

```

22 .SBTTL BIT AND OFFSET DEFINITIONS

23 :DEFINITIONS

24 BGNMOD GLBEQAT

25 EQUALS

26 ; BIT DIFINITIONS

```

(1) 100000 BIT15== 100000
(1) 040000 BIT14== 40000
(1) 020000 BIT13== 20000
(1) 010000 BIT12== 10000
(1) 004000 BIT11== 4000
(1) 002000 BIT10== 2000
(1) 001000 BIT09== 1000
(1) 000400 BIT08== 400
(1) 000200 BIT07== 200
(1) 000100 BIT06== 100
(1) 000040 BIT05== 40
(1) 000020 BIT04== 20
(1) 000010 BIT03== 10
(1) 000004 BIT02== 4
(1) 000002 BIT01== 2
(1) 000001 BIT00== 1

```

```

(1) 001000 BIT9== BIT09
(1) 000400 BIT8== BIT08

```

```

(1) 000200 BIT7== BIT07
(1) 000100 BIT6== BIT06
(1) 000040 BIT5== BIT05
(1) 000020 BIT4== BIT04
(1) 000010 BIT3== BIT03
(1) 000004 BIT2== BIT02
(1) 000002 BIT1== BIT01
(1) 000001 BIT0== BIT00
(1)
(1)
(1) : EVENT FLAG DEFINITIONS
(1) : EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION
(1)
(1) 000040 EF.START== 32. ; START COMMAND WAS ISSUED
(1) 000037 EF.RESTART== 31. ; RESTART COMMAND WAS ISSUED
(1) 000036 EF.CONTINUE== 30. ; CONTINUE COMMAND WAS ISSUED
(1) 000035 EF.NEW== 29. ; A NEW PASS HAS BEEN STARTED
(1) 000034 EF.PWR== 28. ; A POWER-FAIL/POWER-UP OCCURRED
(1)
(1)
(1) : PRIORITY LEVEL DEFINITIONS
(1)
(1) 000340 PRI07== 340
(1) 000300 PRI06== 300
(1) 000240 PRI05== 240
(1) 000200 PRI04== 200
(1) 000140 PRI03== 140
(1) 000100 PRI02== 100
(1) 000040 PRI01== 40
(1) 000000 PRI00== 0
(1)
(1) : OPERATOR FLAG BITS
(1)
(1) 000004 EVL== 4
(1) 000010 LOT== 10
(1) 000020 ADR== 20
(1) 000040 IDU== 40
(1) 000100 ISR== 100
(1) 000200 UAM== 200
(1) 000400 BOE== 400
(1) 001000 PNT== 1000
(1) 002000 PRI== 2000
(1) 004000 IXE== 4000
(1) 010000 IBE== 10000
(1) 020000 IER== 20000
(1) 040000 LOE== 40000
(1) 100000 HOE== 100000
(1)
(1) 30
(1) 31 000000 CS=0 ; CONTROL AND STATUS OFFSET
(1) 32 000002 BA=2 ; BUS ADDRESS OFFSET
(1) 33 000004 DA=4 ; DISK ADDRESS OFFSET
(1) 34 000006 MP=6 ; MULTI PURPOSE OFFSET
(1) 35
(1) 36 : CSR REGISTER
(1) 37
(1) 38 000001 SKDON=BIT0
(1) 39 000001 DRDY=BIT0 ;DRIVE READY

```

CZRLMB0 RL01/02 BD SEC FIL TL  
CZRLMB.MAC 12-DEC-79 14:06

MACY11 30A('052) 17-DEC-79 10:53 PAGE 1-3  
BIT AND OFFSET DEFINITIONS

N 2

SEQ 00

```

40      000100      INTEN=BIT6      ; INTERRUPT ENABLE
41      100000      ERR=BIT15     ; COMPOSITE ERROR
42      040000      DERR=BIT14    ; DRIVE ERROR
43      020000      NXM=BIT13     ; NON-EXISTENT MEMORY ERROR
44      010000      DLT=BIT12     ; DATA LATE
45      004000      DCRC=BIT11    ; DATA CRC ERROR
46      004000      HCRC=BIT11    ; HEADER CRC ERROR
47      010000      HNF=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=BIT1!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      000000      CSR=0
84      000002      DRBT=2
85
86      002230      ENDMOD
87
88

```

90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114

```
.SBTTL MACRO DEFINITIONS

;DELAY EXECUTION OF PROGRAM A SPECIFIED NUMBER OF 100-MILLISECOND TIME COUNTS
.MACRO WAITMS ARG,?WAIT
    MOV     #ARG,DLYCNT      ;INITIALIZE DELAY COUNTER
    ASL     DLYCNT           ;MULTIPLY ARGUMENT BY 2
    ASL     DLYCNT           ;MULTIPLY ARGUMENT BY 2 AGAIN
WAIT:    DELAY #250.         ;IMPLEMENT 25-MS TIME DELAY
    DEC     DLYCNT          ;DECREMENT DELAY COUNT
    BNE     WAIT            ;BRANCH IF TIME DELAY NOT EXPIRED
.ENDM

;DELAY EXECUTION OF PROGRAM A SPECIFIED NUMBER OF 100-MICROSECOND TIME COUNTS
.MACRO WAITUS ARG
    DELAY #ARG              ;IMPLEMENT 100-US TIME DELAY, ARGUMENT SPECIFIES
                          ;/THE NUMBER OF 100-US TIME COUNTS
.ENDM

;ACTIVATE THE CLOCK TO INITIATE THE GENERATION OF CLOCK INTERRUPTS
.MACRO CLKON
    JSR     PC,CLKINI       ;INITIALIZE THE CLOCK
    JSR     PC,CLKST        ;START THE CLOCK
.ENDM
```



```

116          .SBTTL  GLOBAL DATA AND CONSTANTS
117
118 002230      BGNMOD  GLBDAT
119 002230      000000      ERRCNT: .WORD 0          ;ERROR COUNT - HARD
120 002232      000000      SFTCNT: .WORD 0          ;ERROR COUNT - SOFT
121 002234      000000      SKECNT: .WORD 0          ;SEEK ERROR COUNT
122 002236      000000      DERCNT: .WORD 0          ;DRIVE ERROR COUNT
123 002240      000000      WRTCNT: .WORD 0          ;WRITE PASS COUNT PER TRACK
124 002242      000000      RETRY: .WORD 0          ;PRESENT RETRY NUMBER
125 002244      0000C0      BDA: .WORD 0          ;DISK ADDRESS CONTENTS
126 002246      000000      BMP: .WORD 0          ;PRESENT MULTIPURPOSE CONTENTS
127 002250      000000      DCS: .WORD 0          ;CSR ADDRESS
128 002252      000000      E.DCS: .WORD 0          ;CONTENTS OF RLCS AT ERROR
129 002254      000000      E.STAT: .WORD 0          ;STATUS AT FAILURE TIME
130 002256      000000      BBA: .WORD 0          ;PRESENT BUS ADDRESS CONTENTS
131 002260      000000      FUNC: .WORD 0          ;LAST FUNCTION LOADED
132 002262      000000      BCSADR: .WORD 0          ;CSR IMAGE OF LAST COMMAND
133 002264      000000      LSTHDR: .WORD 0          ;LAST POSITION ON DISK
134 002266      000000      PRFLGS: .WORD 0          ;INTERNAL FLAGS
135 002270      000000      LSTDA: .WORD 0          ;DISK ADDRESS AT SOFT ERROR
136 002272      000000      DIFWD: .WORD 0          ;LAST DIFFERENCE WORD OF SEEK
137 002274      000000      SERNM1: .WORD 0          ;SERIAL NUMBER OF CARTRIDGE
138 002276      000000      SERNM2: .WORD 0          ;SERIAL NUMBER OF CARTRIDGE
139 002300      000000      NEWFAC: .WORD 0          ;FLAG TO BUILD A DUMMY FACTORY FILE
140 002302      000000      DRSEL: .WORD 0          ;DRIVE SELECT BITS(8,9)
141 002304      000000      BSECTP: .WORD 0          ;POINTER TO BAD SECTOR FILE DATA STORAGE
142 002306      000000      RSEEK: .WORD 0          ;SEEK IN PROCESS OF RECOVERY
143 002310      000000      SOFTCS: .WORD 0          ;CSR OF SOFT ERROR
144 002312      000000      FWDFLG: .WORD 0          ;SAWTOOTH WRITE CONTROL FLAG
145 002314      000000      CVFLG: .WORD 0          ;'CALL' FLAG FOR VERIFY ROUTINE
146 002316      000000      TDR: .WORD 0          ;TYPE OF DRIVE... RL01=1 RL02=2
147 002320      000000      WRIPG: .WORD 0          ;WRITE IN PROGRESS FLAG
148 002322      000000      PRPOS: .WORD 0          ;PRESENT POSITION ON DISK
149 002324      000000      NEWPOS: .WORD 0          ;NEW DESIRED CYLINDER ADDRESS
150 002326      000000      RECNT: .WORD 0          ;READ ERROR COUNT
151 002330      000000      NXTUNI: .WORD 0          ;POINTER OF UNIT SELECT SLOT IN 'SELTBL'
152 002332      000000      SYMSK: .WORD 0          ;MASK FOR 0-7 DRIVES
153 002334      100177      CYLMSK: .WORD 100177      ;MASK FOR CYLINDER ONLY (RL01)
154 002336      100077      SECMSK: .WORD 100077      ;MASK OUT SECTOR BITS (RL01)
155 002340      000177      CMSK: .WORD 000177      ;CYL MASK FOR RL02
156 002342      000077      SMSK: .WORD 000077      ;SECT MASK FOR RL02
157 002344      000000      PASWD: .WORD 000000      ;PASSWORD (IF=0 THEN NO CHECKING)
158 002346      000000      WRINIT: .WORD 0          ;WRITE INIT FLAG
159 002350      000160      BVEC: .WORD 160          ;VECTOR
160 002352      000240      BPRIOR: .WORD 240          ;PRIORITY 5
161 002354      000000      CLKFRQ: .WORD 0          ;CLOCK FREQUENCY FLAG, 1=60HZ, 2=50HZ
162 002356      000000      CLKTYP: .WORD 0          ;CLOCK TYPE FLAG, 1=P-CLOCK, 2=L-CLOCK
163 002360      000000      CLKADR: .WORD 0          ;POINTER TO ADDRESS OF SUPERVISOR CLOCK TABLE
164 002362      000000      DLYCNT: .WORD 0          ;DELAY COUNTER FOR WAITMS TIMING MACRO
165 002364      000000      CLKSON: .WORD 0          ;'CLOCK ON' INDICATOR
166 002366      000000      CLKCNT: .WORD 0          ;CLOCK COUNTER TO STORE CLOCK TICK COUNT
167 002370      000000      CLKBFR: .WORD 0          ;CLOCK BUFFER TO STORE CLOCK TICK COUNT
168 002372      000000      SYSCLK: .WORD 0          ;FLAG INDICATING PRESENCE OF A SYSTEM CLOCK
169 002374      000000      LOGUNIT: .WORD 0          ;LOGICAL UNIT UNDER TEST
170 002376      000000      CLKFLD: .WORD 0          ;CLOCK FIELD TO CHECK IF LSI-11 CLOCK
171          ;/IS 'TICKING'

```

```
172
173 ;THE FOLLOWING LOCATIONS ARE CLEARED AS A GROUP (DOWN TO 'STFLG')
174 ;THEREFORE DON'T INSERT ANY CONSTANTS
175
176 002400 000000 LSTDR1: .WORD 0 ;BUFFER POINTER OF DRIVE
177 002402 000000 BCSR: .WORD 0 ;CSR FROM P-TABLE
178 002404 000000 BDRSEL: .WORD 0 ;DRIVE UNIT NUMBER FROM P-TABLE
179 002406 000000 HDRFND: .WORD 0 ;FLAG TO INDICATE HDR IN BAD LIST
180 002410 000000 CHKSEC: .WORD 0 ;SECTOR OF ERROR - USED BY BAD SECTOR LOCATION
181 002412 000000 DECNT: .WORD 0 ;DATA ERROR COUNT
182 002414 000000 TEMPO: .WORD 0 ;TEMP LOCATION
183 002416 000000 TEMP1: .WORD 0 ;TEMP LOCATION
184 002420 000000 TEMP2: .WORD 0 ;TEMP LOCATION
185 002422 000000 TEMP3: .WORD 0 ;TEMP LOCATION
186 002424 000000 TICK: .WORD 0 ;STORAGE FOR TICK COUNT
187 002426 000000 SECOND: .WORD 0 ;SECONDS OF SYSTEM CLOCK
188 002430 000000 MINUTE: .WORD 0 ;MINUTES OF SYSTEM CLODK
189 002432 000000 HOUR: .WORD 0 ;HOURS OF SYSTEM CLOCK
190 002434 000000 E.CS: .WORD 0 ;IMAGES OF REGISTERS
191 002436 000000 E.BA: .WORD 0 ;ON INTERRUPT
192 002440 000000 E.DA: .WORD 0
193 002442 000000 E.MP: .WORD 0
194 002444 000000 E.MP1: .WORD 0
195 002446 000000 E.MP2: .WORD 0
196 002450 000000 BUF1: .WORD 0 ;BUFFER FOR FIRST CONTROLLER
197 002452 000000 MAXWC: .WORD 0 ;MAX WORD COUNT DETERMINED BY CORE
198 002454 000000 UUT: .WORD 0 ;NUMBER OF UNITS ON SYSTEM
199 002456 000000 SN1: .WORD 0 ;TYPED SERIAL # - LOW
200 002460 000000 SN2: .WORD 0 ;HIGH
201 002462 000000 WRTLOK: .WORD 0 ;WRITE LOCK FLAG
202 002464 000000 ACCESS: .WORD 0 ;ACCESS PRIV FOR UPDATING
203 002466 000000 PWRFLG: .WORD 0 ;POWER FAIL INDICATOR
204 002470 000000 TRPFLG: .WORD 0 ;TRAP OCCURRENCE FLAG
205 002472 000000 CNTFLG: .WORD 0 ;CONTINUE FLAG
206 002474 000000 STFLG: .WORD 0 ;START FLAG
207 002476 000000 BSFFLG: .WORD 0 ;BAD SECTOR FILE FLAG (FACTORY BSF=0, FIELD BSF=1)
208 002500 000000 CPYCNT: .WORD 0 ;COUNTER FOR DUPLICATING COPIES OF THE 'FIELD'
209 ;/BAD SECTOR FILE ON THE PACK
210 002502 000000 FRSIER: .WORD 0 ;ADDRESS OF ERROR FOUND IN MAIN PROGRAM
211
212 ;END OF MASS CLEAR
213
214 002504 000004 ERRVEC: .WORD 4 ;ERROR VECTOR
215
216 002506 ENDMOD
217
```

```
219 .SBTTL GLOBAL MESSAGES
220
221 002506 BGNMOD GLBTXT
222
223 ;GLOBAL TEXT
224
225
229
230 002506 046122 051503 020072 MRLCS: .ASCIZ 'RLCS: ''
231 002515 050 046122 051503 CRLCS: .ASCIZ ''(RLCS): ''
232 002526 040520 045503 051440 CART: .ASCIZ /PACK SERIAL NO.: /
233 002550 054503 044514 042116 CMSG: .ASCIZ /CYLINDER: /
234 002563 040 042510 042101 HMSG: .ASCIZ / HEAD: /
235 002573 052 025052 025052 STARMSG: .ASCIZ /*****/
236 002675 055 026440 026440 HYPHEN: .ASCIZ /- - - - -/
237 002777 047 044506 046105 TFMSG: .ASCIZ /'FIELD' ENTRIES = /
238 003022 043047 041501 047524 TMMSG: .ASCIZ /'FACTORY' ENTRIES = /
239 003047 047 047523 052106 TSOFT: .ASCIZ /'SOFT' ERRORS FOUND = /
240 003076 044047 051101 023504 THARD: .ASCIZ /'HARD' ERRORS FOUND = /
241 003125 123 043117 020124 MSREC: .ASCIZ /SOFT ERROR RECOVERED.../
242 003155 102 044525 042114 MBLD: .ASCIZ /BUILD A DUMMY BAD SECTOR FILE/
243 003213 102 044525 042114 BUILD: .ASCIZ /BUILD A BAD SECTOR FILE/
244 003243 040 042523 052103 SMSG: .ASCIZ / SECTOR: /
245 003255 101 020124 047105 BSEND: .ASCIZ /AT END OF FILE /
246 003275 123 042505 020113 MSKER: .ASCIZ /SEEK ERROR/
247 003310 047523 052106 042440 MSFER: .ASCIZ /SOFT ERROR ENCOUNTERED/
248 003337 104 044522 042526 MDERS: .ASCIZ /DRIVE ERROR WILL NOT RESET/
249 003372 051104 053111 020105 MRDER: .ASCIZ /DRIVE ERROR RECOVERED/
250 003420 040510 042122 042440 MHDER: .ASCIZ /HARD ERROR/
251 003433 104 044522 042526 NOLOAD: .ASCIZ /DRIVE WOULD NOT LOAD/
252 003460 040520 045503 044440 WRTLCK: .ASCIZ /PACK IS WRITE LOCKED/
253 003505 120 041501 020113 NEWLD: .ASCIZ /PACK WAS JUST LOADED/
254 003532 047516 043040 041501 HWSEC: .ASCIZ /NO FACTORY FILE ENTRIES/
255 003562 047516 043040 041501 NHWSEC: .ASCIZ /NO FACTORY FILE FOUND/
256 003610 047516 043040 042511 SWSEC: .ASCIZ /NO FIELD FILE ENTRIES/
257 003636 047516 043040 042511 NSWSEC: .ASCIZ /NO FIELD FILE FOUND/
258 003662 047516 051440 041525 NOFLDE: .ASCIZ /NO SUCH ENTRY IN 'FIELD' FILE/
259 003720 047503 050115 042514 MDONE: .ASCIZ /COMPLETED.../
260 003735 120 047522 051107 PRGER: .ASCIZ /PROGRAM 'BUG' - DRIVE NOT READY/
261 003775 124 046511 047505 NOCRDY: .ASCIZ /TIMEOUT - NO 'CRDY'/
262 004021 116 020117 051104 NODRIV: .ASCIZ /NO DRIVES/
263 004033 040 051104 053111 DRNM: .ASCIZ / DRIVE: /
264 004044 047105 042524 020122 PASWORD: .ASCIZ /ENTER PASSWORD TO ENABLE BAD SECTOR FILE UPDATES/
265 004125 125 042120 052101 DENIED: .ASCIZ /UPDATING DENIED - INVALID PASSWORD!/
266 004171 105 052116 054522 EXISTS: .ASCIZ /ENTRY ALREADY EXISTS IN BAD SECTOR FILE/
267 004241 122 040505 044504 VERIFY: .ASCIZ /READING PACK/
268 004256 051127 052111 020105 MWRITE: .ASCIZ /WRITE PACK WITH WORST CASE DATA PATTERN/
269 004326 020061 020040 042522 CMD1: .ASCIZ /1 REPORT CONTENTS OF THE BAD SECTOR FILE/
270 004401 062 020040 040440 CMD2: .ASCIZ /2 ADD A SECTOR TO THE 'FIELD' BAD SECTOR FILE/
271 004461 063 020040 042040 CMD3: .ASCIZ /3 DELETE A SECTOR FROM THE 'FIELD' BAD SECTOR FILE/
272 004546 020064 020040 042526 CMD4: .ASCIZ /4 VERIFY PACK - READ ONLY/
273 004602 020065 020040 051127 CMD5: .ASCIZ /5 WRITE PACK WITH WORST CASE DATA PATTERN AND VERIFY/
274 004671 066 020040 046440 CMD6: .ASCIZ /6 MAKE A BAD SECTOR FILE/
275 004724 020067 020040 051120 CMD7: .ASCIZ /7 PRINT HELP MESSAGE/
276 004753 105 052116 051105 CMDDO: .ASCIZ /ENTER COMMAND (1 - 7) - /
277 005004 047503 052116 047105 BSRM: .ASCIZ /CONTENTS OF THE 'FACTORY' BAD SECTOR FILE:/
```

```
278 005057 103 047117 042524 BSRF: .ASCIZ /CONTENTS OF THE 'FIELD' BAD SECTOR FILE:/
279 005130 040502 020104 042522 BADBSF: .ASCIZ /BAD READ OF BAD SECTOR FILE/
280 005164 042101 020104 054503 ABSMSG: .ASCIZ /ADD CYLINDER, SECTOR, & HEAD TO 'FIELD' BAD SECTOR FILE/
281 005254 DELCYL:
282 005254 054503 044514 042116 ABSCYL: .ASCIZ /CYLINDER (0 TO 511.) - /
283 005304 DELSEC:
284 005304 042523 052103 051117 ABSSEC: .ASCIZ /SECTOR (0 TO 39.) - /
285 005331 DELHD:
286 005331 110 040505 020104 ABSHD: .ASCIZ /HEAD (0 OR 1) - /
287 005352 047516 041440 051101 ABSER: .ASCIZ /NO CARTRIDGE SERIAL NO. - ADD ONE?/
288 005415 111 050116 052125 ABSSNL: .ASCIZ /INPUT THE LOW 5 OCTAL DIGITS OF SERIAL NO. /
289 005471 111 050116 052125 ABSSNH: .ASCIZ /INPUT THE HIGH 5 OCTAL DIGITS OF SERIAL NO. /
290 005546 051127 052111 020105 DOWRT: .ASCIZ /WRITE THE UPDATED BAD SECTOR FILE/
291 005610 040503 047116 052117 BADWRT: .ASCIZ /CANNOT UPDATE BAD SECTOR FILE ON PACK/
292 005656 042504 042514 042524 DELMSG: .ASCIZ /DELETE A 'FIELD' BAD SECTOR FILE ENTRY/
293 005725 116 020117 052523 NOENTRY: .ASCIZ /NO SUCH ENTRY TO DELETE!/
294 005756 046122 030460 046440 RL1CLM: .ASCIZ /RLO1 MAX CYLINDER = 255./
295 006007 111 020123 044124 VALSN: .ASCIZ /IS THIS SERIAL NO. VALID/
296 006040 047515 042522 052040 TBLFUL: .ASCIZ /MORE THAN 25. BAD SPOTS FOUND ON THIS PACK./
297 006114 047503 052116 047111 TILLEN: .ASCIZ /CONTINUE TO END OF FILE/
298 006144 051127 052111 020105 MSTWRT: .ASCIZ /WRITE ON ALL SELECTED PACKS/
299 006200 042516 020127 047105 NEWENT: .ASCIZ /NEW ENTRY.../
300 006215 130 042506 020122 ERRAT: .ASCIZ /XFER ERROR AT PACK ADDRESS /
301 006251 122 030114 026461 OVRMAX: .ASCIZ /RLO1-RLO2 CARTRIDGE SPEC ALLOWS MAX OF 16. BAD SECTORS/
302 006340 047506 047125 000104 OK: .ASCIZ /FOUND/
303 006346 041101 053117 020105 INBSF: .ASCIZ /ABOVE SECTOR IS IN BAD SECTOR FILE/
304 006411 103 042510 045503 CKFACT: .ASCIZ /CHECKING FOR 'FACTOR,' FILE.../
305 006450 044103 041505 044513 CKFLD: .ASCIZ /CHECKING FOR 'FIELD' FILE.../
306 006505 104 044522 042526 NOTRDY: .ASCIZ /DRIVE DROPPED - DID NOT RESPOND WITH 'READY'/
307 006562 051104 053111 020105 MDRTYP: .ASCIZ /DRIVE TYPE = RLO/
308 006603 055 051440 053501 SAWFWD: .ASCIZ /- SAWTOOTH FROM CYLINDER 0/
309 006636 020055 040523 052127 SAWREV: .ASCIZ /- SAWTOOTH FROM LAST CYLINDER/
310 006674 051127 052111 047111 WRPKF: .ASCIZ /WRITING PACK FORWARD /
311 006722 051127 052111 047111 WRPKR: .ASCIZ /WRITING PACK REVERSE /
312 006750 051525 020105 044124 THISDRV: .ASCIZ /USE THIS SELECTED UNIT/
313 006777 123 051531 042524 NOCLK: .ASCIZ /SYSTEM CLOCK IS NOT AVAILABLE/
314 007035 122 047125 052040 NOTIM: .ASCIZ /RUN TIMES CANNOT BE REPORTED/
315 007072 051104 053111 020105 NOCTLR: .ASCIZ /DRIVE DROPPED - NO CONTROLLER/
316
317 .NLIST CND,MD,ME
318 .LIST BEX
319 .EVEN
320
321 007130 ENDMOD
322
```

```

324
325      .SBTTL  ERROR MESSAGES
326
327      007130      BGNMOD  GLBERR
328
329      007130      BGNMSG  ERR1
330      007130      010146      MOV      R1,-(SP)      ;SAVE R1
331      ;ROUTINE TO REPORT THE POSITION OF CYLINDER, SECTOR & HEAD
332
333      007132      004537      012502      JSR      R5,PTIME      ;PRINT RUN TIME
334      007136      013737      002410      015334      MOV      CHKSEC,BSFSEC  ;GET THE SECTOR IN ERROR
335      007144      042737      177700      015334      BIC      #177700,BSFSEC ;CLEAR THE JUNK BITS
336      007152      005037      015336      CLR      BSFHD      ;CLEAR THE HEAD #
337      007156      032737      000100      002410      BIT      #100,CHKSEC  ;HEAD 1??
338      007164      001402      BEQ      1$          ;NO
339      007166      005237      015336      INC      BSFHD      ;YES - SET IT TO 1
340      007172      013737      002410      015332      1$:      MOV      CHKSEC,BSFCYL ;GET ADDR AGAIN FOR THE CYLINDER
341      007200      042737      000177      015332      BIC      #177,BSFCYL  ;CLEAR THE HEAD & SECTOR #
342      007206      000337      015332      SWAB    BSFCYL
343      007212      000241      CLC
344      007214      006137      015332      ROL      BSFCYL      ;CLEAR THE 'C' BIT
345      007220      103002      BCC      2$          ;POSITION
346      007222      005237      015332      INC      BSFCYL      ;BR IF DON'T NEED OTHER BIT
347      007226      2$:      PRINTB  #FMT16,#ERRAT,#CMMSG,BSFCYL,#SMSG,BSFSEC,#HMSG,BSFHD ;ADD IN THE LOW ORDER BIT
(14) 007226      013746      015336      MOV      BSFHD,-(SP)
(13) 007232      012746      002563      MOV      #HMSG,-(SP)
(12) 007236      013746      015334      MOV      BSFSEC,-(SP)
(11) 007242      012746      003243      MOV      #SMSG,-(SP)
(10) 007246      013746      015332      MOV      BSFCYL,-(SP)
(9) 007252      012746      002550      MOV      #CMMSG,-(SP)
(8) 007256      012746      006215      MOV      #ERRAT,-(SP)
(7) 007262      012746      007503      MOV      #FMT16,-(SP)
(6) 007266      012746      000010      MOV      #10,-(SP)
(3) 007272      010600      TRAP    C$PNTB
(4) 007274      104414
(4) 007276      062706      000022      ADD      #22,SP
348 007302      004537      025422      JSR      R5,GETDST   ;GET THE DRIVE STATUS
349 007306      010137      002254      MOV      R1,E.STAT
350 007312      PRINTB  #FMT17A,#CRLCS,E.DCS,E.STAT,E.DA
(11) 007312      013746      002440      MOV      E.DA,-(SP)
(10) 007316      013746      002254      MOV      E.STAT,-(SP)
(9) 007322      013746      002252      MOV      E.DCS,-(SP)
(8) 007326      012746      002515      MOV      #CRLCS,-(SP)
(7) 007332      012746      007554      MOV      #FMT17A,-(SP)
(6) 007336      012746      000005      MOV      #5,-(SP)
(3) 007342      010600      MOV      SP,R0
(4) 007344      104414      TRAP    C$PNTB
(4) 007346      062706      000014      ADD      #14,SP
351 007352      PRINTB  #MCRLF
(7) 007352      012746      010041      MOV      #MCRLF,-(SP)
(6) 007356      012746      000001      MOV      #1,-(SP)
(3) 007362      010600      MOV      SP,R0
(4) 007364      104414      TRAP    C$PNTB
(4) 007366      062706      000004      ADD      #4,SP
352 007372      012601      MOV      (SP)+,R1    ;RESET R1
353 007374      ENDMSG

```

```
(3) 007374
(3) 007374 104423
354
355 007376
356 007376 010146
357 007400 004537 025422
358 007404 010137 002254
359 007410
(11) 007410 013746 002440
(10) 007414 013746 002254
(9) 007420 013746 002252
(8) 007424 012746 002515
(7) 007430 012746 007554
(6) 007434 012746 000005
(3) 007440 010600
(4) 007442 104414
(4) 007444 062706 000014
360 007450
(7) 007450 012746 010041
(6) 007454 012746 000001
(3) 007460 010600
(4) 007462 104414
(4) 007464 062706 000004
361 007470 012601
362 007472
(3) 007472
(3) 007472 104423
366 007474 047045 052045 047045 FMT15: .ASCIZ /%N%T%N/
367 007503 045 022516 022524 FMT16: .ASCIZ /%N%T%T%Z3%A.%T%Z2%A.%T%D1%N/
368 007537 045 022516 022524 FMT17: .ASCIZ /%N%T%06%T%01/
369 007554 052045 047445 022466 FMT17A: .ASCIZ /%T%06%A STATUS WAS: %06%A (DA): %06%N/
370 007624 047045 052045 000 FMT18: .ASCIZ /%N%T/
371 007631 045 022516 022516 FMT19: .ASCIZ /%N%N%T/
372 007640 047045 040445 040502 FMT20: .ASCIZ /%N%ABAD SECTOR FILE HAS %Z3%A. ENTRIES/
373 007707 045 022516 022524 FMTSN: .ASCIZ /%N%T%05%05%N/
374 007724 047045 052045 055045 FMTTB: .ASCIZ /%N%T%Z3%A./
375 007737 045 022516 031132 FMTCSH: .ASCIZ /%N%Z2%A.%T%Z3%A.%T%Z2%A.%T%D1/
376 007777 045 022516 022516 FMTMS: .ASCIZ /%N%N%ACOMMANDS AVAILABLE ARE:%N%T/
377 010041 045 000116 MCRLF: .ASCIZ /%N/
378 010044 052045 000 MSG: .ASCIZ /%T/
379 010047 045 022516 052101 TIME: .ASCIZ /%N%ATIME: %Z2%A:%Z2%A:%Z2%A /
380 010105 045 022516 022524 FDTYP: .ASCIZ /%N%T%01%N/
381
385
386 010120 .EVEN
387 010120 ENDMOD
388
389 010120 BGNMOD HPTCODE
390 010120 BGNHW
(3) 010120 000002 .WORD L10002-LSHW/2
391 010122 174400 .WORD 174400 ;CSR BASE ADDRESS DEFAULT
392 010124 000000 .WORD 0 ;DRIVE UNIT NUMBER DEFAULT
393 010126 ENDDHW
(3) 010126 L10002: ENDMOD
394 010126 BGNMOD SPTCODE
395 010126
```

```

396 010126          BGNSW
(3) 010126 000002
397 010130 000001  WRTSAW: .WORD  L10003-L$SW/2
398 010132 000002  WRTLIM: .WORD  1          ;DEFAULT TO SAWTOOTH WRITE CYCLE
399 010134          ENDSW          ;DEFAULT TO 2 WRITE PASSES PER TRACK
(3) 010134          L10003:
400 010134          ENDMOD
401
402 010134          BGNMOD  DSPCODE
403
404 010134          DISPATCH      1
(4) 010134 000001  .WORD  1
(6) 010136 012716  .WORD  11
405
406 010140          ENDMOD
407
408          .SBTTL  STATISTIC CODE
409
410 010140          BGNMOD  RPTCODE
411 010140          BGNRPT
412 010140          ENDRPT
(3) 010140          L10004:
(3) 010140 104425  TRAP      CSRPT
413 010142          ENDMOD
414
415
416          .SBTTL  LOAD PROTECTION TABLE
417
418 010142          BGNPROT
419 010142 000000  .WORD  0          ;P-TABLE OFFSET OF CSR
420 010144 177777  .WORD -1          ;NOT A MASS-BUS DRIVE
421 010146 000010  .WORD 10         ;P-TABLE OFFSET OF DRIVE
422 010150          ENDPROT
423
424

```

CZRLMBO RL01/02 BD SEC FIL TL  
CZRLMB.MAC 12-DEC-79 14:06

MACY11 30A(1052) 17-DEC-79 10:53 J 3 PAGE 1-12  
INITIALIZATION CODE

SEQ 003'

```

426          .SBTTL  INITIALIZATION CODE
427
428 010150   BGNMOD  INITCODE          ;START OF INITIALIZE CODE
429
430 010150   BGNINIT
431
432 010150   SETPRI  #340                ;PRIORITY TO 7 TO INHIBIT INTERRUPTS
(3) 010150   012700 000340             MOV    #340,R0
(3) 010154   104441   TRAP    C$SPRI
433
434 010156   BRESET
(3) 010156   104433   TRAP    C$RESET          ;FOR LSI-11 CPU'S
435 010160   005037 002474             CLR    STFLG
436 010164   005037 002472             CLR    CNTFLG          ;CLEAR CONT
437 010170   005037 002466             CLR    PWRFLG          ;CLEAR THE POWER FAIL FLAG
438          ;CHECK FOR PRESENCE OF A SYSTEM CLOCK
439 010174   CLOCK  P,CLKADR            ;P-CLOCK?
(3) 010174   012700 000120             MOV    #P,R0
(3) 010200   104462   TRAP    C$CLCK
(3) 010202   010037 002360             MOV    R0,CLKADR
440 010206   BNCOMPLETE LCLKCH          ;BRANCH IF NO P-CLOCK
(2) 010206   103006   BCC    LCLKCH
441 010210   012737 000001 002356     MOV    #1,CLKTYP      ;IDENTIFY P-CLOCK TYPE
442 010216   005237 002372             INC    SYSCLK          ;INDICATE PRESENCE OF A SYSTEM CLOCK
443 010222   000522   BR     PWRCH          ;BRANCH TO CHECK POWER
444 010224   LCLKCH: CLOCK  L,CLKADR            ;L-CLOCK?
(3) 010224   012700 000114             MOV    #L,R0
(3) 010230   104462   TRAP    C$CLCK
(3) 010232   010037 002360             MOV    R0,CLKADR
445 010236   BCOMPLETE 1$                      ;BRANCH IF L-CLOCK
(2) 010236   103401   BCS    1$
446 010240   000467   BR     NILCLK          ;ELSE, INDICATE CLOCK IS NOT PRESENT
447 010242   1$: READBUS                          ;CHECK TYPE OF BUS
(3) 010242   104407   TRAP    C$RDBU
448 010244   BNCOMPLETE 2$                      ;BRANCH IF NOT Q-BUS
(2) 010244   103057   BCC    2$
449 010246   005037 002376             CLR    CLKFLD          ;CLEAR CLOCK FIELD FOR STORING 'TICKS'
450 010252   SETVEC  #100,#CLKTIK,#340          ;SET UP L-CLOCK INTERRUPT VECTOR TO CHECK
(7) 010252   012746 000340             MOV    #340,-(SP)
(6) 010256   012746 012124             MOV    #CLKTIK,-(SP)
(5) 010262   012746 000100             MOV    #100,-(SP)
(4) 010266   012746 000003             MOV    #3,-(SP)
(3) 010272   104437   TRAP    C$SVEC
(2) 010274   062706 000010             ADD    #10,SP
451
452 010300   SETPRI  #240                ;/IF CLOCK IS 'TICKING'
(3) 010300   012700 000240             MOV    #240,R0          ;SET PRIORITY TO 5 TO ALLOW CLOCK INTERRUPTS
(3) 010304   104441   TRAP    C$SPRI
453 010306   WAITMS  #5
(3) 010324   012727 000372             MOV    ##250.,(PC)+
(3) 010330   000000   .WORD 0
(3) 010332   013727 002116             MOV    L$DLY,(PC)+
(3) 010336   000000   .WORD 0
(3) 010340   005367 177772             DEC    -6(PC)
(3) 010344   001375   BNE    -4
(3) 010346   005367 177756             DEC    -22(PC)

```



```

(3) 010352 001367 BNE .-20
454 010362 SETPRI #340 ;RESTORE PRIORITY TO 7 TO INHIBIT INTERRUPTS
(3) 010362 012700 000340 MOV #340,R0
(3) 010366 104441 TRAP C$SPRI
455 010370 CLRVEC #100 ;CLEAR L-CLOCK INTERRUPT VECTOR
(3) 010370 012700 000100 MOV #100,R0
(3) 010374 104436 TRAP C$CVEC
456 010376 005737 002376 TST CLKFLD ;L-CLOCK 'TICKS'?
457 010402 001406 BEQ NILCLK ;BRANCH IF NO 'TICKS:
458 010404 012737 000002 002356 2$: MOV #2,CLKTYP ;IDENTIFY L-CLOCK TYPE
459 010412 005237 002372 INC SYSCLK ;INDICATE PRESENCE OF A SYSTEM CLOCK
460 010416 000424 BR PWRCH ;BRANCH TO CHECK POWER
461 010420 NILCLK: PRINTF #FMT15,#NOCLK ;REPORT 'SYSTEM CLOCK IS NOT AVAILABLE'
(8) 010420 012746 006777 MOV #NOCLK,-(SP)
(7) 010424 012746 007474 MOV #FMT15,-(SP)
(6) 010430 012746 000002 MOV #2,-(SP)
(3) 010434 010600 MOV SP,R0
(4) 010436 104417 TRAP C$PNTF
(4) 010440 062706 000006 ADD #6,SP
462 010444 PRINTF #FMT15,#NOTIM ;PRINT 'RUN TIMES CANNOT BE REPORTED'
(8) 010444 012746 007035 MOV #NOTIM,-(SP)
(7) 010450 012746 007474 MOV #FMT15,-(SP)
(6) 010454 012746 000002 MOV #2,-(SP)
(3) 010460 010600 MOV SP,R0
(4) 010462 104417 TRAP C$PNTF
(4) 010464 062706 000006 ADD #6,SP
463 ;POWER FAIL SEQUENCE
464 010470 PWRCH: READEF #EF.PWR
(3) 010470 012700 000034 MOV #EF.PWR,R0
(3) 010474 104447 TRAP C$REFG
465 010476 BNCOMPLETE 3$
(2) 010476 103106 BCC 3$
466 010500 005237 002466 INC PWRFLG ;INDICATE POWER FAIL
467 010504 013702 002454 MOV UUT,R2 ;GET NUMBER OF UNITS SELECTED
468 010510 005302 DEC R2
469 010512 006302 ASL R2
470 010514 006302 ASL R2
471 010516 062702 011162 ADD #SELTBL,R2 ;POINT TO THE CORRECT SLOT
472 010522 012237 002250 11$: MOV (R2)+,DCS ;GET THE DCS ADDRESS
473 010526 011237 002302 MOV (R2),DRSEL ;AND GET THE DRIVE BITS
474 010532 052737 000200 002302 BIS #200,DRSEL ;ADD IN THE CRDY BIT
475 010540 013777 002302 171502 MOV DRSEL,@DCS ;SELECT THE DRIVE
476 010546 012701 000170 MOV #120,R1 ;INITIALIZE WAIT COUNT
477 010552 032777 000001 171470 12$: BIT #1,@DCS ;DRIVE READY UP?
478 010560 001040 BNE 14$ ;YES - RESET DRIVE & HEADS HOME
479 ;WAIT A WHILE
480 010562 WAITMS #10.
(3) 010600 012727 000372 MOV #250.,(PC)+
(3) 010604 000000 .WORD 0
(3) 010606 013727 002116 MOV LSDLY,(PC)+
(3) 010612 000000 .WORD 0
(3) 010614 005367 177772 DEC -6(PC)
(3) 010620 001375 BNE -4
(3) 010622 005367 177756 DEC -22(PC)
(3) 010626 001367 BNE -20
481 010636 005301 DEC R1 ;UPDATE THE TIMER

```

```

482 010640 001344          BNE      12$          ;IF MORE TIME, THEN TRY AGAIN
483
484          ;DRIVE NOT READY IN TIME - KILL THE ENTRY
485 010642 005742          TST      -(R2)          ;CORRECT THE POINTER
486 010644 005022          CLR      (R2)+         ;KILL THE ENTRY WORD FOR DCS
487
488 010646 162702 000004   13$:    SUB      #4,R2          ;POINT TO THE NEXT ENTRY IN LIST
489 010652 022702 011162   CMP      #SELTBL,R2     ;DONE?
490 010656 003721          BLE      11$          ;NO - DO THIS UNIT ALSO
491 010660 000404          BR       15$          ;YES - PROCEED
492
493 010662 004537 025436   14$:    JSR      R5,ISDRST    ;RESET THE DRIVE SELECTED
494 010666 004537 026754   JSP      R5,HDHOME     ;AND BRING THE HEADS HOME
495
496 010672 005737 002372   15$:    TST      SYSCLK      ;CLOCK TICK?
497 010676 001404          BEQ      4$           ;BR IF NO
498 010700
499 010710 000137 011160   4$:     CLKON    POWER      ;YES - SET FOR 1 SEC INTERVALS
500
501          ;'CONTINUE' COMMAND SEQUENCE
502 010714          3$:    READEF   #EF.CONTINUE    ;CONTINUE FROM CONSOLE?
(3) 010714 012700 000036   MOV      #EF.CONTINUE,R0
(3) 010720 104447          TRAP     C$REFG
503 010722          BNCOMPLETE 1$          ;NO, CONTINUE W/ INIT CODE
(2) 010722 103004          BCC      1$
504
505 010724 005237 002472          INC      CNTFLG        ;YES SET CONT FLAG, GO TO END OF INIT
506 010730 000137 011030          JMP      END
507
508 010734 004537 027316   1$:    JSR      R5,CLEAR     ;CLEAR ALL DRIVE STORAGE BUFFERS
509
510 010740 012700 002400   2$:    MOV      #LSTDR1,R0    ;CLEAR FLAGS
511 010744 005020          CLRDAT: CLR      (R0)+
512 010746 020027 002476   CMP      R0,#STFLG+2    ;MASS CLEAR OF GLOBAL DATA AREA
513 010752 001374          BNE      CLRDAT        ;DO TILL TABLE IS ZEROED
514
515 010754 012700 011162   CLRSTB: MOV      #SELTBL,R0
516 010760 012720 177777   MOV      #-1,(R0)+
517 010764 020027 011222   CMP      R0,#STBLE
518 010770 001373          BNE      CLRSTB        ;INIT THE SELECT TABLE
519          ;END OF THE TABLE?
520 010772 013703 002012          MOV      L$UNIT,R3     ;GET NUMBER OF UNITS
521 010776 010337 002454          MOV      R3,UUT        ;SAVE L$UNIT
522 011002 012704 011162   MOV      #SELTBL,R4    ;INIT SELECT TABLE POINTER
523 011006 005001          CLR      R1            ;INIT P-TABLE
524 011010          1$:    GPHARD   R1,R0        ;GET A P-TABLE
(3) 011010 010100          MOV      R1,R0
(3) 011012 104442          TRAP     C$GPHRD
525 011014          BNCOMPLETE 2$
(2) 011014 103002          BCC      2$
526          ;MOVE P-TABLE CONTENTS TO LOCAL STORAGE
527 011016 012024          MOV      (R0)+,(R4)+
528 011020 011024          MOV      (R0),(R4)+
529 011022 005201          2$:    INC      R1
530 011024 005303          DEC      R3
531 011026 001370          BNE      1$           ;DO WHILE

```

```

532
533 011030
534 011030 013704 002454
535 011034 006304
536 011036 006304
537 011040 062704 011162
538 011044 012704 177777
539
540 011050
(3) 011050 012700 000040
(3) 011054 104447
541 011056
(2) 011056 103002
542 011060 005237 002474
543
544 011064
545 011064
(7) 011064 013746 002352
(6) 011070 012746 023646
(5) 011074 013746 002350
(4) 011100 012746 000003
(3) 011104 104437
(2) 011106 062706 000010
546
547 011112 012737 030530 002450
548 011120 012737 002400 002452
549 011126 012737 002450 002256
550 011134 012737 027416 002304
551 011142 005737 002472
552 011146 001004
553
554 011150
555
556
557 011160
558 011160
(3) 011160
(3) 011160 104411
559
560
561 011162 000020
562 011222 177777
563
564 011224
565
566

```

END:

```

MOV UJT,R4
ASL R4
ASL R4
ADD #SELTBL,R4 ;POINT TO THE SELECT TABLE
MOV #-1,R4 ;FORCE A TERMINATE IN THE TABLE
;'START' COMMAND SEQUENCE
READEF #EF.START ;START COMMAND
MOV #EF.START,R0
TRAP CSREFG
BNCOMplete RESTART ;NO. CHK RESTART
BC RESTART
INC STFLG ;SET START INDICATOR

```

RESTART:

```

SETVEC BVEC,#INTR1,BPRIOR ;SET CONTROLLER VECTOR
MOV BPRIOR,-(SP)
MOV #INTR1,-(SP)
MOV BVEC,-(SP)
MOV #3,-(SP)
TRAP CSSEVC
ADD #10,SP

```

FINDBF:

```

MOV #BSFILE,BUF1 ;ALL XFERS TO BSFILE STORAGE
MOV #1280,MAXWC ;MAX XFER SIZE - 1/4 TRACK
MOV #BUF1,BBA ;POINT TO THE DATA STORAGE AREA
MOV #BSECO,BSECP ;POINT TO THE BAD SECTOR FILE DATA
TST CNTFLG ;HERE FROM 'CON' CMD?
BNE POWER ;BR IF TRUE

```

CLKON

```

;ACTIVATE SYSTEM CLOCK TO INITIATE GENERATION
;/OF TIMING INTERVALS

```

POWER:

```

ENDINIT
L10006: TRAP CSINIT

```

```

SELTBL: .BLKW 16.
STBLE: .WORD -1

```

ENDMOD

```

568 .SBTTL AUTO DROP SECTION
569
570 ;THE AUTO DROP SECTION IS CONDITIONALLY EXECUTED AFTER THE INITIALIZATION CODE
571 ;WHEN THE OPERATOR 'ADR' FLAG IS SET. EACH DRIVE IS CHECKED TO DETERMINE IF IT
572 ;RESPONDS WITH 'READY' AND IS DROPPED FROM THE TEST CYCLE IF IT DOES NOT. THE
573 ;HARDWARE TESTS ARE PERFORMED IMMEDIATELY AFTER THE READY STATUS OF ALL DRIVES
574 ;HAVE BEEN CHECKED.
575 BGNAUTO
576 011224 010146 MOV R1,-(SP) ;SAVE CONTENTS OF REGISTERS
577 011226 010246 MOV R2,-(SP)
578 011230 010346 MOV R3,-(SP)
579 011232 013703 002012 MOV L$UNIT,R3 ;INITIALIZE NUMBER OF UNITS
580 011236 012702 011162 MOV #SELTBL,R2 ;INITIALIZE START OF SELECT TABLE
581 011242 005037 002374 CLR LOGUNIT ;CLEAR LOGICAL UNIT NUMBER
582 011246 005037 002470 1$: CLR TRPFLG ;CLEAR TRAP FLAG
583 011252 SETVEC ERRVEC,#TRPHAN,#340 ;SET UP TIME-OUT VECTOR TO DETECT
(7) 011252 012746 000340 MOV #340,-(SP)
(6) 011256 012746 012474 MOV #TRPHAN,-(SP)
(5) 011262 013746 002504 MOV ERRVEC,-(SP)
(4) 011266 012746 000003 MOV #3,-(SP)
(3) 011272 104437 TRAP C$SVEC
(2) 011274 062706 000010 ADD #10,SP
584 ;/NON-EXISTENT CONTROLLER
585 011300 011237 002250 MOV @R2,DCS ;GET CONTROL STATUS REGISTER ADDRESS
586 011304 016237 000002 002302 MOV 2(R2),DRSEL ;GET DRIVE SELECT BITS
587 011312 005777 170732 TST @DCS ;ACCESS CONTROLLER
588 011316 005737 002470 TST TRPFLG ;DID TRAP OCCUR?
589 011322 001441 BEQ 2$ ;BRANCH TO CHECK DRIVE IF TRAP DID NOT
590 ;/OCCUR
591 011324 PRINTF #FMT17,#MRLCS,DCS,#DRNM,<B,DRSEL+1> ;GIVE CONTROL STATUS AND
(11) 011324 005046 CLR -(SP)
(11) 011326 153716 002303 BISB DRSEL+1,(SP)
(10) 011332 012746 004033 MOV #DRNM,-(SP)
(9) 011336 013746 002250 MOV DCS,-(SP)
(8) 011342 012746 002506 MOV #MRLCS,-(SP)
(7) 011346 012746 007537 MOV #FMT17,-(SP)
(6) 011352 012746 000005 MOV #5,-(SP)
(3) 011356 010600 MOV SP,R0
(4) 011360 104417 TRAP C$PNTF
(4) 011362 062706 000014 ADD #14,SP
592 ;/DRIVE INFORMATION
593 011366 PRINTF #FMT15,#NOCTLR ;MSG. 'DROPPING DRIVE - NO CONTROLLER'
(8) 011366 012746 007072 MOV #NOCTLR,-(SP)
(7) 011372 012746 007474 MOV #FMT15,-(SP)
(6) 011376 012746 000002 MOV #2,-(SP)
(3) 011402 010600 MOV SP,R0
(4) 011404 104417 TRAP C$PNTF
(4) 011406 062706 000006 ADD #6,SP
594 011412 DODU LOGUNIT ;DO DROP UNIT ON DRIVE FROM TEST CYCLE
(3) 011412 013700 002374 MOV LOGUNIT,R0
(3) 011416 104451 TRAP C$DODU
595 011420 005022 CLR (R2)+ ;CLEAR CONTROL STATUS REGISTER ADDRESS
596 ;/ENTRY IN SELECT TABLE
597 011422 005022 CLR (R2)+ ;CLEAR DRIVE SELECT ENTRY IN SELECT TABLE
598 011424 000474 BR 5$ ;BRANCH TO ACCESS NEXT DRIVE
599 011426 052737 000200 002302 2$: BIS #200,DRSEL ;ADD IN THE CRDY BIT

```

```

600 011434 013777 002302 170606      MOV      DRSEL,DCS      ;SELECT THE DRIVE
601 011442 012701 000074      MOV      #60.,R1      ;INITIALIZE TIMER
602 011446 032777 000001 170574 3$: BIT      #1,DCS      ;DRIVE READY?
603 011454 001057      BNE      4$           ;BRANCH TO ACCESS NEXT DRIVE IF READY
604 011456      WAITUS  #10.      ;IMPLEMENT A TIME DELAY
(3) 011456 012727 000012      MOV      ##10.,(PC)+
(3) 011462 000000      .WORD   0
(3) 011464 013727 002116      MOV      L$DLY,(PC)+
(3) 011470 000000      .WORD   0
(3) 011472 005367 177772      DEC      -6(PC)
(3) 011476 001375      BNE      -4
(3) 011500 005367 177756      DEC      -22(PC)
(3) 011504 001367      BNE      -20
605 011506 005301      DEC      R1           ;DECREMENT THE TIMER
606 011510 001356      BNE      3$           ;BRANCH IF TIME NOT ELAPSED
607 011512      PRINTF #FMT17,#MRLCS,DCS,#DRNM,<B,DRSEL+1> ;GIVE CONTROL STATUS AND
(11) 011512 005046      CLR      -(SP)
(11) 011514 153716 002303      BISB    DRSEL+1,(SP)
(10) 011520 012746 004033      MOV      #DRNM,-(SP)
(9) 011524 013746 002250      MOV      DCS,-(SP)
(8) 011530 012746 002506      MOV      #MRLCS,-(SP)
(7) 011534 012746 007537      MOV      #FMT17,-(SP)
(6) 011540 012746 000005      MOV      #5,-(SP)
(3) 011544 010600      MOV      SP,R0
(4) 011546 104417      TRAP    C$PNTF
(4) 011550 062706 000014      ADD     #14,SP
608      ;/DRIVE INFORMATION
609 011554      PRINTF #FMT15,#NOTRDY ;MSG. 'DRIVE DROPPED - DID NOT RESPOND
(8) 011554 012746 006505      MOV      #NOTRDY,-(SP)
(7) 011560 012746 007474      MOV      #FMT15,-(SP)
(6) 011564 012746 000002      MOV      #2,-(SP)
(3) 011570 010600      MOV      SP,R0
(4) 011572 104417      TRAP    C$PNTF
(4) 011574 062706 000006      ADD     #6,SP
610      ;/WITH 'READY''
611 011600      DODU    LOGUNIT      ;DO DROP UNIT ON DRIVE FROM TEST CYCLE
(3) 011600 013700 002374      MOV      LOGUNIT,R0
(3) 011604 104451      TRAP    C$DODU
612 011606 005022      CLR      (R2)+
613      ;CLEAR CONTROL STATUS REGISTER ADDRESS
614 011610 005022      CLR      (R2)+      ;/ENTRY IN SELECT TABLE
615 011612 000401      BR      5$           ;CLEAR DRIVE SELECT ENTRY IN SELECT TABLE
616 011614 022222 002374 4$: CMP      (R2)+,(R2)+ ;BRANCH TO ACCESS NEXT DRIVE
617 011616 005237 002374 5$: INC      LOGUNIT    ;ACCESS NEXT DRIVE IN SELECT TABLE
618 011622 005303      DEC      R3           ;INCREMENT LOGICAL UNIT NUMBER
619 011624 001210      BNE      1$           ;DECREMENT DRIVE COUNT
620 011626      CLRVEC ERRVEC      ;BRANCH TO GET NEXT DRIVE IF MORE
(3) 011626 013700 002504      MOV      ERRVEC,R0  ;RELEASE THE ERROR VECTOR
(3) 011632 104436      TRAP    C$VEC
621 011634 012603      MOV      (SP)+,R3
622 011636 012602      MOV      (SP)+,R2
623 011640 012601      MOV      (SP)+,R1
624 011642      ENDAUTO
(3) 011642      L10007:
(3) 011642 104461      TRAP    C$AUTO
625

```

627	011644			BGNMOD	CLNCODE	
628	011644				BGNCLN	
629						
630	011644				SETVEC	ERRVEC,#TRPHAN,#340
(7)	011644	012746	000340		MOV	#340,-(SP)
(6)	011650	012746	012474		MOV	#TRPHAN,-(SP)
(5)	011654	013746	002504		MOV	FRRVEC,-(SP)
(4)	011660	012746	0000C3		MOV	#3,-(SP)
(3)	011664	104437			TRAP	CSSVEC
(2)	011666	062706	000010		ADD	#10,SP
631	011672				SETPRI	#PRI00 ;PRIORITY TO ZERO
(3)	011672	012700	000000		MOV	#PRI00,RO
(3)	011676	104441			TRAP	CSSPRI
632	011700				CLRVEC	BVEC ;RELEASE VECTOR OF FIRST CONTROLLER
(3)	011700	013700	002350		MOV	BVEC,RO
(3)	011704	104436			TRAP	CSCVEC
633						
634	011706			3\$:	CLRVEC	ERRVEC
(3)	011706	013700	002504		MOV	ERRVEC,RO
(3)	011712	104436			TRAP	CSCVEC
635	011714	005737	002372		TST	SYSCLK
636	011720	001400			BEQ	4\$
637	011722			4\$:	BRESET	;TAKE CARE OF LSI-11
(3)	011722	104433			TRAP	C\$RESET
638	011724				ENDCLN	
(3)	011724			L10010:		
(3)	011724	104412			TRAP	C\$CLEAN
639						
640	011726				ENDMOD	
641						
642	011726			BGNMOD	ADDCODE	
643	011726			BGNAU		
644	011726			ENDAU		
(3)	011726			L10011:		
(3)	011726	104452			TRAP	C\$AU
645	011730			ENDMOD		
646						
647	011730			BGNMOD	DROPCODE	
648	011730			BGNDU		
649	011730			ENDDU		
(3)	011730			L10012:		
(3)	011730	104453			TRAP	C\$DU
650	011732			ENDMOD		
651						
652						

```

654 .SBTTL CLOCK INTERRUPT SERVICE ROUTINES
655
656 011732 STARS
(2) :*****
657 :UPDATES TIME EVERY 1/60 SECOND (60 HZ) OR EVERY 1/50 SECOND
658 : (50 HZ)
659 011732 STARS
(2) :*****
660
661 011732 BGNSRV UPDATE
662 011732 010446 MOV R4,-(SP) ;SAVE R4
663 :CLEAR CLOCK INTERRUPT ENABLE TO INHIBIT CLOCK INTERRUPTS DURING UPDATING
664 :OF TIME FIELDS
665 011734 022737 000001 002356 CMP #1,CLKTYP ;P-CLOCK?
666 011742 001004 BNE 1$ ;BRANCH IF NOT P-CLOCK
667 011744 042737 000100 172540 BIC #100,@#172540 ;DISABLE P-CLOCK INTERRUPT FACILITY
668 011752 000403 BR 2$
669 011754 042737 000100 177546 1$: BIC #100,@#177546 ;DISABLE L-CLOCK INTERRUPT FACILITY
670 :UPDATE TIME FIELDS
671 011762 012704 002424 2$: MOV #TICK,R4 ;INITIALIZE TICK ADDRESS
672 011766 005214 INC (R4) ;INCREMENT TICK TIME FIELD
673 011770 023727 002354 000002 CMP CLKFRQ,#2 ;50 HZ CLOCK?
674 011776 001005 BNE 3$ ;NO - BRANCH FOR SERVICING 60 HZ CLOCK
675 012000 021427 000062 CMF (R4),#50. ;((R4))=50?
676 012004 001024 BNE EXIT ;IF NOT,UPDATING IS COMPLETE
677 012006 005014 CLR (R4) ;ELSE,((R4))=0 (RESET COUNT)
678 012010 000404 BR 4$ ;BRANCH TO UPDATE 'SECOND' TIME FIELD
679 012012 021427 000074 3$: CMP (R4),#60. ;((R4))=60?
680 012016 001017 BNE EXIT ;IF NOT,UPDATING IS COMPLETE
681 012020 005014 CLR (R4) ;ELSE,((R4))=0 (RESET COUNT)
682 012022 005724 4$: TST (R4)+ ;(R4)=(R4)+2 (GO TO NEXT TIME FIELD)
683 012024 005214 INC (R4) ;INCREMENT 'SECOND' TIME FIELD
684 012026 021427 000074 CMF (R4),#60. ;((R4))=60?
685 012032 001011 BNE EXIT ;IF NOT,UPDATING IS COMPLETE
686 012034 005014 CLR (R4) ;ELSE,((R4))=0 (RESET COUNT)
687 012036 005724 TST (R4)+ ;ACCESS 'MINUTE' TIME FIELD
688 012040 005214 INC (R4) ;INCREMENT 'MINUTE' TIME FIELD
689 012042 021427 000074 CMF (R4),#60. ;((R4))=60?
690 012046 001003 BNE EX' ;IF NOT,UPDATING IS COMPLETE
691 012050 005014 CLR (R4) ;ELSE,((R4))=0 (RESET COUNT)
692 012052 005724 TST (R4)+ ;ACCESS 'HOUR' TIME FIELD
693 012054 005214 INC (R4) ;INCREMENT 'HOUR' TIME FIELD
694 012056 005337 002370 EXIT: DEC CLKBFR ;COUNT CLOCK TICKS
695 012062 003003 BGT 5$ ;TIME NOT EXPIRED
696 012064 013737 002366 002370 MOV CLKCNT,CLKBFR ;RE-INITIALIZE TIME INCREMENT
697 :RE-ENABLE CLOCK INTERRUPT FACILITY
698 012072 022737 000001 002356 5$: CMP #1,CLKTYP ;P-CLOCK?
699 012100 001004 BNE 6$ ;BRANCH IF NOT P-CLOCK
700 012102 052737 000100 172540 BIS #100,@#172540 ;SET P-CLOCK INTERRUPT ENABLE BIT
701 012110 000403 BR 7$ ;EXIT
702 012112 052737 000100 177546 6$: BIS #100,@#177546 ;SET L-CLOCK INTERRUPT ENABLE BIT
703 012120 012604 7$: MOV (SP)+,R4 ;RESTORE R4
704 012122 ENDSRV
(3) 012122 L10013:
(2) 012122 000002 RTI
705

```

CZRLMBO RL01/02 BD SEC FIL TL  
CZRLMB.MAC 12-DEC-79 14:06

\*ACY11 30A(1052) 17-DEC-79 10:53 E 4 PAGE 1-20  
CLOCK INTERRUPT SERVICE ROUTINES

706  
707 012124  
708  
709 012124 005237 002376  
710  
711  
712 012130  
(3) 012130  
(2) 012130 000002  
713

;L-CLOCK 'TICK' CHECK ROUTINE FOR LSI-11  
BGNSRV CLKTIK

INC CLKFLD

;INCREMENT CLOCK FIELD TO INDICATE  
;/THAT CLOCK IS 'TICKING'

ENDSRV  
L10014:

RTI



715  
716  
717 012132  
718  
719  
720 012132  
(2)  
721  
722  
723 012132  
(2)  
724  
725 012132 010346  
726 012134 022737 000001 002356  
727 012142 001014  
728 012144  
(7) 012144 012746 000340  
(6) 012150 012746 011732  
(5) 012154 012746 000104  
(4) 012160 012746 000003  
(3) 012164 104437  
(2) 012166 062706 000010  
729 012172 000417  
730 012174 022737 000002 002356  
731 012202 001036  
732 012204  
(7) 012204 012746 000340  
(6) 012210 012746 011732  
(5) 012214 012746 000100  
(4) 012220 012746 000003  
(3) 012224 104437  
(2) 012226 062706 000010  
733 012232 013703 002360  
734 012236 022763 000074 000006  
735 012244 001007  
736 012246 012737 000074 002366  
737 012254 012737 000001 002354  
738 012262 000406  
739 012264 012737 000062 002366  
740 012272 012737 000002 002354  
741 012300 012603  
742  
743 012302 000207  
744  
745

```

.SBTTL GLOBAL SUBROUTINES
BGNMOD GLBSUB

STARS
:*****
:SET UP CLOCK INTERRUPT VECTOR, CLOCK COUNT, AND IDENTIFY THE
:CLOCK FREQUENCY
STARS
:*****

CLKINI: MOV R3,-(SP) ;SAVE R3
        CMP #1,CLKTYP ;P-CLOCK?
        BNE LCLK ;BRANCH IF NOT P-CLOCK
        SETVEC #104,#UPDATE,#340 ;SET P-CLOCK INTERRUPT VECTOR
        MOV #340,-(SP)
        MOV #UPDATE,-(SP)
        MOV #104,-(SP)
        MOV #3,-(SP)
        TRAP C$SVEC
        ADD #10,SP
        BR FRQCHK ;BRANCH FOR SYSTEM FREQUENCY CHECK
LCLK:   CMP #2,CLKTYP ;L-CLOCK?
        BNE ENDINI ;BRANCH IF NO CLOCK
        SETVEC #100,#UPDATE,#340 ;SET L-CLOCK INTERRUPT VECTOR
        MOV #340,-(SP)
        MOV #UPDATE,-(SP)
        MOV #100,-(SP)
        MOV #3,-(SP)
        TRAP C$SVEC
        ADD #10,SP
FRQCHK: MOV CLKADR,R3 ;GET BASE ADDRESS OF THE SUPERVISOR CLOCK TABLE
        CMP #60,,6(R3) ;60 HZ?
        BNE FRQ50 ;BRANCH FOR 50 HZ
        MOV #60,,CLKCNT ;INITIALIZE CLOCK COUNT FOR 60 TICKS PER SECOND
        MOV #1,CLKFRQ ;IDENTIFY CLOCK FREQUENCY IS 60 HZ
        BR ENDINI ;RETURN
FRQ50:  MOV #50,,CLKCNT ;INITIALIZE CLOCK COUNT FOR 50 TICKS PER SECOND
        MOV #2,CLKFRQ ;IDENTIFY CLOCK FREQUENCY IS 50 HZ
ENDINI: MOV (SP)+,R3 ;RESTORE R3

RTS PC ;RETURN

```

```

747
748 012304          STARS
(2)                :*****
749                :START CLOCK OPERATION
750 012304          STARS
(2)                :*****
751
752 012304 022737 000002 002356 CLKST:  CMP    #2,CLKTYP      ;L-CLOCK?
753 012312 001006                BNE    1$          ;BRANCH FOR P-CLOCK
754 012314 012737 000100 177546        MOV    #100,@#177546 ;SET INTERRUPT ENABLE BIT TO 1
755 012322 005237 002364                INC    CLKSON      ;INDICATE CLOCK IS 'ON'
756 012326 000414                BR     2$          ;BRANCH TO SET UP TIME INCREMENTS
757 012330 022737 000001 002356 1$:    CMP    #1,CLKTYP    ;P-CLOCK?
758 012336 001013                BNE    3$          ;BRANCH IF NO CLOCK
759 012340 012737 000001 172542        MOV    #1,@#172542 ;SET UP P-CLOCK FOR 1 INTERRUPT PER TICK
760 012346 012737 000115 172540        MOV    #115,@#172540 ;SET INTERRUPT ENABLE,REPEAT INTERRUPT MODE,
761                                ;/LINE FREQUENCY RATE,START CLOCK
762 012354 005237 002364                INC    CLKSON      ;INDICATE CLOCK IS 'ON'
763 012360 013737 002366 002370 2$:    MOV    CLKCNT,CLKBFR ;SET UP TIME INCREMENTS
764 012366 000207                3$:    RTS     PC       ;RETURN
765
766 012370          STARS
(2)                :*****
767                :FIRST & SELDRV -- DRIVE SELECT ROUTINE
768 012370          STARS
(2)                :*****
769
770 012370 012704 011162          FIRST:  MOV    #SELTBL,R4      ;POINT TO THE SELECT TABLE
771 012374 010437 002330                MOV    R4,NXTUNI
772
773 012400 013704 002330          SELDRV: MOV    NXTUNI,R4      ;SETUP THE POINTER
774 012404 005714          10$:    TST    (R4)          ;CHECK FOR A VALID ENTRY
775 012406 100402                BMI    1$          ;OK TO GO ON
776 012410 022124                CMP    (R4)+,(R4)+  ;POINT TO THE NEXT ENTRY SLOT
777 012412 000774                BR     10$         ;AND TRY AGAIN
778
779 012414 012437 002250          1$:    MOV    (R4)+,DCS    ;GET THE CSR ADDR FROM TABLE
780 012420 022737 177777 002250        CMP    #-1,DCS     ;END OF THE TABLE?
781 012426 001001                BNE    2$          ;NO - CONTINUE
782 012430 000416                BR     4$          ;EXIT +1
783
784 012432 012437 002302          2$:    MOV    (R4)+,DRSEL  ;GET THE DRIVE SELECT BITS
785 012436 004537 025422                JSR    R5,GETDST   ;GET THE DRIVE STATUS
786 012442 012737 000001 002316        MOV    #1,TDR     ;DEFAULT TO RL01 TYPE
787 012450 032701 000200                BIT    #BIT7,R1   ;IS IT AN RL02?
788 012454 001403                BEQ    3$          ;NO
789 012456 012737 000002 002316        MOV    #2,TDR     ;YES - SET FOR AN RL02
790
791 012464 022525          3$:    CMP    (R5)+,(R5)+  ;RETURN +2 - NORMAL EXIT
792 012466 010437 002330          4$:    MOV    R4,NXTUNI  ;SAVE THE 'NEXT' SLOT POINTER
793 012472 000205                RTS     R5        ;EXIT
794
795 012474 005237 002470          TRPHAN: INC    TRPFLG
796 012500 000002                RTI
797
798

```

```

799 012502
(2)
800
801 012502
(2)
802
803 012502 005737 002372
804 012506 001416
805 012510
(10) 012510 013746 002426
(9) 012514 013746 002430
(8) 012520 013746 002432
(7) 012524 012746 010047
(6) 012530 012746 000004
(3) 012534 010600
(4) 012536 104414
(4) 012540 062706 000012
806 012544 000205
807 012546
(2)
808
809 012546
(2)
810
811 012546
(11) 012546 005046
(11) 012550 153716 002303
(10) 012554 012746 004033
(9) 012560 013746 002250
(8) 012564 012746 002506
(7) 012570 012746 007537
(6) 012574 012746 000005
(3) 012600 010600
(4) 012602 104417
(4) 012604 062706 000014
812 012610
(9) 012610 013746 002316
(8) 012614 012746 006562
(7) 012620 012746 010105
(6) 012624 012746 000003
(3) 012630 010600
(4) 012632 104417
(4) 012634 062706 000010
813 012640 000205
814
815 012642
(2)
816
817 012642
(2)
818
819 012642 004537 012546
820 012646
(8) 012646 012746 006505
(7) 012652 012746 010044
(6) 012656 012746 000002

```

```

STARS
:*****
:PTIME -- ROUTINE TO PRINT THE SYSTEM RUNTIME IF A CLOCK IS PRESENT
STARS
:*****

PTIME:  TST      SYSCLK      ;CLOCK PRESENT?
        BEQ      1$          ;NO
        PRINTB   #TIME,HOUR,MINUTE,SECOND
        MOV      SECOND,-(SP)
        MOV      MINUTE,-(SP)
        MOV      HOUR,-(SP)
        MOV      #TIME,-(SP)
        MOV      #4,-(SP)
        MOV      SP,R0
        TRAP     C$PNTB
        ADD      #12,SP
1$:     RTS      R5          ;EXIT
STARS
:*****
:DRVID -- ROUTINE TO PRINT THE SELECTED UNIT IDENTIFICATION
STARS
:*****

DRVID:  PRINTF   #FMT17,#MRLCS,DCS,#DRNM,<B,DRSEL+1>
        CLR      -(SP)
        BISB     DRSEL+1,(SP)
        MOV      #DRNM,-(SP)
        MOV      DCS,-(SP)
        MOV      #MRLCS,-(SP)
        MOV      #FMT17,-(SP)
        MOV      #5,-(SP)
        MOV      SP,R0
        TRAP     C$PNTF
        ADD      #14,SP
        PRINTF   #FD1YP,#MDRTYP,TDR
        MOV      TDR,-(SP)
        MOV      #MDRTYP,-(SP)
        MOV      #FD1YP,-(SP)
        MOV      #3,-(SP)
        MOV      SP,R0
        TRAP     C$PNTF
        ADD      #10,SP
        RTS      R5
STARS
:*****
:DRRDY -- ROUTINE TO PRINT THE DRIVE SELECTED ISN'T READY
STARS
:*****

DRRDY:  PRINTF   #FMT17,#MRLCS,DCS,#DRNM,<B,DRSEL+1>
        CLR      -(SP)
        BISB     DRSEL+1,(SP)
        MOV      #DRNM,-(SP)
        MOV      DCS,-(SP)
        MOV      #MRLCS,-(SP)
        MOV      #FMT17,-(SP)
        MOV      #5,-(SP)
        MOV      SP,R0
        TRAP     C$PNTF
        ADD      #14,SP
        PRINTF   #FD1YP,#MDRTYP,TDR
        MOV      TDR,-(SP)
        MOV      #MDRTYP,-(SP)
        MOV      #FD1YP,-(SP)
        MOV      #3,-(SP)
        MOV      SP,R0
        TRAP     C$PNTF
        ADD      #10,SP
        RTS      R5

```

0034

CZRLMBO RL01/02 BD SEC FIL TL  
CZRLMB.MAC 12-DEC-79 14:06

MACV11 30A(10>2) 17-DEC-79 10:53 PAGE 1-24  
GLOBAL SUBROUTINES

SEQ 0047

```

(3) 012662 010600          MOV      SP,R0
(4) 012664 104417          TRAP    C$PNTF
(4) 012666 062706 000006   ADD     #6,SP
821 012672 004537 012700   JSR    R5,DRDRV          ;DROP THE DRIVE SELECTED
822 012676 000205          RTS     R5
823
824 012700          STARS
(2) 012700          :*****
825          :DRDRV -- ROUTINE TO KILL A UNIT ENTRY INTO THE SELTBL AREA IF THE
826          :          PGM DETERMINES A UNIT IS NOT ABLE TO BE USED
827 012700          STARS
(2) 012700          :*****
828
829 012700 013704 002330   DRDRV: MOV     NXTUNI,R4          ;POINT TO THE 'NEXT' UNIT SLOT
830 012704 162704 000004   SUB     #4,R4                ;POINT TO THE CURRENT UNIT
831 012710 005024          CLR    (R4)+                 ;KILL THE ENTRY
832 012712 005024          CLR    (R4)+                 ;KILL DRSEL ENTRY
833 012714 000205          RTS     R5                  ;EXIT
834
835
836 012716          ENDMOD
837
838
839

```

```

841 .SBTTL PROGRAM MAIN LOOP
842
843 012716 BGNTST
844 012716 STARS
(2) ;:*****
845 ;THIS IS WHERE CONTROL IS PASSED AFTER THE INITIAL QUESTIONS HAVE
846 ;BEEN ANSWERED FOR THE P-TABLE STORAGE.
847 012716 STARS
(2) ;:*****
848
849 012716 MTEST:
850 012716 004537 012370 JSR R5,FIRST ;SELECT THE 1ST DRIVE
851 012722 000137 013216 JMP WHATCMD ;NO - UNITS
852 012726 000404 BR 2$
853 012730 004537 012400 1$: JSR R5,SELDRV ;SELECT ANOTHER UNIT
854 012734 000137 013216 JMP WHATCMD ;NO MORE TO SELECT
855 012740 012777 000200 167302 2$: MOV #200,@DCS ;CHECK IF DRIVE THERE
856 012746 053777 002302 167274 BIS DRSEL,@DCS
857 012754 012700 000000 MOV #0.,R0 ;STALL
858 012760 005300 13$: DEC R0
859 012762 001376 BNE 13$
860 012764 004537 025422 JSR R5,GETDST ;GET THE CURRENT DRIVE STATUS
861 012770 010137 002414 MOV R1,TEMPO ;SAVE THE STATUS
862 012774 PRINTF #MCRLF
(7) 012774 012746 010041 MOV #MCRLF,-(SP)
(6) 013000 012746 000001 MOV #1,-(SP)
(3) 013004 010600 MOV SP,R0
(4) 013006 104417 TRAP C$PNTF
(4) 013010 062706 000004 ADD #4,SP
863 013014 004537 012546 JSR R5,DRVID ;TELL OPR THE UNIT SELECTED
864
865 013020 032737 000020 002414 130$: BIT #HOP,TEMPO ;ARE THE HEADS LOADED?
866 013026 001015 BNE 131$ ;BR IF OK
867 013030 PRINTF #FMT18,#NOLOAD ;NO
(8) 013030 012746 003433 MOV #NOLOAD,-(SP)
(7) 013034 012746 007624 MOV #FMT18,-(SP)
(6) 013040 012746 000002 MOV #2,-(SP)
(3) 013044 010600 MOV SP,R0
(4) 013046 104417 TRAP C$PNTF
(4) 013050 062706 000006 ADD #6,SP
868 013054 004537 012700 JSR R5,DRDRV ;DROP THIS DRIVE
869 013060 000452 BR 15$
870 013062 032737 020000 002414 131$: BIT #WL,TEMPO ;IS THE PACK WRITE LOCKED?
871 013070 001414 BEQ 132$ ;BR IF NOT WRT LOCKED
872 013072 PRINTF #FMT18,#WRTLCK ;TELL OPR
(8) 013072 012746 003460 MOV #WRTLCK,-(SP)
(7) 013076 012746 007624 MOV #FMT18,-(SP)
(6) 013102 012746 000002 MOV #2,-(SP)
(3) 013106 010600 MOV SP,R0
(4) 013110 104417 TRAP C$PNTF
(4) 013112 062706 000006 ADD #6,SP
873 013116 005237 002462 INC WRTLCK ;SET THE WRITE LOCK FLAG
874 013122 032737 001000 002414 132$: BIT #VC,TEMPO ;PACK JUST LOADED?
875 013130 001412 BEQ 133$ ;JUMP IF NOT
876 013132 PRINTF #FMT18,#NEWLD ;TELL OPR
(8) 013132 012746 003505 MOV #NEWLD,-(SP)

```

```

(7) 013136 012746 007624      MOV    #FMT18,-(SP)
(6) 013142 012746 000002      MOV    #2,-(SP)
(3) 013146 010600              MOV    SP,R0
(4) 013150 104417              TRAP   C$PNTF
(4) 013152 062706 000006      ADD    #6,SP
877 013156 004537 025436      133$: JSR    R5,ISDRST      ;RESET THE DRIVE
878 013162 004537 025422      JSR    R5,GETDST      ;GET THE DRIVE STATUS AGAIN
879 013166 032777 100000 167054 BIT    #ERR,@DCS      ;COMPOSITE ERROR STILL SET?
880 013174 001404              BEQ    15$            ;NOPE - SKIP OVER
881 013176              ERRDF  170.,MDERS
(4) 013176 104455              TRAP   C$ERDF
(5) 013200 000252              .WORD  170
(5) 013202 003337              .WORD  MDERS
(5) 013204 000000              .WORD  0
882
883 013206              15$:  SETPRI #0          ;PRIORITY TO ZERO
(3) 013206 012700 000000      MOV    #0,R0
(3) 013212 104441              TRAP   C$SPRI
884 013214 000645              BR     1$            ;SELECT THE NEXT UNIT
885
886

```

888  
889  
890 013216  
(2)  
891  
892 013216  
(2)  
893  
894 013216 005737 002474  
895 013222 001551  
896 013224 005737 002344  
897 013230 001440  
898 013232  
(7) 013232 012746 010041  
(6) 013236 012746 000001  
(3) 013242 010600  
(4) 013244 104417  
(4) 013246 062706 000004  
899 013252  
(3) 013252 104443  
(3) 013254 000406  
(4) 013256 002414  
(5) 013260 000022  
(5) 013262 004044  
(5) 013264 177777  
(5) 013266 000001  
(5) 013270 177777  
(3) 013272  
900 013 023737 002344 002414  
901 013500 001414  
902 013302 005237 002464  
903 013306  
(8) 013306 012746 004125  
(7) 013312 012746 007624  
(6) 013316 012746 000002  
(3) 013322 010600  
(4) 013324 104417  
(4) 013326 062706 000006  
904  
905 013332  
(8) 013332 012746 004326  
(7) 013336 012746 007777  
(6) 013342 012746 000002  
(3) 013346 010600  
(4) 013350 104417  
(4) 013352 062706 000006  
906 013356  
(8) 013356 012746 004401  
(7) 013362 012746 007624  
(6) 013366 012746 000002  
(3) 013372 010600  
(4) 013374 104417  
(4) 013376 062706 000006  
907 013402  
(8) 013402 012746 004461  
(7) 013406 012746 007624

```
.SBTTL COMMAND QUERY LOOP

STARS
:*****
:HERE IS THE 'CMD>' QUERY LOOP FOR COMMANDS TO PERFORM
STARS
:*****

WHATCMD: WHATCMD: TST STFLG ;JUST STARTING?
          BEQ NXTCMD ;NO - BYPASS THE STARTING BLURB
          TST PASWD ;DO THE PASSWORD STUFF?
          BEQ HLPMSG ;NO - PRINT THE HELP MESSAGE
          PRINTF #MCRLF
          MOV #MCRLF, -(SP)
          MOV #1, -(SP)
          MOV SP, R0
          TRAP C$PNTF
          ADD #4, SP
          GMANID PASWD, TEMPO, 0, 177777, 1, 177777, NO ;GET THE PASSWORD
          TRAP C$GMAN
          BR 10000$
          .WORD TEMPO
          .WORD T$CODE
          .WORD PASWD
          .WORD 177777
          .WORD T$LOLIM
          .WORD T$HILIM

10000$:
          CMP PASWD, TEMPO ;CORRECT PASSWORD?
          BEQ HLPMSG ;YES
          INC ACCESS ;SET THE DENIED FLAG
          PRINTF #FMT18, #DENIED ;& TELL OPR
          MOV #DENIED, -(SP)
          MOV #FMT18, -(SP)
          MOV #2, -(SP)
          MOV SP, R0
          TRAP C$PNTF
          ADD #6, SP

HLPMSG: PRINTF #FMTMS, #CMD1 ;PRINT THE HELP MESSAGE
          MOV #CMD1, -(SP)
          MOV #FMTMS, -(SP)
          MOV #2, -(SP)
          MOV SP, R0
          TRAP C$PNTF
          ADD #6, SP
          PRINTF #FMT18, #CMD2
          MOV #CMD2, -(SP)
          MOV #FMT18, -(SP)
          MOV #2, -(SP)
          MOV SP, R0
          TRAP C$PNTF
          ADD #6, SP
          PRINTF #FMT18, #CMD3
          MOV #CMD3, -(SP)
          MOV #FMT18, -(SP)
```

```

(6) 013412 012746 000002      MOV      #2,-(SP)
(3) 013416 010600              MOV      SP,RO
(4) 013420 104417              TRAP    (SPNTF)
(4) 013422 062706 000006      ADD      #6,SP
908 013426                      PRINTF  #FMT18,#CMD4
(8) 013426 012746 004546      MOV      #CMD4,-(SP)
(7) 013432 012746 007624      MOV      #FMT18,-(SP)
(6) 013436 012746 000002      MOV      #2,-(SP)
(3) 013442 010600              MOV      SP,RO
(4) 013444 104417              TRAP    (SPNTF)
(4) 013446 062706 000006      ADD      #6,SP
909 013452                      PRINTF  #FMT18,#CMD5
(8) 013452 012746 004602      MOV      #CMD5,-(SP)
(7) 013456 012746 007624      MOV      #FMT18,-(SP)
(6) 013462 012746 000002      MOV      #2,-(SP)
(3) 013466 010600              MOV      SP,RO
(4) 013470 104417              TRAP    (SPNTF)
(4) 013472 062706 000006      ADD      #6,SP
910 013476                      PRINTF  #FMT18,#CMD6
(8) 013476 012746 004671      MOV      #CMD6,-(SP)
(7) 013502 012746 007624      MOV      #FMT18,-(SP)
(6) 013506 012746 000002      MOV      #2,-(SP)
(3) 013512 010600              MOV      SP,RO
(4) 013514 104417              TRAP    (SPNTF)
(4) 013516 062706 000006      ADD      #6,SP
911 013522                      PRINTF  #FMT18,#CMD7
(8) 013522 012746 004724      MOV      #CMD7,-(SP)
(7) 013526 012746 007624      MOV      #FMT18,-(SP)
(6) 013532 012746 000002      MOV      #2,-(SP)
(3) 013536 010600              MOV      SP,RO
(4) 013540 104417              TRAP    (SPNTF)
(4) 013542 062706 000006      ADD      #6,SP
912
913 013546 005037 015330      NXTCMD: CLR      FACNUM          ;CLEAR ENTRY COUNTER
914 013552 005037 015326      CLR      FLDNUM
915 013556                      PRINTF  #MCRLF
(7) 013556 012746 010041      MOV      #MCRLF,-(SP)
(6) 013562 012746 000001      MOV      #1,-(SP)
(3) 013566 010600              MOV      SP,RO
(4) 013570 104417              TRAP    (SPNTF)
(4) 013572 062706 000004      ADD      #4,SP
916 013576                      GMANID  CMDDO,INPUT,D,7,1,7,NO ;PROMPT 'ENTER COMMAND (1-7) -'
(3) 013576 104443              TRAP    (SGMAN)
(3) 013600 000406              BR      10001$
(4) 013602 013650              .WORD  INPUT
(5) 013604 000042              .WORD  T$CODE
(5) 013606 004753              .WORD  CMDDO
(5) 013610 000007              .WORD  7
(5) 013612 000001              .WORD  T$LOLIM
(5) 013614 000007              .WORD  T$HILIM
(3) 013616                      10001$:
917
918 013616 013700 013650      MOV      INPUT,RO          ;GET THE CMD REQUEST TYPED
919 013622 006300              ASL      RO                ;SHIFT FOR PROPER INDEX INTO LIST
920 013624 000170 013630      JMP      @LIST(RO)         ;DO THE FUNCTION REQUESTED
921

```



CZRLMB0 RL01/02 BD SEC FIL TL  
CZRLMB.MAC 12-DEC-79 14:06

MACY11 30A(1052) 17-DEC-79 10:53 N 4  
COMMAND QUERY LOOP PAGE 1-29

922	013630	000000
923	013632	013652
924	013634	015754
925	013636	017352
926	013640	020644
927	013642	020172
928	013644	023030
929	013646	013332
930		
931	013650	000000
932		

LIST: .WORD 0  
 BSRPT  
 BSADD  
 BSDEL  
 BSVERIFY  
 BSWRITE  
 BSMAKE  
 HLPMSG

INPUT: .WORD 0

;NOTHING FOR FUNCTION '0'  
 ; 1 REPORT CONTENTS OF BAD SECTOR FILES  
 ; 2 ADD AN ENTRY INTO 'FIELD' FILE  
 ; 3 DELETE AN ENTRY FROM 'FIELD' FILE  
 ; 4 VERIFY PACK - READ ONLY  
 ; 5 WRITE THE PACK  
 ; 6 MAKE A BAD SECTOR FILE  
 ; 7 PRINT THE COMMANDS AVAILABLE

;STORAGE FOR TYPED COMMAND

934  
935  
936 013652  
(2)  
937  
938  
939  
940  
941  
942 013652  
(2)  
943  
944 013652 004537 012370  
945 013656 000137 013546  
946 013662 000404  
947 013664 004537 012400  
948 013670 000137 013546  
949 013674 004537 025246  
950 013700 005737 002414  
951 013704 001404  
952 013706 004537 012642  
953 013712 000137 013664  
954  
955 013716  
(8) 013716 012746 002573  
(7) 013722 012746 007631  
(6) 013726 012746 000002  
(3) 013732 010600  
(4) 013734 104417  
(4) 013736 062706 000006  
956 013742 004537 023556  
957 013746  
(7) 013746 012746 010041  
(6) 013752 012746 000001  
(3) 013756 010600  
(4) 013760 104417  
(4) 013762 062706 000004  
958 013766 004537 012546  
959 013772  
(7) 013772 012746 010041  
(6) 013776 012746 000001  
(3) 014002 010600  
(4) 014004 104417  
(4) 014006 062706 000004  
960 014012 005037 015350  
961  
962  
963 014016 004537 025032  
964 014022  
(8) 014022 012746 005004  
(7) 014026 012746 007624  
(6) 014032 012746 000002  
(3) 014036 010600  
(4) 014040 104417  
(4) 014042 062706 000006  
965 014046 005037 015330

.SBTTL GLOBAL SUBROUTINES

STARS  
:\*\*\*\*\*  
:THIS IS THE ROUTINE TO REPORT THE CONTENTS OF THE BAD SECTOR FILE  
:FOR THE DRIVE SELECTED. 'BSFILE' CONTAINS AN IMAGE OF THE  
:CARTRIDGE BAD SECTOR FILE. FIRST REPORT THE CARTRIDGE SERIAL  
:NUMBER FOLLOWED BY THE CONTENTS OF THE 'FACTORY' BAD SECTOR FILE  
:AND THEN THE CONTENTS OF THE 'FIELD' BAD SECTOR FILE.  
STARS  
:\*\*\*\*\*

BSRPT: JSR R5,FIRST ;SELECT A DRIVE  
JMP NXTCMD ;NONE AVAIL!  
BR BSRPTL  
BSRPTS: JSR R5,SELDRV ;SELECT THE NEXT UNIT  
JMP NXTCMD ;ALL DONE  
BSRPTL: JSR R5,LOADED ;SEE IF DRIVE READY FOR OPR  
TST TEMPO ;READY?  
BEQ 1\$ ;YES  
JSR R5,DRNRDY  
JMP BSRPTS ;SELECT THE NEXT UNIT  
  
1\$: PRINTF #FMT19,#STARMSG  
MOV #STARMSG,-(SP)  
MOV #FMT19,-(SP)  
MOV #2,-(SP)  
MOV SP,R0  
TRAP CSPNTF  
ADD #6,SP  
JSR R5,RDBDSC ;READ THE BAD SECTOR FILE  
PRINTF #MCRLF  
MOV #MCRLF,-(SP)  
MOV #1,-(SP)  
MOV SP,R0  
TRAP CSPNTF  
ADD #4,SP  
JSR R5,DRVID  
PRINTF #MCRLF  
MOV #MCRLF,-(SP)  
MOV #1,-(SP)  
MOV SP,R0  
TRAP CSPNTF  
ADD #4,SP  
CLR PSNFG ;CLEAR THE PRINT FLAG FOR SER # MSG  
  
;HERE TO REPORT CONTENTS OF THE 'FACTORY' FILE  
BSRFAC: JSR R5,RDFACT ;READ THE FACTORY FILE FROM BD SEC FILE  
PRINTF #FMT18,#BSRM  
MOV #BSRM,-(SP)  
MOV #FMT18,-(SP)  
MOV #2,-(SP)  
MOV SP,R0  
TRAP CSPNTF  
ADD #6,SP  
CLR FACNUM ;INIT THE FACTORY ENTRY COUNTER

```
966 014052 012737 000020 015340      MOV      #16.,SECMAX      ;LAST SECTOR PAIR IN FACTORY FILE
967 014060 005037 015342      CLR      SECNUM          ;POINT TO THE 1ST PAIR OF SECTORS
968 014064 005002          CLR      R2              ;CLEAR THE INDEX INTO THE BSFILE STORAGE
969 014066 004537 015622      JSR      R5,BSFOK        ;FIND A SECTOR TO USE IN FACTORY AREA
970 014072 005737 015346      TST      BSFOK          ;SEE IF ERROR DETECTED
971 014076 001437          BEQ      10$            ;JUMP IF OK
972 014100          PRINTF  #FMT18,#NHWSEC
(8) 014100 012746 003562      MOV      #NHWSEC,-(SP)
(7) 014104 012746 007624      MOV      #FMT18,-(SP)
(6) 014110 012746 000002      MOV      #2,-(SP)
(3) 014114 010600          MOV      SP,R0
(4) 014116 104417          TRAP    C$PNTF
(4) 014120 062706 000006      ADD      #6,SP
973 014124 004537 022570      JSR      R5,NEWBSF       ;BUILD A NEW FILE
974 014130 005737 002414      TST      TEMPO          ;DID I?
975 014134 001405          BEQ      1$            ;NO
976 014136 013737 002414 002300      MOV      TEMPO,NEWFAC   ;SET THE FLAG
977 014144 004537 016770      JSR      R5,WRTBSF      ;WRITE UPDATED 'FIELD' BAD SECTOR FILE
978                                ;/ON THE PACK IF REQUESTED
979 014150          1$: PRINTF  #FMT18,#HYPHEN
(8) 014150 012746 002675      MOV      #HYPHEN,-(SP)
(7) 014154 012746 007624      MOV      #FMT18,-(SP)
(6) 014160 012746 000002      MOV      #2,-(SP)
(3) 014164 010600          MOV      SP,R0
(4) 014166 104417          TRAP    C$PNTF
(4) 014170 062706 000006      ADD      #6,SP
980 014174 000544          BR       BSRFLD         ;DO THE FIELD REPORT
981
982                                ;START PROCESSING THE ENTRIES
983                                10$:
984 014176          PRINTF  #FMTSN,#CART,SERNM2,SERNM1
(10) 014176 013746 002274      MOV      SERNM1,-(SP)
(9) 014202 013746 002276      MOV      SERNM2,-(SP)
(8) 014206 012746 002526      MOV      #CART,-(SP)
(7) 014212 012746 007707      MOV      #FMTSN,-(SP)
(6) 014216 012746 000004      MOV      #4,-(SP)
(3) 014222 010600          MOV      SP,R0
(4) 014224 104417          TRAP    C$PNTF
(4) 014226 062706 000012      ADD      #12,SP
985 014232 005237 015350      INC      PSNFG          ;SET THE FLAG
986 014236 005037 002416      CLR      TEMP1
987 014242 016203 030530 11$: MOV      BSFILE(R2),R3   ;GET THE CYLINDER # FROM ENTRY
988 014246 005703          TST      R3            ;SEE IF ITS OK TO USE
989 014250 100002          BPL     2$            ;OK
990 014252 000137 015352      JMP      NOFACT         ;WHOOOPS...ERROR
991 014256 005237 015330 2$: INC      FACNUM        ;COUNT THIS ENTRY
992 014262 022737 000176 015330      CMP      #126.,FACNUM   ;END OF FILE LIMIT?
993 014270 001506          BEQ     BSRFLD         ;YUP
994 014272 022737 000062 015330      CMP      #50.,FACNUM    ;TIME TO QUIT PRINTING?
995 014300 001040          BNE     21$           ;NO
996 014302 005737 002416          TST     TEMP1         ;PRINTED ERROR MESSAGE YET?
997 014306 001035          BNE     21$           ;YUP
998 014310          PRINTF  #FMT19,#OVRMAX ;TELL OPR OVER LIMIT
(8) 014310 012746 006251      MOV      #OVRMAX,-(SP)
(7) 014314 012746 007631      MOV      #FMT19,-(SP)
(6) 014320 012746 000002      MOV      #2,-(SP)
```

```
(3) 014324 010600      MOV      SP,RO
(4) 014326 104417      TRAP     C$PNTF
(4) 014330 062706 000006  ADD     #6,SP
999 014334          PRINTF  #MCRLF
(7) 014334 012746 010041  MOV     #MCRLF,-(SP)
(6) 014340 012746 000001  MOV     #1,-(SP)
(3) 014344 010600      MOV     SP,RO
(4) 014346 104417      TRAP     C$PNTF
(4) 014350 062706 000004  ADD     #4,SP
1000 014354          GMANIL  TILLEND,TEMPO,177777,NO
(3) 014354 104443      TRAP     C$GMAN
(3) 014356 000404      BR      10002$
(4) 014360 002414      .WORD   TEMPO
(5) 014362 000120      .WORD   T$CODE
(5) 014364 006114      .WORD   TILLEND
(5) 014366 177777      .WORD   177777
(3) 014370          10002$:
1001 014370 005737 002414  TST     TEMPO          ;NO?
1002 014374 001444      BEQ     BSRFLD        ;QUIT PRINTING ENTRIES
1003 014376 005237 002416  INC     TEMP1         ;SET THE PRINT ERROR FLAG
1004 014402 010337 015332  21$:  MOV     R3,BSFCYL     ;SAVE THE CYLINDER NUMBER
1005 014406 005722      TST     (R2)+         ;POINT TO HEAD & SECTOR ENTRY
1006 014410 016203 030530  MOV     BSFILE(R2),R3 ;GET IT
1007 014414 110337 015334  MOV     R3,BSFSEC     ;SAVE THE SECTOR NUMBER
1008 014420 000303      SWAB    R3           ;PUT THE HEAD # IN LOW BYTE
1009 014422 110337 015336  MOV     R3,BSFHD      ;SAVE THE HEAD NUMBER
1010 014426 005722      TST     (R2)+         ;POINT TO THE NEXT ENTRY
1011 014430          PRINTF  #FMTC$H,FACNUM,#CM$SG,BSFCYL,#SM$SG,BSFSEC,#HM$SG,BSFHD
(14) 014430 013746 015336  MOV     BSFHD,-(SP)
(13) 014434 012746 002563  MOV     #HM$SG,-(SP)
(12) 014440 013746 015334  MOV     BSFSEC,-(SP)
(11) 014444 012746 003243  MOV     #SM$SG,-(SP)
(10) 014450 013746 015332  MOV     BSFCYL,-(SP)
(9) 014454 012746 002550  MOV     #CM$SG,-(SP)
(8) 014460 013746 015330  MOV     FACNUM,-(SP)
(7) 014464 012746 007737  MOV     #FMTC$H,-(SP)
(6) 014470 012746 000010  MOV     #10,-(SP)
(3) 014474 010600      MOV     SP,RO
(4) 014476 104417      TRAP     C$PNTF
(4) 014500 062706 000022  ADD     #22,SP
1012 014504 000656      BR      11$          ;PROCESS THE NEXT ENTRY
1013
1014          ;HERE TO REPORT THE CONTENTS OF THE 'FIELD' FILE
1015 014506 004537 025076  BSRFLD: JSR     R5,RDFIELD ;GET THE FIELD BD SEC FILE
1016 014512 005002      CLR     R2           ;POINT TO THE 1ST SECTOR OF THE 'FIELD' FILE
1017 014514 012737 000020 015340  MOV     #16.,SECMAX  ;SET THE LAST USABLE SECTOR NUMBER
1018 014522 005037 015342      CLR     SECNUM       ;POINT TO THE 1ST SECTOR IN FIELD FILE
1019 014526 005037 015326      CLR     FLDNUM       ;CLEAR THE FIELD ENTRY COUNTER
1020 014532 005037 015350      CLR     PSNFG        ;CLEAR THE PRINT FLAG FOR SERIAL #
1021 014536          PRINTF  #FMT18,#BSRF
(8) 014536 012746 005057  MOV     #BSRF,-(SP)
(7) 014542 012746 007624  MOV     #FMT18,-(SP)
(6) 014546 012746 000002  MOV     #2,-(SP)
(3) 014552 010600      MOV     SP,RO
(4) 014554 104417      TRAP     C$PNTF
(4) 014556 062706 000006  ADD     #6,SP
```

```
1022 014562 004537 015622 JSR R5,BSFOK ;FIND A SECTOR TO USE IN THE FIELD AREA
1023 014566 005737 015346 TST BSFOK ;ANY ERROR DETECTED?
1024 014572 001434 BEQ 10$ ;JUMP IF ...
1025 014574 PRINTF #FMT18,#NSWSEC
(8) 014574 012746 003636 MOV #NSWSEC,-(SP)
(7) 014600 012746 007624 MOV #FMT18,-(SP)
(6) 014604 012746 000002 MOV #2,-(SP)
(3) 014610 010600 MOV SP,R0
(4) 014612 104417 TRAP C$PNTF
(4) 014614 062706 000006 ADD #6,SP
1026 014620 004537 022570 JSR R5,NEWBSF ;BUILD A NEW FILE
1027 014624 005737 002414 TST TEMPO ;DID I?
1028 014630 001402 BEQ 1$ ;NO
1029 014632 004537 016770 JSR R5,WRTBSF ;WRITE UPDATED 'FIELD' BAD SECTOR FILE
1030 ;/ON THE PACK IF REQUESTED
1031 014636 1$: PRINTF #FMT18,#HYPHEN
(8) 014636 012746 002675 MOV #HYPHEN,-(SP)
(7) 014642 012746 007624 MOV #FMT18,-(SP)
(6) 014646 012746 000002 MOV #2,-(SP)
(3) 014652 010600 MOV SP,R0
(4) 014654 104417 TRAP C$PNTF
(4) 014656 062706 000006 ADD #6,SP
1032 014662 000544 BR BSRTOT ;PRINT THE TOTALS FOUND
1033
1034 ;HERE TO PROCESS ENTRIES FROM THE FIELD FILE
1035 014664 10$: PRINTF #FMTSN,#CART,SERNM2,SERNM1
1036 014664 MOV SERNM1,-(SP)
(10) 014664 013746 002274 MOV SERNM2,-(SP)
(9) 014670 013746 002276 MOV #CART,-(SP)
(8) 014674 012746 002526 MOV #FMTSN,-(SP)
(7) 014700 012746 007707 MOV #4,-(SP)
(6) 014704 012746 000004 MOV SP,R0
(3) 014710 010600 TRAP C$PNTF
(4) 014712 104417 ADD #12,SP
(4) 014714 062706 000012 INC PSNFG ;SET THE PRINT FLAG
1037 014720 005237 015350 CLR TEMP1
1038 014724 005037 002416 11$: MOV BSFILE(R2),R3 ;GET THE CYLINDER # FROM ENTRY
1039 014730 016203 030530 TST R3 ;SEE IF ITS OK TO USE
1040 014734 005703 BPL 2$ ;OK
1041 014736 100002 JMP NOFIELD ;ERROR!
1042 014740 000137 015440 INC FLDNUM ;COUNT THIS ENTRY
1043 014744 005237 015326 CMP #126.,FLDNUM ;END OF FIELD ENTRY LIMIT?
1044 014750 022737 000176 015326 BEQ BSRTOT ;YES
1045 014756 001506 CMP #50.,FLDNUM ;TIME TO QUIT PRINTING?
1046 014760 022737 000062 015326 BNE 21$ ;NO
1047 014766 001040 TST TEMP1 ;PRINT THE ERROR MESSAGE?
1048 014770 005737 002416 BNE 21$ ;NO
1049 014774 001035 PRINTF #FMT19,#OVRMAX ;YES - TELL OPR
1050 014776 (8) 014776 012746 006251 MOV #OVRMAX,-(SP)
(7) 015002 012746 007631 MOV #FMT19,-(SP)
(6) 015006 012746 000002 MOV #2,-(SP)
(3) 015012 010600 MOV SP,R0
(4) 015014 104417 TRAP C$PNTF
(4) 015016 062706 000006 ADD #6,SP
1051 015022 PRINTF #MCRLF
```

```
(7) 015022 012746 010041      MOV      #MCRLF,-(SP)
(6) 015026 012746 000001      MOV      #1,-(SP)
(3) 015032 010600              MOV      SP,R0
(4) 015034 104417              TRAP     C$PNTF
(4) 015036 062706 000004      ADD      #4,SP
1052 015042                    GMANIL   TILLEND,TEMPO,177777,NO
(3) 015042 104443              TRAP     C$GMAN
(3) 015044 000404              BR       10003$
(4) 015046 002414              .WORD   TEMPO
(5) 015050 000120              .WORD   T$CODE
(5) 015052 006114              .WORD   TILLEND
(5) 015054 177777              .WORD   177777
(3) 015056                    10003$:
1053 015056 005737 002414      TST      TEMPO                ;QUIT?
1054 015062 001444              BEQ      BSRTOT                ;YUP
1055 015064 005237 002416      INC      TEMP1                ;SET THE PRINT FLAG
1056 015070 010337 015332      21$:  MOV      R3,BSFCYL         ;SAVE THE CYLINDER NUMBER
1057 015074 005722              TST      (R2)+                ;POINT TO HEAD & SECTOR ENTRY
1058 015076 016203 030530      MOV      BSFILE(R2),R3        ;GET IT
1059 015102 110337 015334      MOV      R3,BSFSEC           ;SAVE THE SECTOR NUMBER
1060 015106 000303              SWAB     R3                    ;PUT THE HEAD # IN LOW BYTE
1061 015110 110337 015336      MOV      R3,BSFHD            ;SAVE THE HEAD NUMBER
1062 015114 005722              TST      (R2)+                ;POINT TO THE NEXT ENTRY
1063 015116                    PRINTF   #FMTCSH,FLDNUM,#CMMSG,BSFCYL,#SMSG,BSFSEC,#HMSG,BSFHD
(14) 015116 013746 015336      MOV      BSFHD,-(SP)
(13) 015122 012746 002563      MOV      #HMSG,-(SP)
(12) 015126 013746 015334      MOV      BSFSEC,-(SP)
(11) 015132 012746 003243      MOV      #SMSG,-(SP)
(10) 015136 013746 015332      MOV      BSFCYL,-(SP)
(9) 015142 012746 002550      MOV      #CMMSG,-(SP)
(8) 015146 013746 015326      MOV      FLDNUM,-(SP)
(7) 015152 012746 007737      MOV      #FMTCSH,-(SP)
(6) 015156 012746 000010      MOV      #10,-(SP)
(3) 015162 010600              MOV      SP,R0
(4) 015164 104417              TRAP     C$PNTF
(4) 015166 062706 000022      ADD      #22,SP
1064 015172 000656              BR       11$                    ;PROCESS THE NEXT ENTRY
1065
1066
1067
1068 015174                    ;PRINT THE TOTALS FROM EACH SECTION
(9) 015174 013746 015330      BSRTOT: PRINTF #FMTTB,#TMMSG,FACNUM
(8) 015200 012746 003022      MOV      FACNUM,-(SP)
(7) 015204 012746 007724      MOV      #TMMSG,-(SP)
(6) 015210 012746 000003      MOV      #FMTTB,-(SP)
(3) 015214 010600              MOV      #3,-(SP)
(4) 015216 104417              MOV      SP,R0
(4) 015220 062706 000010      TRAP     C$PNTF
1069 015224                    ADD      #10,SP
(9) 015224 013746 015326      PRINTF   #FMTTB,#TFMSG,FLDNUM
(8) 015230 012746 002777      MOV      FLDNUM,-(SP)
(7) 015234 012746 007724      MOV      #TFMSG,-(SP)
(6) 015240 012746 000003      MOV      #FMTTB,-(SP)
(3) 015244 010600              MOV      #3,-(SP)
(4) 015246 104417              MOV      SP,R0
(4) 015250 062706 000010      TRAP     C$PNTF
(4) 015250 062706 000010      ADD      #10,SP
```

1070 015254  
(8) 015254 012746 002573  
(7) 015260 012746 007624  
(6) 015264 012746 000002  
(3) 015270 010600  
(4) 015272 104417  
(4) 015274 062706 000006  
1071 015300  
(7) 015300 012746 010041  
(6) 015304 012746 000001  
(3) 015310 010600  
(4) 015312 104417  
(4) 015314 062706 000004  
1072 015320 000137 013664  
1073  
1074  
1075  
1076 015324 000000  
1077 015326 000000  
1078 015330 000000  
1079 015332 000000  
1080 015334 000000  
1081 015336 000000  
1082 015340 000000  
1083 015342 000000  
1084 015344 000000  
1085 015346 000000  
1086 015350 000000  
1087  
1088  
1089  
1090 015352 005737 015330  
1091 015356 001014  
1092 015360 005037 015330  
1093 015364  
(8) 015364 012746 003532  
(7) 015370 012746 007624  
(6) 015374 012746 000002  
(3) 015400 010600  
(4) 015402 104417  
(4) 015404 062706 000006  
1094 015410  
(8) 015410 012746 002675  
(7) 015414 012746 007624  
(6) 015420 012746 000002  
(3) 015424 010600  
(4) 015426 104417  
(4) 015430 062706 000006  
1095 015434 000137 014506  
1096  
1097  
1098  
1099 015440 005737 015326  
1100 015444 001014  
1101 015446 005037 015326  
1102 015452

```
PRINTF #FMT18,#STARMMSG
MOV #STARMMSG,-(SP)
MOV #FMT18,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C$PNTF
ADD #6,SP
PRINTF #MCRLF
MOV #MCRLF,-(SP)
MOV #1,-(SP)
MOV SP,R0
TRAP C$PNTF
ADD #4,SP
JMP BSRPTS ;SELECT NEXT UNIT

;HERE IS THE STORAGE FOR THIS ROUTINE
NOSNUM: .WORD 0 ;HAVE SERIAL # FLAG
FLDNUM: .WORD 0 ;NUMBER OF CURRENT FIELD ENTRY
FACNUM: .WORD 0 ;NUMBER OF THE CURRENT FACTORY ENTRY
BSFCYL: .WORD 0 ;CURRENT CYLINDER FROM ENTRY IN PROCESS
BSFSEC: .WORD 0 ;CURRENT SECTOR FROM ENTRY
BSFHD: .WORD 0 ;CURRENT SURFACE (HEAD) FROM ENTRY IN PROCESS
SECMAX: .WORD 0 ;LAST USABLE SECTOR NUMBER IN SELECTED SECTION
SECNUM: .WORD 0 ;CURRENT SECTOR BEING USED TO EXTRACT ENTRIES
SECOLD: .WORD 0 ;START ADDR OF THE 'FOUND' SECTOR IN BAD SEC FILE
BSFOKF: .WORD 0 ;ERROR DETECT FLAG
PSNFG: .WORD 0 ;PRINT FLAG FOR SERIAL #

;HERE IF AT THE END OF THE FACTORY FILE
NOFACT: TST FACNUM ;WAS ANY ENTRY DETECTED?
BNE 1$ ;YES
CLR FACNUM ;CLEAR THE ENTRY COUNTER FOR FACTORY SECTORS
PRINTF #FMT18,#HWSEC
MOV #HWSEC,-(SP)
MOV #FMT18,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C$PNTF
ADD #6,SP
1$: PRINTF #FMT18,#HYPHEN
MOV #HYPHEN,-(SP)
MOV #FMT18,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C$PNTF
ADD #6,SP
JMP BSRFLD ;DO THE FIELD SECTION

;HERE IF AT THE END OF THE FIELD FILE
NOFIELD: TST FLDNUM ;ANY FIELD ENTRIES?
BNE 1$ ;YES
CLR FLDNUM ;NO - CLEAR THE ENTRY COUNTER
PRINTF #FMT18,#SWSEC
```

```

(8) 015452 012746 003610      MOV      #SWSEC,-(SP)
(7) 015456 012746 007624      MOV      #FMT18,-(SP)
(6) 015462 012746 000002      MOV      #2,-(SP)
(3) 015466 010600      MOV      SP,R0
(4) 015470 104417      TRAP     C$PNTF
(4) 015472 062706 000006      ADD      #6,SP
1103 015476      1$: PRINTF #FMT18,#HYPHEN
(8) 015476 012746 002675      MOV      #HYPHEN,-(SP)
(7) 015502 012746 007624      MOV      #FMT18,-(SP)
(6) 015506 012746 000002      MOV      #2,-(SP)
(3) 015512 010600      MOV      SP,R0
(4) 015514 104417      TRAP     C$PNTF
(4) 015516 062706 000006      ADD      #6,SP
1104 015522 000137 015174      JMP      BSRTOT      ;DO THE TOTALS
1105
1106      ;HERE IF NO SERIAL NUMBER OR 2 0'S NOT DETECTED IN 1ST 4 WORDS OF SECTOR
1107
1108 015526 062737 000004 015342 NOMSEC: ADD      #4,SECNUM      ;UPDATE THE SECTOR NUMBER TO ACCESS NEXT BSF COPY
1109 015534 023737 015342 015340      CMP      SECNUM,SECMAX      ;AT THE END OF SECTION?
1110 015542 101416      BLOS     1$      ;BRANCH IF OK
1111 015544      PRINTF #FMT18,#BSEND
(8) 015544 012746 003255      MOV      #BSEND,-(SP)
(7) 015550 012746 007624      MOV      #FMT18,-(SP)
(6) 015554 012746 000002      MOV      #2,-(SP)
(3) 015560 010600      MOV      SP,R0
(4) 015562 104417      TRAP     C$PNTF
(4) 015564 062706 000006      ADD      #6,SP
1112 015570 052737 177777 015346      BIS      #177777,BSFOKF      ;SET THE ERROR FLAG
1113 015576 000465      BR      BSFOKX      ;EXIT THE SECTOR FIND ROUTINE WITH ERROR SET
1114 015600 005737 002476      1$: TST     BSFFLG      ;IS BAD SECTOR FILE WRITTEN BY FIELD?
1115 015604 001003      BNE     2$      ;YES - BRANCH IF FIELD BAD SECTOR FILE
1116 015606 004537 025032      JSR     R5,RDFACT      ;ELSE, READ FACTORY BAD SECTOR FILE
1117 015612 000403      BR      BSFOK      ;CHECK IF THIS SECTOR IS O.K.
1118 015614 004537 025076      2$: JSR     R5,RDFIELD      ;READ FIELD BAD SECTOR FILE
1119 015620 000400      BR      BSFOK      ;CHECK IF THIS SECTOR IS O.K.

```



```
1121 ;HERE IS THE SECTOR FIND ROUTINE FOR THE BAD SECTOR FILE
1122
1123 015622 005037 015346 BSFOK: CLR BSFOKF ;CLEAR THE ERROR FLAG
1124 015626 012737 030530 015344 MOV #BSFILE,SECOLD ;GET BASIC ADDRESS
1125 015634 060237 015344 ADD R2,SECOLD ;ADD IN THE OFFSET
1126 015640 005762 030530 TST BSFILE(R2) ;SEE IF ANY SERIAL NUMBER
1127 015644 100730 BMI NOMSEC ;IF -1 JUMP TO ERROR SERVICE
1128 015646 001006 BNE 1$ ;IF >0 THEN HAVE A NUMBER
1129 015650 005762 030532 TST BSFILE+2(R2) ;ANY SER. NUM IN WORD 2?
1130 015654 001003 BNE 1$ ;BR IF OK
1131 015656 052737 177777 015324 BIS #177777,NOSNUM ;SET THE NO SERIAL NUMBER FLAG
1132 015664 022762 177777 031126 1$: CMP #-1,BSFILE+254.(R2) ;END OF SECTOR OK ALSO?
1133 015672 001315 BNE NOMSEC ;NO - NO MATCH HERE!
1134 015674 022762 177777 031130 CMP #-1,BSFILE+256.(R2) ;END OK HERE TOO?
1135 015702 001311 BNE NOMSEC ;NO - DON'T HAVE A BAD SECT FILE YET!
1136 015704 016237 030530 002274 MOV BSFILE(R2),SERNM1 ;SAVE LOW ORDER OF SERIAL #
1137 015712 005722 TST (R2)+ ;POINT TO HIGH ORDER OF SERIAL #
1138 015714 005762 030530 TST BSFILE(R2) ;SEE IF LEGAL
1139 015720 100702 BMI NOMSEC ;NO - ERROR
1140 015722 016237 030530 002276 MOV BSFILE(R2),SERNM2 ;SAVE HIGH ORDER OF SERIAL #
1141 015730 005722 TST (R2)+ ;POINT TO A BLANK ENTRY
1142 015732 005762 030530 TST BSFILE(R2) ;SEE IF LEGAL (=0)
1143 015736 001273 BNE NOMSEC ;NO - ERROR
1144 015740 005722 TST (R2)+ ;POINT TO LAST CHECK ENTRY SPOT
1145 015742 005762 030530 TST BSFILE(R2) ;SEE IF LEGAL (=0)
1146 015746 001267 BNE NOMSEC ;NO - ERROR
1147 015750 005722 TST (R2)+ ;POINT TO 1ST VALID ENTRY IN SECTOR
1148 015752 000205 BSFOKX: RTS R5 ;EXIT - R2 POINTS TO THE OFFSET VALUE
1149 ;AND SECNUM = CURRENT SECTOR IN FILE
```

```

1151 015754          STARS
(2)
1152
1153 015754          :*****
(2)                :HERE IS THE ROUTINE TO ADD AN ENTRY INTO THE 'FIELD' BAD SECTOR FILE
1154                STARS
(2)                :*****
1155 015754          BSADD: PRINTF  #FMT19,#ABSMSG
(8) 015754 012746 005164      MOV      #ABSMSG,-(SP)
(7) 015760 012746 007631      MOV      #FMT19,-(SP)
(6) 015764 012746 000002      MOV      #2,-(SP)
(3) 015770 010600      MOV      SP,R0
(4) 015772 104417      TRAP    C$PNTF
(4) 015774 062706 000006      ADD     #6,SP
1156 016000 004537 012370      JSR     R5,FIRST      ;SELECT 1ST UNIT
1157 016004 000137 013546      JMP     NXTCMD        ;NONE AVAIL.!
1158 016010 000404      BR      BSADDL
1159 016012 004537 012400      BSADDS: JSR     R5,SELDRV ;SELECT THE NEXT UNIT
1160 016016 000137 013546      JMP     NXTCMD        ;ALL DONE
1161 016022 004537 025246      PSADDL: JSR     R5,LOADED ;DRV READY?
1162 016026 005737 002414      TST     TEMPO        ;WELL?
1163 016032 001403      BEQ     1$           ;YES
1164 016034 004537 012642      JSR     R5,DRNRDY
1165 016040 000764      BR      BSADDS        ;SELECT THE NEXT UNIT
1166
1167 016042 004537 012546      1$:     JSR     R5,DRVID  ;TELL OPR WHAT DRIVE
1168 016046 004537 023656      JSR     R5,RDBDSC    ;GET A FRESH COPY OF THE BAD SECTOR FILE
1169 016052 005737 002464      TST     ACCESS      ;ALLOWED TO DO IT?
1170 016056 001414      BEQ     BSATD        ;YES
1171 016060          PRINTF  #FMT18,#DENIED
(8) 016060 012746 004125      MOV     #DENIED,-(SP)
(7) 016064 012746 007624      MOV     #FMT18,-(SP)
(6) 016070 012746 000002      MOV     #2,-(SP)
(3) 016074 010600      MOV     SP,R0
(4) 016076 104417      TRAP    C$PNTF
(4) 016100 062706 000006      ADD     #6,SP
1172 016104 000137 013546      JMP     NXTCMD        ;TELL OPR NOT ALLOWED & EXIT
1173
1174 016110          BSATD:  GMANIL  THISDRV,TEMPO,1,NO
(3) 016110 104443      TRAP    C$GMAN
(3) 016112 000404      BR      10004$
(4) 016114 002414      .WORD  TEMPO
(5) 016116 000120      .WORD  T$CODE
(5) 016120 006750      .WORD  THISDRV
(5) 016122 000001      .WORD  1
(3) 016124
1175 016124 005737 002414      10004$: TST     TEMPO
1176 016130 001730      BEQ     BSADDS        ;BRANCH IF NOT TO USE THIS DRIVE
1177
1178 016132 005737 015324      BSASN:  TST     NOSNUM  ;SEE IF NEED A SERIAL NUMBER
1179 016136 001470      BEQ     GETCYL      ;JUMP IF HAVE A NUMBER
1180 016140          PRINTF  #MCRLF
(7) 016140 012746 010041      MOV     #MCRLF,-(SP)
(6) 016144 012746 000001      MOV     #1,-(SP)
(3) 016150 010600      MOV     SP,R0
(4) 016152 104417      TRAP    C$PNTF
(4) 016154 062706 000004      ADD     #4,SP

```

1181 016160  
 (3) 016160 104443  
 (3) 016162 000404  
 (4) 016164 002414  
 (5) 016166 000120  
 (5) 016170 005352  
 (5) 016172 000001  
 (3) 016174  
 1182 016174 005737 002414  
 1183 016200 001447  
 1184 016202  
 (3) 016202 104443  
 (3) 016204 000406  
 (4) 016206 002456  
 (5) 016210 000022  
 (5) 016212 005415  
 (5) 016214 077777  
 (5) 016216 000001  
 (5) 016220 077777  
 (3) 016222  
 1185 016222  
 (3) 016222 104443  
 (3) 016224 000406  
 (4) 016226 002460  
 (5) 016230 000022  
 (5) 016232 005471  
 (5) 016234 077777  
 (5) 016236 000000  
 (5) 016240 077777  
 (3) 016242  
 1186 016242  
 (10) 016242 013746 002456  
 (9) 016246 013746 002460  
 (8) 016252 012746 002526  
 (7) 016256 012746 007707  
 (6) 016262 012746 000004  
 (3) 016266 010600  
 (4) 016270 104417  
 (4) 016272 062706 000012  
 1187 016276  
 (3) 016276 104443  
 (3) 016300 000404  
 (4) 016302 002414  
 (5) 016304 000120  
 (5) 016306 006007  
 (5) 016310 177777  
 (3) 016312  
 1188 016312 005737 002414  
 1189 016316 001720  
 1190  
 1191

GETSN: GMANIL ABSSER,TEMPO,1,NO  
 TRAP C\$GMAN  
 BR 10005\$  
 .WORD TEMPO  
 .WORD T\$CODE  
 .WORD ABSSER  
 .WORD 1  
 10005\$:  
 TST TEMPO ;SEE IF YES (=1)  
 BEQ GETCYL ;BRANCH IF NO  
 GMANID ABSSNL,SN1,0,77777,1,77777,NO  
 TRAP C\$GMAN  
 BR 10006\$  
 .WORD SN1  
 .WORD T\$CODE  
 .WORD ABSSNL  
 .WORD 77777  
 .WORD T\$LOLIM  
 .WORD T\$HILIM  
 10006\$:  
 GMANID ABSSNH,SN2,0,77777,0,77777,NO  
 TRAP C\$GMAN  
 BR 10007\$  
 .WORD SN2  
 .WORD T\$CODE  
 .WORD ABSSNH  
 .WORD 77777  
 .WORD T\$LOLIM  
 .WORD T\$HILIM  
 10007\$:  
 PRINTF #FMTSN,#CART,SN2,SN1  
 MOV SN1,-(SP)  
 MOV SN2,-(SP)  
 MOV #CART,-(SP)  
 MOV #FMTSN,-(SP)  
 MOV #4,-(SP)  
 MOV SP,RO  
 TRAP C\$PNTF  
 ADD #12,SP  
 GMANIL VALSN,TEMPO,177777,NO  
 TRAP C\$GMAN  
 BR 10010\$  
 .WORD TEMPO  
 .WORD T\$CODE  
 .WORD VALSN  
 .WORD 177777  
 10010\$:  
 TST TEMPO ;JMP IF NO  
 BEQ GETSN ;NO VALID SERIAL NUMBER - ASK AGAIN

1193  
1194 016320  
(3) 016320 104443  
(3) 016322 000406  
(4) 016324 015332  
(5) 016326 000042  
(5) 016330 005254  
(5) 016332 000777  
(5) 016334 000000  
(5) 016336 000777  
(3) 016340  
1195  
1196  
1197 016340 022737 000001 0023'6  
1198 016346 001017  
1199 016350 022737 000377 015332  
1200 016356 103013  
1201 016360  
(8) 016360 012746 005756  
(7) 016364 012746 007631  
(6) 016370 012746 000002  
(3) 016374 010600  
(4) 016376 104417  
(4) 016400 062706 000006  
1202 016404 000745  
1203 016406  
(3) 016406 104443  
(3) 016410 000406  
(4) 016412 015334  
(5) 016414 000042  
(5) 016416 005304  
(5) 016420 000077  
(5) 016422 000000  
(5) 016424 000047  
(3) 016426  
1204 016426  
(3) 016426 104443  
(3) 016430 000406  
(4) 016432 015336  
(5) 016434 000042  
(5) 016436 005331  
(5) 016440 000003  
(5) 016442 000000  
(5) 016444 000001  
(3) 016446  
1205  
1206  
1207  
1208 016446 013700 015332  
1209 016452 000300  
1210 016454 000241  
1211 016456 006000  
1212 016460 103002  
1213 016462 052700 100000  
1214 016466 053700 015334  
1215 016472 005737 015336

GETCYL: GMANID ABS CYL,BSF CYL,D,777,0,511.,NO  
TRAP C\$GMAN  
BR 10011\$  
.WORD BSF CYL  
.WORD T\$CODE  
.WORD ABS CYL  
.WORD 777  
.WORD T\$LOLIM  
.WORD T\$HILIM

10011\$:  
:CHECK FOR A VALID RLO1 CYL IF DRIVE=RLO1

CMP #1,TDR :RLO1=1  
BNE GETSEC :SKIP CHECK IF RLO2  
CMP #255.,BSF CYL :VALID RLO1 CYLINDER?  
BHIS GETSEC :YES  
PRINTF #FMT19,#RLO1CLM :NO - TELL OPR  
MOV #RLO1CLM,-(SP)  
MOV #FMT19,-(SP)  
MOV #2,-(SP)  
MOV SP,RO  
TRAP C\$PNTF  
ADD #6,SP

BR GETCYL :GET CYL AGAIN  
GETSEC: GMANID ABS SEC,BSF SEC,D,77,0,39.,NO  
TRAP C\$GMAN  
BR 10012\$  
.WORD BSF SEC  
.WORD T\$CODE  
.WORD ABS SEC  
.WORD 77  
.WORD T\$LOLIM  
.WORD T\$HILIM

10012\$:  
GMANID ABS HD,BSF HD,D,3,0,1,NO  
TRAP C\$GMAN  
BR 10013\$  
.WORD BSF HD  
.WORD T\$CODE  
.WORD ABS HD  
.WORD 3  
.WORD T\$LOLIM  
.WORD T\$HILIM

10013\$:  
:CHECK TO SEE IF THE NEW ENTRY ALREADY EXISTS IN THE FILE

MOV BSF CYL,RO :GET THE CYL TYPED  
SWAB RO  
CLC :CLEAR THE 'C' BIT  
ROR RO  
BCC 1\$ :BR IF DON'T NEED THE EXTRA BIT  
BIS #BIT15,RO :ADD HIGH ORDER BIT IN CYL #  
1\$: BIS BSF SEC,RO :ADD IN THE SECTOR NUMBER  
TST BSF HD :ON HEAD 0??

1216	016476	001402			BEQ	ACKENT		:BR IF HEAD 0
1217	016500	052700	000100		BIS	#100,R0		:MAKE IT HEAD #1
1218	016504	010037	002410		ACKENT: MOV	R0,CHKSEC		:SAVE FOR THE CHECK
1219	016510	004537	027340		JSR	R5,CKBDSC		:CHECK TO SEE IF ALREADY IN BAD SECT FILE
1220	016514	005737	002406		TST	HDRFND		:HEADER IN FILE?
1221	016520	001414			BEQ	1\$		:BR IF NOT IN FILE
1222	016522				PRINTF	#FMT18,#EXISTS		:TELL OPR ENTRY IN FILE NOW
(8)	016522	012746	004171		MOV	#EXISTS,-(SP)		
(7)	016526	012746	007624		MOV	#FMT18,-(SP)		
(6)	016532	012746	000002		MOV	#2,-(SP)		
(3)	016536	010600			MOV	SP,R0		
(4)	016540	104417			TRAP	(SPNTF		
(4)	016542	062706	000006		ADD	#6,SP		
1223	016546	000137	016012		JMP	BSADDS		:SELECT THE NEXT UNIT
1224								
1225								
1226								
1227								
1228								
1229	016552				1\$: PRINTF	#FMT18,#NEWENT		:TELL OPR IT IS A NEW ENTRY
(8)	016552	012746	006200		MOV	#NEWENT,-(SP)		
(7)	016556	012746	007624		MOV	#FMT18,-(SP)		
(6)	016562	012746	000002		MOV	#2,-(SP)		
(3)	016566	010600			MOV	SP,R0		
(4)	016570	104417			TRAP	(SPNTF		
(4)	016572	062706	000006		ADD	#6,SP		
1230	016576	005037	015342		CLR	SECNUM		:START THE SEARCH AT SECTOR 20.
1231	016602	005002			CLR	R2		:POINT TO THE STARTING AREA IN BSFILE
1232	016604	012737	000020	015340	MOV	#16.,SECMAX		:THIS IS THE LAST AVAIL. SECTOR PAIR
1233	016612	005037	015326		CLR	FLDNUM		:START AT ENTRY #1
1234	016616	004537	015622		JSR	R5,BSFOK		:FIND A 'FIELD' SECTOR AREA
1235	016622	005737	015346		TST	BSFOKF		:ON A SECTOR?
1236	016626	001421			BEQ	2\$		:YES
1237	016630				PRINTF	#FMT18,#NSWSEC		:NO - TELL OPR
(8)	016630	012746	003636		MOV	#NSWSEC,-(SP)		
(7)	016634	012746	007624		MOV	#FMT18,-(SP)		
(6)	016640	012746	000002		MOV	#2,-(SP)		
(3)	016644	010600			MOV	SP,R0		
(4)	016646	104417			TRAP	(SPNTF		
(4)	016650	062706	000006		ADD	#6,SP		
1238	016654	004537	022570		JSR	R5,NEWBSF		:ASK OPR IF TIME TO MAKE A 'FIELD' BSF
1239	016660	005737	002414		TST	TEMPO		:WAS A FILE BUILT?
1240	016664	001002			BNE	2\$		:YES - CONTINUE
1241	016666	000137	016012		JMP	BSADDS		:SELECT THE NEXT UNIT
1242								
1243	016672	005737	015324		2\$: TST	NOSNUM		:PACK HAVE A SERIAL # ??
1244	016676	001406			BEQ	21\$		:YUP
1245	016700	013777	002456	176436	MOV	SN1,@SECOLD		:NO - SAVE LOW 5 #
1246	016706	013777	002460	176432	MOV	SN2,@SECOLD+2		:SAVE HIGH 5 #
1247	016714	005237	015326		21\$: INC	FLDNUM		:COUNT THIS ENTRY TO BE TESTED
1248	016720	005762	030530		TST	BSFILE(R2)		:SEE IF A FREE SLOT
1249	016724	100403			BMI	3\$		:I FOUND IT...
1250	016726	062702	000004		ADD	#4,R2		:POINT TO THE NEXT ENTRY
1251	016732	000757			BR	2\$		:AND TRY THE NEXT SLOT
1252								
1253	016734	013762	015332	030530	3\$: MOV	BSFCYL,BSFILE(R2)		:INSERT THE CYLINDER NUMBER

```

1254 016742 013703 015336      MOV    BSFHD,R3      ;GET THE SELECTED HEAD
1255 016746 000303              SWAB   R3            ;SWAP BYTES TO POSITION THE HD BIT
1256 016750 063703 015334      ADD    BSFSEC,R3     ;R3 NOW HAS COMPLETE HD & SEC ENTRY
1257 016754 010362 030532      MOV    R3,BSFILE+2(R2) ;INSERT 2ND HALF OF ENTRY
1258
1259                               ;INSERT THE ENTRY INTO REST OF THE 'FIELD' FILE
1260
1261 016760 004537 016770      48:   JSR    R5,WRTBSF ;WRITE UPDATED 'FIELD' BAD SECTOR FILE
1262                               ;/ON THE PACK IF REQUESTED
1263
1264 016764 000137 016012      JMP    BSADDS        ;SELECT THE NEXT UNIT

```

1266 016770  
(2)  
1267  
1268  
1269  
1270 016770  
(2)  
1271  
1272 016770 010146  
1273 016772  
(7) 016772 012746 010041  
(6) 016776 012746 000001  
(3) 017002 010600  
(4) 017004 104417  
(4) 017006 062706 000004  
1274 017012  
(3) 017012 062706  
(3) 017014 000004  
(4) 017016 002414  
(5) 017020 000120  
(5) 017022 005546  
(5) 017024 177777  
(3) 017026  
1275 017026 005737 002414  
1276 017032 001545  
1277 017034 004537 025246  
1278 017040 005737 002414  
1279 017044 001413  
1280 017046  
(8) 017046 012746 006505  
(7) 017052 012746 007624  
(6) 017056 012746 000002  
(3) 017062 010600  
(4) 017064 104417  
(4) 017066 062706 000006  
1281 017072 000525  
1282 017074 004537 025422  
1283 017100 032701 020000  
1284 017104 001413  
1285 017106  
(8) 017106 012746 003460  
(7) 017112 012746 007624  
(6) 017116 012746 000002  
(3) 017122 010600  
(4) 017124 104417  
(4) 017126 062706 000006  
1286 017132 000505  
1287 017134 012737 000005 002500  
1288  
1289 017142 012737 177000 002246  
1290 017150 012737 077724 002244  
1291 017156 022737 000001 002316  
1292 017164 001403  
1293 017166 012737 177724 002244  
1294 017174 012737 000012 002260  
1295 017202 005737 002300

STARS  
:\*\*\*\*\*  
:HERE IS WHERE THE 'FIELD' FILE IS WRITTEN ON THE PACK.  
:THE OPERATOR IS ASKED IF IT IS TIME TO UPDATE THE PACK...IF NOT, THEN  
:THIS CODE IS ABORTED.  
STARS  
:\*\*\*\*\*  
WRTBSF: MOV R1,-(SP) ;SAVE R1  
PRINTF #MCRLF  
MOV #MCRLF,-(SP)  
MOV #1,-(SP)  
MOV SP,R0  
TRAP CSPNTF  
ADD #4,SP  
GMANIL DOWRT,TEMPO,177777,NO  
TRAP CSGMAN  
BR 10014\$  
.WORD TEMPO  
.WORD TSCODE  
.WORD DOWRT  
.WORD 177777  
10014\$: TST TEMPO ;YES? (=1)  
BEQ 3\$ ;EXIT IF 'NO'  
JSR R5,LOADED ;READY?  
TST TEMPO ;WELL?  
BEQ 11\$ ;YES  
PRINTF #FMT18,#NOTRDY ;NO  
MOV #NOTRDY,-(SP)  
MOV #FMT18,-(SP)  
MOV #2,-(SP)  
MOV SP,R0  
TRAP CSPNTF  
ADD #6,SP  
BR 3\$  
11\$: JSR R5,GETDST  
BIT #WL,R1 ;DRIVE WRITE LOCKED?  
BEQ 5\$ ;NO  
PRINTF #FMT18,#WRTLCK ;YES  
MOV #WRTLCK,-(SP)  
MOV #FMT18,-(SP)  
MOV #2,-(SP)  
MOV SP,R0  
TRAP CSPNTF  
ADD #6,SP  
BR 3\$  
5\$: MOV #5,CPYCNT ;INITIALIZE COPY COUNT FOR DUPLICATION OF  
;THE 'FIELD' BAD SECTOR FILE ON THE PACK  
12\$: MOV #-512,,BMP ;SET UP THE WORD COUNT  
MOV #77724,BDA ;START THE WRITE AT SECTOR 20. (RL01)  
CMP #1,TDR ;RL02?  
BEQ 1\$ ;JUMP IF RL01  
MOV #177724,BDA ;START AT SECTOR 20. FOR RL02  
1\$: MOV #WR,'E,FUNC ;LOAD THE FUNCTION  
TST NEWFAC ;MAKING A DUMMY 'FACTORY' FILE?

```

1296 017206 001405          BEQ      13$          ;NO
1297 017210 042737 000077 002244    BIC      #77,BDA      ;YES - START AT SECTOR 00
1298 017216 005037 002300          CLR      NEWFAC      ;CLEAR THE FLAG ALSO
1299 017222 005237 002320          13$:    INC      WRIPG      ;SET THE WRITE IN PROGRESS FLAG
1300 017226 004537 023534          JSR      R5,LDFUNC   ;DO THE WRITE OF THE UPDATED 'FIELD'
1301                                ;/BAD SECTOR FILE
1302 017232 004537 025310          JSR      R5,WTRDY   ;WAIT FOR READY
1303 017236 005337 002500          DEC      CPYCNT     ;DECREMENT COPY COUNT
1304 017242 005777 163002          TST      @DCS       ;WAS THE TRANSFER GOOD?
1305 017246 100407          BMI      4$        ;BRANCH TO REPORT 'CANNOT UPDATE BAD SECTOR'
1306                                ;/FILE ON PACK''
1307 017250 005737 002500          TST      CPYCNT     ;IS ENTIRE 'FIELD' BAD SECTOR FILE
1308                                ;/WRITTEN ON THE PACK?
1309 017254 001420          BEQ      2$        ;BRANCH TO PRINT TIME AND REPORT
1310                                ;/OPERATION IS COMPLETED
1311 017256 062737 000004 002244    ADD      #4,BDA      ;ELSE, ADD OFFSET TO DISK ADDRESS REGISTER TO
1312                                ;/ACCESS THE NEXT SECTOR GROUP IN WHICH TO
1313                                ;/DUPLICATE THE BAD SECTOR FILE
1314 017264 000726          BR       12$       ;BRANCH TO REPEAT WRITE OPERATION
1315
1316                                ;HERE IF AN ERROR DETECTED WHILE UPDATING THE MEDIA
1317
1318 017266          4$:    PRINTF #FMT18,#BADWRT          ;REPORT 'CANNOT UPDATE BAD SECTOR FILE ON PACK''
    (8) 017266 012746 005610    MOV      #BADWRT,-(SP)
    (7) 017272 012746 007624    MOV      #FMT18,-(SP)
    (6) 017276 012746 000002    MOV      #2,-(SP)
    (3) 017302 010600          MOV      SP,R0
    (4) 017304 104417          TRAP    C$PNTF
    (4) 017306 062706 000006    ADD      #6,SP
1319 017312 000137 013546          JMP      NXTCMD     ;BACK TO THE QUERY LOOP
1320 017316 004537 012502          2$:    JSR      R5,PTIME ;PRINT THE SYS RUN TIME
1321 017322          PRINTF #FMT18,#MDONE ;TELL OPR - DONE
    (8) 017322 012746 003720    MOV      #MDONE,-(SP)
    (7) 017326 012746 007624    MOV      #FMT18,-(SP)
    (6) 017332 012746 000002    MOV      #2,-(SP)
    (3) 017336 010600          MOV      SP,R0
    (4) 017340 104417          TRAP    C$PNTF
    (4) 017342 062706 000006    ADD      #6,SP
1322 017346 012601          3$:    MOV      (SP)+,R1 ;RESET R1
1323 017350 000205          RTS      R5        ;EXIT ROUTINE
    
```



1325 017352  
(2)  
1326  
1327  
1328 017352  
(2)  
1329  
1330 017352  
(8) 017352 012746 005656  
(7) 017356 012746 007631  
(6) 017362 012746 000002  
(3) 017366 010600  
(4) 017370 104417  
(4) 0173 2 062706 000006  
1331 017376 004537 012370  
1332 017402 000137 013546  
1333 017406 000404  
1334 017410 004537 012400  
1335 017414 000137 013546  
1336 017420 004537 025246  
1337 017424 005737 002414  
1338 017430 001403  
1339 017432 004537 012642  
1340 017436 000764  
1341  
1342 017440 004537 012546  
1343 017444 004537 023656  
1344 017450 005737 002464  
1345 017454 001414  
1346 017456  
(8) 017456 012746 004125  
(7) 017462 012746 007624  
(6) 017466 012746 000002  
(3) 017472 010600  
(4) 017474 104417  
(4) 017476 062706 000006  
1347 017502 000137 013546  
1348  
1349 017506  
(3) 017506 104443  
(3) 017510 000404  
(4) 017512 002414  
(5) 017514 000120  
(5) 017516 006750  
(5) 017520 000001  
(3) 017522  
1350 017522 005737 002414  
1351 017526 001730  
1352  
1353 017530  
(7) 017530 012746 010041  
(6) 017534 012746 000001  
(3) 017540 010600  
(4) 017542 104417  
(4) 017544 062706 000004  
1354 017550

```
STARS
:*****
:HERE IS THE CODE TO SERVICE REMOVING AN ENTRY FROM THE 'FIELD' BAD
:SECTOR FILE
STARS
:*****

BSDEL: PRINTF #FMT19,#DELMSG ;TELL OPR ABOUT TO DELETE...
      MOV #DELMSG,-(SP)
      MOV #FMT19,-(SP)
      MOV #2,-(SP)
      MOV SP,R0
      TRAP C$PNTF
      ADD #6,SP
      JSR R5,FIRST ;SELECT THE 1ST UNIT
      JMP NXTCMD ;NONE AVAIL.!
BSDELS: JSR R5,SELDRV ;SELECT NEXT UNIT
        JMP NXTCMD ;ALL DONE
BSDELL: JSR R5,LOADED ;READY?
        TST TEMPO
        BEQ 1$ ;YES
        JSR R5,DRNRDY
        BR BSDELS ;SELECT THE NEXT UNIT

1$: JSR R5,DR 'D ;TELL OPR WHAT DRIVE SELECTED
    JSR R5,RL 'SC ;GET A FRESH COPY OF THE BAD SEC FILE
    TST ACCESS ;AL OWED TO PROCEED?
    BEQ BSDELTD ;YES
    PRINTF #FMT18,#DENIED ;NO - TELL OPR
    MOV #DENIED,-(SP)
    MOV #FMT18,-(SP)
    MOV #2,-(SP)
    MOV SP,R0
    TRAP C$PNTF
    ADD #6,SP
    JMP NXTCMD ;BACK TO THE QUERY LOOP

BSDELTD: GMANIL THISDRV,TEMPO,1,NO
        TRAP C$GMAN
        BR 10015$
        .WORD TEMPO
        .WORD T$CODE
        .WORD THISDRV
        .WORD 1
10015$: TST TEMPO
        BEQ BSDELS ;RE-SELECT IF NOT THIS DRIVE

BSDEL1: PRINTF #MCRLF
        MOV #MCRLF,-(SP)
        MOV #1,-(SP)
        MOV SP,R0
        TRAP C$PNTF
        ADD #4,SP
        GMANID DELCYL,BSFCYL,D,777,0,511.,NO
```

(3)	017550	104443		TRAP	CSGMAN	
(3)	017552	000406		BR	10016\$	
(4)	017554	015332		.WORD	BSFCYL	
(5)	017556	000042		.WORD	T\$CODE	
(5)	017560	005254		.WORD	DEL CYL	
(5)	017562	000777		.WORD	777	
(5)	017564	000000		.WORD	T\$LOLIM	
(5)	017566	000777		.WORD	T\$HILIM	
(3)	017570					
1355	017570			10016\$:		
(3)	017570	104443		GMANID	DELSEC,BSFSEC,D,77,0,39.,NO	
(3)	017572	000406		TRAP	CSGMAN	
(4)	017574	015334		BR	10017\$	
(5)	017576	000042		.WORD	BSFSEC	
(5)	017600	005304		.WORD	T\$CODE	
(5)	017602	000077		.WORD	DELSEC	
(5)	017604	000000		.WORD	77	
(5)	017606	000047		.WORD	T\$LOLIM	
(3)	017610			.WORD	T\$HILIM	
1356	017610			10017\$:		
(3)	017610	104443		GMANID	DELHD,BSFHD,D,3,0,1,NO	
(3)	017612	000406		TRAP	CSGMAN	
(4)	017614	015336		BR	10020\$	
(5)	017616	000042		.WORD	BSFHD	
(5)	017620	005331		.WORD	T\$CODE	
(5)	017622	000003		.WORD	DELHD	
(5)	017624	000000		.WORD	3	
(5)	017626	000001		.WORD	T\$LOLIM	
(3)	017630			.WORD	T\$HILIM	
1357	017630	013700	015332	10020\$:		
1358	017634	000300		MOV	BSFCYL,RO	:COPY THE CYL TO REMOVE
1359	017636	000241		SWAB	RO	:PUT IT IN HIGH BYTE
1360	017640	006000		CLC		
1361	017642	103002		ROR	RO	
1362	017644	052700	100000	BCC	1\$	:BR IF DON'T WANT ANOTHER BIT
1363	017650	053700	015334	BIS	#BIT15,RO	:ADD IN HIGH ORDER CYL BIT
1364	017654	005737	015336	BIS	BSFSEC,RO	:ADD IN THE SECTOR NUMBER
1365	017660	001402		TST	BSFHD	:ON HEAD 0??
1366	017662	052700	000100	BEQ	2\$	:YES
1367	017666	010037	002410	BIS	#100,RO	:NO - POINT TO HEAD 1
1368	017672	004537	027340	MOV	RO,CHKSEC	:SAVE THE COMPACTED DISK ADDRESS
1369	017676	005737	002406	JSR	R5,CKBDS	:CHECK TO SEE IF ENTRY EXISTS
1370	017702	001014		TST	HDRFND	:FOUND?
1371	017704			BNE	10\$	:YES
(8)	017704	012746	005725	PRINTF	#FMT18,#NOENTRY	:NO
(7)	017710	012746	007624	MOV	#NOENTRY,-(SP)	
(6)	017714	012746	000002	MOV	#FMT18,-(SP)	
(3)	017720	010600		MOV	#2,-(SP)	
(4)	017722	104417		MOV	SP,RO	
(4)	017724	062706	000006	TRAP	C\$PNTF	
1372	017730	000137	017410	ADD	#6,SP	
1373				JMP	BSDELS	:SELECT THE NEXT UNIT
1374	017734	004537	025076	10\$:		
1375	017740	005002		JSR	R5,RDFIELD	:GET THE FIELD BAD SEC FILE
1376	017742	005037	015342	CLR	R2	
1377	017746	005037	015326	CLR	SECNUM	
				CLR	FLDNUM	

```

1378 017752 012737 000020 015340      MOV      #16.,SECMAX
1379 017760 004537 015622      JSR      R5,BSFOK          ;POINT TO A WORK AREA
1380 017764 005737 015346      TST      BSFOK           ;POINTING TO A VALID AREA?
1381 017770 001421      BEQ      11$             ;YES - PROCEED
1382 017772      PRINTF  #FMT18,#NSWSEC    ;TELL OPR THAT ERROR EXISTS
(8) 017772 012746 003636      MOV      #NSWSEC,-(SP)
(7) 017776 012746 007624      MOV      #FMT18,-(SP)
(6) 020002 012746 000002      MOV      #2,-(SP)
(3) 020006 010600      MOV      SP,R0
(4) 020010 104417      TRAP    C$PNTF
(4) 020012 062706 000006      ADD      #6,SP
1383 020016 004537 022570      JSR      R5,NEWBSF       ;SEE IF OPR WANTS TO MAKE A FILE
1384 020022 005737 002414      TST      TEMPO           ;PROCEED IF A 'FIELD' FILE BUILT
1385 020026 001002      BNE      11$             ;BR - FILE WAS BUILT
1386 020030 000137 017410      JMP      BSDELS          ;SELECT THE NEXT UNIT
1387
1388 020034 023762 015332 030530 11$:  CMP      BSFCYL,BSFILE(R2) ;AT CORRECT ENTRY?
1389 020042 001027      BNE      20$             ;NOPE! UPDATE POINTER
1390 020044 013737 015336 002416      MOV      BSFHD,TEMP1    ;GET THE HEAD SELECTED
1391 020052 000337 002416      SWAB    TEMP1           ;PUT IT IN HIGH BYTE
1392 020056 053737 015334 002416      BIS      3SFSEC,TEMP1   ;ADD IN THE SECTOR BITS
1393 020064 023762 002416 030532      CMP      TEMP1,BSFILE+2(R2) ;CORRECT SECTOR TOO?
1394 020072 001013      BNE      20$             ;NO - UPDATE POINTER
1395
1396      ;HAVE THE ENTRY SLOT NOW ... KILL THE ENTRY & MOVE ALL OTHERS UP 1
1397
1398 020074 016262 030534 030530 12$:  MOV      BSFILE+4(R2),BSFILE(R2) ;MOVE NEXT CYL ENTRY UP
1399 020102 016262 030536 030532      MOV      BSFILE+6(R2),BSFILE+2(R2) ;MOVE NEXT SECT ENTRY UP
1400 020110 005762 030534      TST      BSFILE+4(R2)   ;END OF ENTRIES YET?
1401 020114 100422      BMI      3$             ;YUP - EXIT
1402 020116 022222      CMP      (R2)+,(R2)+    ;POINT TO THE NEXT SLOT OF ENTRIES
1403 020120 000765      BR       12$           ;AND DO AGAIN
1404
1405      ;UPDATE THE ENTRY SLOT POINTER
1406
1407 020122 022222 20$:  CMP      (R2)+,(R2)+    ;UPDATE POINTER BY 2 LOCATIONS
1408 020124 005762 030530      TST      BSFILE(R2)    ;END OF ENTRIES?
1409 020130 100341      BPL      11$           ;NO - LOOK AT THIS SLOT
1410
1411      ;HERE IF NO 'FIELD' ENTRY DETECTED ON THE PACK
1412
1413 020132      PRINTF  #FMT18,#NOFLDE    ;TELL OPR NO 'FIELD' ENTRY HERE
(8) 020132 012746 003662      MOV      #NOFLDE,-(SP)
(7) 020136 012746 007624      MOV      #FMT18,-(SP)
(6) 020142 012746 000002      MOV      #2,-(SP)
(3) 020146 010600      MOV      SP,R0
(4) 020150 104417      TRAP    C$PNTF
(4) 020152 062706 000006      ADD      #6,SP
1414 020156 000137 017410      JMP      BSDELS          ;SELECT THE NEXT UNIT
1415
1416      ;HERE TO CLEAR THIS ENTRY FROM REST OF FIELD BAD SECTOR FILE
1417      ;WILL COPY THIS MODIFIED SECTOR PAIR INTO THE ENTIRE 'FIELD' BAD SEC FILE
1418
1419 020162 004537 016770 3$:  JSR      R5,WRTBSF       ;WRITE UPDATED 'FIELD' BAD SECTOR FILE
1420      ;/ON THE PACK IF REQUESTED
1421 020166 000137 017410      JMP      BSDELS          ;SELECT THE NEXT UNIT

```

```

1423 020172
(2)
1424
1425
1426
1427
1428 020172
(2)
1429
1430 020172
(8) 020172 012746 004256
(7) 020176 012746 007631
(6) 020202 012746 000002
(3) 020206 010600
(4) 020210 104417
(4) 020212 062706 000006
1431 020216
(3) 020216 104443
(3) 020220 000404
(4) 020222 002414
(5) 020224 000120
(5) 020226 006144
(5) 020230 177777
(3) 020232
1432 020232 005737 002414
1433 020236 001402
1434 020240 004537 012370
1435 020244 000137 013546
1436 020250 000404
1437 020252 004537 012400
1438 020256 000137 013546
1439
1440 020262 004537 012546
1441 020266 004537 025246
1442 020272 005737 002414
1443 020276 001403
1444 020300 004537 012642
1445 020304 000762
1446
1447 020306 004537 012502
1448 020312 004537 025422
1449 020316 032701 020000
1450 020322 001416
1451 020324 004537 012546
1452 020330
(8) 020330 012746 003460
(7) 020334 012746 010044
(6) 020340 012746 000002
(3) 020344 010600
(4) 020346 104417
(4) 020350 062706 000006
1453 020354 000137 020252
1454
1455 020360
(7) 020360 012746 010041
(6) 020364 012746 000001

```

```

STARS
:*****
:BSWRITE -- ROUTINE TO WRITE THE WHOLE PACK WITH THE WORST CASE DATA PATTERN
:          FOR THE RL01/2 THEN ISSUE THE 'VERIFY' (READ PACK) COMMAND. THIS
:          WILL CHECK THE PACK FOR BAD SPOTS AND COMPARE THE FOUND ENTRIES
:          WITH THE EXISTING BAD SECTOR FILE.
STARS
:*****
BSWRITE: PRINTF #FMT19,#MWRITE ;TELL OPR WHAT IS HAPPENING
        MOV #MWRITE,-(SP)
        MOV #FMT19,-(SP)
        MOV #2,-(SP)
        MOV SP,R0
        TRAP C$PNTF
        ADD #6,SP
        GMANIL MSTWRT,TEMPO,177777,NO
        TRAP C$GMAN
        BR 10021$
        .WORD TEMPO
        .WORD T$CODE
        .WORD MSTWRT
        .WORD 177777
10021$
        TST TEMPO
        BEQ 1$
        JSR R5,FIRST ;QUIT IF CAN'T WRITE ON ALL PACKS
        ;SELECT THE 1ST UNIT
1$:     JMP NXTCMD ;NONE AVAIL.!
        BR BSWRTL
BSWRTL: JSR R5,SELDRV ;SELECT THE NEXT UNIT
        JMP NXTCMD ;ALL DONE
BSWRTL: JSR R5,DRVID ;TELL OPR WHAT DRIVE SELECTED
        JSR R5,LOADED
        TST TEMPO
        BEQ 1$ ;DRV READY
        JSR R5,DRNRDY ;TELL OPR NOT READY
        BR BSWRTL ;SELECT THE NEXT UNIT
1$:     JSR R5,PTIME ;PRINT THE RUN TIME
        JSR R5,GETDST ;GET STATUS OF DRV
        BIT #WL,R1 ;WRITE LOCKED?
        BEQ 2$ ;NO
        JSR R5,DRVID ;TELL THE DRIVE ID
        PRINTF #MSG,#WRTLCK ;YES
        MOV #WRTLCK,-(SP)
        MOV #MSG,-(SP)
        MOV #2,-(SP)
        MOV SP,R0
        TRAP C$PNTF
        ADD #6,SP
11$:    JMP BSWRTL ;SELECT THE NEXT UNIT
2$:     PRINTF #MCRLF
        MOV #MCRLF,-(SP)
        MOV #1,-(SP)

```

(3)	020370	010600		MOV	SP,RO	
(4)	020372	104417		TRAP	C\$PNTF	
(4)	020374	062706	000004	ADD	#4,SP	
1456	020400	004537	020612	JSR	R5,CLRBSN	;CLEAR THE TEMP STORAGE FOR HARD ERRORS
1457	020404	005037	002232	CLR	SFTCNT	;CLEAR THE SOFT ERROR COUNTER
1458	020410	005037	002230	CLR	ERRCNT	;CLEAR THE HARD ERROR COUNTER
1459						
1460	020414			PRINTF	#FMT18,#WRPKF	;PRINT WRITE PACK FWD
(8)	020414	012746	006674	MOV	#WRPKF,-(SP)	
(7)	020420	012746	007624	MOV	#FMT18,-(SP)	
(6)	020424	012746	000002	MOV	#2,-(SP)	
(3)	020430	010600		MOV	SP,RO	
(4)	020432	104417		TRAP	C\$PNTF	
(4)	020434	062706	000006	ADD	#6,SP	
1461	020440	005737	010130	TST	WRISAW	;SAWTOOTH WRT?
1462	020444	001412		BEQ	3\$	;NO
1463	020446			PRINTF	#MSG,#SAWFWD	;YES - TELL OPR
(8)	020446	012746	006603	MOV	#SAWFWD,-(SP)	
(7)	020452	012746	010044	MOV	#MSG,-(SP)	
(6)	020456	012746	000002	MOV	#2,-(SP)	
(3)	020462	010600		MOV	SP,RO	
(4)	020464	104417		TRAP	C\$PNTF	
(4)	020466	062706	000006	ADD	#6,SP	
1464	020472	005037	002312	CLR	FWDFLG	;SET CONTROL FOR FWD SAWTOOTH WRITE
1465	020476	004537	025514	JSR	R5,WRPACK	;WRITE THE PACK
1466	020502	004537	020636	JSR	R5,CVERIFY	;CALL THE VERIFY ROUTINE
1467						
1468	020506	004537	012502	JSR	R5,PTIME	;TELL THE HALF TIME
1469	020512			PRINTF	#FMT18,#WRPKR	;TELL OPR WRT PACK REVERSE
(8)	020512	012746	006722	MOV	#WRPKR,-(SP)	
(7)	020516	012746	007624	MOV	#FMT18,-(SP)	
(6)	020522	012746	000002	MOV	#2,-(SP)	
(3)	020526	010600		MOV	SP,RO	
(4)	020530	104417		TRAP	C\$PNTF	
(4)	020532	062706	000006	ADD	#6,SP	
1470	020536	005737	010130	TST	WRISAW	;SAWTOOTH WRT?
1471	020542	001412		BEQ	4\$	;NO
1472	020544			PRINTF	#MSG,#SAWREV	;YES
(8)	020544	012746	006636	MOV	#SAWREV,-(SP)	
(7)	020550	012746	010044	MOV	#MSG,-(SP)	
(6)	020554	012746	000002	MOV	#2,-(SP)	
(3)	020560	010600		MOV	SP,RO	
(4)	020562	104417		TRAP	C\$PNTF	
(4)	020564	062706	000006	ADD	#6,SP	
1473	020570	005237	002312	INC	FWDFLG	;SET CONTROL FOR REVERSE SAWTOOTH WRT
1474	020574	004537	025514	JSR	R5,WRPACK	;WRITE THE PACK
1475	020600	004537	020636	JSR	R5,CVERIFY	;CALL THE VERIFY ROUTINE
1476	020604	004537	021110	JSR	R5,ENDRD1	;PRINT THE TOTALS OF ERRORS DETECTED
1477	020610	000620		BR	BSWRTS	;SELECT THE NEXT DRIVE

3\$:

4\$:

1479  
1480  
1481  
1482 020612 010146  
1483 020614 012701 030014  
1484 020620 012721 177777  
1485 020624 022701 030410  
1486 020630 001373  
1487 020632 012601  
1488 020634 000205

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

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

```

1490 020636 STARS
(2) :*****
1491 :BSVERIFY -- ROUTINE TO READ THE PACK TO FIND BAD SPOTS. SPOTS THAT
1492 : ARE 'BAD' AFTER 16 RETRYS TO RECOVER THE DATA WILL BE ENTERED
1493 : INTO A TEMPORARY AREA FOR LATER INSERTION INTO THE REAL BAD
1494 : SECTOR FILE (UNDER THE OPERATOR'S CONTROL).
1495 020636 STARS
(2) :*****
1496
1497 020636 005237 002314 CVERIFY: INC CVFLG ;SET THE 'CALLED' FLAG
1498 020642 000402 BR COMVER ;GO TO THE COMMON VERIFY CODE
1499
1500 020644 005037 002314 BSVERIFY: CLR CVFLG ;CLEAR THF 'CALLED' FLAG
1501
1502 020650 COMVER: PRINTF #FMT19,#VERIFY ;MSG. 'READING PACK'
(8) 020650 012746 004241 MOV #VERIFY,-(SP)
(7) 020654 012746 007631 MOV #FMT19,-(SP)
(6) 020660 012746 000002 MOV #2,-(SP)
(3) 020664 010600 MOV SP,R0
(4) 020666 104417 TRAP CSPNTF
(4) 020670 062706 000006 ADD #6,SP
1503 020674 005737 002314 TST CVFLG ;'CALLED'?
1504 020700 001011 BNE BSVERL ;YES - SKIP SELECT CODE
1505 020702 004537 012370 JSR R5,FIRST ;NO - SELECT THE 1ST UNIT
1506 020706 000137 013546 JMP NXTCMD ;NONE AVAIL.!
1507 020712 000404 BR BSVERL
1508 020714 004537 012400 BSVERS: JSR R5,SELDRV ;SELECT THE NEXT UNIT
1509 020720 000137 013546 JMP NXTCMD ;ALL DONE
1510 020724 004537 025246 BSVERL: JSR R5,LOADED ;DRV RDY?
1511 020730 005737 002414 TST TEMPO
1512 020734 001410 BEQ 1$ ;YES
1513 020736 004537 012642 JSR R5,DRNRDY
1514 020742 005737 002314 TST CVFLG ;'CALLED'?
1515 020746 001002 BNE 10$ ;YES
1516 020750 000137 020714 JMP BSVERS ;SELECT THE NEXT UNIT
1517 020754 000205 10$: RTS R5 ;NO - EXIT NOW
1518
1519 020756 1$: PRINTF #MCRLF
(7) 020756 012746 010041 MOV #MCRLF,-(SP)
(6) 020762 012746 000001 MOV #1,-(SP)
(3) 020766 010600 MOV SP,R0
(4) 020770 104417 TRAP CSPNTF
(4) 020772 062706 000004 ADD #4,SP
1520 020776 022737 000005 013650 CMP #5,INPUT ;HERE FROM 'WRITE' COMMAND?
1521 021004 001406 BEQ 11$ ;YES
1522 021006 004537 020612 JSR R5,CLRBSN ;NO - INIT HARD ERROR STORAGE AREA
1523 021012 005037 002232 CLR SFTCNT ;CLEAR THE SOFT ERROR COUNTER
1524 021016 005037 002230 CLR ERRCNT ;CLEAR THE HARD ERROR COUNT
1525 021022 004537 023656 11$: JSR R5,RDBDSC ;GET A FRESH COPY OF THE BAD SECTOR FILE
1526 021026 004537 026754 JSR R5,HDHOME ;PUT THE HEADS OVER CYLINDER 0
1527 021032 012737 002450 002256 MOV #BUF1,BBA ;POINT TO THE BUFFER FOR READ/WRITE
1528 021040 012737 175400 002246 MOV #-1280.,BMP ;SAVE THE WC FOR 10 SECTORS
1529 021046 005037 002416 CLR TEMP1 ;START AT HEAD 0
1530 021052 005001 CLR R1 ;START AT CYLINDER 0
1531
1532 021054 022737 000001 002316 CONREAD: CMP #1,TDR ;DRIVE = RL01?

```

```

1533 021062 001101      BNE      CRD2      ;NO - MUST BE AN RL02
1534 021064 022701 077600  CMP      #077600,R1 ;AT RL01 LAST CYL?
1535 021070 001101      BNE      STREAD    ;NO - READ THIS TRACK
1536
1537 021072 005737 002416  CRD1:   TST      TEMP1 ;ON LAST CYL ... IS IT LAST TRACK?
1538 021076 001476      BEQ      STREAD    ;NO - DO THE READ
1539
1540 021100 005737 002314  ENDRD:  TST      CVFLG ;'CALLED'?
1541 021104 001401      BEQ      ENDRD1   ;NO - PROCEED WITH THE TOTALS PRINTOUT
1542 021106 000205      RTS      R5       ;YES - EXIT NOW
1543 021110 004537 012502  ENDRD1: JSR      R5,PTIME ;PRINT THE DONE READING TIME
1544 021114      PRINTF  #MSG,#MDONE ;TELL OPR ALL DONE
      (8) 021114 012746 003720  MOV      #MDONE,-(SP)
      (7) 021120 012746 010044  MOV      #MSG,-(SP)
      (6) 021124 012746 000002  MOV      #2,-(SP)
      (3) 021130 010600      MOV      SP,R0
      (4) 021132 104417      TRAP     C$PNTF
      (4) 021134 062706 000006  ADD      #6,SP
1545 021140 004537 012546  JSR      R5,DRVID ;TELL OPR WHICH DRIVE
1546 021144      PRINTF  #FMTTB,#TSOFT,SFTCNT ;PRINT TOTAL 'SOFT' ERRORS
      (9) 021144 013746 002232  MOV      SFTCNT,-(SP)
      (8) 021150 012746 003047  MOV      #TSOFT,-(SP)
      (7) 021154 012746 007724  MOV      #FMTTB,-(SP)
      (6) 021160 012746 000003  MOV      #3,-(SP)
      (3) 021164 010600      MOV      SP,R0
      (4) 021166 104417      TRAP     C$PNTF
      (4) 021170 062706 000010  ADD      #10,SP
1547 021174      PRINTF  #FMTTB,#THARD,ERRCNT ;PRINT TOTAL 'HARD' ERRORS
      (9) 021174 013746 002230  MOV      ERRCNT,-(SP)
      (8) 021200 012746 003076  MOV      #THARD,-(SP)
      (7) 021204 012746 007724  MOV      #FMTTB,-(SP)
      (6) 021210 012746 000003  MOV      #3,-(SP)
      (3) 021214 010600      MOV      SP,R0
      (4) 021216 104417      TRAP     C$PNTF
      (4) 021220 062706 000010  ADD      #10,SP
1548 021224      PRINTF  #FMT18,#HYPHEN
      (8) 021224 012746 002675  MOV      #HYPHEN,-(SP)
      (7) 021230 012746 007624  MOV      #FMT18,-(SP)
      (6) 021234 012746 000002  MOV      #2,-(SP)
      (3) 021240 010600      MOV      SP,R0
      (4) 021242 104417      TRAP     C$PNTF
      (4) 021244 062706 000006  ADD      #6,SP
1549 021250 004537 022162  JSR      R5,ADDFND ;SEE IF OPR WANTS TO UPDATE BAD SEC FILE
1550 021254 005737 002314  TST      CVFLG    ;'CALLED'?
1551 021260 001001      BNE      1$       ;YES
1552 021262 000614      BR       BSVERS   ;SELECT THE NEXT UNIT
1553 021264 000205      RTS      R5       ;NO - EXIT NOW
1554
1555
1556 021266 022701 177600  ;HERE TO CHECK THE END OF AN RL02
1557 021272 001677  CRD2:   CMP      #177600,R1 ;RL02 LAST CYL?
      BEQ      CRD1   ;YES - CHECK TO SEE IF LAST TRACK TOC

```



0003

ZRLMB0 RL01/02 BD SEC FIL TL  
ZRLMB.MAC 12-DEC-79 14:06

MACV11 30A(1052) 17-DEC-79 10:53 L 6 PAGE 1-53  
GLOBAL SUBROUTINES

```

1559                                     ;HERE TO READ THE TRACK SELECTED...WILL TRY 10 SECTORS AT A TIME
1560                                     ;IF AN ERROR IS DETECTED, WILL THEN TRY TO RECOVER BY READING ONE
1561                                     ;SECTOR AT A TIME. A SECTOR IS DEEMED 'HARD ERROR' AFTER 16 RETRYS.
1562                                     ;ALL BAD SPOTS WILL ENTER A TEMP BAD SEC FILE STORAGE AREA...TO BE ADDED
1563                                     ;TO THE REAL BAD SECTOR FILE AFTER WHOLE PACK HAS BEEN READ.
1564
1565 021274 005002                          STREAD: CLR      R2                ;START AT SECTOR 0 ON THIS TRACK
1566 021276 005037 002412                  CLR      DECNT                ;INITIALIZE ERROR RECOVERY COUNTER
1567 021302 010137 002244                  SRD1:  MOV     R1,BDA           ;INSERT THE CYL # INTO DISK ADDR
1568 021306 053737 002416 002244          BIS     TEMP1,BDA            ;ADD THE HEAD NUMBER (0 OR 1)
1569 021314 050237 002244                  BIS     R2,BDA                ;ADD THE SECTOR NUMBER
1570 021320 012737 000014 00226C          MOV     #READ,FUNCTION       ;GET A READ FUNCTION
1571 021326 004537 023534                  JSR     R5,LDFUNC            ;ISSUE THE READ CMD
1572 021332 004537 025310                  JSR     R5,WTRDY             ;WAIT FOR READ TO FINISH
1573
1574 021336 005777 60706                    TST     @DCS                  ;ANY ERROR ON THE READ?
1575 021342 100041                          BPL     NXTSEC                ;BR IF OK
1576
1577                                     ;HERE IF AN ERROR DETECTED ON THE READ ... READ ONE SECTOR AT A TIME
1578                                     ;TILL WHOLE TRACK HAS BEEN READ
1579
1580                                     PRINTF #MCRLF
1581 (7) 021344 012746 010041                MOV     #MCRLF,-(SP)
1582 (6) 021350 012746 000001                MOV     #1,-(SP)
1583 (3) 021354 010600                          MOV     SP,R0
1584 (4) 021356 104417                          TRAP   C$PNTF
1585 (4) 021360 062706 000004                ADD     #4,SP
1581 021364 017737 160660 002252          MOV     @DCS,E.DCS           ;GET THE ERROR DETECTED
1582 021372 013703 002250                    MOV     DCS,R3                ;GET THE BASE ADDRESS FOR RLCS
1583 021376 016337 000004 002440          MOV     DA(R3),E.DA           ;SAVE THE DISK ADDRESS AT ERROR
1584 021404 005337 002440                    DEC     E.DA                   ;SECTOR IS PREVIOUS FROM INDICATED
1585 021410 013737 002440 002410          MOV     E.DA,CHKSEC           ;STORE ERROR ADDRESS FOR RECOVERY LOOP
1586 021416 013737 002410 002502          MOV     CHKSEC,FRSTER        ;TELL OPR ABOUT THE ERROR DETECTED
1587 021424                                ERRSOFT 400,MSFER,ERR1
1588 (4) 021424 104457                          TRAP   C$ERSOFT
1589 (5) 021426 000620                          .WORD  400
1590 (5) 021430 003310                          .WORD  MSFER
1591 (5) 021432 007130                          .WORD  ERR1
1588 021434 004537 025436                    JSR     R5,ISDRST             ;ISSUE A DRIVE RESET TO CLEAR THE ERROR
1589 021440 005237 002232                    INC     SFTCNT                ;ADD TO SOFT ERROR TALLY
1590 021444 000446                          BR     ONESEC                  ;RECOVER THE TRACK DATA...SLOWLY

```

ZRLMBG RL01/02 BC SEC FIL TL  
 ZRLMB.MAC 12-DEC-79 14:06

MACV11 30A(1052) 17-DEC-79 10:53 PAGE 1-54  
 GLOBAL SUBROUTINES

M 6

```

1592                                     ;HERE TO SELECT THE NEXT SECTOR ADDR TO READ FROM ON THIS TRACK
1593                                     ;HERE IF NO ERROR DETECTED ON PREV. READ CMD
1594
1595 021446 062702 000012          NXTSEC: ADD      #10.,R2          ;POINT TO THE NEXT SPOT ON THE TRACK
1596 021452 022702 000050          CMP      #40.,R2          ;END OF THE TRACK?
1597 021456 001311          BNE      SRD1            ;NO - DO THE READ
1598
1599                                     ;HERE TO SELECT THE NEXT TRACK TO READ ... WILL DO A SEEK TO NEXT HEAD
1600                                     ;OR TO THE NEXT CYLINDER.
1601
1602 021460 005737 002416          NXTTRK: TST     TEMP1          ;ON HEAD #1 NOW?
1603 021464 001427          BEQ      5$              ;NO - SEEK TO NEXT TRACK SAME CYL
1604 021466 005037 002416          CLR      TEMP1          ;SET FOR NEXT CYL HEAD 0
1605 021472 062701 000200          ADD      #200,R1         ;POINT TO THE NEXT CYLINDER
1606 021476 042701 000177          BIC      #177,R1        ;CLEAR UNEXPECTED JUNK BITS
1607 021502 012737 000200 002244  MOV      #200,BDA
1608
1609 021510 052737 000005 002244 4$:  BIS      #SIGN!MK,BDA      ;SET FOR A SEEK CMD
1610 021516 012737 000006 002260  MOV      #SEEK,FUNC       ;GET THE SEEK CMD
1611 021524 004537 023534          JSR      R5,LDFUNC       ;ISSUE THE SEEK
1612 021530 004537 025310          JSR      R5,WTRDY        ;WAIT TILL READY
1613 021534 010137 002322          MOV      R1,PRPOS        ;SAVE THE PRESENT POSITION ON DISK
1614 021540 000137 021054          JMP      CONREAD         ;CONTINUE READING THE PACK
1615
1616 021544 012737 000100 002416 5$:  MOV      #HEAD,TEMP1      ;SAVE HEAD SELECT STATUS
1617 021552 012737 000020 002244  MOV      #SKHS,BDA       ;SET FOR SEEK TO NEXT TRACK SAME CYL
1618 021560 000753          BR       4$              ;ISSUE THE SEEK

```

```

1620 021562 STARS
(2) :*****
1621 :HERE TO TRY AND RECOVER THE DATA ON SFELECTED TRACK BY READING 1 SECTOR
1622 :AT A TIME. SECTOR WILL BE MARKED 'BAD' AFTER 16 RETRYS AND NO RECOVERY.
1623 021562 STARS
(2) :*****
1624 :*****
1625 021562 005002 ONESEC: CLR R2 ;START AT SECTOR 0 ON THIS TRACK
1626 021564 012737 177600 002246 MOV #-128.,BMP ;SET THE WC AT 1 SECTOR'S WORTH
1627 021572 013737 002322 002244 1$: MOV PRPOS,BDA ;GET THE CYL # TO START AT
1628 021600 050237 002244 BIS R2,BDA ;ADD IN THE SECTOR NUMBER
1629 021604 053737 002416 002244 BIS TEMP1,BDA ;AND THE TRACK (HEAD 0 OR 1)
1630
1631 :READ A SECTOR
1632 021612 012737 000014 002260 2$: MOV #READ,FUNC ;GET A READ FUNCTION
1633 021620 004537 023534 JSR R5,LDFUNC ;ISSUE THE READ
1634 021624 004537 025310 JSR R5,WTRDY ;WAIT FOR READY
1635
1636 021630 005777 160414 TST @DCS ;THIS SECTOR READ OK?
1637 021634 100106 BPL 3$ ;BE IF OK - SELECT NEXT SECTOR
1638
1639 :ERROR IN THIS SECTOR - TRY A MAX OF 16 TIMES TO RECOVER
1640
1641 021636 017737 160406 002252 MOV @DCS,E.DCS ;SAVE THE DETECTED ERROR
1642 021644 023737 002244 002502 CMP BDA,FRSTER ;DID WE REPORT THIS IN MAIN PROGRAM?
1643 021652 001425 BEQ 10$ ;YES - SKIP
1644 021654 005737 002412 TST DECNT ;DID WE REPORT IT YET IN RECOVERY LOOP?
1645 021660 001022 BNE 10$ ;YES - SKIP
1646 021662 PRINTF #MCRLF ;ELSE REPORT SOFT ERROR NOW
(7) 021662 012746 010041 MOV #MCRLF,-(SP)
(6) 021666 012746 000001 MOV #1,-(SP)
(3) 021672 010600 MOV SP,R0
(4) 021674 104417 TRAP C$PNTF
(4) 021676 062706 000004 ADD #4,SP
1647 021702 013737 002244 002410 MOV BDA,CHKSEC ;GET ERROR ADDRESS FOR PRINOUT
1648 021710 013737 002244 002440 MOV BDA,E.DA
1649 021716 ERRSOFT 420.,MSFER,ERR1
(4) 021716 104457 TRAP C$ERSOFT
(5) 021720 000644 .WORD 420
(5) 021722 003310 .WORD MSFER
(5) 021724 007130 .WORD ERR1
1650 021726 005237 002412 10$: INC DECNT ;COUNT THIS RETRY
1651 021732 013737 002244 002410 MOV BDA,CHKSEC ;SEE IF THIS SECTOR IS ALREADY IN
1652 021740 004537 027340 JSR R5,CKBDSC ;THE BAD SECTOR FILE
1653 021744 005737 002406 TST HDRFND ;IN THE FILE NOW?
1654 021750 001423 BEQ 21$ ;BR IF ERROR
1655 021752 PRINTF #FMT18,#INBSF ;TELL OPR SECT IS IN BSF ALREADY
(8) 021752 012746 006346 MOV #INBSF,-(SP)
(7) 021756 012746 007624 MOV #FMT18,-(SP)
(6) 021762 012746 000002 MOV #2,-(SP)
(3) 021766 010600 MOV SP,R0
(4) 021770 104417 TRAP C$PNTF
(4) 021772 062706 000006 ADD #6,SP
1656 021776 PRINTF #MCRLF
(7) 021776 012746 010041 MOV #MCRLF,-(SP)
(6) 022002 012746 000001 MOV #1,-(SP)

```

```
(3) 022006 010600      MOV      SP,R0
(4) 022010 104417      TRAP    C$PNTF
(4) 022012 062706 000004  ADD     #4,SP
1657 022016 000445      BR      30$      ;DO THE NEXT SECTOR ON THIS TRACK
1658
1659 022020 022737 000020 002412 21$:  CMP     #16.,DECNT ;TIME TO MARK IT AS A BAD SPOT?
1660 022026 001403      BEQ     22$      ;YES
1661 022030 004537 025436      JSR     R5,ISDRST ;NO - ISSUE A DRIVE RESET
1662 022034 000666      BR      2$      ;AND CONTINUE
1663 022036 005337 002232 22$:  DEC     SFTCNT    ;DELETE THIS HARD ERROR FROM SOFT ERROR TALLY
1664 022042 004537 026604      JSR     R5,INBAD  ;YES - MAKE A TEMP BAD SPOT ENTRY
1665 022046 004537 025436 23$:  JSR     R5,ISDRST ;RESET THE DRIVE FOR THE NEXT FJUNCTION
1666
1667      ;HERE TO SELECT THE NEXT SECTOR TO RECOVER IN THIS TRACK
1668
1669 022052 005737 002412 3$:  TST     DECNT      ;ANY DETECTED?
1670 022056 001433      BEQ     31$      ;OR IF NONE THIS SECTOR
1671 022060 005037 002412      CLR     DECNT      ;CLEAR LOOP COUNTER FOR NEXT SECTOR
1672 022064      PRINTF #FMT18,#MSREC ;TELL OPR 'RECOVERED'
(8) 022064 012746 003125      MOV     #MSREC,-(SP)
(7) 022070 012746 007624      MOV     #FMT18,-(SP)
(6) 022074 012746 000002      MOV     #2,-(SP)
(3) 022100 010600      MOV     SP,R0
(4) 022102 104417      TRAP    C$PNTF
(4) 022104 062706 000006  ADD     #6,SP
1673 022110      PRINTF #MCRLF
(7) 022110 012746 010041      MOV     #MCRLF,-(SP)
(6) 022114 012746 000001      MOV     #1,-(SP)
(3) 022120 010600      MOV     SP,R0
(4) 022122 104417      TRAP    C$PNTF
(4) 022124 062706 000004  ADD     #4,SP
1674 022130 000406      BR      31$
1675 022132 005337 002232 30$:  DEC     SFTCNT    ;ADJUST COUNTERS BECAUSE SECTOR-
1676 022136 005037 002412      CLR     DECNT      ;IN-ERROR ALREADY IN BSF
1677 022142 004537 025436      JSR     R5,ISDRST ;RESET THE DRIVE
1678
1679 022146 005202 31$:  INC     R2          ;POINT TO THE NEXT SECTOR
1680 022150 022702 000050      CMP     #40.,R2   ;END OF THIS TRACK?
1681 022154 001206      BNE     1$          ;NO - READ THIS SECTOR
1682 022156 000137 021460      JMP     NXTTRK    ;ELSE BACK TO NORMAL 10 SECTOR READS
1683
1684 022162      STARS
(2)      ;:*****
1685      ;ADDEND -- ROUTINE TO ASK OPR IF THE NEW BAD SPOTS FOUND BY THE
1686      ;      'WRITE' COMMAND OR THE 'VERIFY' COMMAND IS TO BE ADDED TO THE
1687      ;      BAD SECTOR FILE ON THE PACK.
1688 022162      STARS
(2)      ;:*****
1689
1690 022162 005737 002464  ADDEND: TST     ACCESS ;ALLOWED TO UPDATE THE PACK?
1691 022166 001177      BNE     ADDFEX    ;NO - EXIT NOW
1692 022170 004537 023656      JSR     R5,RDBDSC ;GET A FRESH COPY OF THE BAD SECTOR FILE
1693 022174 004537 025076      JSR     R5,RDFIELD ;GET A CORE COPY OF THE 'FIELD' FILE
1694 022200 012701 030014      MOV     #BSECN,R1 ;POINT TO THE NEW ENTRY TABLE
1695 022204 005737 002230      TST     ERRCNT    ;SEE IF ANY NEW BAD SPOTS
1696 022210 001566      BEQ     ADDFEX    ;EXIT IF NONE DETECTED
```

```

1697 022212 005002          CLR      R2          ;CLEAR AN INDEX INTO BAD SECTOR FILE
1698 022214 005037 015342    CLR      SECNUM     ;START AT THE 1ST SECTOR IN 'FIELD'
1699 022220 012737 000020    015340  MOV      #16,SECMAX ;SETUP THE LIMIT
1700 022226 004537 015622    JSR      R5,BSFOK   ;SET R2 TO POINT INTO THE FILE
1701 022232 005737 015346    TST      BSFOKF     ;SEE IF FIELD FILE EXISTS
1702 022236 001420          BEQ      1$         ;BR IF OK
1703 022240          PRINTF  #FMT18,#NSWSEC ;TELL OPR THAT NO 'FIELD' FILE EXISTS
(8) 022240 012746 003636    MOV      #NSWSEC,-(SP)
(7) 022244 012746 007624    MOV      #FMT18,-(SP)
(6) 022250 012746 000002    MOV      #2,-(SP)
(3) 022254 010600          MOV      SP,R0
(4) 022256 104417          TRAP    C$PNTF
(4) 022260 062706 000006    ADD      #6,SP
1704 022264 004537 022570    JSR      R5,NEWBSF  ;ASK IF TIME TO MAKE A 'FIELD' FILE
1705 022270 005737 002414    TST      TEMPO      ;WAS A 'FIELD' FILE BUILT?
1706 022274 001001          BNE     1$         ;BR IF YES
1707 022276 000533          BR      ADDFEX     ;NO - EXIT
1708 022300 011137 002410    1$:     MOV      (R1),CHKSEC ;GET AN ENTRY
1709 022304 023727 002410    177777  CMP      CHKSEC,#-1 ;DONE?
1710 022312 001523          BEQ      4$         ;YES - UPDATE REST OF 'FIELD' FILE
1711
1712          ;HERE TO SEE IF ENTRY ALREADY EXISTS...SHOULDN'T
1713
1714 022314 004537 027340          JSR      R5,CKBDSC  ;WELL...
1715 022320 005737 002406          TST      HDRFND     ;FIND IN LIST ALREADY?
1716 022324 001114          BNE     3$         ;YES - LOOK AT THE NEXT ENTRY
1717
1718          ;HERE TO ASK OPR IF THIS ENTRY TO BE ADDED TO BAD SEC FILE
1719
1720 022326 011137 015332          MOV      (R1),BSFCYL ;GET DA FOR CYL #
1721 022332 042737 000177    015332  BIC      #177,BSFCYL ;CLEAR HEAD & SECTOR #
1722 022340 000337 015332          SWAB    BSFCYL
1723 022344 000241          CLC
1724 022346 006137 015332          ROL     BSFCYL
1725 022352 103002          BCC     1$
1726 022354 005237 015332          INC     BSFCYL     ;ADD IN LOW ORDER CYL #
1727 022360 011137 015334    11$:     MOV      (R1),BSFSEC ;GET DA FOR SEC VALUE
1728 022364 042737 177700    015334  BIC      #177700,BSFSEC ;CLEAR CYLINDER # & HEAD
1729 022372 005037 015336          CLR     BSFHD      ;START AT HEAD 0
1730 022376 032711 000100          BIT     #100,(R1)  ;HEAD 1?
1731 022402 001402          BEQ     2$         ;NO - ITS HEAD 0
1732 022404 005237 015336          INC     BSFHD      ;POINT TO HEAD 1
1733 022410    2$:     PRINTF  #FMT16,#NEWENT,#CMMSG,BSFCYL,#SMSG,BSFSEC,#HMSG,BSFHD
(14) 022410 013746 015336    MOV      BSFHD,-(SP)
(13) 022414 012746 002563    MOV      #HMSG,-(SP)
(12) 022420 013746 015334    MOV      BSFSEC,-(SP)
(11) 022424 012746 003243    MOV      #SMSG,-(SP)
(10) 022430 013746 015332    MOV      BSFCYL,-(SP)
(9) 022434 012746 002550    MOV      #CMMSG,-(SP)
(8) 022440 012746 006200    MOV      #NEWENT,-(SP)
(7) 022444 012746 007503    MOV      #FMT16,-(SP)
(6) 022450 012746 000010    MOV      #10,-(SP)
(3) 022454 010600          MOV      SP,R0
(4) 022456 104417          TRAP    C$PNTF
(4) 022460 062706 000022    ADD      #22,SP
1734 022464          GMANIL  ABSMSG,TEMPO,177777,NO ;ASK OPR IF OK TO ENTER

```

```
(3) 022464 104443          TRAP  CS$GMAN
(3) 022466 0J0404          BR    10022$
(4) 022470 002414          .WORD TEMF0
(5) 022472 000120          .WORD T$CODE
(5) 022474 005164          .WORD ABSMSG
(5) 022475 177777          .WORD 177777
(3) 022500
1735 022500 005737 002414    10022$: TST  TEMPO          ;BR IF NO
1736 022504 001424          BEQ   3$             ;NO - GET THE NEXT ENTRY
1737
1738
1739          ;HERE TO INSERT THIS SPOT IN THE BAD SECTOR FILE
1740
1741 022506 005762 030530    21$:  TST  BSFILE(R2) ;SEE IF A FREE SPOT
1742 022512 100403          BMI   22$          ;BR IF FOUND A FREE ENTRY
1743 022514 062702 000004    ADD   #4,R2        ;POINT TO THE NEXT ENTRY SLOT
1744 022520 000772          BR    21$          ;AND TRY AGAIN
1745
1746 022522 013762 015332 030530 22$:  MOV  BSFCYL,BSFILE(R2) ;INSERT THE CYL # INTO FILE
1747 022530 011162 030532    MOV  (R1),BSFILE+2(R2) ;ADD THE SECTOR NUM & HEAD
1748 022534 042762 177700 030532  BIC  #177700,BSFILE+2(R2) ;CLEAR CYL # AND HEAD
1749 022542 005737 015336    TST  BSFHD         ;IS IT HEAD ONE?
1750 022546 001403          BEQ   3$             ;NO - SKIP
1751 022550 052762 000400 030532  BIS  #400,BSFILE+2(R2) ;YES - SET BIT 8 FOR HEAD ONE
1752
1753          ;HERE TO UPDATE THE POINTER INTO THE TEMP BAD SEC TABLE
1754 022556 005721          3$:  TST  (R1)+       ;UPDATE THE POINTER
1755 022560 000647          BR    1$             ;PROCESS THIS ENTRY
1756
1757 022562 004537 016770    4$:  JSR  R5,WRTBSF   ;WRITE UPDATED 'FIELD' BAD SECTOR FILE
1758          ;/ON THE PACK IF REQUESTED
1759
1760          ;HERE TO EXIT THIS PHASE
1761 022566 000205          ADDFEX: RTS  R5     ;EXIT
1762
1763
```

1765 022570  
(2)  
1766  
1767  
1768 022570  
(2)  
1769  
1770 022570 010146  
1771 022572  
(3) 022572 104443  
(3) 022574 000404  
(4) 022576 002414  
(5) 022600 000120  
(5) 022602 003155  
(5) 022604 177777  
(3) 022606  
1772 022606 005737 002414  
1773 022612 001502  
1774  
1775  
1776 022614 012701 030530  
1777 022620 010137 015344  
1778 022624 012721 177, 7  
1779 022630 022701 033130  
1780 022634 001373  
1781 022636 012701 030530  
1782 022642 005021  
1783 022644 005021  
1784 022646 005021  
1785 022650 005011  
1786 022652 005737 002456  
1787 022656 001020  
1788  
1789 022660  
(3) 022660 104443  
(3) 022662 000406  
(4) 022664 002456  
(5) 022666 000022  
(5) 022670 005415  
(5) 022672 077777  
(5) 022674 000001  
(5) 022676 077777  
(3) 022700  
1790 022700  
(3) 022700 104443  
(3) 022702 000406  
(4) 022704 002460  
(5) 022706 000022  
(5) 022710 005471  
(5) 022712 077777  
(5) 022714 000000  
(5) 022716 077777  
(3) 022720  
1791  
1792 022720  
(10) 022720 013746 002456

```
STARS
:*****
:NEWBSF -- ROUTINE TO ASK OPR IF TIME TO CREATE A BAD SECTOR
:          FILE IF THE AREA CAN'T BE RECOGNIZED AS A BAD SECTOR FILE.
STARS
:*****

NEWBSF: MOV     R1,-(SP)           ;SAVE R1
        GMANIL  MBLD,TEMPO,177777,NO
        TRAP    C$GMAN
        BR      10023$
        .WORD   TEMPO
        .WORD   T$CODE
        .WORD   MBLD
        .WORD   177777

10023$: TST     TEMPO             ;BR IF NO
        BEQ    2$              ;EXIT

:HERE TO INIT THE BSFILE STORAGE FOR BUILDING A FILE
1$: MOV   #BSFILE,R1          ;SETUP A POINTER
        MOV   R1,SECOLD       ;POINT TO THE START OF THE 'UPDATE' AREA
11$: MOV  #-1,(R1)+          ;INIT A LOCATION
        CMP  #BSFILE+1280.,R1 ;DONE??
        BNE  11$             ;NO - PROCEED TO INIT
        MOV  #BSFILE,R1      ;GET START AGAIN
        CLR  (R1)+           ;CLEAR
        CLR  (R1)+
        CLR  (R1)+
        CLR  (R1)
        TST  SN1             ;ALREADY HAVE A SERIAL NUMBER?
        BNE  13$            ;YES - TELL OPR WHAT IT IS

12$: GMANID ABSSNL,SN1,0,77777,1,77777,NO ;GET SERIAL # LOW 5
        TRAP  C$GMAN
        BR    10024$
        .WORD SN1
        .WORD T$CODE
        .WORD ABSSNL
        .WORD 77777
        .WORD T$LOLIM
        .WORD T$HILIM

10024$: GMANID ABSSNH,SN2,0,77777,0,77777,NO ;GET SERIAL # HIGH 5
        TRAP  C$GMAN
        BR    10025$
        .WORD SN2
        .WORD T$CODE
        .WORD ABSSNH
        .WORD 77777
        .WORD T$LOLIM
        .WORD T$HILIM

10025$: PRINTF #FMTSN,#CART,SN2,SN1
        MOV   SN1,-(SP)
```

(9)	022724	013746	002460		MOV	SN2,-(SP)	
(8)	022730	012746	002526		MOV	#CART,-(SP)	
(7)	022734	012746	007707		MOV	#FMTSN,-(SP)	
(6)	022740	012746	000004		MOV	#4,-(SP)	
(3)	022744	010600			MOV	SP,R0	
(4)	022746	104417			TRAP	C\$PNTF	
(4)	022750	062706	000012		ADD	#12,SP	
1793	022754				GMANIL	VALSN,TEMPO,177777,NO	
(3)	022754	104443			TRAP	C\$GMAN	
(3)	022756	000404			BR	10026\$	
(4)	022760	002414			.WORD	TEMPO	
(5)	022762	000120			.WORD	T\$CODE	
(5)	022764	006007			.WORD	VALSN	
(5)	022766	177777			.WORD	177777	
(3)	022770			10026\$:			
1794	022770	005737	002414		TST	TEMPO	;SEE IF TYPED IN SERIAL NUMBER IS OK
1795	022774	001731			BEQ	12\$	;NO - GET A NEW SERIAL NUMBER
1796	022776	013737	002456	030530	MOV	SN1,BSFILE	;SAVE THE SERIAL NUMBER LOW 5
1797	023004	013737	002460	030532	MOV	SN2,BSFILE+2	;AND SERIAL NUMBER HIGH 5
1798	023012	005237	002414		INC	TEMPO	;INDICATE FILE BUILT - 1 SECTOR
1799	023016	000205			RTS	R5	
1800							
1801	023020	005037	002414	2\$:	CLR	TEMPO	;INDICATE NO FILE BUILT
1802	023024	012601			MOV	(SP)+,R1	
1803	023026	000205			RTS	R5	



1805 023030  
(2)  
1806  
1807  
1808  
1809  
1810 023030  
(2)  
1811  
1812 023030  
(8) 023030 012746 003213  
(7) 023034 012746 007631  
(6) 023040 012746 000002  
(3) 023044 010600  
(4) 023046 104417  
(4) 023050 062706 000006  
1813 023054 004537 012370  
1814 023060 000137 013546  
1815 023064 000404  
1816 023066 004537 012400  
1817 023072 000137 013546  
1818 023076 004537 025246  
1819 023102 005737 002414  
1820 023106 001403  
1821 023110 004537 012642  
1822 023114 000764  
1823  
1824 023116 004537 012546  
1825 023122 005737 002464  
1826 023126 001414  
1827 023130  
(8) 023130 012746 004125  
(7) 023134 012746 007624  
(6) 023140 012746 000002  
(3) 023144 010600  
(4) 023146 104417  
(4) 023150 062706 000006  
1828 023154 000137 013546  
1829  
1830 023160  
(3) 023160 104443  
(3) 023162 000404  
(4) 023164 002414  
(5) 023166 000120  
(5) 023170 006750  
(5) 023172 000001  
(3) 023174  
1831 023174 005737 002414  
1832 023200 001732  
1833  
1834 023202 004537 023656  
1835 023206 004537 025032  
1836 023212  
(8) 023212 012746 006411  
(7) 023216 012746 007631  
(6) 023222 012746 000002

STARS  
:\*\*\*\*\*  
:BSMAKE -- ROUTINE TO CREATE A 'FACTORY' OR 'FIELD' BAD SECTOR FILE.  
: THIS ROUTINE ABORTS IF 'UPDATE' ACCESS TO THE PACK IS DENIED.  
: THE 'FACTORY' FILE WILL CONTAIN ONLY THE DUMMY HEADERS...NO  
: ENTRYS CAN BE PUT IN THIS AREA!  
STARS  
:\*\*\*\*\*  
BSMAKE: PRINTF #FMT19,#BUILD ;TELL OPR WHAT IS HAPPENING  
MOV #BUILD,-(SP)  
MOV #FMT19,-(SP)  
MOV #2,-(SP)  
MOV SP,R0  
TRAP C\$PNTF  
ADD #6,SP  
JSR R5,FIRST ;SELECT THE 1ST UNIT  
JMP NXTCMD ;NONE AVAIL.!  
BR BSMKL  
BSMKS: JSR R5,SELDRV ;SELECT THE NEXT UNIT  
JMP NXTCMD ;ALL DONE  
BSMKL: JSR R5,LOADED ;DRV RDY?  
TST TEMPO  
BEQ 1\$ ;YES  
JSR R5,DRNRDY  
BR BSMKS ;SELECT THE NEXT UNIT  
1\$: JSR R5,DRVID ;TELL OPR WHAT DRIVE SELECTED  
TST ACCESS ;ALLOWED TO UPDATE PACK?  
BEQ 10\$ ;YES - PROCEED  
PRINTF #FMT18,#DENIED ;NO - TELL OPR  
MOV #DENIED,-(SP)  
MOV #FMT18,-(SP)  
MOV #2,-(SP)  
MOV SP,R0  
TRAP C\$PNTF  
ADD #6,SP  
JMP NXTCMD ;QUIT NOW  
10\$: GMANIL THISDRV,TEMPO,1,NO  
TRAP C\$GMAN  
BR 10027\$  
.WORD TEMPO  
.WORD T\$CODE  
.WORD THISDRV  
.WORD 1  
10027\$: TST TEMPO  
BEQ BSMKS ;RE-SELECT IF NOT THIS DRIVE  
11\$: JSR R5,RDBDSC ;GET A FRESH COPY OF THE 'BAD SEC FILE'  
JSR R5,RDFACT ;THEN A CORE IMAGE OF THE FACTORY FILE  
PRINTF #FMT19,#CKFACT ;TELL OPR CHECKING FACT FILE  
MOV #CKFACT,-(SP)  
MOV #FMT19,-(SP)  
MOV #2,-(SP)

```
(3) 023226 010600      MOV      SP,R0
(4) 023230 104417      TRAP     C$PNTF
(4) 023232 062706 000006  ADD      #6,SP
1837 023236 005002      CLR      R2          ;POINT TO 1ST WORD IN CORE IMAGE
1838 023240 005037 015342  CLR      SECNUM     ;START AT 1ST SECTOR PAIR IN FILE
1839 023244 012737 000020 015340  MOV      #16,SECMAX ;STOP AT THIS SECT PAIR
1840 023252 004537 015622  JSR      R5,BSFOK   ;SEE IF ANY RECOGNIZED 'FACTORY' FILE
1841 023256 005737 015346  TST      BSFOKF    ;WELL???
1842 023262 001013      BNE      12$        ;NO - ASK IF TIME TO MAKE ONE
1843 023264      PRINTF  #MSG,#OK      ;MSG TO OPR 'FOUND'
(8) 023264 012746 006340  MOV      #OK,-(SP)
(7) 023270 012746 010044  MOV      #MSG,-(SP)
(6) 023274 012746 000002  MOV      #2,-(SP)
(3) 023300 010600      MOV      SP,R0
(4) 023302 104417      TRAP     C$PNTF
(4) 023304 062706 000006  ADD      #6,SP
1844 023310 000424      BR       2$          ;JUMP OVER BUILD CODE
1845
1846      ;HERE TO BUILD A DUMMY 'FACTORY' FILE SO AT LEAST ONE EXISTS...WILL NOT
1847      ;CONTAIN ANY ENTRIES!
1848
1849 023312      12$: PRINTF  #MSG,#NHWSEC ;TELL OPR THAT NO 'FACTORY' EXISTS
(8) 023312 012746 003562  MOV      #NHWSEC,-(SP)
(7) 023316 012746 010044  MOV      #MSG,-(SP)
(6) 023322 012746 000002  MOV      #2,-(SP)
(3) 023326 010600      MOV      SP,R0
(4) 023330 104417      TRAP     C$PNTF
(4) 023332 062706 000006  ADD      #6,SP
1850 023336 004537 022570  JSR      R5,NEWBSF  ;ASK IF TIME TO BUILD ONE
1851 023342 005737 002414  TST      TEMPO     ;DID I MAKE A DUMMY FILE?
1852 023346 001405      BEQ      2$        ;NO - CHECK ON THE 'FIELD' FILE
1853 023350 013737 002414 002300  MOV      TEMPO,NEWFAC ;SET FACTORY FLAG
1854 023356 004537 016770  JSR      R5,WRTBSF ;WRITE UPDATED 'FIELD' BAD SECTOR FILE
1855      ;/ON THE PACK IF REQUESTED
1856
```

```

1858
1859
1860 023362 004537 025076
1861 023366
(8) 023366 012746 006450
(7) 023372 012746 007631
(6) 023376 012746 000002
(3) 023402 010600
(4) 023404 104417
(4) 023406 062706 000006
1862 023412 005002
1863 023414 005037 015342
1864 023420 012737 000020 015340
1865 023426 004537 015622
1866 023432 005737 015346
1867 023436 001013
1868 023440
(8) 023440 012746 006340
(7) 023444 012746 010044
(6) 023450 012746 000002
(3) 023454 010600
(4) 023456 104417
(4) 023460 062706 000006
1869 023464 000421
1870
1871 023466
(8) 023466 012746 003636
(7) 023472 012746 010044
(6) 023476 012746 000002
(3) 023502 010600
(4) 023504 104417
(4) 023506 062706 000006
1872 023512 004537 022570
1873 023516 005737 002414
1874 023522 001402
1875 023524 004537 016770
1876
1877
1878 023530 000137 023066

```

```

;HERE TO SEE IF A 'FIELD' FILE HAS TO BE BUILT
2$: JSR R5,RDFIELD ;GET A CORE IMAGE OF THE 'FIELD' FILE
PRINTF #FMT19,#CKFLD ;TELL OPR CHECKING FOR FIELD FILE
MOV #CKFLD,-(SP)
MOV #FMT19,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C$PNTF
ADD #6,SP
CLR R2 ;START AT 1ST WORD IN BUFFER
CLR SECNUM ;AND 1ST SECTOR PAIR OF FILE
MOV #16,SECMAX ;SETUP THE LIMIT FOR SEARCH
JSR R5,BSFOK ;POINT TO A VALID AREA
TST BSFOK ;FIND THE 'FIELD' AREA?
BNE 21$ ;NO - ASK IF TIME TO MAKE ONE
PRINTF #MSG,#OK ;TELL OPR 'FOUND' FILE
MOV #OK,-(SP)
MOV #MSG,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C$PNTF
ADD #6,SP
BR 4$ ;PROCEED

21$: PRINTF #MSG,#NSWSEC ;TELL OPR NO 'FIELD' FILE
MOV #NSWSEC,-(SP)
MOV #MSG,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP L$PNTF
ADD #6,SP
JSR R5,NEWBSF ;ASK OPR IF TIME TO BUILD A FILE
TST TEMPO ;BUILT A FILE?
BEQ 4$ ;BR IF NO
3$: JSR R5,WRTBSF ;WRITE UPDATED 'FIELD' BAD SECTOR FILE
;ON THE PACK IF REQUESTED

4$: JMP BSMKS ;SELECT THE NEXT UNIT

```

```

1880          .SBTTL  ROUTINE TO LOAD FUNCTION
1881          :CALL   JSR    R5,LDFUNC
1882          :ALL INFORMATION MUST BE SET UP IN DRIVE BUFFER
1883          :R4 HAS POINTER TO BUFFER
1884
1885 023534 013703 002250  LDFUNC: MOV    DCS,R3          ;GET CSR FOR DRIVE
1886 023540 032713 000200  BIT    #BIT7,(R3)       ;CAN WE ISSUE COMMAND?
1887 023544 001004          BNE    1$              ;YES, GO ISSUE COMMAND
1888
1889 023546          ERRSF  200.,PRGER      ;THIS ERROR SHOULD NEVER PRINT
(4) 023546 104454  TRAP   C$ERSF
(5) 023550 000310  .WORD  200
(5) 023552 003735  .WORD  PRGER
(5) 023554 000000  .WORD  0
1890
1891 023556 017763 156474 000002 1$:  MOV    @BBA,BA(R3)      ;LOAD BUS ADDRESS REGISTER
1892 023564 013763 002244 000004  MOV    BDA,DA(R3)      ;LOAD DISK ADDRESS REGISTER
1893 023572 013763 002246 000006  MOV    BMP,MP(R3)      ;LOAD MULTI-PURPOSE REGISTER
1894 023600 013737 002260 002262  MOV    FUNC,BCSADR     ;GET FUNCTION
1895 023606 053737 002302 002262  BIS    DRSEL,BCSADR    ;SET DRIVE SELECT BITS
1896 023614 052737 000201 002262  BIS    #CRDY!DRDY,BCSADR ;SET CRDY & DRDY IN IMAGE
1897 023622 042737 002000 002262  BIC    #OPI,BCSADR     ;WE'RE CLEAR BIT 10 FOR DRIVE 7-4 (OKAY?)
1898 023630 013763 002262 000000  MOV    BCSADR,CS(R3)   ;LOAD CSR
1899 023636 042763 000200 000000  BIC    #CRDY,CS(R3)   ;ISSUE FUNCTION
1900 023644 000205          RTS    R5              ;EXIT
1901
1902          .SBTTL  INTERRUPT SERVICE ROUTINE
1903 023646          BGNSRV !INTR1
1904
1905 023646 042777 000100 156374 INTR1: BIC    #INTEN,@DCS
1906 023654          ENDSRV
(3) 023654          L10016:
(2) 023654 000002          RTI

```

```

1908 .SBTTL BAD SECTOR FILE ROUTINE
1909
1910 023656 STARS
(2) :*****
1911 :ROUTINE TO RECOVER BAD SECTOR FILE AND SAVE IT FOR
1912 :COMPARISON UPON ERROR ON READS/Writes. WE WILL ONLY
1913 :RESERVE SPACE FOR 16 BAD SECTORS PER DRIVE.
1914 :WE WILL ISSUE A DRIVE RESET FIRST, READ HEADER, POSITION
1915 :TO LAST TRACK (CYLINDER 255, SURFACE 1) AND READ IN
1916 :THE FIRST SECTOR FOR FACTORY BAD, AND THE 20TH FOR
1917 :FIELD BAD SECTORS. R4 WILL CONTAIN THE BUFFER POINTER
1918 :TO THE DRIVE WE WANT TO READ
1919
1920 :CALL JSR R5,RDBDSC
1921 023656 STARS
(2) :*****
1922
1923 023656 010046 RDBDSC: MOV R0,-(SP) ;SAVE REGISTERS
1924 023660 010146 MOV R1,-(SP) ;
1925 023662 010246 MOV R2,-(SP) ;
1926 023664 010346 MOV R3,-(SP) ;
1927 023666 004537 025436 21$: JSR R5,ISDRST ;RESET THE DRIVE - GET STATUS AND CLEAR ERROR REG.
1928 023672 012737 000010 002260 MOV #RDHDR,FUNC ;READ HEADER TO FIND POSITION
1929 023700 004537 023534 JSR R5,LDFUNC ;ON DISK
1930 023704 004537 025310 JSR R5,WTRDY
1931 023710 005777 156334 TST @DCS ;ERROR DETECTED?
1932 023714 100016 BPL 22$ ;NO
1933 023716 017737 156326 002252 MOV @DCS,E.DCS ;YES - SAVE THE RLCS STATUS
1934 023724 013703 002250 MOV DCS,R3 ;GET THE BASE ADDRESS FOR RLCS
1935 023730 016337 000004 002440 MOV DA(R3),E.DA ;SAVE THE DISK ADDRESS AT ERROR
1936 023736 ERRHRD 500.,MHDER,ERR2
(4) 023736 104456 TRAP C$ERRHD
(5) 023740 000764 .WORD 500
(5) 023742 003420 .WORD MHDER
(5) 023744 007376 .WORD ERR2
1937 023746 000137 025020 JMP 9$ ;FORCED EXIT
1938
1939 023752 016300 000006 22$: MOV MP(R3),R0 ;GET HEADER AND CALCULATE
1940 023756 022737 000001 002316 CMP #1,TDR ;RL02 TYPE DRIVE?
1941 023764 001005 BNE 23$ ;JUMP IF RL02
1942 023766 043700 002334 BIC CYLSK,R0 ;HERE FOR RL01 - GET CYL ADDRESS (BITS 7-14)
1943 023772 012701 077600 MOV #77600,R1 ;INITIALIZE FOR CYL 255
1944 023776 000404 BR 25$
1945 024000 043700 002340 23$: BIC C$SK,R0 ;HERE FOR RL02
1946 024004 012701 177600 MOV #177600,R1 ;INITIALIZE FOR CYL 510
1947 024010 160001 25$: SUB R0,R1 ;GET DIFFERENCE FROM PRESENT CYL ADDRESS TO CYL 255
1948 024012 010137 002244 MOV R1,BDA ;INITIALIZE DAR WITH DISK ADDRESS DIFFERENCE
1949 024016 052737 000025 002244 BIS #SKHS!SIGN!MK,BDA
1950 024024 012737 000006 002260 MOV #SEEK,FUNC
1951 024032 004537 023534 JSR R5,LDFUNC ;SEEK TO THE BAD SEC FILE CYLINDER
1952 024036 004537 025310 JSR R5,WTRDY ;WAIT FOR DONE
1953 024042 005777 156202 TST @DCS ;ERROR DETECTED?
1954 024046 100016 BPL 26$ ;NO
1955 024050 017737 156174 002252 MOV @DCS,E.DCS ;YES - SAVE THE RLCS STATUS
1956 024056 013703 002250 MOV DCS,R3 ;GET THE BASE ADDRESS FOR RLCS
1957 024062 016337 000004 002440 MOV DA(R3),E.DA ;SAVE THE DISK ADDRESS AT ERROR

```

```

1958 024070          ERRHRD 510.,MHDER,ERR2
      (4) 024070 104456 TRAP   C$ERHRD
      (5) 024072 000776 .WORD 510
      (5) 024074 003420 .WORD MHDER
      (5) 024076 007376 .WORD ERR2
1959 024100 000137 025020 JMP   9$          ;FORCED EXIT
1960
1961 024104 012737 000010 002260 26$: MOV   #RDHDR,FUNC
1962 024112 004537 023534 JSR   R5,LDFUNC ;READ THE HEADER ON THIS CYLINDER
1963 024116 004537 025310 JSR   R5,WTRDY
1964 024122 005777 156122 TST   @DCS      ;ERROR DETECTED?
1965 024126 100016 BPL   27$      ;NO
1966 024130 017737 156114 002252 MOV   @DCS,E.DCS ;YES - SAVE THE ERROR STATUS IN RLCS
1967 024136 013703 002250 MOV   DCS,R3    ;GET THE BASE ADDRESS FOR RLCS
1968 024142 016337 000004 002440 MOV   DA(R3),E.DA ;SAVE THE DISK ADDRESS AT ERROR
1969 024150          ERRHRD 520.,MHDER,ERR2
      (4) 024150 104456 TRAP   C$ERHRD
      (5) 024152 001010 .WORD 520
      (5) 024154 003420 .WORD MHDER
      (5) 024156 007376 .WORD ERR2
1970 024160 000137 025020 JMP   9$          ;FORCED EXIT
1971
1972 024164 016300 000006          27$: MOV   MP(R3),R0 ;NO ERROR - GET THE HEADER WORD
1973 024170 042700 000077 BIC   #77,R0    ;CLEAR THE SECTOR BITS FROM HEADER WORD
1974 024174 022737 000001 002316 CMP   #1,TDR   ;RL01?
1975 024202 001007 BNE   300$     ;NO - JMP FOR RL02
1976 024204 022700 077700 CMP   #77700,R0 ;HERE FOR RL01 - LAST TRACK?
1977 024210 001226 BNE   21$     ;NO - TRY AGAIN
1978 024212 012737 077700 002244 MOV   #77700,BDA ;YES - SET FOR A READ CMD
1979 024220 000406 BR    555$
1980 024222 022700 177700 300$: CMP   #177700,R0 ;HERE FOR RL02 - LAST TRACK (511.)?
1981 024226 001217 BNE   21$     ;NO - TRY AGAIN
1982 024230 012737 177700 002244 MOV   #177700,BDA
1983 024236 012737 177400 002246 555$: MOV   #-256.,BMP ;DO A 1 SECTOR PAIR READ
1984 024244 012737 000014 002260 MOV   #READ,FUNC ;READ DATA FUNCTION
1985
1986 024252 005037 002422 CLR   TEMP3    ;MANUFACTURING/FIELD FILE SWITCH
1987 024256 005037 002414 CLR   TEMPO
1988 024262 005037 002412 CLR   DECNT    ;CLEAR THE COUNT OF ENTRYS DETECTED
1989 024266 013702 002304 MOV   BSECT,R2 ;INITIALIZE LIST TO ALL 1'S
1990 024272 012700 000176 MOV   #126.,R0 ;126 ENTRIES
1991 024276 012722 177777 11$: MOV   #-1,(R2)+ ;INIT ENTRY TO -1
1992 024302 005300 DEC   R0
1993 024304 001374 BNE   11$     ;EXIT IF STORAGE INITED
1994
1995 024306 013702 002304 MOV   BSECT,R2 ;GET LIST TO STORE
1996 024312 012700 000031 MOV   #25.,R0  ;25 ENTRIES
1997 024316 004537 023534 4$: JSR   R5,LDFUNC ;ISSUE THE READ CMD
1998 024322 004537 025310 JSR   R5,WTRDY ;WAIT TILL SECTOR READ
1999
2000 024326 005777 155716 TST   @DCS    ;WAS THE READ GOOD?
2001 024332 100065 BPL   3$      ;YES
2002
2003 024334 004537 025436 JSR   R5,ISDRST ;NO - RESET THE DRIVE
2004 024340 062737 000004 002244 ADD   #4,BDA   ;NEXT SECTOR
2005 024346 005737 002422 TST   TEMP3   ;MANUFACTURING OR FIELD BAD

```

```

2006 024352 001424          BEQ      5$          ;MANUFACTURING
2007 024354 022737 000001 002316    CMP      #1,TDR     ;RL01=1
2008 024362 001024          BNE      400$      ;MUST BE AN RL02
2009 024364 022737 077750 002244    CMP      #77750,BDA ;END OF FACTORY FILE
2010 024372 001351          BNE      4$        ;NO - READ NEXT SECTOR
2011 024374          41$: PRINTF  #FMT18,#SWSEC ;TELL OPR NO 'FIELD' FILE ON PACK
    (8) 024374 012746 003610    MOV      #SWSEC,-(SP)
    (7) 024400 012746 007524    MOV      #FMT18,-(SP)
    (6) 024404 012746 000002    MOV      #2,-(SP)
    (3) 024410 010600          MOV      SP,R0
    (4) 024412 104417          TRAP    (SPNTF)
    (4) 024414 062706 000006    ADD      #6,SP
2012 024420 000137 024766    JMP      7$          ;EXIT
2013
2014 024424 023727 002244 077724  5$:  CMP      BDA,#77724 ;AT END OF MANUFACTURING BAD
2015 024432 000410          BR       55$
2016 024434 022737 177750 002244  400$: CMP      #177750,BDA ;AT END OF FIELD BAD FOR RL02
2017 024442 001325          BNE      4$        ;NO GO BACK FOR NEXT
2018 024444 000753          BR       41$      ;PRINT 'FIELD' ERROR
2019 024446 023727 002244 177724  55$: CMP      BDA,#177724 ;AT END OF MANUFACTURING BAD
2020 024454 001320          BNE      4$        ;NO, GET THE NEXT SECTOR PAIR
2021 024456          56$: PRINTF  #FMT18,#HWSEC ;TELL OPR NO 'FACTORY' FILE ON PACK
    (8) 024456 012746 003532    MOV      #HWSEC,-(SP)
    (7) 024462 012746 007624    MOV      #FMT18,-(SP)
    (6) 024466 012746 000002    MOV      #2,-(SP)
    (3) 024472 010600          MOV      SP,R0
    (4) 024474 104417          TRAP    (SPNTF)
    (4) 024476 062706 000006    ADD      #6,SP
2022 024502 000137 024766    JMP      7$          ;EXIT & HEADS HOME
2023
2024 024506 017701 155544  3$:  MOV      @BBA,R1    ;START OF LIST
2025 024512 005037 015324    CLR      NOSNUM    ;CLEAR THE FOUND SERIAL NUMBER FLAG
2026 024516 005721          TST      (R1)+     ;SEE IF A SERIAL NUMBER PRESENT
2027 024520 001005          BNE      31$      ;YUP - SN WORD 0 >0
2028 024522 005721          TST      (R1)+     ;NO ... SEE IF SN WORD 1 =0
2029 024524 001004          BNF      32$      ;OK - SOME SERIAL NUM PRESENT
2030 024526 005237 015324    JAC      NOSNUM    ;NO - SET THE 'NO SERIAL NUMBER' FLAG
2031 024532 000401          BR       32$
2032 024534 005721          31$: TST      (R1)+     ;SKIP OVER THE 2ND SERIAL NUM WORD
2033 024536 022121          32$: CMP      (R1)+,(R1)+ ;SKIP PAST THE 'BLANK' WORDS
2034 024540 012137 002416  1$:  MOV      (R1)+,TEMP1 ;GET CYLINDER ENTRY WORD
2035 024544 100446          BMI      2$        ;IF MINUS - END OF BAD SECTORS
2036 024546 005237 002412    INC      DECNT     ;COUNT THIS ENTRY IN THE FILE
2037 024552 012137 002420    MOV      (R1)+,TEMP2 ;GET HEAD AND SECTOR
2038 024556 000337 002416    SWAB    TEMP1     ;PUT CYLINDER IN HIGH BYTE
2039 024562 000241          CLC
2040 024564 006037 002416    ROR     TEMP1     ;ALIGN THE BITS
2041 024570 103003          BCC     111$      ;NEED ANOTHER BIT?
2042 024572 052737 100000 002416  111$: BIS      #BIT15,TEMP1 ;YES
2043 024600 013712 002416          MOV      TEMP1,(R2) ;STORE OFF CYLINDER PART
2044 024604 013737 002420 002416    MOV      TEMP2,TEMP1 ;GET SECTOR
2045 024612 042737 177700 002416    BIC      #177700,TEMP1 ;LEAVE ONLY SECTOR
2046 024620 053712 002416          BIS      TEMP1,(R2) ;SET IN SECTOR BITS
2047 024624 042737 177377 002420    BIC      #177377,TEMP2 ;CLEAR ALL EXCEPT HEAD BIT
2048 024632 006237 002420          ASR     TEMP2
2049 024636 006237 002420          ASR     TEMP2
  
```

```

2050 024642 053722 002420          BIS      TEMP2,(R2)+    ;SET IN HEAD
2051 024646 005300          DEC      R0
2052 024650 001333          BNE     1$
2053 024652 005737 002414          TST     TEMPO        ;PRINT A MESSAGE?
2054 024656 001330          BNE     1$           ;NO
2055 024660 000423          BR      6$
2056
2057 024662 005737 002422          2$:    TST     TEMP3        ;SWITCH TO FIELD BAD OR QUI*
2058 024666 001037          BNE     7$           ;QUIT, 7$
2059 024670 022737 000001 002316  CMP     #1,TDR       ;RL01=1
2060 024676 001004          BNE     350$        ;MUST BE RL02
2061 024700 012737 077724 002214  MOV     #77724,BDA   ;START AT FIELD SECTOR
2062 024706 000403          BR      36$
2063 024710 012737 177724 002244 350$:  MOV     #177724,BDA ;START OF FIELD AREA FOR RL02
2064 024716 012737 000001 002422 36$:  MOV     #1,TEMP3
2065 024724 000137 024316          JMP     4$
2066 024730          6$:    PRINTF #FMT18,#TBLFUL
      (8) 024730 012746 006040          MOV     #TBLFUL,-(SP)
      (7) 024734 012746 007624          MOV     #FMT18,-(SP)
      (6) 024740 012746 000002          MOV     #2,-(SP)
      (3) 024744 010600          MOV     SP,R0
      (4) 024746 104417          TRAP   C$PNTF
      (4) 024750 062706 000006          ADD     #6,SP
2067 024754 005237 002414          INC     TEMPO        ;SET THE PRINT FLAG
2068 024760 012700 000170          MOV     #120.,R0    ;RESET THE COUNTER
2069 024764 000665          BR      1$           ;AND CONTINUE
2070
2071 024766 005737 002414          7$:    TST     TEMPO        ;OVER 25. ENTRIES?
2072 024772 001412          BEQ     9$           ;NO
2073 024774          PRINTF #FMT20,DECNT ;PRINT # ENTRIES IN FILE
      (8) 024774 013746 002412          MOV     DECNT,-(SP)
      (7) 025000 012746 007640          MOV     #FMT20,-(SP)
      (6) 025004 012746 000002          MOV     #2,-(SP)
      (3) 025010 010600          MOV     SP,R0
      (4) 025012 104417          TRAP   C$PNTF
      (4) 025014 062706 000006          ADD     #6,SP
2074 025020 012603          9$:    MOV     (SP)+,R3
2075 025022 012602          MOV     (SP)+,R2
2076 025024 012601          MOV     (SP)+,R1
2077 025026 012600          MOV     (SP)+,R0
2078 025030 000205          RTS     R5
2079
2080          ;ROUTINE TO READ THE 'FACTORY' FILE FROM THE BAD SECTOR FILE
2081
2082 025032 005037 002476          RDFACT: CLR     BSFFLG        ;CLEAR BSF FLAG TO DENOTE 'FACTORY' ENTRIES
2083 025036 004537 025222          JSR     R5,CLRBSF    ;CLEAR THE BSFILE STORAGE AREA
2084 025042 012737 177000 002246  MOV     #-512.,BMP   ;SAVE THE WORD COUNT
2085 025050 012737 077700 002244  MOV     #77700,BDA   ;AND THE DISK ADDR FOR FACTORY FILE
2086 025056 022737 000001 002316  CMP     #1,TDR       ;IS IT AN RL02?
2087 025064 001426          BEQ     RDBSFILE     ;NO - READ THE FILE
2088 025066 012737 177700 002244  MOV     #177700,BDA  ;HERE FOR RL02
2089 025074 000422          BR      RDBSFILE     ;THEN READ THEE FILE
2090
2091          ;ROUTINE TO READ THE 'FIELD' FILE FROM THE BAD SECTOR FILE
2092 025076 012737 000001 002476  RDFIELD: MOV     #1,BSFFLG ;MAKE BSF FLAG EQUAL TO 1 TO DENOTE 'FIELD' ENTRIES
2093 025104 004537 025222          JSR     R5,CLRBSF    ;CLEAR THE BSFILE STORAGE AREA

```



```
2094 025110 012737 177000 002246      MOV      #-512.,BMP      ;SAVE THE WORD COUNT
2095 025116 012737 077724 002244      MOV      #77724,BDA    ;AND THE DISK ADDR FOR 'FIELD' FILE
2096 025124 022737 000001 002316      CMP      #1,TDR        ;IS DRIVE A RL02?
2097 025132 001403                BEQ      RDBSFIL        ;NO - READ THE FILE
2098 025134 012737 177724 002244      MOV      #177724,BDA   ;HERE FOR RL02
2099
2100 025142 063737 015342 002244  RDBSFIL: ADD  SECNUM,BDA ;ADD OFFSET TO DAR TO ACCESS APPROPRIATE SECTORS
2101 025150 012737 000014 002260      MOV      #READ,FUNC    ;SAVE THE COMMAND
2102 025156 004537 023534                JSR      R5,LDFUNC      ;AND ISSUE IT
2103 025162 004537 025310                JSR      R5,WTRDY      ;THEN WAIT FOR READY
2104 025166 005777 155056                TST      @DCS          ;WAS THERE ANY ERROR?
2105 025172 100012                BPL      RDBSEX        ;NO - EXIT
2106
2107 025174                PRINTF  #FMT19,#BADBSF ;TELL THE OPR AN ERROR OCCURRED
(8) 025174 012746 005130      MOV      #BADBSF,-(SP)
(7) 025200 012746 007631      MOV      #FMT19,-(SP)
(6) 025204 012746 000002      MOV      #2,-(SP)
(3) 025210 010600      MOV      SP,R0
(4) 025212 104417      TRAP    C$PNTF
(4) 025214 062706 000006      ADD      #6,SP
2108
2109 025220 000205                RDBSEX: RTS      R5      ;EXIT
2110
2111 025222 010146                CLRBSF: MOV      R1,-(SP) ;SAVE R1
2112 025224 012701 030530      MOV      #BSFILE,R1    ;SET UP A POINTER
2113 025230 012721 177777 1$:      MOV      #-1,(R1)+     ;SET BUFFER & POINT TO NEXT
2114 025234 022701 033130      CMP      #BSFILE+1280.,R1 ;DONE?
2115 025240 001373                BNE      1$            ;NO - INIT THE NEXT ADDR
2116 025242 012601      MOV      (SP)+,R1      ;RESET R1
2117 025244 000205                RTS      R5            ;EXIT
2118
2119 025246                STARS
(2) :*****
2120 :LOADED -- CHECK FOR DRV READY
2121 025246                STARS
(2) :*****
2122
2123 025246 010146                LOADED: MOV      R1,-(SP) ;SAVE R1
2124 025250 004537 025422      JSR      R5,GETDST     ;GET DRV STATUS
2125 025254 005037 002414      CLR      TEMPO        ;CLEAR THE FLAG
2126 025260 032701 000020      BIT      #HOP,R1      ;HEADS OVER PACK?
2127 025264 001002                BNE      1$            ;YES
2128 025266 005237 002414      INC      TEMPO        ;NO
2129 025272 032701 000010 1$:      BIT      #BRHM,R1     ;BRUSHES HOME?
2130 025276 001002                BNE      2$            ;YES
2131 025300 005237 002414      INC      TEMPO
2132 025304 012601 2$:      MOV      (SP)+,R1
2133 025306 000205                RTS      R5            ;EXIT
```

2135  
 2136  
 2137  
 2138  
 2139  
 2140  
 2141 025310 010046  
 2142 025312 010146  
 2143 025314 012701 001750  
 2144 025320  
 (3) 025320 012727 000002  
 (3) 025324 000000  
 (3) 025326 013727 002116  
 (3) 025332 000000  
 (3) 025334 005367 177772  
 (3) 025340 001375  
 (3) 025342 005367 177756  
 (3) 025346 001367  
 2145 025350 032777 000200 154672  
 2146 025356 001016  
 2147 025360 005301  
 2148 025362 001356  
 2149 025364 017737 154660 002252  
 2150 025372 013703 002250  
 2151 025376 016337 000004 002440  
 2152 025404  
 (4) 025404 104455  
 (5) 025406 000156  
 (5) 025410 003775  
 (5) 025412 007376  
 2153 025414 012601  
 2154 025416 012600  
 2155 025420 000205  
 2156  
 2157  
 2158  
 2159  
 2160  
 2161  
 2162  
 2163 025422 013703 002250  
 2164 025426 012763 000003 000004  
 2165 025434 000405  
 2166 025436 013703 002250  
 2167 025442 012763 000013 000004  
 2168 025450 012763 000204 000000  
 2169 025456 053763 002302 000000  
 2170 025464 042763 000200 000000  
 2171 025472 004537 025310  
 2172 025476 022763 000013 000004  
 2173 025504 001402  
 2174 025506 016301 000006  
 2175 025512 000205  
 2176

.SBTTL ROUTINE TO WAIT FOR CONTROLLER READY  
 ;ROUTINE TO WAIT FOR CONTROLLER READY UNDER FLAG  
 ;MODE. USED IN INITIALIZE PORTION OF PROGRAM, I.E.,  
 ;GETTING BAD SECTOR FILE, WRITING PACK INITIALLY.

WTRDY: MOV R0, -(SP) ;SAVE REGISTERS  
 MOV R1, -(SP)  
 MOV #1000., R1 ;WAIT A WHILE  
 1\$: WAITUS #2  
 MOV ##2., (PC)+  
 .WORD 0  
 MOV LSDLY, (PC)+  
 .WORD 0  
 DEC -6(PC)  
 BNE -4  
 DEC -22(PC)  
 BNE -20  
 BIT #RDRDY, @DCS ;READY SET?  
 BNE 2\$ ;YES, EXIT  
 DEC R1 ;TIMED OUT?  
 BNE 1\$ ;NO GO BACK  
 MOV @DCS, E.DCS ;SAVE THE STATUS FOR ERROR REPORT  
 MOV DCS, R3 ;GET THE BASE ADDRESS FOR RLCS  
 MOV DA(R3), E.DA ;SAVE THE DISK ADDRESS AT ERROR  
 ERRDF 110., NOCRDY, ERR2  
 TRAP C\$ERDF  
 .WORD 110  
 .WORD NOCRDY  
 .WORD ERR2  
 2\$: MOV (SP)+, R1 ;RESTORE REGISTERS  
 MOV (SP)+, R0  
 RTS R5

.SBTTL GET STATUS/DRIVE RESET ROUTINE  
 ;ROUTINE TO ISSUE DRIVE RESET  
 ;ALSO GET STATUS. R1 HAS STATUS IF GS  
 ;USES R3, DOES NOT SAVE IT

GETDST: MOV DCS, R3 ;GET CSR ADDRESS  
 MOV #GSBIT, DA(R3) ;INITIALIZE DAR FOR GET STATUS COMMAND  
 BR CSTUFF

ISDRST: MOV DCS, R3 ;GET CSR ADDRESS  
 MOV #DRST, DA(R3) ;INIT DAR FOR GET STATUS COMMAND AND CLEAR ERR REG

CSTUFF: MOV #CRDY!GSTAT, CS(R3) ;SET CONTROLLER READY AND GET STATUS FUNCTION  
 BIS DRSEL, CS(R3) ;SELECT THE DRIVE  
 BIC #CRDY, CS(R3) ;PERFORM THE GET STATUS COMMAND  
 JSR R5, WTRDY ;WAIT FOR CONTROLLER READY  
 CMP #DRST, DA(R3) ;RESET THE DRIVE?  
 BEQ 1\$ ;NO - EXIT  
 MOV MP(R3), R1 ;ELSE, GET THE STATUS WORD  
 1\$: RTS R5

```
2178 .SBTTL ROUTINE TO WRITE PACKS INITIALLY
2179
2180 ;ROUTINE TO WRITE PACK WITH PATTERN, ALL TRACKS WILL BE
2181 ;WRITTEN (EXCEPT BAD SECTOR TRACK)
2182 ;FORMAT IS # OF WORDS (WORD 1), PATTERN ADDRESS (WORD 2)
2183 ;PATTERN (WORDS 3 - 128)
2184 ;WE WILL ATTEMPT TO WRITE MULTIPLE SECTORS AT A TIME
2185 ;(10 SECTORS). IF AN ERROR OCCURS WE WILL THEN
2186 ;WRITE INDIVIDUAL SECTORS FOR THAT TRACK. WE DO WRITES,
2187 ;READS AND INCORE COMPARISONS TO VERIFY.
2188
2189 ;
2190 ;CALL JSR R5,WRPACK
2191
2192 025514 010046 WRPACK: MOV R0,-(SP) ;SAVE REGISTERS
2193 025516 010146 MOV R1,-(SP)
2194 025520 010246 MOV R2,-(SP)
2195 025522 010346 MOV R3,-(SP)
2196 025524 010446 MOV R4,-(SP)
2197 025526 013746 002256 MOV BBA,-(SP)
2198 025532 004537 026754 1$: JSR R5,HDHOME ;HEADS HOME
2199 025536 012737 002450 002256 MOV #BUF1,BBA
2200 025544 012737 175400 002246 MOV #-1280.,BMP ;INITIALIZE TO WRITE 10 SECTORS
2201 025552 004537 030412 JSR R5,WRBUF ;GENERATE THE WC DATA PATTERN
2202
2203 ;NOW ACTUALLY WRITE DATA OUT ON PACK, WILL NOT WRITE LAST
2204 ;TRACK
2205
2206 025556 005001 CLR R1 ;R1=CYL 000
2207 025560 005004 CLR R4 ;START AT 1ST CYLINDER
2208 025562 005737 002312 TST FWDFLG ;FORWARD DIRECTION?
2209 025566 001410 BEQ 2$ ;YES
2210 025570 012704 000776 MOV #510.,R4 ;SET FOR THE LAST CYL (RL02)
2211 025574 022737 000001 002316 CMP #1,TDR ;DRIVE = RL01?
2212 025602 001002 BNE 2$ ;NO - DA IS OK
2213 025604 042704 177400 BIC #177400,R4 ;YES - MAX CYL IS 255.
2214 025610 000137 026152 2$: JMP SKWRT ;SEEK TO THE START CYLINDER
2215
2216 025614 022737 000001 002316 CONWR: CMP #1,TDR ;RL01=1
2217 025622 001007 BNE 13$ ;MUST BE AN RL02
2218 025624 022701 077600 CMP #077600,R1 ;RL01 LAST CYLINDER?
2219 025630 001020 BNE STWRT ;NO - PROCEED TO WRITE TRACK
2220 025632 005737 002416 12$: TST TEMP1 ;ON HEAD 1 LAST TRACK?
2221 025636 001415 BEQ STWRT ;NO - WRITE HEAD 0 LAST TRACK
2222 025640 000404 BR ENDWR
2223 025642 022701 177600 13$: CMP #177600,R1 ;LAST CYL FOR RL02?
2224 025646 001011 BNE STWRT ;NO - GO WRITE TRACK
2225 025650 00077J BR 12$ ;YES - TEST FOR LAST TRACK ON LAST CYL
2226
2227 ;HERE WHEN ALL DONE WRITING THE PACK
2228
2229 025652 012637 002256 ENDWR: MOV (SP)+,BBA
2230 025656 012604 MOV (SP)+,R4
2231 025660 012603 MOV (SP)+,R3
2232 025662 012602 MOV (SP)+,R2
2233 025664 012601 MOV (SP)+,R1
```

```

2234 025666 012600          MOV    (SP)+,R0
2235 025670 000205          RTS    R5                ;END EXIT
2236
2237          ;THIS PORTION WILL WRITE THE PACK USING MULTIPLE SECTORS. IF AN
2238          ;ERROR OCCURS WE WILL GO TO 2$ AND INDIVIDUAL SECTORS.
2239          ;IF AFTER 3 RETRYS ON A SECTOR NO RECOVERY CAN BE MADE, THEN THE SECTOR WILL
2240          ;BE MARKED 'BAD' IN THE TEMP BAD SEC FILE STORAGE AREA.
2241
2242 025672 005002          STWRT: CLR    R2                ;INITIAL SECTOR 0
2243 025674 005037 002412          CLR    DECNT                ;INITIALIZE ERROR LOOP COUNTER
2244 025700 010137 002244          SWRT1: MOV   R1,BDA           ;SET UP CYLINDER
2245 025704 053737 002416 002244          BIS    TEMP1,BDA           ;INSERT THE HEAD NUMBER (0 OR 1)
2246 025712 050237 002244          BIS    R2,BDA             ;ADD IN THE SECTOR NUMBER
2247 025716 012737 000012 002260          MOV   #WRITE,FUNC         ;WRITE CMD
2248 025724 004537 023534          JSR   R5,LDFUNC           ;ISSUE THE WRITE
2249 025730 004537 025310          JSR   R5,WTRDY           ;WAIT FOR READY
2250
2251 025734 005777 154310          TST   @DCS                ;ERROR DETECTED?
2252 025740 100041          BPL   WNXSEC              ;BR IF NO ERROR - GET NEXT SECTOR
2253
2254          ;HERE IF AN ERROR WAS DETECTED - GOING TO WRITE THE TRACK ONE SECTOR
2255          ;AT A TIME ... >3 RETRYS = 'BAD' SECTOR
2256
2257 025742          PRINTF #MCRLF
(7) 025742 012746 010041          MOV   #MCRLF,-(SP)
(6) 025746 012746 000001          MOV   #1,-(SP)
(3) 025752 010600          MOV   SP,R0
(4) 025754 104417          TRAP  C$PNTF
(4) 025756 062706 000004          ADD   #4,SP
2258 025762 017737 154262 002252          MOV   @DCS,E.DCS         ;SAVE THE ERROR DETECTED
2259 025770 013703 002250          MOV   DCS,R3             ;GET THE BASE ADDRESS FOR RLCS
2260 025774 016337 000004 002440          MOV   DA(R3),E.DA        ;SAVE THE DISK ADDRESS AT ERROR
2261 026002 005337 002440          DEC   E.DA               ;SECTOR IS PREVIOUS FROM INDICATED
2262 026006 013737 002440 002410          MOV   E.DA,CHKSEC
2263 026014 013737 002410 002502          MOV   CHKSEC,FRSTER     ;STORE ERROR ADDRESS FOR ERROR LOOP
2264 026022          ERRSOFT 410.,MSFER,ERR1
(4) 026022 104457          TRAP  C$ERSOFT
(5) 026024 000632          .WORD 410
(5) 026026 003310          .WORD MSFER
(5) 026030 007130          .WORD ERR1
2265 026032 004537 025436          JSR   R5,ISDRST         ;RESET THE DRIVE
2266 026036 005237 002232          INC   SFTCNT            ;ADD TO SOFT ERROR TALLY
2267 026042 000471          BR    W1SEC             ;WRITE 1 SECTOR AT A TIME
2268
2269          ;HERE TO SELECT THE NEXT SECTOR GROUP ON THIS TRACK
2270
2271 026044 062702 000012          WNXSEC: ADD   #10.,R2        ;NEXT GROUP
2272 026050 022702 000050          CMP   #40.,R2          ;DONE?
2273 026054 001311          BNE   SWRT1            ;NO, GO BACK
2274 026056 005237 002240          INC   WRTCNT           ;COUNT THIS WRITE PASS ON SELECTED TRK
2275 026062 023737 002240 010132          CMP   WRTCNT,WRTLIM    ;AT LIMIT FOR THIS TRACK?
2276 026070 001300          BNE   STWRT           ;NO - DO THIS TRACK AGAIN
2277 026072 005037 002240          CLR   WRTCNT          ;YES - CLEAR THE PASS COUNTER
2278
2279          ;HERE TO SELECT THE NEXT TRACK WITH A SEEK CMD
2280

```

```

2281 026076 005737 010130      WNXTRK: TST      WRTSAW      ;DOING A SAWTOOTH WRITE CYCLE?
2282 026102 001410              BEQ      3$              ;NO - DO INCREMENTAL
2283 026104 005737 002312      TST      FWDFLG      ;SAWTOOTH FWD WRT?
2284 026110 001003              BNE      2$              ;NO - DOING REVERSE WRT
2285
2286 026112 004537 026754      1$:      JSR      R5,HDHOME ;YES - SET THE HEADS OVER CYL #000
2287 026116 000402              BR
2288
2289 026120 004537 027052      2$:      JSR      R5,HDLAST  ;SET THE HEADS OVER THE LAST CYL
2290
2291 026124 005737 002416      3$:      TST      TEMP1      ;DOING HEAD 0 ??
2292 026130 001432              BEQ      5$              ;YES - SET FOR HEAD #1
2293 026132 005737 002312      TST      FWDFLG      ;FWD WRITE?
2294 026136 001404              BEQ      31$             ;YES - R4 IS AN UPCOUNTER?
2295 026140 005304              DEC      R4              ;NO - DOWNCOUNT R4 (CYL COUNTER)
2296 026142 002003              BGE      32$             ;PROCEED IF STILL HAVE SOME TO DO
2297 026144 000137 025652              JMP      ENDWR          ;JUST COMPLETED THE PACK
2298
2299 026150 005204              31$:     INC      R4              ;POINT TO THE NEXT CYLINDER (FWD DIRECTION)
2300
2301              026152
2302 026152 005037 002416      SKWRT=. 32$:     CLR      TEMP1      ;SET POINTER BACK TO HEAD #0
2303
2304 026156 010401              4$:      MOV      R4,R1          ;GET THE CYLINDER #
2305 026160 000301              SWAB     R1              ;POSITION THE BITS FOR DIRECT LOADING
2306 026162 000241              CLC
2307 026164 006001              ROR      R1              ;INTO THE DA REGISTER
2308 026166 103002              BCC      41$            ;FOR THE SEEK TO THE PROPER
2309 026170 052701 100000              BIS      #BIT15,R1      ;CYLINDER
2310 026174 010137 002324      41$:     MOV      R1,NEWPOS      ;SET THE DESIRED DISK ADDRESS
2311 026200 053737 002416 002324      BIS      TEMP1,NEWPOS   ;ADD IN THE SELECTED HEAD BIT
2312 026206 004537 027136      JSR      R5,SKFNC      ;ISSUE THE SEEK TO THE DESIRED CYLINDER/HEAD
2313 026212 000137 025614      JMP      CONWR          ;AND CONTINUE WRITING THE PACK
2314
2315 026216 012737 000100 002416      5$:      MOV      #HEAD,TEMP1   ;POINT TO HEAD #1
2316 026224 000754              BR      4$              ;AND SEEK THERE
2317
2318              ;IF AN ERROR OCCURS THEN WE COME HERE AND DO THE TRACK SECTOR
2319              ;BY SECTOR.
2320
2321 026226 005002              W1SEC:  CLR      R2              ;R2 = SECTOR
2322 026230 012737 177600 002246      MOV      #-128.,BMP     ;LOAD WORD COUNT
2323 026236 013737 002322 002244      1$:      MOV      PRPOS,BDA     ;SETUP DISK ADDRESS
2324 026244 053737 002416 002244      BIS      TEMP1,BDA     ;ADD IN THE HEAD NUMBER (0 OR 1)
2325 026252 050237 002244      BIS      R2,BDA        ;ADD IN THE SECTOR NUMBER
2326
2327              ;HERE TO WRITE A SECTOR
2328
2329 026256 012737 000012 002260      2$:      MOV      #WRITE,FUNC    ;WRITE FUNCTION
2330 026264 004537 023534      JSR      R5,LDFUNC     ;ISSUE THE WRITE
2331 026270 004537 025310      JSR      R5,WTRDY      ;WAIT FOR WRITE TO FINISH
2332
2333 026274 005777 153750              TST      @DCC           ;ERROR ON WRITE?
2334 026300 100114              BPL      3$              ;NO - SETUP FOR NEXT SECTOR
2335
2336              ;HERE IF ERROR ON 1 SECTOR WRITE

```

```

2337
2338 026302 017737 153742 002252      MOV      @DCS,E.DCS      ;SAVE THE DETECTED ERROR
2339 026310 023737 002244 002502      CMP      BDA,FRSTER    ;DID WE REPORT ERROR IN MAIN PROGRAM?
2340 026316 001425                BEQ      10$           ;YES - SKIP
2341 026320 005737 002412                TST      DECNT         ;DID WE REPORT IT ONCE IN ERROR LOOP?
2342 026324 001022                BNE      10$           ;YES - SKIP
2343 026326                PRINTF  #MCRLF          ;ELSE REPORT IT NOW
(7) 026326 012746 010041      MOV      #MCRLF,-(SP)
(6) 026332 012746 000001      MOV      #1,-(SP)
(3) 026336 010600                MOV      SP,R0
(4) 026340 104417                TRAP    C$PNTF
(4) 026342 062706 000004      ADD      #4,SP
2344 026346 013737 002244 002410      MOV      BDA,CHKSEC
2345 026354 013737 002244 002440      MOV      BDA,E.DA
2346 026362                ERRSOFT 430.,MSFER,ERR1
(4) 026362 104457                TRAP    C$ERSOFT
(5) 026364 000656                .WORD  430
(5) 026366 003310                .WORD  MSFER
(5) 026370 007130                .WORD  ERR1
2347 026372 005237 002412                INC      DECNT         ;NO, GIVE IT ONE MORE TRY
2348 026376 013737 002244 002410 10$:      MOV      BDA,CHKSEC    ;CHECK IF SECTOR IS IN
2349 026404 004537 027340                JSR      R5,CKBDSC     ;BAD SECTOR FILE
2350 026410 005737 002406                TST      HDRFND       ;IF SET, IT WAS
2351 026414 001431                BEQ      21$           ;NO MATCH
2352 026416                PRINTF  #FMT18,#INBSF  ;TELL OPR SECT IN FILE ALREADY
(8) 026416 012746 006346      MOV      #INBSF,-(SP)
(7) 026422 012746 007624      MOV      #FMT18,-(SP)
(6) 026426 012746 000002      MOV      #2,-(SP)
(3) 026432 010600                MOV      SP,R0
(4) 026434 104417                TRAP    C$PNTF
(4) 026436 062706 000006      ADD      #6,SP
2353 026442                PRINTF  #MCRLF
(7) 026442 012746 010041      MOV      #MCRLF,-(SP)
(6) 026446 012746 000001      MOV      #1,-(SP)
(3) 026452 010600                MOV      SP,R0
(4) 026454 104417                TRAP    C$PNTF
(4) 026456 062706 000004      ADD      #4,SP
2354 026462 005337 002232      DEC      SFTCNT       ;ADJUST COUNTERS BECAUSE SECTOR-
2355 026466 005037 002412      CLR      DECNT        ;IN-ERROR ALREADY IN BSF
2356 026472 004537 025436      JSR      R5,ISDRST    ;RESET THE DRIVE
2357 026476 000434                BR      31$           ;WORK ON NEXT SECTOR
2358
2359 026500 022737 000004 002412 21$:      CMP      #4,DECNT     ;IT MAY HAVE BEEN NOISE.
2360 026506 001403                BEQ      22$           ;HARD ERROR?
2361 026510 004537 025436      JSR      R5,ISDRST    ;NO - ISSUE A DRIVE RESET
2362 026514 000660                BR      2$            ;AND TRY AGAIN
2363
2364 026516 005337 002232                DEC      SFTCNT       ;DELETE THIS HARD ERROR FROM SOFT ERROR TALLY
2365 026522 004537 026604                JSR      R5,INBAD     ;TELL OPR & PUT IT IN TEMP STORAGE
2366 026526 004537 025436                JSR      R5,ISDRST    ;RESET THE DRIVE
2367 026532 005737 002412                TST      DECNT        ;ANY RECOVERY HERE?
2368 026536 001414                BEQ      31$          ;NO
2369 026540                PRINTF  #FMT18,#MSREC  ;YES - TELL OPR 'RECOVERED'
(8) 026540 012746 003125      MOV      #MSREC,-(SP)
(7) 026544 012746 007624      MOV      #FMT18,-(SP)
(6) 026550 012746 000002      MOV      #2,-(SP)

```

```

(3) 026554 010600      MOV      SP,R0
(4) 026556 104417      TRAP    C$PNTF
(4) 026560 062706 000006  ADD     #6,SP
2370 026564 005037 002412  CLR     DECNT      ;CLEAR LOOP COUNTER FOR NEXT SECTOR
2371
2372      ;SELECT THE NEXT SECTOR
2373
2374 026570 005202      31$:   INC     R2      ;POINT TO THE NEXT SECTOR
2375 026572 020227 000050  CMP     R2,#40.    ;END OF THE TRACK?
2376 026576 002617      BLT     1$         ;NO - DO THIS SECTOR
2377 026600 000137 026076  JMP     WNXTRK     ;YES - DO NEXT TRACK

```

186

CZRLMBO RL01/02 BD SEC FIL TL  
CZRLMB.MAC 12-DEC-79 14:06

MACV11 30A(1052) 17-DEC-79 10:53 PAGE 4-6  
ROUTINE TO WRITE PACKS INITIALLY

1 8

2379 026604  
(2)  
2380  
2381  
2382  
2383 026604  
(2)  
2384  
2385 026604 010146  
2386 026606 016337 000000 002434  
2387 026614 016337 000002 002436  
2388 026622 016337 000004 002440  
2389 026630 016337 000006 002442  
2390 026636 016337 000006 002444  
2391 026644 016337 000006 002446  
2392 026652 005037 002412  
2393 026656 005337 002410  
2394 026662  
(4) 026662 104456  
(5) 026664 000454  
(5) 026666 003420  
(5) 026670 007130  
2395 026672 005237 002230  
2396 026676 012701 030014  
2397 026702 005711  
2398 026704 100417  
2399 026706 005721  
2400 026710 022701 030076  
2401 026714 001372  
2402 026716  
(8) 026716 012746 006040  
(7) 026722 012746 007624  
(6) 026726 012746 000002  
(3) 026732 010600  
(4) 026734 104417  
(4) 026736 062706 000006  
2403 026742 000402  
2404 026744 013711 002410  
2405 026750 012601  
2406 026752 000205

```

STARS
:*****
:INBAD -- ROUTINE TO INSERT THE BAD SECTOR FOUND INTO A TEMP
:          BAD SECTOR FILE AND TELL OPR THAT A BAD SECTOR (HARD ERR)
:          WAS DETECTED.
STARS
:*****
INBAD:  MOV     R1,-(SP)           ;SAVE R1
        MOV     CS(R3),E.CS
        MOV     BA(R3),E.BA
        MOV     DA(R3),E.DA
        MOV     MP(R3),E.MP
        MOV     MP(R3),E.MP1
        MOV     MP(R3),E.MP2
        CLR     DECNT           ;CLEAR CURRENT SOFT ERROR COUNT
        DEC     CHKSEC         ;SECTOR IS PREVIOUS FROM INDICATED
        ERRHRD 300.,MHDER,ERR1
        TRAP   C$ERHRD
        .WORD  300
        .WORD  MHDER
        .WORD  ERR1
        INC     ERRCNT         ;UPDATE THE HARD ERROR COUNT
        MOV     #BSECN,R1     ;POINT TO THE BAD SECTOR TEMP STORAGE
IBDN:   TST     (R1)          ;LOOK FOR A SPOT TO INSERT ENTRY
        BMI    IBDN1         ;BR IF FOUND ONE
        TST     (R1)+        ;POINT TO NEXT ENTRY ADDR
        CMP     #BSECN+50.,R1 ;END OF TABLE?
        BNE    IBDN         ;NO - TRY THIS ENTRY SLOT
        PRINTF #FMT18,#TBLFUL ;YES - TELL OPR END OF TABLE (25. ENTRYS FOUND)
        MOV     #TBLFUL,-(SP)
        MOV     #FMT18,-(SP)
        MOV     #2,-(SP)
        MOV     SP,R0
        TRAP   C$PNTF
        ADD     #6,SP
        BR     IBDN2
IBDN1:  MOV     CHKSEC,(R1)   ;EXIT
IBDN2:  MOV     (SP)+,R1     ;SAVE THE ENTRY IN TABLE
        RTS     R5          ;RESET R1

```



```

2409          .SBTTL  HEADS HOME ROUTINE
2410
2411          ;ROUTINE TO BRING HEADS OVER TRACK 0
2412
2413 026754 010046          HDHOME: MOV    R0,-(SP)          ;SAVE R0
2414 026756 012737 000010 002260  MOV    #RDHDR,FUNC      ;READ HEADER
2415 026764 004537 023534          JSR    R5,LDFUNC        ;GO DO IT.
2416 026770 004537 025310          JSR    R5,WTRDY
2417
2418 026774 016300          MOV    MP(R3),R0        ;GET HEADER
2419 027000 042700          BIC    #177,R0         ;ONLY CYLINDER
2420 027004 010037 002244          MOV    R0,BDA         ;MOVE IT TO BUFFERED DA
2421 027010 052737 000001 002244  BIS    #MK,BDA        ;SET MARKER
2422 027016 012737 000006 002260  MOV    #SEEK,FUNC     ;LOAD SEEK
2423 027024 004537 023534          JSR    R5,LDFUNC        ;SEEK.
2424 027030 004537 025310          JSR    R5,WTRDY        ;WAIT.
2425 027034 013737 002322 002264  MOV    PRPOS,LSTHDR
2426 027042 005037 002322          CLR    PRPOS          ;SET BUFFER TO HOME
2427 027046 012600          MOV    (SP)+,R0
2428 027050 000205          RTS    R5
2429
2430          ;ROUTINE TO SET THE HEADS OVER THE LAST CYLINDER
2431
2432 027052 012737 000010 002260  HDLAST: MOV    #RDHDR,FUNC ;SET TO READ THE CURRENT POSITION
2433 027060 004537 023534          JSR    R5,LDFUNC        ;READ HEADERS
2434 027064 004537 025310          JSR    R5,WTRDY        ;WAIT TILL DONE
2435
2436 027070 016337 000006 002322  MOV    MP(R3),PRPOS   ;GET THE CURRENT POSITION
2437 027076 042737 000177 002322  BIC    #177,PRPOS     ;SAVE ONLY THE CYL BITS
2438 027104 012737 177600 002322  MOV    #177600,NEWPOS ;SET LAST CYL FOR RL02
2439 027112 022737 000001 002316  CMP    #1,TDR         ;DRIVE = RL01?
2440 027120 001003          BNE    1$            ;NO - MUST BE RL02
2441 027122 012737 077600 002324  MOV    #77600,NEWPOS  ;YES - SET RL01 LAST TRACK ADDRESS
2442 027130 004537 027136          JSR    R5,JKFNC       ;SEEK TO THE LAST TRACK
2443 027134 000205          RTS    R5
    
```

```

2445          .SBTTL  SEEK ROUTINE
2446          :ROUTINE TO SEEK TO A CYLINDER POINTED TO BY 'NEWPOS' FROM A CYLINDER
2447          :POINTED TO BY 'PRPOS'
2448          :EXITS WITH PRPOS CONTAINING THE NEW CYLINDER ADDRESS
2449
2450 027136 010146          SKFNC:  MOV    R1,-(SP)          ;SAVE R1
2451 027140 010246          MOV    R2,-(SP)          ;SAVE R2
2452 027142 013702 002324  MOV    NEWPOS,R2        ;SET THE DESIRED CYL
2453 027146 013701 002322  MOV    PRPOS,R1         ;GET THE CURRENT POSITION
2454 027152 042701 000177  BIC    #177,R1          ;CLEAR THE HEAD/SECTOR BITS
2455 027156 042702 000177  BIC    #177,R2
2456 027162 160102          SUB    R1,R2            ;CALC THE DIFFERENCE
2457 027164 103002          BC    1$
2458 027166 005402          NEG    R2                ;MAKE DIFFERENCE A POSITIVE NUMBER
2459 027170 000402          BR    2$
2460 027172 052702 000004  1$:  BIS    #4,R2            ;SET THE DIRECTION BIT
2461 027176 052702 000001  2$:  BIS    #MK,R2          ;SET THE SEEK MARKFR BIT
2462 027202 032737 000100 002324  BIT    #HEAD,NEWPOS     ;GO TO HEAD #1?
2463 027210 001402          BEQ    3$                ;NO
2464 027212 052702 000020  BIS    #SKHS,R2         ;YES - SELECT THE HEAD BIT
2465 027216 010237 002244  3$:  MOV    R2,BDA           ;SAVE THE DA
2466 027222 010237 002272  MOV    R2,DIFWD         ;ALSO AS DIFFERENCE WORD
2467 027226 012737 000006 002260  MOV    #SEEK,FUNC       ;SET TO DO A SEEK FUNCTION
2468 027234 004537 023534  JSR    R5,LDFUNC        ;ISSUE THE SEFK
2469 027240 004537 025310  JSR    R5,WTRDY        ;WAIT TILL READY SET
2470 027244 005777 153000  TST    @DCS             ;SEEK ERROR DETECTED?
2471 027250 100014          BPL    31$               ;NO
2472 027252 017737 152772 002252  MOV    @DCS,E.DCS       ;YES - SAVE THE ERROR STATUS
2473 027260 013703 002250  MOV    DCS,R3           ;GET THE BASE ADDRESS FOR RLCS
2474 027264 016337 000004 002440  MOV    DA(R3),E.DA      ;SAVE THE DISK ADDRESS AT ERROR
2475 027272          ERRHRD 530.,MSKER,ERR2
2476 (4) 027272 104456          TRAP  C$ERRHRD
2477 (5) 027274 001022          .WORD 530
2478 (5) 027276 003275          .WORD MSKER
2479 (5) 027300 007376          .WORD ERR2
2480
2481
2482          2477 027302 013737 002324 002322  31$:  MOV    NEWPOS,PRPOS     ;UPDATE THE CURRENT POSITION WORD
2483          2478 027310 012602          4$:  MOV    (SP)+,R2        ;RESET R2
2484          2479 027312 012601          MOV    (SP)+,R1        ;RESET R1
2485          2480 027314 000205          RTS    R5              ;EXIT
2486
2487          :ROUTINE TO CLEAR ALL DRIVE INFORMATION USED ON START OR
2488          :RESTART IF CALLED. CAN BE USED TO CLEAR INDIVIDUAL DRIVE
2489          :INFORMATION BY BITMAP FOLLOWING CALL.
2490          :CALL JSR R5,CLEAR
2491          :
2492
2493          2489 027316 010446          CLEAR: MOV    R4,-(SP)          ;SAVE R4
2494          2490 027320 012704 002230  MOV    #ERRCNT,R4       ;POINT TO THE 1ST TO CLEAR
2495          2491 027324 005024          2$:  CLR    (R4)+           ;CLEAR
2496          2492 027326 020427 002326  CMP    R4,#RECNTR       ;AT END OF BUFFER
2497          2493 027332 001374          BNE    2$                ;NO, GO TO 2$
2498          2494 027334 012604          4$:  MOV    (SP)+,R4        ;RESTORE CURRENT BUFFER POINTER
2499          2495 027336 000205          RTS    R5              ;EXIT
2500          2496

```

2497  
2498  
2499  
2500  
2501  
2502  
2503 027340 005037 002406  
2504 027344 010046  
2505 027346 010146  
2506 027350 012700 000177  
2507 027354 013701 002304  
2508 027360 022711 177777  
2509 027364 001411  
2510 027366 023711 002410  
2511 027372 001404  
2512 027374 005721  
2513 027376 005300  
2514 027400 001367  
2515 027402 000402  
2516  
2517 027404 005237 002406  
2518  
2519 027410 012601  
2520 027412 012600  
2521 027414 000205  
2522  
2523 027416  
(2)  
2524 027416  
(2)  
2525  
2526  
2527 027416 000176  
2528 030012 177777  
2529  
2530 030014 000176  
2531 030410 177777  
2532 030412  
(2)  
2533 030412  
(2)  
2534  
2535 030412  
(2)  
2536  
2537  
2538 030412  
(2)  
2539  
2540 030412 010146  
2541 030414 010246  
2542 030416 010346  
2543  
2544 030420 013701 002246  
2545 030424 013702 002450  
2546 030430 012703 030462

```

.SBTTL ROUTINE TO CHECK FOR BAD SECTOR
:ROUTINE TO MATCH BAD SECTOR.....BDA IS SECTOR WE ARE LOOKING
:FOR IN LIST POINTED TO BY BSECT.....HDRFND IS SET IF WE FIND IT.
:
CKBDSC: CLR      HDRFND      ;CLEAR FLAG
        MOV      R0,-(SP)    ;SAVE R0
        MOV      R1,-(SP)    ;SAVE R1
        MOV      #127,R0     ;127 ENTRIES
1$:     MOV      BSECT,R1    ;GET WHERE WE'RE LOOKING
2$:     CMP      #-1,(R1)    ;END OF ENTRIES?
        BEQ      4$         ;BRANCH IF AT END
        CMP      CHKSEC,(R1) ;HAVE WE GOT A MATCH
        BEQ      3$         ;THEN GO SET INDICATOR, ELSE
        TST     (R1)+
        DEC     R0
        BNE     2$
        BR      4$

3$:     INC     HDRFND      ;SET FLAG FOUND

4$:     MOV     (SP)+,R1
        MOV     (SP)+,R0
        RTS    R5

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
1$:     MOV     BUF,R2     ;GET THE BUFFER ADDRESS
        MOV     #WCPAT,R3  ;GET THE STARTING ADDRESS OF THE DATA PATTERN

```

2547 030434 012322  
2548 030436 005201  
2549 030440 001404  
2550 030442 022703 030522  
2551 030446 001372  
2552 030450 000767  
2553 030452 012603  
2554 030454 012602  
2555 030456 012601  
2556 030460 000205  
2557  
2558  
2559  
2560  
2561 030462 155555  
2562 030464 066666  
2563 030466 133333  
2564 030470 155555  
2565 030472 066666  
2566 030474 133333  
2567 030476 155555  
2568 030500 066666  
2569 030502 133333  
2570 030504 155555  
2571 030506 066666  
2572 030510 133333  
2573 030512 155555  
2574 030514 066666  
2575 030516 133333  
2576 030520 155555  
2577  
2578 030522 000240  
2579 030524  
(3) 030524  
(3) 030524 104401  
2580 030526 000000  
2581  
2582 030530  
(2)  
2583 030530  
(2)  
2584 030530 002400  
2585  
2586 035530 177777  
2587  
2588 035532  
(2)  
2589 035532  
(2)  
2590  
2591 035532  
2592 035532  
(3) 035532 000011  
2593 035534  
(4) 035534 000031  
(4) 035536 035556

```
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?  
BNE 2$ ;NO - STORE THE NEXT FROM DATA TABLE  
BR 1$ ;YES - RESET THE DATA TABLE POINTER  
3$: MOV (SP)+,R3 ;RESET R3  
MOV (SP)+,R2  
MOV (SP)+,R1  
RTS R5 ;EXIT
```

;WORST CASE PATTERN USED IN WRITING

```
WCPAT: .WORD 155555  
.WORD 066666  
.WORD 133333  
.WORD 155555  
.WORD 066666  
.WORD 133333  
.WORD 155555  
.WORD 066666  
.WORD 133333  
.WORD 155555  
.WORD 066666  
.WORD 133333  
.WORD 155555  
.WORD 066666  
.WORD 133333  
.WORD 155555
```

```
ENDOFPROGRAM: NOP  
ENDTST  
L10015: TRAP C$ETST  
HALT
```

STARS  
:\*\*\*\*\*

```
STARS  
:*****  
BSFILE: .BLKW 1280. ;STORAGE FOR BAD SECTOR FILE DATA  
;/(1280. WORDS - 10 SECTORS - 1/4 TRACK)  
.WORD -1 ;END OF STORAGE
```

STARS  
:\*\*\*\*\*  
STARS  
:\*\*\*\*\*

```
BGNMOD HRDPRM  
BGNHRD  
.WORD L10017-L$HARD/2  
GPRMA CSRMSG,CSR,0,160000,177776,YES  
.WORD T$CODE  
.WORD CSRMSG
```

```
(4) 035540 160000 .WORD T$LOLIM
(4) 035542 177776 .WORD T$HILIM
2594 035544 GPRMD DRMSG,DRBT,0,03400,0,7,YES
(4) 035544 001032 .WORD T$CODE
(4) 035546 035572 .WORD DRMSG
(4) 035550 003400 .WORD 03400
(4) 035552 000000 .WORD T$LOLIM
(4) 035554 000007 .WORD T$HILIM
2595 035556 ENHDRD
(2) .EVEN
(3) 035556 L10017:
2596
2600
2601 035556 052502 020123 042101 CSRMSG: .ASCIZ /BUS ADDRESS/
2602 035572 051104 053111 000105 DRMSG: .ASCIZ /DRIVE/
2603
2607
2608 .EVEN
2609
2610 035600 ENDMOD
2611
2612 035600 BGNMOD SFTPRM
2613 035600 BGNSFT
(3) 035600 000010 .WORD L10020-L$SOFT/2
2614 035602 GPRML DSWRT,0,1,YES
(4) 035602 000130 .WORD T$CODE
(4) 035604 035622 .WORD DSWRT
(4) 035606 000001 .WORD 1
2615 035610 GPRMD DWCNT,2,D,177777,1,177777,YES
(4) 035610 001052 .WORD T$CODE
(4) 035612 035650 .WORD DWCNT
(4) 035614 177777 .WORD 177777
(4) 035616 000001 .WORD T$LOLIM
(4) 035620 177777 .WORD T$HILIM
2616 035622 ENDSFT
(2) .EVEN
(3) 035622 L10020:
2620 .EVEN
2621 035622 040523 052127 047517 DSWRT: .ASCIZ /SAWTOOTH WRITE CYCLE?/
2622 035650 051127 052111 020105 DWCNT: .ASCIZ /WRITE CYCLES PER TRACK?/
2623
2627 035700 ENDMOD
2628 035700 LASTAD
(2) .EVEN
(4) 035700 000000 .WORD 0
(4) 035702 000000 .WORD 0
(3) 035704 L$LAST::
2629
2630 000001 .END
```



















CZRLMBO RL01/02 BD SEC FIL TL  
CZRLMB.MAC 12-DEC-79 14:06

MACY11 30A(1052) 17-DEC-79 10:53 J 9  
PAGE 5-8  
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ

L\$DESP	002076	G	15#				
L\$DEVP	002060	G	15#				
L\$DISP	010136	G	15	40#			
L\$DLY	002116	G	15#	453	480	604	2144
L\$DTP	002040	G	15#				
L\$DTYP	002034	G	15#				
L\$DU	011730	G	15	648#			
L\$DUT	002072	G	15#				
L\$DVTY	002216	G	15	21#			
L\$EF	002052	G	15#				
L\$ENVI	002044	G	15#				
L\$ETP	002102	G	15#				
L\$EXP1	002046	G	15#				
L\$EXP4	002064	G	15#				
L\$EXP5	002066	G	15#				
L\$HARD	035534	G	15	2592#			
L\$HIME	002120	G	15#				
L\$HPCP	002016	G	15#				
L\$HPTP	002022	G	15#				
L\$HW	010122	G	15	390#			
L\$ICP	002104	G	15#				
L\$INIT	010150	G	15	430#			
L\$LADP	002026	G	15#				
L\$LAST	035704	G	15	2628#			
L\$LOAD	002100	G	15#				
L\$LUN	002074	G	15#				
L\$MREV	002050	G	15#				
L\$NAME	002000	G	15#				
L\$PRIO	002042	G	15#				
L\$PROT	010142	G	15	418#			
L\$PRT	002112	G	15#				
L\$REPP	002062	G	15#				
L\$REV	002010	G	15#				
L\$RPT	010140	G	15	411#			
L\$SOFT	035602	G	15	2613#			
L\$SPC	002056	G	15#				
L\$SPCP	002020	G	15#				
L\$SPTP	002024	G	15#				
L\$STA	002030	G	15#				
L\$SW	010130	G	15	396#			
L\$TEST	002114	G	15#				
L\$TIML	002014	G	15#				
L\$UNIT	002012	G	15#	520	579		
L1000	007374		353#				
L10001	007472		362#				
L10002	010126		390	393#			
L10003	010134		396	399#			
L10004	010140		412#				
L10006	011160		558#				
L10007	011642		624#				
L10010	011724		638#				
L10011	011726		644#				
L10012	011730		649#				
L10013	012122		704#				
L10014	012130		712#				
L10015	030524		2579#				









TBLFUL	006040	2595#	2616#	2402										
TDR	002316	296#	2066	789*	812	1197	1291	1532	1940	1974	2007	2059	2086	2096
TEMPO	002414	146#	786*	2439										
		2211	2216	865	870	874	899	900	950	974	976	1000	1001	1027
		182#	861*	1162	1174	1175	1181	1182	1187	1188	1239	1274	1275	1278
		1052	1053	1350	1384	1431	1432	1442	1511	1705	1734	1735	1771	1772
		1337	1349	1798*	1801*	1819	1830	1831	1851	1853	1873	1987*	2053	2067*
		1793	1794	2128*	2131*									
TEMP1	002416	2071	2125*	996	1003*	1038*	1048	1055*	1390*	1391*	1392*	1393	1529*	1537
		183#	986*	1604*	1616*	1629	2034*	2038*	2040*	2042*	2043	2044*	2045*	2046
		1568	1602	2291	2302*	2311	2315*	2324						
		2220	2245	2044	2047*	2048*	2049*	2050						
TEMP2	002420	184#	2037*	2005	2057	2064*								
TEMP3	002422	185#	1986*											
TFMSG	002777	237#	1069											
THARD	003076	240#	1547											
THISDR	006750	312#	1174	1349	1830									
TICK	002424	186#	671											
TILLEN	006114	297#	1000	1052										
TIME	010047	379#	805											
TMMSG	003022	238#	1068											
TRPFLG	002470	204#	582*	588	795*									
TRPHAN	012474	583	630	795#										
TSOFT	003047	239#	1546											
TSARGC=	000002	15#	347#	350#	351#	359#	360#	461#	462#	591#	593#	607#	609#	805#
		811#	812#	820#	862#	867#	872#	876#	898#	903#	905#	906#	907#	908#
		909#	910#	911#	915#	955#	957#	959#	964#	972#	979#	984#	998#	999#
		1011#	1021#	1025#	1031#	1036#	1050#	1051#	1063#	1068#	1069#	1070*	1071#	1093#
		1094#	1102#	1103#	1111#	1155#	1171#	1180#	1186#	1201#	1222#	1229#	1237#	1273#
		1280#	1285#	1318#	1321#	1330#	1346#	1353#	1371#	1382#	1413#	1430#	1452#	1455#
		1460#	1463#	1469#	1472#	1502#	1519#	1544#	1546#	1547#	1548#	1580#	1646#	1655#
		1656#	1672#	1673#	1703#	1733#	1792#	1812#	1827#	1836#	1843#	1849#	1861#	1868#
		1871#	2011#	2021#	2066#	2073#	2107#	2257#	2343#	2352#	2353#	2369#	2402#	
TS CODE=	001052	899#	916#	1000#	1052#	1174#	1181#	1184#	1185#	1187#	1194#	1203#	1204#	1274#
		1349#	1354#	1355#	1356#	1431#	1734#	1771#	1789#	1790#	1793#	1830#	2593#	2594#
		2614#	2615#											
TSERRN=	001022	7#	881#	1587#	1649#	1889#	1936#	1958#	1969#	2152#	2264#	2346#	2394#	2475#
TSEXCP=	000000	899#	916#	1184#	1185#	1194#	1203#	1204#	1354#	1355#	1356#	1789#	1790#	2593#
		2594#	2615#											
TSGMAN=	000000	7#	899#	916#	1184#	1185#	1194#	1203#	1204#	1354#	1355#	1356#	1789#	1790#
TSHILI=	177777	899#	916#	1184#	1185#	1194#	1203#	1204#	1354#	1355#	1356#	1789#	1790#	2593#
		2594#	2615#											
TSLAST=	000001	7#	2628#											
TSLOLI=	000001	899#	916#	1184#	1185#	1194#	1203#	1204#	1354#	1355#	1356#	1789#	1790#	2593#
		2594#	2615#											
TSLSYM=	010000	7#	353	362	393	399	412	558	624	638	644	649	704	712
		1906	2579	2595	2616									
TSLTNO=	000001	2628#												
TSNEST=	177777	7#	14#	16#	27#	86#	118#	216#	221#	321#	327#	329#	353#	355#
		362#	387#	389#	390#	393#	394#	395#	396#	399#	400#	402#	406#	410#
		411#	412#	413#	418#	422#	428#	430#	558#	564#	575#	624#	627#	628#
		638#	640#	642#	643#	644#	645#	647#	648#	649#	650#	661#	704#	707#
		712#	717#	836#	843#	1903#	1906#	2579#	2591#	2592#	2595#	2610#	2612#	2613#
TSNSO =	000000	2616#	2627#											
		14#	16	27#	86	118#	216	221#	321	327#	387	389#	394	395#
		400	402#	406	410#	413	418#	422	428#	564	575#	624	627#	640



VALSN	006007	295#	1187	1793															
VC =	001000	61#	874																
VERIFY	004241	267#	1502																
WCPAT	030462	2546	2550	2561#															
WDE =	100000	55#																	
WGE =	002000	60#																	
WHATCM	013216	851	854	894#															
WL -	020000	57#	870	1283	1449														
WNXSEC	026044	2252	2271#																
WNXTRK	026075	2281#	2377																
WRBUF	030412	2201	2540#																
WRCHK =	000002	69#																	
WRINIT	002346	158#																	
WRIPG	002320	147#	1299*																
WRITE =	000012	73#	1294	2247	2329														
WRACK	025514	1465	1474	2192#															
WRPKF	006674	310#	1460																
WRPKR	006722	311#	1469																
WRTBSF	016770	977	1029	1261	1272#	1419	1757	1854	1875										
WRTCNT	002240	123#	2274*	2275	2277*														
WRTLCK	003460	252#	872	1285	1452														
WRTLIM	010132	398#	2275																
WRTLOK	002462	201#	873*																
WRTSAW	010130	397#	1461	1470	2281														
WTRDY	025310	1302	1572	1612	1634	1930	1952	1963	1998	2103	2141#	2171	2249	2331					
		2416	2424	2434	2469														
W1SEC	026226	2267	2321#																
XSALWA=	000000	7#																	
XSALS=	000040	7#																	
XSOFFS=	000400	7#																	
XSTRUE-	000020	7#																	
.	= 035704	4#	19#	386#	453	480	561#	604	2144	2301	2527#	2530#	2584#						











REDEF	464	502	540												
SETPRI	432	452	454	631	883										
SETVER	450	545	583	630	728	732									
STARS	656	659	720	723	748	750	766	768	799	801	807	809	815	817	824
	827	844	847	890	892	936	942	1151	1153	1266	1270	1325	1328	1423	1428
	1490	1495	1620	1623	1684	1688	1765	1768	1805	1810	1910	1921	2119	2121	2379
	2383	2523	2524	2532	2533	2535	2538	2582	2583	2588	2589				
SVC	5#	7													
WAITMS	93#	453	480												
WAITUS	103#	604	2144												

. ABS. 035704 000

ERRORS DETECTED: 0

.CZRLMB.LST/CRF=SVC33/ML.CZRLMB.MAC  
RUN-TIME: 121 116 10 SECONDS  
RUN-TIME RATIO: 502/248=2.0  
CORE USED: 16K (31 PAGE)