

RL01

**RL01 DRIVE COMPATIBILITY
MD-11-DZRLF-A
TEST**

**EP-DZRLF-A-DL
COPYRIGHT © 1977
FICHE 1 OF 1**

**JAN 1978
digital
MADE IN USA**

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZRLF-A-D
PRODUCT NAME: RLO1 DRIVE COMPATABILITY TEST
DATE CREATED: 11 OCTOBER 1977
MAINTAINER: DIAGNOSTIC ENGINEERING
AUTHOR: D. DEKNIS

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

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

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

COPYRIGHT (C) 1977, DIGITAL EQUIPMENT CORPORATION

1.0 GENERAL INFORMATION

1.1 PROGRAM ABSTRACT

1.1.1 STRUCTURE OF PROGRAM

THIS DIAGNOSTIC OCCUPIES 14.5K WORDS OF MEMORY AND IS COMPATIBLE WITH BOTH XXDP AND ACT. IT CAN BE RUN STANDALONE UNDER XXDP, AND CAN BE CHAINED UNDER XXDP, ACT AND APT IN ACT MODE (SEE "CREATE CORE IMAGE" COMMAND BELOW FOR DETAILS OF CHAINING PROCEDURE). IT IS A SINGLE PROGRAM FROM THE STANDPOINT OF THE DIAGNOSTIC USER, BUT WE HAVE INCORPORATED INTO IT A CONTROL MODULE WHICH WILL LATER BE RELEASED INDEPENDENTLY AS A DIAGNOSTIC SUPERVISOR.

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

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

1.1.2 DIAGNOSTIC INFORMATION

THE RLO1 DRIVE COMPATABILITY TEST IS A PDP-11 (LSI-11) BASED PROGRAM THAT WILL TEST INTERCHANGABILITY OF CARTRIDGES BETWEEN DRIVES. THE TEST PERFORMS WRITES, READS, OVERWRITES, ADJACENT CYLINDER WRITES TO PROVE COMPATABILITY.

1.2 SYSTEM REQUIREMENTS

1.2.1 HARDWARE REQUIREMENTS

PDP-11/LSI-11 PROCESSOR WITH 16K OR MORE OF CORE
 CONSOLE DEVICE (LA30, LA36, VT50, ETC.)
 RL11/RLV11 CONTROLLER(S)
 1 - 4 RLO1 DRIVES
 1 RLO1K CARTRIDGE WITH BAD SECTOR FILE
 KW11P, KW11L (OPTIONAL)
 LINEPRINTER (OPTIONAL)

1.2.2 SOFTWARE REQUIREMENTS

MAINDEC-11-DZRLF-A

1.3 RELATED DOCUMENTS AND STANDARDS

1.4 DIAGNOSTIC HIERARCHY PREREQUISITES

THE RLO1 SUBSYSTEM(S) SHOULD HAVE SUCCESSFULLY RUN THE FOLLOWING PROGRAMS:

MAINDEC-11-DZRLA
MAINDEC-11-DZRLB
MAINDEC-11-DZRLC
MAINDEC-11-DZRLD
MAINDEC-11-DZRLE

1.5 ASSUMPTIONS

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

2.0 OPERATING INSTRUCTIONS

2.1 LOADING AND STARTING PROCEDURES

2.1.1 LOADING PROCEDURES

FOLLOW STANDARD DEC PROCEDURES TO LOAD THE PROGRAM. (XXDP, ABSOLUTE LOADER, UPD1, UPD2)

2.1.2 STARTING PROCEDURES

THE PROGRAM STARTS AT LOCATION 200. USE STANDARD DEC PROCEDURES TO START THE PROGRAM.

2.1.3 STEPS FOR QUICK AND SIMPLE EXECUTION

THE DIAGNOSTIC CAN BE EXECUTED STANDALONE WITHOUT READING THE REMAINDER OF THIS DOCUMENT AS FOLLOWS:

- A) LOAD THE DIAGNOSTIC
- B) START AT ADDRESS 200
- C) ANSWER THE HARDWARE QUESTIONS
- D) RECEIVE PROMPT (DS A>)
- E) ENTER STA<CR>
- F) ANSWER HARDWARE QUESTIONS
- G) GET END OF PASS MESSAGES OR ERROR MESSAGES
- H) TO END EXECUTION, ENTER CONTROL/C

2. 2 SPECIAL ENVIRONMENTS

THE ENVIRONMENTS THIS PROGRAM WILL RUN IN ARE XXDP, ACT, AND SLIDE.

2. 3 PROGRAM OPTIONS

2. 3. 1 START COMMAND

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

2. 3. 1. 1 TESTS SWITCH (/TESTS: <TEST-LIST>)

<TEST-LIST> IS A SEQUENCE OF DECIMAL NUMBERS (1: 2 ETC.) OR RANGES OF DECIMAL NUMBERS (1-5: 8-10 ETC.) SEPARATED BY COLONS, SPECIFYING WHICH TESTS IT IS DESIRED BE EXECUTED. THE TEST 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 ORDER OF SPECIFICATION. THE DEFAULT IS TO EXECUTE ALL TESTS. ON THIS AND ALL SWITCHES, THE ANGLE BRACKETS <> ARE PUNCTUATION USED IN THE DEFINITION ONLY, AND ARE NOT TO BE TYPED BY THE OPERATOR. SEE EXAMPLE AT END OF 2. 3. 1.

2. 3. 1. 2 PASS SWITCH (/PASS: <PASS-CNT>)

<PASS-CNT> IS A DECIMAL NUMBER INDICATING THE DESIRED NUMBER OF PASSES. A PASS IS DEFINED AS THE EXECUTION OF THE FULL DIAGNOSTIC (ALL SELECTED TESTS) AGAINST ALL UNITS SUBMITTED. THE DEFAULT IS NON-ENDING EXECUTION: IE, EXIT IS ACCOMPLISHED EITHER BY TYPING A CONTROL/C OR BY A HALT ON ERROR BEING ENCOUNTERED, IN WHICH CASE WE RETURN TO COMMAND MODE. SEE EXAMPLE AT END OF 2. 3. 1.

2. 3. 1. 3 FLAGS SWITCH (/FLAGS: <FLAG-LIST>)

<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
IDR	INHIBIT DROPPING OF UNITS BY DIAGNOSTIC

THE FLAGS NAMED OR EQUATED TO 1 ARE SET, THOSE EQUATED TO 0

ARE CLEARED. A FLAG NOT SPECIFIED IS CLEARED. IF THE FLAGS SWITCH
IS NOT GIVEN ALL FLAGS ARE CLEARED. SEE EXAMPLE AT END OF 2.3.1.

F 1

SEQ 0005

2.3.1.4 END OF PASS SWITCH (/EOP: <INCR>)

<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. SEE EXAMPLE AT END OF 2.3.1.

2.3.1.5 EFFECT OF COMMAND

THE EFFECT OF THE START COMMAND IS TO INITIATE THE HARDWARE PARAMETER
DIALOGUE, THE SOFTWARE PARAMETER DIALOGUE, AND THEN THE DIAGNOSTIC
TESTS THEMSELVES. (THERE ARE NO SOFTWARE QUESTIONS FOR THIS PROGRAM.)

THE HARDWARE PARAMETER DIALOGUE COMMENCES WITH THE QUESTION
"# UNITS?" TO WHICH THE OPERATOR REPLIES WITH A DECIMAL
NUMBER N FROM 1 TO 64. THE TERM "UNIT" REFERS TO THE DEVICE
TO WHICH THIS SERIES OF DIAGNOSTICS IS DEDICATED. FOLLOWING THIS ARE
THE QUESTIONS WHEREBY THE P-TABLES THEMSELVES WILL BE BUILT. EACH
P-TABLE IS A CORE-RESIDENT TABLE CONTAINING
ALL THE HARDWARE INFORMATION FOR ONE UNIT. THE OPERATOR MUST SUPPLY
N (NUMBER OF UNITS) VALUES FOR EACH QUESTION. HE MAY
DO THIS BY GIVING ONE ANSWER TO EACH QUESTION (IN WHICH CASE THE
SERIES OF QUESTIONS WILL BE POSED N TIMES) OR BY GIVING N VALUES,
SEPARATED BY COMMAS, TO EACH QUESTION (SERIES WILL BE POSED ONCE).
EACH QUESTION IS FOLLOWED BY THE RESPONSE RADIX (D FOR DECIMAL, B
FOR BINARY, O FOR OCTAL, L FOR YES/NO) IN PARENTHESES AND THE DEFAULT
VALUE AFTER THE PARENTHESES.

FOLLOWING THE HARDWARE QUESTIONS ARE THE SOFTWARE QUESTIONS TO
BUILD THE SOFTWARE TABLES, WHICH DEFINE THE MODE (QUICK VERIFY ETC.)
THAT THE DIAGNOSTIC WILL EXECUTE IN.

AT THE POINT WHERE THE QUESTION "# UNITS?" IS ANSWERED,
CORE STORAGE IS ALLOCATED FOR THE P-TABLES, AND IF THERE IS NOT ENOUGH
TO ACCOMMODATE THEM THE MESSAGE "TOO MANY UNITS" IS ISSUED. IN THIS CASE
THE DIAGNOSTIC MUST BE EXECUTED MORE THAN ONCE TO TEST ALL UNITS.

EXAMPLE:

STA/TESTS: 1: 2-4: 6: 8-10/PASS: 3/FLAGS: IER: HOE=1: UAM: LOE

THIS COMMAND WILL CAUSE THREE PASSES TO BE MADE, EACH PASS CONSISTING
OF TESTS 1, 2, 3, 4, 6, 8, 9, AND 10 EXECUTED AGAINST ALL UNITS. THERE IS NO
DIFFERENCE BETWEEN SAYING <FLAG> AND SAYING <FLAG=1>. THE
NOTATION <FLAG=0> IS MEANINGFUL ONLY ON A COMMAND OTHER THAN
START TO CLEAR A FLAG THAT WAS PREVIOUSLY SET. NOTE THAT
ON ALL COMMANDS ONLY THE FIRST THREE LETTERS ARE SCANNED.

2.3.2 RESTART COMMAND

RES(TART)/TESTS: <TEST-LIST>/PASS: <PASS-CNT>/FLAGS: <FLAG-LIST>/UNITS: <UNIT-LIST>

2.3.2.1 TESTS, PASS, AND FLAGS SWITCHES

<TEST-LIST>, <PASS-CNT>, AND <FLAG-LIST> ARE AS IN THE START COMMAND.

2. 3. 2. 2 UNITS SWITCH (/UNITS: <UNIT-LIST>)

<UNIT-LIST> IS A SEQUENCE OF DECIMAL NUMBERS (1,2 ETC.) OR RANGES OF DECIMAL NUMBERS (1-5, 8-10 ETC.) SEPARATED BY COLONS, INDICATING WHICH UNITS IT IS DESIRED BE TESTED. THE NUMBERS MAY RANGE FROM 1 THRU N (N IS THE NUMBER OF UNITS SPECIFIED IN THE PREVIOUS START COMMAND). THE NUMBER INDICATES THE POSITION OF THE P-TABLE AS THE DATA WAS ENTERED DURING THE HARDWARE DIAGLOGUE. THE UNITS WHICH ARE SELECTED MUST NOT HAVE BEEN DROPPED BY THE DROP COMMAND. SEE THE DISCUSSION OF ADD AND DROP COMMANDS BELOW. DEFAULT IS TO TEST ALL UNITS WHICH HAVE NOT BEEN DROPPED BY A DROP COMMAND.

2. 3. 2. 3 EFFECT OF COMMAND

THE RESTART COMMAND DIFFERS FROM THE START COMMAND IN THAT THE P-TABLES FROM THE PREVIOUS START COMMAND (THERE MUST HAVE BEEN ONE) ARE USED, INSTEAD OF NEW ONES BEING BUILT. THE UNITS SWITCH GIVES THE ABILITY TO SELECT A SUBSET OF THESE. THE SOFTWARE DIALOGUE MAY OPTIONALLY BE RE-EXECUTED (OPERATOR WILL BE ASKED). THE COMMAND CAN BE USED AFTER COMMAND MODE HAS BEEN REENTERED IN ANY OF THE THREE NORMAL WAYS: A) THE REQUESTED NUMBER OF PASSES HAVE BEEN MADE B) AN ERROR WAS ENCOUNTERED WITH THE HALT ON ERROR FLAG SET C) A CONTROL/C WAS ENTERED BY THE OPERATOR.

2. 3. 3 CONTINUE COMMAND

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

2. 3. 3. 1 PASS SWITCH (/PASS: <PASS-CNT>)

<PASS-CNT> IS SAME AS IN START COMMAND, BUT THE DEFAULT IS THE UNSATISFIED PASS-CNT FROM THE PREVIOUS START OR RESTART. IF NONE REMAINS, THE DEFAULT IS NON-ENDING EXECUTION.

2. 3. 3. 2 FLAG SWITCH (/FLAGS: <FLAG-LIST>)

<FLAG-LIST> IS SAME AS IN START COMMAND, BUT UNSPECIFIED FLAGS RETAIN THEIR CURRENT VALUE.

2. 3. 3. 3 EFFECT OF COMMAND

THE CONTINUE COMMAND WILL ACT LIKE A RESTART COMMAND IN THIS PARTICULAR PROGRAM.

2. 3. 4 PROCEED COMMAND

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

2. 3. 4. 1 FLAGS SWITCH (/FLAGS: <FLAG-LIST>)

H 1

<FLAG-LIST> IS AS IN THE START COMMAND, BUT UNSPECIFIED FLAGS RETAIN THEIR CURRENT VALUE.

SEQ 0007

2.3.4.2 EFFECT OF COMMAND

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

2.3.5 CREATE CORE IMAGE COMMAND

CC1/TESTS: <TEST-LIST>/PASS: <PASS-CNT>/FLAGS: <FLAG-LIST>

2.3.5.1 TESTS, PASS, AND FLAGS SWITCHES

<TEST-LIST>, <PASS-CNT>, <FLAG-LIST>, AND ARE AS IN THE START COMMAND, EXCEPT THAT THE UAM (UNATTENDED MODE) FLAG DEFAULTS TO THE SET POSITION.

2.3.5.2 EFFECT OF COMMAND

THIS PROGRAM IS NOT CHAINABLE DUE TO THE MANUAL INTERVENTION INVOLVED.

2.3.6 ADD COMMAND

ADD/UNITS: <UNIT-LIST>

2.3.6.1 UNITS SWITCH (/UNITS: <UNIT-LIST>)

<UNIT-LIST> IS AS IN THE RESTART COMMAND.

2.3.6.2 EFFECT OF COMMAND

THE UNITS SPECIFIED ARE ADDED TO THE TEST SEQUENCE. EACH UNIT MUST HAVE A P-TABLE IN MEMORY DUE TO AN EARLIER HARDWARE DIALOGUE. THIS COMMAND MUST BE FOLLOWED BY A RESTART OR CONTINUE. THE UNITS SWITCH MUST BE SPECIFIED. THE ADD COMMAND IS MEANINGFUL ONLY FOR UNITS THAT WERE PREVIOUSLY DROPPED.

2.3.7 DROP COMMAND

DRO(P)/UNITS: <UNIT-LIST>

2.3.7.1 UNITS SWITCH (/UNITS: <UNIT-LIST>)

<UNIT-LIST> IS AS IN THE RESTART COMMAND.

2.3.7.2 EFFECT OF COMMAND

1 1

THE UNITS SPECIFIED WILL BE DROPPED FROM TESTING. THE UNITS WILL BE RESELECTED ONLY BY THE EXECUTION OF AN ADD OR START COMMAND. THE UNITS SWITCH MUST BE ENTERED. THIS COMMAND MUST BE FOLLOWED BY A RESTART OR A CONTINUE COMMAND.

2.3.8 PRINT COMMAND

 PRI(NT)

2.3.8.1 EFFECT OF COMMAND

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

2.3.9 DISPLAY COMMAND

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

2.3.9.1 UNITS SWITCH (/UNITS: <UNIT-LIST>)

<UNIT-LIST> IS AS IN THE RESTART COMMAND.

2.3.9.2 EFFECT OF COMMAND

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.

2.3.10 FLAGS COMMAND

 FLA(GS)

2.3.10.1 EFFECT OF COMMAND

THE CURRENT SETTINGS OF ALL FLAGS ARE PRINTED.

2.3.11 ZFLAGS COMMAND

 ZFL(AGS)

2.3.11.1 EFFECT OF COMMAND

ALL FLAGS ARE CLEARED.

2.3.12 HARDWARE PARAMETERS

J 1
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.

SEQ 0009

RL11 (L) Y?

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

BUS ADDRESS (0) 174400?

ANSWER WITH THE BUS ADDRESS OF THE CONTROLLER.

VECTOR (0) 330?

ANSWER WITH THE INTERRUPT VECTOR OF THE CONTROLLER.

BR LEVEL (0) 5?

ANSWER WITH THE INTERRUPT PRIORITY OF THE CONTROLLER.

DRIVE (0) 0?

ANSWER WITH THE DRIVE(S) CONNECTED TO THE CONTROLLER.

2. 3. 13 SOFTWARE PARAMETERS

THERE ARE NO SOFTWARE PARAMETERS.

2. 3. 14 EXTENDED DISCUSSION OF P-TABLE DIALOGUE

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

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

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

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

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

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

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

NOW LET US SEE HOW WE COULD USE THESE CAPABILITIES TO CONSTRUCT A SET OF P-TABLES. ASSUME THAT WE HAVE 64 UNITS, AND THAT THERE ARE THREE HARDWARE PARAMETERS FOR EACH (THREE SLOTS IN THE P-TABLE, THREE HARDWARE QUESTIONS IN THE DIALOGUE). LET THE DESIRED VALUE FOR THE FIRST PARAMETER BE THE NUMBER 75 FOR ALL 64 TABLES. LET THE DESIRED VALUE FOR THE SECOND PARAMETER BE EQUAL TO THE UNIT NUMBER (1,2,3,...,64) EXCEPT FOR UNIT 50, WHICH SHOULD RECEIVE THE VALUE 49. LET THE DESIRED VALUE FOR THE THIRD PARAMETER BE THE NUMBER 76 FOR THE FIRST 20 UNITS AND THE NUMBER 77 FOR THE LAST 44 UNITS.

THE FOLLOWING DIALOGUE WOULD ACCOMPLISH THIS GOAL:

UNITS (D) ? 64

UNIT 1

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

UNIT 21

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

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

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

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

2.4 EXECUTION TIMES

THE EXECUTION TIME IS VARIABLE, DEPENDENT ON OPERATOR.

3.0 ERROR INFORMATION

ERROR INFORMATION IS COMPLETE IN GIVING ALL INFORMATION NECESSARY. ALL REGISTERS ARE GIVEN AS WELL AS TRACK, SECTOR AND DRIVES INVOLVED IN ERROR.

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

DZRL? XXX ERR YYYYY TST ZZZ SUB PPP PC: RRRRRR

WHERE:

? IS PROGRAM LETTER
XXX IS SFT - SOFT ERROR
HRD - HARD ERROR
DV FAT - DEVICE FATAL ERROR
SYS FAT - SYSTEM FATAL ERROR
YYYYY IS THE ERROR NUMBER
ZZZ IS THE TEST NUMBER
PPP IS THE SUBTEST NUMBER
RRRRR IS THE PROGRAM LISTING LOCATION

ERRORS GIVE THE REGISTER CONTENTS BEFORE AND AFTER THE ERROR ALONG WITH A ONE LINE DESCRIPTION AND RELEVANT DATA.

EXAMPLE:

ONE LINE DESCRIPTION
(OPTIONAL SECOND LINE)
(OPTIONAL THIRD LINE)
BEFORE CS: XXXXXX BA: XXXXXX DA: XXXXXX MP: XXXXXX
AFTER CS: XXXXXX BA: XXXXXX DA: XXXXXX MP: XXXXXX
OTHER PERTINENT INFORMATION IS GIVEN AT THIS TIME.

REGISTER DESCRIPTIONS CAN BE FOUND IN SECTION 5.0.
ERROR DESCRIPTIONS

ERROR READING SECTOR

ERROR WAS ENCOUNTERED WHILE TRYING TO READ VERIFY THE SECTOR AFTER IT WAS WRITTEN BY THE SAME DRIVE.

MINIMUM OF TWO DRIVES REQUIRED

THE PROGRAM REQUIRES AT LEAST TWO DRIVES TO PROVE COMPATIBILITY.

MAXIMUM OF FOUR DRIVES ALLOWED

THE PROGRAM ONLY ALLOWS A MAXIMUM OF FOUR DRIVES.

CAN'T FIND FIVE ADJACENT TRACKS

THE PROGRAM REQUIRES TEN SETS OF FIVE ADJACENT TRACKS AT PREDETERMINED SPOTS ACROSS THE PACK. IT WAS UNABLE TO FIND FIVE COMPLETELY GOOD ADJACENT TRACKS IN THE LIMITS GIVEN.

ERROR WRITING SECTOR

AN ERROR WAS ENCOUNTERED WHILE TRYING TO WRITE THE GIVEN SECTOR.

OVERWRITE ERROR

AN ERROR WAS ENCOUNTERED WHILE TRYING TO READ DATA AFTER AN OVERWRITE BY ONE DRIVE. BOTH DRIVES INVOLVED ARE GIVEN.

READ RECOVERY ERROR

AN ERROR WAS ENCOUNTERED WHILE TRYING TO RECOVER ANOTHER DRIVES DATA.

ADJACENT TRACK TEST

AN ERROR WAS ENCOUNTERED WHILE IN THE ADJACENT TEST PART. A FURTHER DESCRIPTION IS GIVEN.

3.2 ERROR HALTS

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

4.0 PERFORMANCE AND PROGRESS REPORTS

4.1 PERFORMANCE REPORTS

THIS PROGRAM WILL NOT GIVE ANY PERFORMANCE REPORTS.

4.2 PROGRESS REPORTS

THIS PROGRAM WILL NOT GIVE ANY PROGRESS REPORTS.

5.0 DEVICE INFORMATION TABLES

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

RLCS - CONTROL AND STATUS REGISTER (XXXXX0)

BIT 15 - COMPOSITE ERROR
 BIT 14 - DRIVE ERROR
 BIT 13 - NON EXISTANT MEMORY ERROR
 BIT 12 - HEADER NOT FOUND (WITH BIT 10 SET)
 - DATA LATL (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

PLBA - BUS ADDRESS REGISTER (XXXXX2)

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

RLDA - DISK ADDRESS REGISTER (XXXXX4)

FOR READ/WRITE FUNCTIONS

BIT 15 - MUST BE ZERO(0)
 BIT 14-7 - CYLINDER ADDRESS FOR TRANSFER
 BIT 6 - SURFACE FOR TRANSFER
 BIT 5-0 - SECTOR FOR TRANSFER (0-47)

FOR SEEK FUNCTION

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

FOR GET STATUS FUNCTION

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

RLMP - MULTIPURPOSE REGISTER

FOR READ/WRITE FUNCTION

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

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

HAS DRIVE STATUS

BIT 15 - WRITE DATA ERROR
 BIT 14 - CURRENT HEAD ERROR(CHE)
 BIT 13 - WRITE LOCK STATUS(WL)
 BIT 12 - SEEK TIME OUT(SKTO)
 BIT 11 - SPIN ERROR(SPE)
 BIT 10 - WRITE GATE ERROR(WGE)
 BIT 9 - VOLUME CHECK(VC)
 BIT 8 - DRIVE SELECT ERROR(DSE)
 BIT 7 - RESERVED(O)
 BIT 6 - SURFACE
 BIT 5 - COVER OPEN
 BIT 4 - HEADS HOME
 BIT 3 - BRUSHES HOME
 BIT 2-0 - STATE BITS
 0 - LOAD STATE
 1 - SPIN UP
 2 - BRUSH CYCLE
 3 - LOAD HEADS
 4 - SEEK - TRACK COUNTING
 5 - SEEK - LINEAR MODE
 6 - UNLOAD HEADS
 7 - SPIN DOWN

6.0 TEST DESCRIPTION

THE FOLLOWING IS A BREIF DESCRIPTION OF THE WAY THE PROGRAM EXECUTES. THE PROGRAM WILL CHECK COMPATIBILITY BETWEEN 2 - 4 DRIVES USING THE SAME RLO1K CARTRIDGE. THE PROGRAM WILL ASK THE OPERATOR TO SEQUENCE THE PACK BETWEEN THE DRIVES GIVEN IN THE FOLLOWING MANNER.

PLACE PACK IN DRIVE N ON CONTROLLER X AND LOAD
 UNLOAD DRIVE N ON CONTROLLER X
 PLACE PACK IN DRIVE N+1 ON CONTROLLER X AND LOAD
 UNLOAD DRIVE N+1 ON CONTROLLER X
 ETC.....

THE PROGRAM WILL SEQUENCE IN THE ORDER THAT WAS GIVEN IN THE
HARDWARE QUESTIONS. I. E.

DRIVE ? 0,1,2,3

PROGRAM WILL SEQUENCE 0,1,2,3,2,1,0

DRIVE ? 1,0,3,2

PROGRAM WILL SEQUENCE 1,0,3,2,3,0,1

WHEN THE FIRST DRIVE IS LOADED THE PROGRAM WILL ATTEMPT TO
FIND TEN SETS OF FIVE ADJACENT TRACKS AT PREDETERMINED SPOTS
THAT CONTAIN NO BAD SECTORS USING THE BAD SECTOR FILE.
THE 10 SPOTS ARE: ON BOTH SURFACES, INNER, OUTER, MIDDLE,
ONE QUARTER AND THREE QUARTERS. AFTER THIS IS
DONE THE OVERWRITE TEST IS PREPARED (FIRST DRIVE CAN'T OVERWRITE)
AS WELL AS THE ADJACENT TEST.

AS THE PACK IS CYCLED BETWEEN DRIVES THE FOLLOWING CHECKS
ARE MADE:

EACH DRIVE CAN OVERWRITE EACH OTHER DRIVE

EACH DRIVE CAN RECOVER EACH OTHERS DATA

EACH DRIVE CAN WRITE ADJACENT TO EVERY OTHER
DRIVE WITHOUT DISTURBING THE OTHER'S DATA.

READS AND WRITES TAKE PLACE AFTER SEEKS FROM
BOTH DIRECTIONS.

ADJACENT WRITES TAKE PLACE TO BOTH SIDES OF EACH WRITE

TESTS ARE PERFORMED AT ALL TEN SPOTS ACROSS THE PACK.

MAIN. MACY11 30(1046) 06-DEC-77 18:18
DZRLFA.P11 23-NOV-77 13:47

0 2

TABLE OF CONTENTS

SEQ 0016

2831	GLOBAL EQUATES SECTION
2889	GLOBAL DATA SECTION
3058	GLOBAL TEXT SECTION
3098	GLOBAL ERROR REPORT SECTION
3205	INITIALIZATION SECTION
3504	GLOBAL SUBROUTINES SECTION

2806			.ENABLE AMA
2807			.ENABLE ABS
2808			.NLIST ME,CND,MD
2810			
2811	002000		. =2000
2812			
2813	002000		SVC
2814		000000	SVCINS=0
2815		000000	SVCTAG=0
2816			
2817			
2818			
2819	002000		POINTER NONE
2820			
2821			
2822	002000		BGNMOD MDHEDR
2823	002000		HEADER DZRLF,A,0
(5)	002000	104	.ASCII @D@
(5)	002001	132	.ASCII @Z@
(5)	002002	122	.ASCII @R@
(5)	002003	114	.ASCII @L@
(5)	002004	106	.ASCII @F@
(6)	002005	000	.BYTE 0
(6)	002006	000	.BYTE 0
(5)	002007	000	.BYTE 0
(4)	002010	101	.ASCII @AA@
(4)	002011	060	.ASCII @0@
(4)	002012	001	.BYTE C\$REVISION
(3)	002013	006	.BYTE C\$EDIT
(4)	002014	000000	.WORD 0
(4)	002016	000000	.WORD
(4)	002020	000000	.WORD
(4)	002022	000000	.WORD
(4)	002024	000000	.WORD 0
(5)	002026	000000	.WORD 0
(4)	002030	000000	.WORD 0
(4)	002032	000000	.WORD 0
(4)	002034	000000	.WORD 0
(4)	002036	000000	.WORD 0
(4)	002040	022040	.WORD LSDISPATCH
(4)	002042	022042	.WORD LSINIT
(4)	002044	023314	.WORD LSCLEAN
(4)	002046	032012	.WORD LSHARD
(4)	002050	000000	.WORD 0
(4)	002052	002104	.WORD LSDVTYP
(4)	002054	000000	.WORD 0
(4)	002056	022024	.WORD L\$HW
(4)	002060	000000	.WORD 0
(4)	002062	002102	.WORD L\$DR
(4)	002064	002102	.WORD L\$DRST
(4)	002066	000000	.WORD 0
(4)	002070	000000	.WORD 0
(4)	002072	000000	.WORD 0
(4)	002074	000000	.WORD 0
(4)	002076	032134	.WORD L\$LAST
2824			

2825 002100 ENDMOD
2826 002100 DEVREG
(5) 002100 000000 .WORD 0
(2) 002102 000001 .BLKW
2827
2828 002104 DEVTYP <RLO1>
(3) 002104 046122 030460 000 .ASCIZ @RLO1@
(2) 002112 .EVEN

2829
2830
2831 .SBTTL GLOBAL EQUATES SECTION

2832
2833 ;DEFINITIONS

2834
2835
2836 002112 BGNMOD GLBEQAT

2837
2838 002112 EQUALS

(1) ;
(1) ; BIT DEFINITIONS

(1) ;
(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) ;
(1) 001000 BIT9== BIT09

(1) 000400 BIT8== BIT08

(1) 000200 BIT7== BIT07

(1) 000100 BIT6== BIT06

(1) 000040 BIT5== BIT05

(1) 000020 BIT4== BIT04

(1) 000010 BIT3== BIT03

(1) 000004 BIT2== BIT02

(1) 000002 BIT1== BIT01

(1) 000001 BIT0== BIT00

(1) ;

(1) ; EVENT FLAG DEFINITIONS

(1) ; EF32: EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION

(1) ; EF16: EFO1 AVAILABLE FOR PROGRAM USE

(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)      000020      EF16==      16.
(1)      000017      EF15==      15.
(1)      000016      EF14==      14.
(1)      000015      EF13==      13.
(1)      000014      EF12==      12.
(1)      000013      EF11==      11.
(1)      000012      EF10==      10.
(1)      000011      EF09==      9.
(1)      000010      EF08==      8.
(1)      000007      EF07==      7.
(1)      000006      EF06==      6.
(1)      000005      EF05==      5.
(1)      000004      EF04==      4.
(1)      000003      EF03==      3.
(1)      000002      EF02==      2.
(1)      000001      EF01==      1.
(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

2839
2840      000000      CS=0      ; CONTROL AND STATUS OFFSET
2841      000002      BA=2      ; BUSADDRESS OFFSET
2842      000004      DA=4      ; DISK ADDRESS OFFSET
2843      000006      MP=6      ; MULTI PURPOSE OFFSET
2844      ; CONSTANT OFFSETS FOR INDIVIDUAL DRIVE BUFFERS
2845
2846      000000      CSR=0      ; CONTROLLER ADDRESS
2847      000002      VEC=2      ; VECTOR OF CONTROLLER
2848      000004      DSB=4      ; DRIVE SELECT
2849      000006      PAT=6      ; PATTERN UNIQUE TO DRIVE
2850
2851
2852      000001      DRDY=BIT0      ; DRIVE READY
2853      000100      INTEN=BIT6      ; INTERRUPT ENABLE
2854      100000      ERR=BIT15      ; COMPOSITE ERROR
2855      040000      DERR=BIT14      ; DRIVE ERROR
2856      020000      NXM=BIT13      ; NON-EXISTANT MEMORY ERROR
2857      010000      DLT=BIT12      ; DATA LATE
2858      004000      DCRC=BIT11      ; DATA CRC ERROR
2859      004000      HCRC=BIT11      ; HEADER CRC ERROR
2860      010000      HNF=BIT12      ; HEADER NOT FOUND ERROR
2861      002000      OPI=BIT10      ; OPERATION INCOMPLETE ERROR
2862      000200      CRDY=BIT7      ; CONTROLLER READY
2863      000040      BA17=BIT5      ; EXTENDED BUS ADDRESS BIT 17
  
```

2864	000020	BA16=BIT4	; EXTENDED BUS ADDRESS BIT 16
2865	000002	CRSET=BIT1	; CONTROLLER RESET FUNCTION CODE
2866	000004	GSTAT=BIT2	; GET DRIVE STATUS FUNCTION CODE
2867	000006	SEEK=BIT1!BIT2	; SEEK FUNCTION CODE
2868	000010	RDHDR=BIT3	; READ HEADER FUNCTION CODE
2869	000012	WRITE=BIT3!BIT1	; WRITE FUNCTION CODE
2870	000014	READ=BIT3!BIT2	; READ FUNCTION CODE
2871	000013	DRST=BIT3!BIT1!BIT0	; DRIVE RESET COMMAND CODE FOR DRIVE COMMAND WORD
2872	000003	GSBIT=BIT1!BIT0	; GET STATUS COMMAND CODE FOR DRIVE COMMAND WORD
2873	000001	MK=BIT0	; MARKER BIT FOR DRIVE COMMAND WORD(SEEK,GET STATUS)
2874	000004	SIGN=BIT2	; DIRECTION FOR SEEK(0=AWAY FROM SPINDLE)
2875	000020	SKHS=BIT4	; HEAD SELECT FOR SEEK
2876	000100	HEAD=BIT6	; HEAD SELECT FOR READ,WRITE,GET STATUS
2877			
2878			; OFFSET FOR HARDWARE P-TABLE
2879			
2880	000000	CSR=0	
2881	000002	VECT=2	
2882	000004	PRIOR=4	
2883	000006	DRBT=6	
2884	000010	RLCNT=10	
2885			
2886			
2887			
2888	002112		

ENDMOD
 SBTTL GLOBAL DATA SECTION

2894	002112	BGNMOD	GLBDAT	
2895				
2896	002112	000000	HDRFND:	. WORD 0 ; 1=HEADER IN BAD SECTOR LIST
2897	002114	000	OUT10:	. BYTE 0
2898	002115	000	OUT20:	. BYTE 0
2899	002116	000	OUT30:	. BYTE 0
2900	002117	000	OUT40:	. BYTE 0
2901	002120	000	OUT50:	. BYTE 0
2902	002121	000	OUT11:	. BYTE 0
2903	002122	000	OUT21:	. BYTE 0
2904	002123	000	OUT31:	. BYTE 0
2905	002124	000	OUT41:	. BYTE 0
2906	002125	000	OUT51:	. BYTE 0
2907	002126	000	OQU10:	. BYTE 0
2908	002127	000	OQU20:	. BYTE 0
2909	002130	000	OQU30:	. BYTE 0
2910	002131	000	OQU40:	. BYTE 0
2911	002132	000	OQU50:	. BYTE 0
2912	002133	000	OQU11:	. BYTE 0
2913	002134	000	OQU21:	. BYTE 0
2914	002135	000	OQU31:	. BYTE 0
2915	002136	000	OQU41:	. BYTE 0
2916	002137	000	OQU51:	. BYTE 0
2917	002140	000	MID10:	. BYTE 0
2918	002141	000	MID20:	. BYTE 0
2919	002142	000	MID30:	. BYTE 0

2920	002143	000	MID40:	BYTE	0
2921	002144	000	MID50:	BYTE	0
2922	002145	000	MID11:	BYTE	0
2923	002146	000	MID21:	BYTE	0
2924	002147	000	MID31:	BYTE	0
2925	002150	000	MID41:	BYTE	0
2926	002151	000	MID51:	BYTE	0
2927	002152	000	TQU10:	BYTE	0
2928	002153	000	TQU20:	BYTE	0
2929	002154	000	TQU30:	BYTE	0
2930	002155	000	TQU40:	BYTE	0
2931	002156	000	TQU50:	BYTE	0
2932	002157	000	TQU11:	BYTE	0
2933	002160	000	TQU21:	BYTE	0
2934	002161	000	TQU31:	BYTE	0
2935	002162	000	TQU41:	BYTE	0
2936	002163	000	TQU51:	BYTE	0
2937	002164	000	INN10:	BYTE	0
2938	002165	000	INN20:	BYTE	0
2939	002166	000	INN30:	BYTE	0
2940	002167	000	INN40:	BYTE	0
2941	002170	000	INN50:	BYTE	0
2942	002171	000	INN11:	BYTE	0
2943	002172	000	INN21:	BYTE	0
2944	002173	000	INN31:	BYTE	0
2945	002174	000	INN41:	BYTE	0
2946	002175	000	INN51:	BYTE	0
2947			EVEN		
2948					
2949			;SECTOR LIST FOR LAST DRIVE WRITTEN		
2950			;MAP OF 16 SECTOR DRIVE BITS		
2951					
2952	002176	000020	SECLST:	BLKW	16.
2953					
2954			;BUFFER TABLE FOR 24 X 5 MATRIX USED FOR ADJACENT CYLINDER TESTING.		
2955					
2956	002236	000170	SECBUF:	BLKW	5*24.
2957					
2958					
2959			;LIST OF TRACKS USED TO OVERWRITE TEST.		
2960			;FIRST FIVE ARE BYTE ADDRESSES OF TOP SURFACE.		
2961			;LAST FIVE ARE BYTE ADDRESSES OF BOTTOM SURFACE		
2962					
2963	002616	002116	OVWTRK:	OUT30	
2964	002620	002130		OQU30	
2965	002622	002142		MID30	
2966	002624	002154		TQU30	
2967	002626	002166		INN30	
2968	002630	002123		OUT31	
2969	002632	002135		OQU31	
2970	002634	002147		MID31	
2971	002636	002161		TQU31	
2972	002640	002173		INN31	
2973					
2974	002642	152525	PATLST:	WORD	152525
2975	002644	133333		WORD	133333

GLOBAL DATA SECTION

2976	002646	066666	.WORD	066666	
2977	002650	155555	.WORD	155555	
2978					
2979	002652	000000	FOWR: .WORD	0	
2980	002654	000000	FADJ: .WORD	0	
2981	002656	000000	TEMP: .WORD	0	
2982	002660	000000	LSTCLR: .WORD	0	; LAST CONTROLLER
2983	002662	000000	REASON: .WORD	0	; DRIVE ERROR REASON
2984	002664	000000	ERFLG: .WORD	0	; ERROR FLAG
2985	002666	000000	STFLG: .WORD	0	; PROGRAM START UP FLAG
2986	002670	000000	ADJLOC: .WORD	0	; TRACK INDEX FOR ADJ. CYL TEST
2987	002672	000000	ADJFLG: .WORD	0	; FLAG FOR ADJ. STORE OR RETRIEVE
2988	002674	000000	ADJDIR: .WORD	0	; ADJACENT SEEK DIRECTION
2989	002676	000000	DRSTAT: .WORD	0	
2990	002700	000000	HSFLG: .WORD	0	
2991	002702	000000	OSECT: .WORD	0	
2992	002704	000000	HEAD01: .WORD	0	; SURFACE FLAG
2993	002706	000000	DIRC: .WORD	0	; DIRECTION OF SEEK
2994	002710	000000	DESCYL: .WORD	0	; PACK SURFACE
2995	002712	000000	REVSK: .WORD	0	; REVERSE SEEK
2996	002714	000000	FORSK: .WORD	0	; FORWARD SEEK
2997	002716	000000	UUT: .WORD	0	; UNIT UNDER TEST
2998	002720	000000	SECT: .WORD	0	; SECTOR
2999	002722	000000	LSTDRV: .WORD	0	; LAST DRIVE
3000	002724	000000	GDATA: .WORD	0	; GOOD DATA
3001	002726	000000	BDATA: .WORD	0	; BAD DATA
3002	002730	000000	WCOUNT: .WORD	0	; WORD COUNT
3003	002732	000000	SEWRD: .WORD	0	; SECTOR WORD
3004	002734	000000	OFFSET: .WORD	0	; INCREMENT
3005	002736	000000	LSTTRK: .WORD	0	; LAST TRACK OF SEARCH
3006	002740	000000	FRTTRK: .WORD	0	; FIRST TRACK OF SEARCH
3007	002742	000000	PRSTRK: .WORD	0	; PRESENT TRACK
3008	002744	000000	SURFACE: .WORD	0	; SURFACE
3009	002746	000000	TRKFND: .WORD	0	; TRACK FOUND
3010	002750	000000	TRKCNT: .WORD	0	; TRACK COUNT
3011	002752	000000	E. CS: .WORD	0	; IMAGE OF CSR
3012	002754	000000	E. BA: .WORD	0	; IMAGE OF BUS ADDRESS
3013	002756	000000	E. DA: .WORD	0	; IMAGE OF DISK ADDRESS
3014	002760	000000	E. MP: .WORD	0	; IMAGE OF MULTI-PURPOSE WORD 1
3015	002762	000000	E. MP1: .WORD	0	; " " " " " 2
3016	002764	000000	E. MP2: .WORD	0	; " " " " " 3
3017	002766	000000	BCS: .WORD	0	; COMMAND LOADED
3018	002770	000000	BBA: .WORD	0	; BUS ADDRESS LOADED
3019	002772	000000	BDA: .WORD	0	; DISK ADDRESS LOADED
3020	002774	000000	BMP: .WORD	0	; WORD COUNT LOADED
3021	002776	000000	SERNM1: .WORD	0	; SERIAL NUMBER OF CARTRIDGE
3022	003000	000000	SERNM2: .WORD	0	; " " " "
3023	003002	000000	ADJTRK: .WORD	0	; INSIDE/OUTSIDE FLAG
3024	003004	000000	ADJUUT: .WORD	0	; UUT FOR "ADJCYL"
3025	003006	000000	ADJLC2: .WORD	0	; TEMP LOC FOR "ADJCYL"
3026	003010	000000	ADJLC3: .WORD	0	; " " " "
3027	003012	000000	ADJLC4: .WORD	0	; " " " "
3028	003014	000000	STSEC1: .WORD	0	; SECTORS TO WRITE "ADJCYL"
3029	003016	000000	STSEC: .WORD	0	; " " " "
3030	003020	006000	BUF: .BLKW	3072.	; BUFFER FOR 24 SECTOR READS
3031					

```

3032 017020          DRBUF:          ;DRIVE INFORMATION BUFFERS
3033
3037
3045
(1) 017020 000000          CSR          ;CONTROLLER ADDRESS
(1) 017022 000002          VEC          ;VECTOR
(1) 017024 000004          DSB          ;DRIVE SELECT BITS
(1) 017026 000006          PAT          ;PATTERN UNIQUE TO DRIVE
(1)
(1)
(1) 017030 000000          CSR          ;CONTROLLER ADDRESS
(1) 017032 000002          VEC          ;VECTOR
(1) 017034 000004          DSB          ;DRIVE SELECT BITS
(1) 017036 000006          PAT          ;PATTERN UNIQUE TO DRIVE
(1)
(1)
(1) 017040 000000          CSR          ;CONTROLLER ADDRESS
(1) 017042 000002          VEC          ;VECTOR
(1) 017044 000004          DSB          ;DRIVE SELECT BITS
(1) 017046 000006          PAT          ;PATTERN UNIQUE TO DRIVE
(1)
(1)
(1) 017050 000000          CSR          ;CONTROLLER ADDRESS
(1) 017052 000002          VEC          ;VECTOR
(1) 017054 000004          DSB          ;DRIVE SELECT BITS
(1) 017056 000006          PAT          ;PATTERN UNIQUE TO DRIVE
(1)
3046
3050 017060 000000          ENDBUF: .WORD 0          ;END OF DRIVE BUFFERS
3051
3052
3053 017062          ENDMOD
3054
3055
3056
3057
3058          .SBTTL GLOBAL TEXT SECTION
3059 017062          BGNMOD GLBTXT
3060
3061          ;GLOBAL TEXT
3062
3063
3067
3068 017062 047103 046124 020122 CNTTOT: .ASCIZ /CNTLR TIMED OUT/
3069 017102 051105 047522 020122 INITWR: .ASCIZ /ERROR ON RECOVERING INITIAL WRITE BY FIRST DRIVE /
3070 017164 051105 047522 020122 DCKER: .ASCIZ /ERROR ON READ/
3071 017202 044515 044516 052515 FEW: .ASCIZ /MINIMUM OF TWO DRIVES REQUIRED/
3072 017241 115 054101 046511 MANY: .ASCIZ /MAXIMUM OF FOUR DRIVES ALLOWED/
3073 017300 042524 052123 040440 NONE: .ASCIZ /TEST ABORTED - CAN'T FIND ANY GOOD SPOTS/
3074 017351 124 054522 047111 OVMES: .ASCIZ /TRYING TO OVERWRITE DRIVE /
3075 017404 051124 044531 043516 RECMS: .ASCIZ /TRYING TO READ DATA WRITTEN BY DRIVE /
3076 017452 040503 023516 020124 ERRFND: .ASCIZ /CAN'T FIND FIVE ADJACENT TRACKS/
3077 017512 053117 051105 051127 OVWER: .ASCIZ /OVERWRITE ERROR/
3078 017532 042522 042101 051040 RECER: .ASCIZ /READ RECOVERY ERROR/
3079 017556 051105 047522 020122 FUNERR: .ASCIZ /ERROR IN SEEK OPERATION/
3080 017606 044515 020123 042523 SKER: .ASCIZ /MIS SEEK ERROR/
  
```

3081	017625	106	051117	040527	FWD:	.ASCIZ	/FORWARD/
3082	017635	122	053105	051105	REV:	.ASCIZ	/REVERSE/
3083	017645	105	051122	051117	WRIT1:	.ASCIZ	/ERROR WRITING SECTOR/
3084	017672	051105	047522	020122	READ1:	.ASCIZ	/ERROR READING SECTOR/
3085	017717	101	045104	041501	ADJTXT:	.ASCIZ	/ADJACENT CYLINDER TEST/
3086							
3087							
3088							
3089							
3093							
3094					.EVEN		
3095							
3096	017746				ENDMOD		
3097							
3098					.SBTTL	GLOBAL ERROR REPORT SECTION	
3099	017746				BGNMOD	GLBERR	
3100							
3101	017746				BGNMSG	ERR1	
3102							
3103	017746				PRINTB	#FRM10, FRTTRK, LSTTRK, SURFACE	
(10)	017746	013746	002744		MOV	SURFACE, -(SP)	
(9)	017752	013746	002736		MOV	LSTTRK, -(SP)	
(8)	017756	013746	002740		MOV	FRTTRK, -(SP)	
(7)	017762	012746	021260		MOV	#FRM10, -(SP)	
(6)	017766	012746	000004		MOV	#4, -(SP)	
(3)	017772	010600			MOV	SP, R0	
(4)	017774	104014			EMT	C\$PNTB	
(4)	017776	062706	000012		ADD	#12, SP	
3104							
3105	020002				ENDMSG		
(3)	020002				L10000:		
(3)	020002	104023			EMT	C\$MSG	
3106							
3107	020004				BGNMSG	ERR2	
3108	020004				PRINTB	#FRM4, CSR(R4), <B, DSB+1(R4)>	
(9)	020004	005046			CLR	-(SP)	
(9)	020006	156416	000005		BISB	DSB+1(R4), (SP)	
(8)	020012	016446	000000		MOV	CSR(R4), -(SP)	
(7)	020016	012746	020761		MOV	#FRM4, -(SP)	
(6)	020022	012746	000003		MOV	#3, -(SP)	
(3)	020026	010600			MOV	SP, R0	
(4)	020030	104014			EMT	C\$PNTB	
(4)	020032	062706	000010		ADD	#10, SP	
3109	020036	004737	025134		JSR	PC, REGDMP ; REGISTER DUMP ROUTINE	
3110	020042				ENDMSG		
(3)	020042				L10001:		
(3)	020042	104023			EMT	C\$MSG	
3111							
3112	020044				BGNMSG	ERR3	
3113	020044				PRINTB	#FRM4, CSR(R4), <B, DSB+1(R4)>	
(9)	020044	005046			CLR	-(SP)	
(9)	020046	156416	000005		BISB	DSB+1(R4), (SP)	
(8)	020052	016446	000000		MOV	CSR(R4), -(SP)	
(7)	020056	012746	020761		MOV	#FRM4, -(SP)	
(6)	020062	012746	000003		MOV	#3, -(SP)	
(3)	020066	010600			MOV	SP, R0	

(4)	020070	104014		EMT	C\$PNTB
(4)	020072	062706	000010	ADD	#10, SP
3114	020076	004737	025134	JSR	PC, REGDMP ; REGISTER DUMP ROUTINE
3115	020102			PRINTB	#FRM5, <B, DESCYL+1>, <B, DESCYL>, SECT
(10)	020102	013746	002720	MOV	SECT, -(SP)
(9)	020106	005046		CLR	-(SP)
(9)	020110	153716	002710	BISB	DESCYL, (SP)
(8)	020114	005046		CLR	-(SP)
(8)	020116	153716	002711	BISB	DESCYL+1, (SP)
(7)	020122	012746	021022	MOV	#FRM5, -(SP)
(6)	020126	012746	000004	MOV	#4, -(SP)
(3)	020132	010600		MOV	SP, R0
(4)	020134	104014		EMT	C\$PNTB
(4)	020136	062706	000012	ADD	#12, SP
3116	020142			PRINTB	#FRM16, CSR(R3), <B, DSB+1(R3)>
(9)	020142	005046		CLR	-(SP)
(9)	020144	156316	000005	BISB	DSB+1(R3), (SP)
(8)	020150	016346	000000	MOV	CSR(R3), -(SP)
(7)	020154	012746	021611	MOV	#FRM16, -(SP)
(6)	020160	012746	000003	MOV	#3, -(SP)
(3)	020164	010600		MOV	SP, R0
(4)	020166	104014		EMT	C\$PNTB
(4)	020170	062706	000010	ADD	#10, SP
3117					
3118	020174			ENDMSG	
(3)	020174				
(3)	020174	104023		L10002: EMT	C\$MSG
3119					
3120	020176			BGNMSG	ERR4
3121					
3122	020176			PRINTB	#FRM4, CSR(R4), <B, DSB+1(R4)>
(9)	020176	005046		CLR	-(SP)
(9)	020200	156416	000005	BISB	DSB+1(R4), (SP)
(8)	020204	016446	000000	MOV	CSR(R4), -(SP)
(7)	020210	012746	020761	MOV	#FRM4, -(SP)
(6)	020214	012746	000003	MOV	#3, -(SP)
(3)	020220	010600		MOV	SP, R0
(4)	020222	104014		EMT	C\$PNTB
(4)	020224	062706	000010	ADD	#10, SP
3123	020230	004737	025134	JSR	PC, REGDMP ; REGISTER DUMP ROUTINE
3124	020234			PRINTB	#FRM5, <B, DESCYL+1>, <B, DESCYL>, SECT
(10)	020234	013746	002720	MOV	SECT, -(SP)
(9)	020240	005046		CLR	-(SP)
(9)	020242	153716	002710	BISB	DESCYL, (SP)
(8)	020246	005046		CLR	-(SP)
(8)	020250	153716	002711	BISB	DESCYL+1, (SP)
(7)	020254	012746	021022	MOV	#FRM5, -(SP)
(6)	020260	012746	000004	MOV	#4, -(SP)
(3)	020264	010600		MOV	SP, R0
(4)	020266	104014		EMT	C\$PNTB
(4)	020270	062706	000012	ADD	#12, SP
3125	020274			PRINTB	#FRM6, REASON, LSTDRV, LSTCLR, LSTDRV
(11)	020274	013746	002722	MOV	LSTDRV, -(SP)
(10)	020300	013746	002660	MOV	LSTCLR, -(SP)
(9)	020304	013746	002722	MOV	LSTDRV, -(SP)
(8)	020310	013746	002662	MOV	REASON, -(SP)

(7)	020314	012746	021071	MOV	#FRM6, -(SP)
(6)	020320	012746	000005	MOV	#5, -(SP)
(3)	020324	010600		MOV	SP, R0
(4)	020326	104014		EMT	C\$PNTB
(4)	020330	062706	000014	ADD	#14, SP
3126	020334			PRINTB	#FRM7, DIRC
(8)	020334	013746	002706	MOV	DIRC, -(SP)
(7)	020340	012746	021112	MOV	#FRM7, -(SP)
(6)	020344	012746	000002	MOV	#2, -(SP)
(3)	020350	010600		MOV	SP, R0
(4)	020352	104014		EMT	C\$PNTB
(4)	020354	062706	000006	ADD	#6, SP
3127					
3128	020360			ENDMSG	
(3)	020360			L10003:	
(3)	020360	104023		EMT	C\$MSG
3129					
3130	020362			BGNMSG	ERR5
3131	020362			PRINTB	#FRM4, CSR(R4), <B, DSB+1(R4)>
(9)	020362	005046		CLR	-(SP)
(9)	020364	156416	000005	BISB	DSB+1(R4), (SP)
(8)	020370	016446	000000	MOV	CSR(R4), -(SP)
(7)	020374	012746	020761	MOV	#FRM4, -(SP)
(6)	020400	012746	000003	MOV	#3, -(SP)
(3)	020404	010600		MOV	SP, R0
(4)	020406	104014		EMT	C\$PNTB
(4)	020410	062706	000010	ADD	#10, SP
3132	020414	004737	025134	JSR	PC, REGDMP
3133	020420			ENDMSG	
(3)	020420			L10004:	
(3)	020420	104023		EMT	C\$MSG
3134					
3135	020422			BGNMSG	ERR6
3136	020422			PRINTB	#FRM4, CSR(R4), <B, DSB+1(R4)>
(9)	020422	005046		CLR	-(SP)
(9)	020424	156416	000005	BISB	DSB+1(R4), (SP)
(8)	020430	016446	000000	MOV	CSR(R4), -(SP)
(7)	020434	012746	020761	MOV	#FRM4, -(SP)
(6)	020440	012746	000003	MOV	#3, -(SP)
(3)	020444	010600		MOV	SP, R0
(4)	020446	104014		EMT	C\$PNTB
(4)	020450	062706	000010	ADD	#10, SP
3137	020454	004737	025134	JSR	PC, REGDMP
3138	020460			PRINTB	#FRM17, R1, E. MP
(9)	020460	013746	002760	MOV	E. MP, -(SP)
(8)	020464	010146		MOV	R1, -(SP)
(7)	020466	012746	021676	MOV	#FRM17, -(SP)
(6)	020472	012746	000003	MOV	#3, -(SP)
(3)	020476	010600		MOV	SP, R0
(4)	020500	104014		EMT	C\$PNTB
(4)	020502	062706	000010	ADD	#10, SP
3139	020506			ENDMSG	
(3)	020506			L10005:	
(3)	020506	104023		EMT	C\$MSG
3140					
3141					

```

3142
3143          ;FORMAT STATEMENTS
3144
3148
3149 020510 047045 040445 047125 FRM1: .ASCIZ /%N%AUNLOAD DRIVE %01%A ON CONTROLLER %06%A AND REMOVE PACK%/
3150 020605      045 022516 050101 FRM2: .ASCIZ /%N%APLACE PACK IN DRIVE %01%A ON CONTROLLER %06%A AND LOAD%/
3151 020702 047045 040445 051127 FRM3: .ASCIZ !%N%A WRONG PACK # IS %05%05%A # S/B %05%05%A!
3152 020761      045 041501 047117 FRM4: .ASCIZ /%A CONTROLLER: %06%A DRIVE: %01%A/
3153 021022 040445 042510 042101 FRM5: .ASCIZ /%A HEAD: %01%A CYL: %Z3%A SECTOR: %Z2%A/
3154 021071      045 022524 030517 FRM6: .ASCIZ /%T%01%A ON %06%A/
3155 021112 040445 042523 045505 FRM7: .ASCIZ /%A SEEK DIRECTION: %T%A DATA: %N%/
3156 021152 040445 047527 042122 FRM8: .ASCIZ !%A WORD: %Z3%A S/B: %06%A WAS: %06%A!
3157 021216 042045 022463 020101 FRM9: .ASCIZ /%D3%A WORDS BAD OUT OF 128 READ%/
3158 021260 040445 042502 053524 FRM10: .ASCIZ /%A BETWEEN %Z3%A - %Z3%A HEAD: %01%A/
3159 021324 047045 040445 053520 FRM11: .ASCIZ /%N%A PWR FAIL NOT SUPPORTED%/
3160 021361      045 041101 043105 FRM12: .ASCIZ /%A BEFORE CS: %06%A BA: %06%A DA: %06%A MP: %06%/
3161 021440 047045 040445 043101 FRM13: .ASCIZ /%N%A AFTER CS: %06%A BA: %06%A DA: %06%A MP: %06%A/
3162 021523      045 022516 020101 FRM14: .ASCIZ /%N%A DRIVE STATUS: %06%/
3163 021552 047045 040445 040503 FRM15: .ASCIZ /%N%A CAN'T FIND BAD SECTOR FILE/
3164 021611      045 040501 045104 FRM16: .ASCIZ /%A ADJACENT WRITTEN BY CONTROLLER: %06%A DRIVE: %01%A/
3165 021676 040445 054105 023520 FRM17: .ASCIZ /%A EXP'D: %06%A REC'D: %06%A/
3166 021732 047045 040445 047125 FRM18: .ASCIZ /%N%A UNLOAD ALL DRIVES TO BE USED%/
3167 021775      045 022516 020101 ENDPAS: .ASCIZ /%N%A END OF PASS%/
3168
3169
3173
3174
3175
3176
3177
3178          . EVEN
3179 022022          ENDMOD
3180
3181 022022          BGNMOD HPTCODE
3182
3183 022022          BGNHW
3184 (3) 022022 000005          . WORD L10006-L5HW/2
3185 022024 174400          . WORD 174400
3186 022026 000330          . WORD 330
3187 022030 000240          . WORD 240
3188 022032 000000          . WORD 0
3189 022034 000001          . WORD 1
3190
3191 022036          ENDHW
3192 (3) 022036          L10006:
3193 022036          ENDMOD
3194
3195
3196 022036          BGNMOD DSPCODE
3197
3198 022036          DISPATCH 1
3199 (4) 022036 000001          . WORD 1
      (6) 022040 031046          . WORD T1
  
```

```

3200 022042          ENDMOD
3201
3202
3203
3204
3205          .SBTTL  INITIALIZATION SECTION
3206 022042          BGNMOD  INITCODE
3207
3208 022042          BGNINIT
3209
3210 022042          SETPRI  #340
(3) 022042 012700 000340  MOV    #340,R0
(3) 022046 104041          EMT    C$SPRI
3211
3212 022050 023727 002014 000002  CMP    LSUNIT,#2          ;MORE THAN TWO
3213 022056 002005          BGE    90$                ;YES, OKAY
3214
3215 022060          ERRSF   19,FEW
(3) 022060 104421          TRAP  T$ERCODE
(5) 022062 000023          .WORD 19
(5) 022064 017202          .WORD FEW
3216 022066 000137 023270          JMP    CMPENA            ;CLEAN CODE WHEN < 2 DRIVES
3217
3218 022072 023727 002014 000004 90$:  CMP    LSUNIT,#4          ;MORE THAN FOUR
3219 022100 003405          BLE    91$                ;NO, OKAY
3220
3221 022102          ERRSF   20,MANY
(3) 022102 104421          TRAP  T$ERCODE
(5) 022104 000024          .WORD 20
(5) 022106 017241          .WORD MANY
3222 022110 000137 023270          JMP    CMPENA            ;CLEAN CODE WHEN > 4 DRIVES
3223
3224 022114 013737 002014 002716 91$:  MOV    LSUNIT,UUT        ;GET NUMBER OF UNITS
3225 022122 005001          CLR    R1                ;INIT P-TABLE
3226 022124 012704 017020          MOV    #DRBUF,R4        ;SET UP DRIVE BUFFER
3227 022130 012702 002642          MOV    #PATLST,R2       ;GET LIST OF PATTERNS
3228 022134 005737 002716          1$:  TST    UUT                ;ANY P-TABLES LEFT?
3229 022140 001422          BEQ    END                ;NO, GO TO END
3230 022142          GPHARD  R1,R0            ;GET A P-TABLE
(3) 022142 010100          MOV    R1,R0
(3) 022144 104042          EMT    C$GPHRD
3231 022146 012064 000000          MOV    (R0)+,CSR(R4)    ;GET CSR
3232 022152 012064 000002          MOV    (R0)+,VEC(R4)    ;GET VECTOR
3233 022156 005720          TST    (R0)+            ;SKIP PAST BR
3234 022160 011064 000004          MOV    (R0),DSB(R4)     ;GET DRIVE
3235 022164 011264 000006          MOV    (R2),PAT(R4)
3236 022170 005722          TST    (R2)+
3237 022172 005201          INC    R1                ;NEXT P TABLE
3238 022174 005337 002716          DEC    UUT                ;NEXT DRIVE
3239 022200 062704 000010          ADD    #PAT+2,R4
3240 022204 000753          BR     1$
3241 022206 013737 002014 002716  END:  MOV    LSUNIT,UUT        ;GET BEGINNING OF BUFFER
3242 022214 012704 017020          MOV    #DRBUF,R4
3243 022220 005037 002654          CLR    FADJ              ;CLEAR ADJ. TEST FLAG
3244 022224 005037 002652          CLR    FOWR              ;CLEAR OVERWRITE FLAG
3245 022230          READEF #EF.PWR
  
```

```

(3) 022230 012700 000034      MOV    #EF.PWR,RO
(3) 022234 104050              EMT    C$REFG
3246 022236                    BNCOMLETE  SETUP
(2) 022236 103010              BCC    SETUP
3247 022240                    PRINTF #FRM11
(7) 022240 012746 021324      MOV    #FRM11,-(SP)
(6) 022244 012746 000001      MOV    #1,-(SP)
(3) 022250 010600              MOV    SP,RO
(4) 022252 104017              EMT    C$PNTF
(4) 022254 062706 000004      ADD    #4,SP
3248
3249
3250
3251
3252
3253
3254
3255
3256
3257
3258
3259
3260
3261
3262
3263 022260 005237 002666      SETUP: INC    STFLG           ;INDICATE A START COMMAND
3264 022264 012737 177777 002776  MOV    #-1,SERNM1
3265 022272 012737 177777 003000  MOV    #-1,SERNM2
3266 022300                    PRINTF #FRM18
(7) 022300 012746 021732      MOV    #FRM18,-(SP)
(6) 022304 012746 000001      MOV    #1,-(SP)
(3) 022310 010600              MOV    SP,RO
(4) 022312 104017              EMT    C$PNTF
(4) 022314 062706 000004      ADD    #4,SP
3267 022320 004537 030450      JSR    R5,LOAD           ; TELL OPERATOR TO LOAD
3268 022324 004537 030020      JSR    R5,SERNUM        ; GET SERIAL NUMBER
3269 022330 004537 027364      JSR    R5,MERGE         ; MERGE BAD SECTOR FILES
3270 022334 012701 002114      MOV    #OUT10,R1        ; INITIALIZE ALL TRACKS
3271 022340 012700 000031      MOV    #25,RO
3272 022344 012721 177777      15:  MOV    #177777,(R1)+
3273 022350 005300              DEC    RO
3274 022352 001374              BNE    15
3275
3276 022354 004537 027552      JSR    R5,FNDTRK        ; TRY TO FIND FIVE TRACKS
3277 022360 000001              1
3278 022362 000000              0
3279 022364 000 020          .BYTE 0,16             ; INWARD SEARCH
3280
3281 022366 005737 002746      .TOP SURFACE
3282 022372 001005              .TRACK RANGE
3283
3284 022374                    TST    TRKFND           ; WAS SEARCH SUCCESSFUL????
(3) 022374 104463              BNE    25              ; YES
(5) 022376 000012
(5) 022400 017452
(5) 022402 017746
ERRHRD 10,ERRFND,ERR1 ;NO TRACKS
TRAP   TSERCODE
WORD  10
WORD  ERRFND
WORD  ERR1

```

```
3285 022404 000404 BR 35
3286
3287 022406 012700 002114 25: MOV #OUT10,RO ; STORE AWAY TRACKS FOUND
3288 022412 004537 027764 JSR R5, FIXCYL
3289
3290 022416 004537 027552 35: JSR R5, FNDTRK ; TRY TO FIND FIVE TRACKS
3291 022422 000001 1 ; INWARD SEARCH
3292 022424 000001 1 ; BOTTOM SURFACE
3293 022426 000 020 . BYTE 0, 16. ; TRACK RANGE
3294
3295 022430 005737 002746 TST TRKFND ; WAS SEARCH SUCCESSFUL???.
3296 022434 001005 BNE 45 ; YES
3297
3298 022436 ERRHRD 10, ERRFND, ERR1 ; NO TRACKS
(3) 022436 104463 TRAP TSERCODE
(5) 022440 000012 . WORD 10
(5) 022442 017452 . WORD ERRFND
(5) 022444 017746 . WORD ERR1
3299 022446 000404 BR 55
3300
3301 022450 012700 002121 45: MOV #OUT11,RO ; STORE TRACKS AWAY
3302 022454 004537 027764 JSR R5, FIXCYL
3303 022460 004537 027552 55: JSR R5, FNDTRK ; FIND NEXT 5 TRACK
3304 022464 177777 -1 ; OUTWARD SEARCH
3305 022466 000000 0 ; TOP SURFACE
3306 022470 376 356 . BYTE 254, 238. ; TRACK RANGE
3307
3308 022472 005737 002746 TST TRKFND ; WAS SEARCH SUCCESSFUL?
3309 022476 001005 BNE 65 ; YES
3310
3311 022500 ERRHRD 10, ERRFND, ERR1 ; NO TRACKS
(3) 022500 104463 TRAP TSERCODE
(5) 022502 000012 . WORD 10
(5) 022504 017452 . WORD ERRFND
(5) 022506 017746 . WORD ERR1
3312 022510 000404 BR 75 ; SKIP
3313
3314 022512 012700 002164 65: MOV #INN10,RO ; STORE AWAY TRACKS FOUND
3315 022516 004537 027764 JSR R5, FIXCYL
3316
3317 022522 004537 027552 75: JSR R5, FNDTRK ; NEXT SET
3318 022526 177777 -1 ; OUTWARD SEARCH
3319 022530 000001 1 ; BOTTOM SURFACE
3320 022532 376 356 . BYTE 254, 238. ; TRACK RANGE
3321
3322 022534 005737 002746 TST TRKFND ; SEARCH SUCCESSFUL?
3323 022540 001005 BNE 85 ; YES
3324
3325 022542 ERRHRD 10, ERRFND, ERR1 ; NO TRACKS
(3) 022542 104463 TRAP TSERCODE
(5) 022544 000012 . WORD 10
(5) 022546 017452 . WORD ERRFND
(5) 022550 017746 . WORD ERR1
3326 022552 000404 BR 95
3327
3328 022554 012700 002171 85: MOV #INN11,RO ; STORE AWAY TRACKS FOUND
```

INITIALIZATION SECTION

3329	022560	004537	027764		JSR	R5, FIXCYL	
3330							
3331	022564	004537	027552	95:	JSR	R5, FNDTRK	; NEXT SET
3332	022570	000001			1		; INWARD SEARCH
3333	022572	000000			0		; TOP SURFACE
3334	022574	176	210		. BYTE	126. , 136.	; TRACK RANGE
3335							
3336	022576	005737	002746		TST	TRKFND	; DID WE FIND A SET
3337	022602	001015			BNE	105	; YES
3338							
3339	022604	004537	027552		JSR	R5, FNDTRK	; NEXT SET (OTHER SIDE)
3340	022610	177777			-1		; OUTWARD SEARCH
3341	022612	000000			0		; TOP SURFACE
3342	022614	202	170		. BYTE	130. , 120.	; TRACK RANGE
3343							
3344	022616	005737	002746		TST	TRKFND	; DID WE FIND A SET
3345	022622	001005			BNE	105	; YES
3346							
3347	022624				ERRHRD	10. , ERRFND, ERR1	; NO TRACKS
(3)	022624	104463			TRAP	T\$ERCODE	
(5)	022626	000012			. WORD	10	
(5)	022630	017452			. WORD	ERRFND	
(5)	022632	017746			. WORD	ERR1	
3348	022634	000404			BR	115	
3349							
3350	022636	012700	002140	105:	MOV	#MID10, R0	; STORE AWAY
3351	022642	004537	027764		JSR	R5, FIXCYL	
3352							
3353	022646	004537	027552	115:	JSR	R5, FNDTRK	; NEXT SET
3354	022652	000001			1		; INWARD SEARCH
3355	022654	000001			1		; BOTTOM SURFACE
3356	022656	176	210		. BYTE	126. , 136.	; RANGE
3357							
3358	022660	005737	002746		TST	TRKFND	; SUCCESS?
3359	022664	001015			BNE	125	; YES
3360							
3361	022666	004537	027552		JSR	R5, FNDTRK	; LOOK THE OTHER SIDE
3362	022672	177777			-1		; OUTWARD
3363	022674	000001			1		; BOTTOM SURFACE
3364	022676	202	170		. BYTE	130. , 120	; RANGE
3365							
3366	022700	005737	002746		TST	TRKFND	; SUCCESS?
3367	022704	001005			BNE	125	; YES
3368							
3369	022706				ERRHRD	10. , ERRFND, ERR1	; NO TRACKS
(3)	022706	104463			TRAP	T\$ERCODE	
(5)	022710	000012			. WORD	10	
(5)	022712	017452			. WORD	ERRFND	
(5)	022714	017746			. WORD	ERR1	
3370	022716	000404			BR	135	
3371							
3372	022720	012700	002145	125:	MOV	#MID11, R0	; STORE AWAY THE TRACKS FOUND
3373	022724	004537	027764		JSR	R5, FIXCYL	
3374							
3375	022730	004537	027552	135:	JSR	R5, FNDTRK	; NEXT SET
3376	022734	000001			1		; INWARD

3377	022736	000000			0		; TOP SURFACE
3378	022740	076	110		. BYTE	62. , 72.	; RANGE
3379							
3380	022742	005737	002746		TST	TRKFND	; SUCCESS?
3381	022746	001015			BNE	145	; YES
3382							
3383	022750	004537	027552		JSR	R5, FNDTRK	; LOOK OTHER SIDE
3384	022754	177777			-1		; OUTWARD
3385	022756	000000			0		; TOP SURFACE
3386	022760	102	070		. BYTE	66. , 56.	; RANGE
3387							
3388	022762	005737	002746		TST	TRKFND	; SUCCESS?
3389	022766	001005			BNE	145	; YES
3390							
3391	022770				ERRHRD	10. , ERRFND, ERR1	; NO TRACKS
(3)	022770	104463			TRAP	TSERCODE	
(5)	022772	000012			. WORD	10	
(5)	022774	017452			. WORD	ERRFND	
(5)	022776	017746			. WORD	ERR1	
3392	023000	000404			BR	155	
3393							
3394	023002	012700	002126	145:	MOV	#0QU10, R0	; STORE AWAY NEXT SET
3395	023006	004537	027764		JSR	R5, FIXCYL	
3396							
3397	023012	004537	027552	155:	JSR	R5, FNDTRK	; LOOK FOR NEXT SET
3398	023016	000001			1		; INWARD
3399	023020	000001			1		; BOTTOM
3400	023022	076	110		. BYTE	62. , 72.	; RANGE
3401							
3402	023024	005737	002746		TST	TRKFND	; SUCCESS?
3403	023030	001015			BNE	165	; YES
3404							
3405	023032	004537	027552		JSR	R5, FNDTRK	; LOOK FOR ANOTHER SET
3406	023036	177777			-1		; OUTWARD
3407	023040	000001			1		; BOTTOM
3408	023042	102	070		. BYTE	66. , 56.	; RANGE
3409							
3410	023044	005737	002746		TST	TRKFND	; SUCCESS?
3411	023050	001005			BNE	165	; YES
3412							
3413	023052				ERRHRD	10. , ERRFND, ERR1	; NO TRACKS
(3)	023052	104463			TRAP	TSERCODE	
(5)	023054	000012			. WORD	10	
(5)	023056	017452			. WORD	ERRFND	
(5)	023060	017746			. WORD	ERR1	
3414	023062	000404			BR	175	
3415							
3416	023064	012700	002133	165:	MOV	#0QU11, R0	; STORE AWAY TRACKS
3417	023070	004537	027764		JSR	R5, FIXCYL	
3418							
3419	023074	004537	027552	175:	JSR	R5, FNDTRK	; NEXT SET OF TRACKS
3420	023100	000001			1		; INWARD
3421	023102	000000			0		; TOP SURFACE
3422	023104	276	310		. BYTE	190. , 200	; RANGE
3423							
3424	023106	005737	002746		TST	TRKFND	; SUCCESS?

3425	023112	001015		BNE	185		: YES
3426							
3427	023114	004537	027552	JSR	R5, FNDTRK		: LOOK OTHER SIDE
3428	023120	177777		-1			: OUTWARD SEARCH
3429	023122	000000		0			: TOP
3430	023124	302	270	. BYTE	194. , 184.		: RANGE
3431							
3432	023126	005737	002746	TST	TRKFND		: SUCCESS
3433	023132	001005		BNE	185		: YES
3434							
3435	023134			ERRHRD	10. , ERRFND, ERR1		: NO TRACKS
(3)	023134	104463		TRAP	T\$ERCODE		
(5)	023136	000012		. WORD	10		
(5)	023140	017452		. WORD	ERRFND		
(5)	023142	017746		. WORD	ERR1		
3436	023144	000404		BR	195		
3437							
3438	023146	012700	002152	185:	MOV	#TQU10, R0	: STORE TRACKS AWAY
3439	023152	004537	027764		JSR	R5, FIXCYL	
3440							
3441	023156	004537	027552	195:	JSR	R5, FNDTRK	: NEXT SET
3442	023162	000001		1			: INWARD
3443	023164	000001		1			: BOTTOM SURFACE
3444	023166	276	310	. BYTE	190. , 200.		: RANGE
3445							
3446	023170	005737	002746	TST	TRKFND		: SUCCESS?
3447	023174	001015		BNE	205		: YES
3448							
3449	023176	004537	027552	JSR	R5, FNDTRK		: OTHER SET
3450	023202	177777		-1			: OUTWARD
3451	023204	000001		1			: BOTTOM SURFACE
3452	023206	302	270	. BYTE	194. , 184.		: RANGE
3453							
3454	023210	005737	002746	TST	TRKFND		: SUCCESS
3455	023214	001005		BNE	205		: YES
3456							
3457	023216			ERRHRD	10. , ERRFND, ERR1		: NO TRACKS
(3)	023216	104463		TRAP	T\$ERCODE		
(5)	023220	000012		. WORD	10		
(5)	023222	017452		. WORD	ERRFND		
(5)	023224	017746		. WORD	ERR1		
3458	023226	000404		BR	215		
3459							
3460	023230	012700	002157	205:	MOV	#TQU11, R0	: STORE SET AWAY
3461	023234	004537	027764		JSR	R5, FIXCYL	
3462							
3463	023240	012700	002114	215:	MOV	#JUT10, R0	: DID WE FIND ANY AT ALL
3464	023244	012701	000062		MOV	#50. , R1	
3465	023250	122720	000377	225:	CMPB	#377, (R0)+	
3466	023254	001016		BNE	EXIT		
3467	023256	005301		DEC	R1		
3468	023260	001373		BNE	225		
3469	023262			ERRSF	3. , NONE		
(3)	023262	104421		TRAP	T\$ERCODE		
(5)	023264	000003		. WORD	3		
(5)	023266	017300		. WORD	NONE		

INITIALIZATION SECTION

```
3470 023270 005001      CMPENA: CLR      R1
3471 023272 013700 002014  MOV      LSUNIT,R0
3472 023276              245:  DODU      R1          ; DO DROP UNIT
      (3) 023276 010100      MOV      R1,R0
      (3) 023300 104053      EMT      CSDODU
3473 023302 005201      INC      R1
3474 023304 005300      DEC      R0
3475 023306 001373      BNE      245
3476 023310              DOCLN
      (3) 023310 104044      EMT      CSDCLN
3477
3478
3479 023312      EXIT:
3480
3481 023312      L10007: ENDINIT
      (3) 023312              EMT      CSINIT
      (3) 023312 104011
3482
3483 023314      ENDMOD
3484
3485 023314      BGNMOD CLNCODE
3486
3487
3488 023314      BGNCLN
3489
3490
3491
3492 023314 000240      NOP
3493
3494
3495 023316      L10010: ENDCLN
      (3) 023316              EMT      CSCLEAN
      (3) 023316 104012
3496
3497 023320      ENDMOD
3498
3499 023320      BGNMOD DRPCODE
3500 023320      BGNDU
3501 023320 000240      NOP
3502 023322      ENDDU
      (3) 023322
      (3) 023322 104055      L10011: EMT      CSDU
3503 023324      ENDMOD
3504      SBTTL GLOBAL SUBROUTINES SECTION
3505 023324      BGNMOD GLBSUB
3506
3507      .ALL COMMON OR GLOBAL SUBROUTINES GO HERE
3508
3509      :ROUTINE TO PERFORM OVERWRITE
3510      :CALL: JSR      R5,OVWPER
3511      :          SECTORS TO WRITE FORWARD
3512      :          SECTORS TO WRITE REVERSE
3513
3514 023324 010046      OVWPER. MOV      R0,-(SP)      .SAVE R0, R1, R2, R3
3515 023326 010146      MOV      R1,-(SP)
3516 023330 010246      MOV      R2,-(SP)
```



```

3573
3574 023556 010046          WRSEC:  MOV    R0,-(SP)      ;SAVE R0
3575 023560 010146          MOV    R1,-(SP)      ;SAVE R1
3576 023562 010246          MOV    R2,-(SP)      ;SAVE R2
3577 023564 012701 003020  MOV    #BUF,R1       ;WRITE PATTERN INTO
3578 023570 012702 000200  MOV    #128.,R2      ;MEMORY THAT WE
3579 023574 016421 000006  25:   MOV    PAT(R4),(R1)+ ;WILL WRITE ONTO
3580 023600 005302          DEC    R2            ;PACK FOR THIS
3581 023602 001374          BNE    25            ;DRIVE
3582 023604 012701 100000  MOV    #100000,R1    ;MASK FOR BIT MAP
3583 023610 153702 002710  BISB  DESCYL,R2      ;GET CYLINDER
3584 023614 000302          SWAB  R2            ;PUT IN HIGH BYTE
3585 023616 006002          ROR   R2            ;ALIGN FOR DISK ADDRESS
3586 023620 032737 000400 002710 BIT    #400,DESCYL   ;WHICH SURFACE
3587 023626 001402          BEQ   35            ;0, SKIP
3588 023630 052702 000100  BIS   #HEAD,R2      ;SET BOTTOM HEAD
3589 023634 052502          35:   BIS   (R5)+,R2    ;START AT SECTOR 36
3590 023636 030103          45:   BIT   R1,R3      ;WRITE THIS SECTOR?
3591 023640 001452          BEQ   55            ;NO
3592
3593 023642 005037 002700  CLR   HSFLG         ;
3594 023646 012737 177600 002774 MOV    #-128.,BMP    ;LOAD WORD COUNT
3595 023654 010237 002772  MOV    R2,BDA       ;LOAD DISK ADDRESS
3596 023660 010237 002656  MOV    R2,TEMP      ;SAVE DISK ADDRESS
3597 023664 042702 177700  BIC   #177700,R2    ;
3598 023670 020227 000047  CMP   R2,#39        ;
3599 023674 003403          BLE  65            ;
3600 023676 162737 000050 002772 SUB    #40.,BDA      ;
3601 023704 012737 003020 002770 65:   MOV    #BUF,BBA     ;LOAD BUS ADDRESS
3602 023712 013702 002656  MOV    TEMP,R2      ;RESTORE DISK ADDRESS
3603 023716 004537 030550 115:  JSR   R5,LDFUNC     ;GO WRITE
3604 023722 000012          WRITE
3605 023724 005737 002664  TST   ERFLG         ;ERROR IN WRITING
3606 023730 001416          BEQ   55            ;NO, OKAY
3607 023732 005737 002700  TST   HSFLG         ;
3608 023736 001007          BNE  105           ;
3609 023740          ERRSFT 100.,WRIT1,ERR2
   (3) 023740 104464          TRAP  T$ERCODE
   (5) 023742 000144          .WORD 100
   (5) 023744 017645          .WORD WRIT1
   (5) 023746 020004          .WORD ERR2
3610 023750 005237 002700  INC   HSFLG         ;
3611 023754 000760          BR   115           ;
3612 023756          105:  ERRHRD 110.,WRIT1,ERR2
   (3) 023756 104463          TRAP  T$ERCODE
   (5) 023760 000156          .WORD 110
   (5) 023762 017645          .WORD WRIT1
   (5) 023764 020004          .WORD ERR2
3613
3614 023766 005202          55:   INC   R2            ;NEXT SECTOR
3615 023770 000241          CLC                    ;CLEAR CARRY BIT
3616 023772 006001          ROR   R1            ;DONE?
3617 023774 103320          BCC  45            ;NO GO BACK
3618 023776 012602          MOV   (SP)+,R2      ;REGISTERS AND EXIT
3619 024000 012601          MOV   (SP)+,R1
3620 024002 012600          MOV   (SP)+,R0
  
```

3621	024004	000205			RTS	R5	
3622							
3623	024006	005037	003002		ADJCYL: CLR	ADJTRK	; INSIDE/OUTSIDE TRACK FLAG
3624	024012	005037	002704		CLR	HEAD01	; INIT TO TOP SURFACE
3625	024016	012737	000001	003004	MOV	#1,ADJUUT	; START OF TRACK LIST
3626	024024	012701	002114		21\$: MOV	#OUT10,R1	
3627	024030	012537	002670		20\$: MOV	(R5)+,ADJLOC	; PICK UP TRACK OFFSET
3628	024034	001003			BNE	1\$; IS THERE ONE?
3629	024036	005037	002674		CLR	ADJDIR	
3630	024042	000205			RTS	R5	; NO EXIT
3631	024044	012537	003006		1\$: MOV	(R5)+,ADJLC2	; YES, GET REST OF INFO
3632	024050	012537	003010		MOV	(R5)+,ADJLC3	
3633	024054	012537	003012		MOV	(R5)+,ADJLC4	
3634	024060	113700	002670		2\$: MOV	ADJLOC,R0	; GET OFFSET
3635	024064	012737	000020	003016	MOV	#16.,STSEC	; STARTING SECTOR IS 16
3636							
3637	024072	010102			MOV	R1,R2	; GET START INTO R2
3638							
3639	024074	005300			3\$: DEC	R0	; DOWN COUNT OFFSET
3640	024076	001414			BEQ	4\$; FOUND IT?
3641							
3642	024100	105722			TSTB	(R2)+	; INDEX (R2)
3643	024102	062737	000042	003016	ADD	#34.,STSEC	; NO, NEXT SECTOR
3644	024110	022737	000050	003016	CMP	#40.,STSEC	
3645	024116	003366			BGT	3\$	
3646	024120	162737	000050	003016	SUB	#40.,STSEC	
3647	024126	000762			BR	3\$; BACK FOR NEXT
3648							
3649	024130	121227	000377		4\$: CMPB	(R2),#377	; LEGAL TRACK?
3650	024134	001002			BNE	5\$; YES, CONTINUE
3651							
3652	024136	000137	024560		JMP	13\$; NO PICK UP NEXT SET
3653							
3654	024142	005037	002710		5\$: CLR	DESCYL	; SET UP FOR OUTER TRACK
3655							
3656	024146	005737	002704		TST	HEAD01	; WHICH HEAD?
3657	024152	001403			BEQ	6\$; TOP, SKIP
3658							
3659	024154	052737	000400	002710	BIS	#400,DESCYL	; LOWER HEAD, SET IT!
3660							
3661	024162	004537	024706		6\$: JSR	R5,SKCYL	; SEEK TO OUTER TRACK
3662							
3663	024166	111237	002710		MOV	(R2),DESCYL	; GET DESIRED TRACK
3664							
3665	024172	004537	024706		JSR	R5,SKCYL	; SEEK TO IT
3666	024176	012737	017625	002706	MOV	#FWD,DIRC	; SEEK DIRECTION
3667	024204	113703	002671		MOV	ADJLOC+1,R3	; GET SECTORS TO WRITE
3668	024210	000303			SWAB	R3	; ALIGN IT
3669	024212	042703	000377		BIC	#377,R3	; CLEAR OUT HIGH BYTE
3670							
3671	024216	022737	000047	003016	CMP	#39.,STSEC	; OVER FORTY?
3672	024224	002003			BGE	7\$; NO, CONTINUE
3673							
3674	024226	162737	000050	003016	SUB	#40.,STSEC	; YES BACK IT UP
3675	024234	013737	003016	024246	7\$: MOV	STSEC,8\$; STARTING SECTOR
3676							

3677	024242	004537	023556		JSR	R5, WRSEC	;WRITE SECTORS	
3678	024246	000000		85:	.WORD	0		
3679	024250	013737	024246	024262	MOV	85, 1085		
3680	024256	004537	026524		JSR	R5, VAJWR	;VERIFY THIS WRITE	
3681	024262	000000		1085:	.WORD	0		
3682	024264	013737	024262	024276	MOV	1085, 2085		
3683	024272	004537	026760		JSR	R5, BSVWR		
3684	024276	000000		2085:	.WORD	0		
3685	024300	013737	003016	003014	MOV	STSEC, STSEC1	;GET OTHER SECTORS TO WRITE	
3686	024306	062737	000010	003014	ADD	#8, STSEC1	;8 SECTORS GONE BY	
3687	024314	022737	000047	003014	CMP	#39, STSEC1	;GONE PAST 40?	
3688	024322	002003			BGE	95	;NO, OKAY	
3689								
3690	024324	162737	000050	003014	SUB	#40, STSEC1	;YES BACK IT UP	
3691								
3692	024332	013703	003006		95:	MOV	ADJLC2, R3	;GET SECTORS TO WRITE
3693								
3694	024336	013737	003014	024350	MOV	STSEC1, 105	;STARTING SECTORS	
3695								
3696	024344	004537	023556		JSR	R5, WRSEC	;WRITE SECTORS	
3697	024350	000000		105:	.WORD	0		
3698	024352	013737	024350	024364	MOV	105, 1105		
3699	024360	004537	026524		JSR	R5, VAJWR	;VERIFY THIS WRITE	
3700	024364	000000		1105:	.WORD	0		
3701	024366	013737	024364	024400	MOV	1105, 2105		
3702	024374	004537	026760		JSR	R5, BSVWR	;VERIFY ADJ CYL + 1	
3703	024400	000000		2105:	.WORD	0		
3704	024402	112737	000377	002710	MOVB	#377, DESCYL	;SEEK TO INNER TRACK	
3705	024410	004537	024706		JSR	R5, SKCYL		
3706								
3707	024414	111237	002710		MOVB	(R2), DESCYL	;SEEK BACK TO PROPER TRACK	
3708								
3709	024420	004537	024706		JSR	R5, SKCYL	;SEEK TO PROPER CYLINDER	
3710	024424	012737	017635	002706	MOV	#REV, DIRC	;SEEK DIRECTION	
3711	024432	113703	003011		MOVB	ADJLC3+1, R3	;GET SECTORS TO WRITE	
3712								
3713	024436	000303			SWAB	R3	;ALIGN IT	
3714	024440	042703	000377		BIC	#377, R3	;CLEAR OUT HIGH BYTE	
3715	024444	013737	003016	024456	MOV	STSEC, 115		
3716								
3717	024452	004537	023556		JSR	R5, WRSEC	;WRITE PROPER SECTOR	
3718	024456	000000		115:	.WORD	0		
3719								
3720	024460	013737	024456	024472	MOV	115, 1115		
3721	024466	004537	026524		JSR	R5, VAJWR	;VERIFY THIS WRITE	
3722	024472	000000		1115:	.WORD	0		
3723	024474	013737	024472	024506	MOV	1115, 2115		
3724	024502	004537	026760		JSR	R5, BSVWR		
3725	024506	000000		2115:	.WORD	0		
3726	024510	013703	003012		MOV	ADJLC4, R3	;GET SECTORS	
3727	024514	013737	003014	024526	MOV	STSEC1, 125	;GET SECTORS TO WRITE	
3728								
3729	024522	004537	023556		JSR	R5, WRSEC	;WRITE PROPER SECTORS	
3730	024526	000000		125:	.WORD	0		
3731								
3732								

```

OUTERR MACY11 30(1046) 06-DEC-77 18:18 PAGE 83-22          N 3
DZRLFA P11 23-NOV-77 13:47          GLOBAL SUBROUTINES SECTION          SEQ 0039

3733 024530 013737 024526 024542          MOV      125,1125
3734 024536 004537 026524          JSR      R5,VAJWR          ;VERIFY THIS WRITE
3735 024542 000000          1125:   .WORD      0
3736
3737
3738 024544 013737 024542 024556          MOV      1125,2125
3739 024552 004537 026760          JSR      R5,BSVWR          ;VERIFY ADJ CYLINDERS + 1
3740 024556 000000          2125:   .WORD      0
3741
3742
3743 024560 005737 002704          135:    TST      HEAD01          ;WHICH HEAD WERE WE DOING?
3744 024564 001003          BNE      145
3745 024566 005237 002704          INC      HEAD01
3746 024572 000402          BR       995
3747 024574 005037 002704          145:    CLR      HEAD01          ;NEXT SET OF TRACKS
3748 024600 062701 000005          995:    ADD      #5,R1          ;NEXT SET OF TRACKS
3749 024604 020127 002175          CMP      R1,#INN51          ;END OF LIST
3750 024610 002002          BGE      185          ;END OF TRACK LIST
3751 024612 000137 024060          JMP      25          ;NO GO BACK
3752
3753          ;AT END OF TRACK LIST NEXT GROUP OF WRITES
3754
3755 024616 005737 002654          155:    TST      FADJ          ;FIRST SET?
3756 024622 001403          BEQ      155          ;NO, CONTINUE
3757 024624 005037 002654          CLR      FADJ          ;YES, CLEAR FIRST
3758 024630 000421          BR       175          ;EXIT
3759 024632 005737 003002          155:    TST      ADJTRK          ;DONE BOTH INSIDE OUTSIDE
3760 024636 001004          BNE      165          ;TRACKS, YES 165
3761 024640 005237 003002          INC      ADJTRK          ;NO, SET INSIDE FLAG
3762 024644 000137 024024          JMP      215          ;GO DO INSIDE TRACK
3763 024650 005037 003002          165:    CLR      ADJTRK          ;BACK TO OUTSIDE TRACK
3764 024654 005237 003004          INC      ADJUUT          ;DONE WITH ANOTHER
3765 024660 023737 003004 002716          CMP      ADJUUT,UUT          ;DONE TABLE FOR ALL UUT?
3766 024666 001402          BEQ      175          ;YES, FOR EXIT
3767 024670 000137 024024          JMP      215          ;NO, GO BACK FOR NEXT
3768 024674 005725          175:    TST      (R5)+          ;BUMP EXIT TO END OF
3769 024676 001376          BNE      175          ;TABLE FOR PROPER RETURN
3770 024700 005037 002674          CLR      ADJDIR          ;EXIT
3771 024704 000205          RTS      R5
3772
3773
3774          ;ROUTINE TO SEEK TO A DESIRED CYLINDER
3775          ;CALL: JSR      R5,SKCYL
3776          ;ROUTINE HAS DESIRED CYLINDER IN LOC "DESCYL"
3777          ;
3778          ;
3779 024706 010146          SKCYL:  MOV      R1,-(SP)          ;SAVE R1
3780 024710 004537 030550          905:    JSR      R5,LDFUNC          ;GET PRESENT POSITION
3781 024714 000010          RDHDR
3782
3783 024716 005737 002664          TST      ERFLG          ;ERROR FLAG SET
3784 024722 001074          BNE      55          ;YES, SKIP
3785
3786 024724 005001          CLR      R1
3787 024726 153701 002710          BISB    DESCYL,R1          ;GET DESIRED CYLINDER
3788 024732 000301          SWAB    R1          ;GET IN HIGH BYTE

```

```

3789 024734 006001          ROR      R1          ;ALIGN IT
3790 024736 042737 000177 002760 BIC      #177,E.MP   ;CLEAR PRESENT HD: SEC
3791 024744 163701 002760      SUB      E.MP,R1     ;CALCULATE DIFFERENCE WORD
3792 024750 100002          BPL      1$         ;IF POSITIVE SET DIRECTION
3793 024752 005401          NEG      R1         ;NEGATE
3794 024754 000402          GR       2$         ;SKIP SETTING DIRECTION
3795 024756 052701 000004      1$:     BIS      #SIGN,R1 ;SET FOR FORWARD SEEK
3796 024762 052701 000001      2$:     BIS      #MK,R1  ;SET MARKER BIT
3797 024766 032737 000400 002710 BIT      #400,DESCYL ;WHICH HEAD
3798 024774 001402          BEQ      3$         ;TOP
3799 024776 052701 000020      BIS      #SKHS,R1  ;BOTTOM
3800 025002 010137 002772      3$:     MOV      R1,BDA   ;LOAD DIFFERENCE WORD
3801 025006 004537 030550      JSR      R5,LDFUNC ;EXECUTE SEEK
3802 025012 000006
3803
3804 025014 005737 002664      TST      ERFLG     ;ERROR?
3805 025020 001035          BNE      5$         ;YES, SKIP
3806
3807 025022 004537 030550      JSR      R5,LDFUNC ;VERIFY POSITION?
3808 025026 000010      RDHDR
3809 025030 005737 002664      TST      ERFLG
3810 025034 001027          BNE      5$
3811 025036 042737 000077 002760 BIC      #77,E.MP   ;VERIFY POSITION
3812 025044 005001          CLR      R1
3813 025046 153701 002710      BISB    DESCYL,R1 ;IS CORRECT AND IF
3814 025052 000301          SWAB    R1         ;NOT CORRECT THEN
3815 025054 006001          ROR      R1         ;RESEEK
3816 025056 032737 000400 002710 BIT      #400,DESCYL
3817 025064 001402          BEQ      4$
3818 025066 052701 000100      BIS      #HEAD,R1
3819 025072 020137 002760      4$:     CMP      R1,E.MP
3820 025076 001414          BEQ      6$
3821
3822 025100          ERRDF   12.,SKER,ERR6 ;SEEK ERROR
(3) 025100 104462      TRAP    T$ERCODE
(5) 025102 000014      .WORD  12
(5) 025104 017606      .WORD  SKER
(5) 025106 020422      .WORD  ERR6
3823 025110 000137 024710      JMP     90$
3824
3825 025114          5$:     ERRDF   13.,FUNERR,ERR5 ;FUNCTION ERROR IN SEEK
(3) 025114 104462      TRAP    T$ERCODE
(5) 025116 000015      .WORD  13
(5) 025120 017556      .WORD  FUNERR
(5) 025122 020362      .WORD  ERR5
3826 025124 000137 024710      JMP     90$
3827 025130 012601      6$:     MOV      (SP)+,R1 ;CANT GET THERE
3828 025132 000205      RTS     R5         ;EXIT
3829
3830
3831          ;ROUTINE TO PERFORM REGISTER PRINTOUT DUMP
3832          ;CALL: JSR PC,REGDMP
3833 025134          REGDMP: PRINTB #FRM12,BCS,BBA,BDA,BMP
(11) 025134 013746 002774      MOV     BMP,-(SP)
(10) 025140 013746 002772      MOV     BDA,-(SP)
(9) 025144 013746 002770      MOV     BBA,-(SP)
  
```

```

(8) 025150 013746 002766      MOV      BCS, -(SP)
(7) 025154 012746 021361      MOV      #FRM12, -(SP)
(6) 025160 012746 000005      MOV      #5, -(SP)
(3) 025164 010600              MOV      SP, R0
(4) 025166 104014              EMT      C$PNTB
(4) 025170 062706 000014      ADD      #14, SP
3834 025174              PRINTB   #FRM13, E. CS, E. BA, E. DA, E. MP
(11) 025174 013746 002760      MOV      E. MP, -(SP)
(10) 025200 013746 002756      MOV      E. DA, -(SP)
(9) 025204 013746 002754      MOV      E. BA, -(SP)
(8) 025210 013746 002752      MOV      E. CS, -(SP)
(7) 025214 012746 021440      MOV      #FRM13, -(SP)
(6) 025220 012746 000005      MOV      #5, -(SP)
(3) 025224 010600              MOV      SP, R0
(4) 025226 104014              EMT      C$PNTB
(4) 025230 062706 000014      ADD      #14, SP
3835 025234 032737 040000 002752  BIT      #BIT14, E. CS
3836 025242 001437              BEQ      1$
3837 025244 016403 000000      MOV      CSR(R4), R3
3838 025250 012763 000013 000004  MOV      #13, DA(R3)
3839 025256 012737 000004 002766  MOV      #4, BCS
3840 025264 056437 000004 002766  BIS      DSB(R4), BCS
3841 025272 013763 002766 000000  MOV      BCS, CS(R3)
3842 025300 032763 000200 000000 2$: BIT      #200, CS(R3)
3843 025306 001774              BEQ      2$
3844 025310 016337 000006 002676  MOV      MP(R3), DRSTAT
3845 025316              PRINTB   #FRM14, DRSTAT
(8) 025316 013746 002676      MOV      DRSTAT, -(SP)
(7) 025322 012746 021523      MOV      #FRM14, -(SP)
(6) 025326 012746 000002      MOV      #2, -(SP)
(3) 025332 010600              MOV      SP, R0
(4) 025334 104014              EMT      C$PNTB
(4) 025336 062706 000006      ADD      #6, SP
3846 025342 000207 1$: RTS      PC
3847
3848 ;ROUTINE TO SET DRIVE IN SECTOR LIST
3849 ;CALL: JSR R5, SETLST ;R0 HAS SECTOR
3850 ;DRIVE GOTTEN FROM R4
3851
3852 025344 010146 SETLST: MOV      R1, -(SP) ;SAVE R1
3853
3854 025346 162700 000034 SUB      #28, R0 ;START LIST AT 0
3855 025352 100002 BPL      3$
3856 025354 062700 000050 ADD      #40, R0
3857 025360 012701 002176 3$: MOV      #SECLST, R1 ;BEGINNING OF SECTOR LIST
3858 025364 005700 1$: TST      R0 ;FOUND SECTOR?
3859 025366 001403 BEQ      2$ ;BRANCH IF YES
3860 025370 005300 DEC      R0 ;DECREMENT SECTOR
3861 025372 005721 TST      (R1)+ ;NEXT ENTRY IN LIST
3862 025374 000773 BR       1$ ;GO BACK
3863 025376 010411 2$: MOV      R4, (R1) ;STORE DRIVE BITS IN LIST
3864 025400 012601 MOV      (SP)+, R1 ;RESTORE R1
3865 025402 000205 RTS      R5
3866
3867 ;ROUTINE TO STORE OR RETRIEVE ADJACENT CYLINDER SECTOR DRIVE
3868 ;INFORMATION FROM THE 24X5 "SECLST" BUFFER.
    
```

```

3869 ;ENTER WITH RO = SECTOR REQUEST
3870 ;EXIT WITH RO = ADJACENT CYLINDER DRIVE INFORMATION FOR SECTOR
3871 ;EXIT WITH RO = 0 IF SECTOR REQUESTED IS NOT IN BUFFER MAP
3872 ;CALL 1: JSR R5,RSADJS
3873 ; ;RETRIEVE SECTOR INFO.
3874 ;CALL 2: JSR R5,RSADJS
3875 ; ;STORE SECTOR INFO.
3876 025404 010146 RSADJS: MOV R1,-(SP)
3877 025406 010246 MOV R2,-(SP)
3878 025410 010346 MOV R3,-(SP)
3879 025412 042700 177700 BIC #177700,R0 ;SAVE SECTOR BITS
3880 025416 012537 002672 MOV (R5)+,ADJFLG ;SAVE RETRIEVE/STORE FLAG
3881 025422 012701 000001 MOV #1,R1 ;START WITH TRACK (N-2)
3882 025426 012702 002236 MOV #SECBUF,R2 ;START OF 24X5 BUFFER
3883 025432 012703 000020 MOV #16,R3 ;SECTOR 16 START FOR (N-2) TRACK
3884 025436 123701 002670 15: CMPB ADJLOC,R1 ;CHECK TRACK INDEX
3885 025442 001413 BEQ 25 ;
3886 025444 005201 INC R1 ;INDEX TRACK REFERENCE
3887 025446 062702 000060 ADD #48,R2 ;UPDATE BUFFER TO NEXT TRACK REF.
3888 025452 062703 000042 ADD #34,R3 ;UPDATE SECTOR START FOR NEXT TRACK
3889 025456 020327 000050 CMP R3,#40
3890 025462 002765 BLT 15
3891 025464 162703 000050 SUB #40,R3
3892 025470 000762 BR 15 ;
3893 025472 012701 000030 25: MOV #24,R1 ;SET COUNTER FOR 24 SECTORS
3894 025476 020003 35: CMP R0,R3 ;COMPARE SECTOR TO SECTOR TABLE
3895 025500 001413 BEQ 55 ;YES,STORE OR RETRIEVE SECTOR INFO.
3896 025502 005722 TST (R2)+ ;INDEX SECLST BUFFER IN WORD FORMAT
3897 025504 005203 INC R3 ;INDEX SECTOR COUNT
3898 025506 020327 000047 CMP R3,#39 ;COMPARE SECTOR COUNT FOR <40
3899 025512 003402 BLE 45
3900 025514 162703 000050 SUB #40,R3 ;KEEP SECTOR COUNT<40
3901 025520 005301 45: DEC R1 ;PASSED 24 SECTORS?
3902 025522 001365 BNE 35 ;COMPARE NEXT SECTOR
3903 025524 005000 CLR R0 ;SETUP RO FOR EXIT
3904 025526 000405 BR 75 ;EXIT ROUTINE,SECTOR NOT FOUND
3905 025530 005737 002672 55: TST ADJFLG ;FLAG=0 FOR RETRIEVE
3906 025534 001401 BEQ 65
3907 025536 010412 MOV R4,(R2) ;STORE DRIVE INFO. INTO BUFFER
3908 025540 011200 65: MOV (R2),R0 ;SAVE DRIVE INFO. INTO RO FOR EXIT
3909 025542 012603 75: MOV (SP)+,R3
  
```

GLOBAL SUBROUTINES SECTION

```

3911 025544 012602          MOV      (SP)+,R2
3912 025546 012601          MOV      (SP)+,R1
3913 025550 000205          RTS      R5              ;EXIT
3914
3915          ;ROUTINE TO LOCATE DRIVE THAT WROTE SECTOR LAST
3916          ;CALL: JSR      R5,FNDDRV          ;R0-CONTAINS SECTOR
3917          ;ON EXIT R0-DRIVE
3918
3919 025552 010146          FNDDRV: MOV      R1,-(SP)          ;SAVE R1
3920 025554 162700 000034      SUB      #28.,R0          ;START LIST AT 0
3921 025560 100002          BPL      3$
3922 025562 062700 000050      ADD      #40.,R0
3923 025566 012701 002176      3$: MOV      #SECLST,R1          ;START OF LIST
3924 025572 005700          1$: TST      R0              ;FOUND SECTOR?
3925 025574 001403          BEQ      2$              ;YES, GET DRIVE #, EXIT
3926 025576 005300          DEC      R0              ;NO, DOWN COUNT SECTOR
3927 025600 005721          TST      (R1)+           ;NEXT ENTRY IN LIST
3928 025602 000773          BR       1$              ;GO BACK
3929 025604 011100          2$: MOV      (R1),R0          ;GET DRIVE BUFFER POINTER
3930 025606 012601          MOV      (SP)+,R1        ;RESTORE R1
3931 025610 000205          RTS      R5              ;EXIT
3932
3933
3934          ;ROUTINE TO VERIFY THAT THE OVERWRITE DID ACTUALLY OVERWRITE THE
3935          ;PREVIOUS DATA ON THE PACK.
3936
3937          ;CALL: JSR      R5,VEROW          USES R3 AS BIT MAP OF SECTORS TO
3938          ;CHECK. R3 IS LOADED PRIOR TO
3939          ;WRITING SECTORS.
3940
3941 025612 010046          VEROW: MOV      R0,-(SP)          ;SAVE REGISTER CONTENTS
3942 025614 010146          MOV      R1,-(SP)
3943 025616 010246          MOV      R2,-(SP)
3944 025620 012737 000034 002720      MOV      #28.,SECT          ;START VERIFY AT SECTOR 28
3945 025626 012701 100000      MOV      #100000,R1        ;BIT MASK FOR VERIFICATION
3946 025632 016437 000006 002724      MOV      PAT(R4),GDATA     ;GET PATTERN FOR THIS DRIVE
3947
3948 025640 012737 177600 002774 1$: MOV      #-128.,BMP        ;SET UP READ-ONE SECTOR
3949 025646 012737 003020 002770      MOV      #BUF,BBA         ;BUS ADDRESS
3950 025654 042737 000077 002772 2$: BIC      #77,BDA         ;CLEAR OUT SECTOR BITS
3951 025662 053737 002720 002772      BIS      SECT,BDA         ;SET SECTOR
3952 025670 030103          BIT      R1,R3           ;DO WE READ THIS ONE?
3953 025672 001521          BEQ      5$              ;NO, BRANCH
3954 025674 004537 030550      JSR      R5,LDFUNC        ;READ
3955 025700 000014          READ
3956
3957 025702 005737 002752          TST      E.CS            ;ERROR
3958 025706 100107          BPL      4$              ;NO CONTINUE
3959
3960 025710 005737 002652          TST      F0WR           ;INITIAL WRITE
3961 025714 001412          BEQ      21$            ;NO
3962 025716 012737 017102 002662      MOV      #INITWR,REASON    ;SETUP INITIAL WRITE OF SECTOR
3963 025724 016437 000000 002660      MOV      CSR(R4),LSTCLR
3964 025732 016437 000005 002722      MOV      DSB+1(R4),LSTDRV
3965 025740 000415          BR       22$
3966 025742 012737 017351 002662 21$: MOV      #OVMS,REASON     ;SET MESSAGE FOR OVERWRITE

```

GLOBAL SUBROUTINES SECTION

```

3967 025750 013700 002720      MOV      SECT,R0      ;FIND DRIVE THAT LAST WROTE
3968 025754 004537 025552      JSR      R5,FNDDRV   ;SECTOR
3969 025760 016037 000000 002660  MOV      CSR(R0),LSTCLR ;GET IT'S CSR
3970 025766 116037 000005 002722  MOV      DSB+1(R0),LSTDRV ;GET THE DRIVE
3971 025774      22$:  ERRDF  13.,OVWER,ERR4 ;PRINT ERROR
      (3) 025774 104462      TRAP   T$ERCODE
      (5) 025776 000015      .WORD  13
      (5) 026000 017512      .WORD  OVWER
      (5) 026002 020176      .WORD  ERR4
3972 026004 005037 002730      CLR      WCOUNT     ;CLEAR BAD WORD COUNT W/IN SECTOR
3973 026010 005037 002732      CLR      SECWRD      ;CLEAR WORD IN SECTOR
3974 026014 012702 003020      MOV      #BUF,R2     ;GET BUFFER START
3975 026020 023712 002724      3$:  CMP      GDATA,(R2) ;IS DATA CORRECT?
3976 026024 001417      BEQ      31$        ;YES CHECK NEXT
3977 026026 005237 002730      INC      WCOUNT     ;NO ACCOUNT FOR IT
3978 026032      PRINTF #FRM8,SECWRD,GDATA,(R2)
      (10) 026032 011246      MOV      (R2),-(SP)
      (9) 026034 013746 002724      MOV      GDATA,-(SP)
      (8) 026040 013746 002732      MOV      SECWRD,-(SP)
      (7) 026044 012746 021152      MOV      #FRM8,-(SP)
      (6) 026050 012746 000004      MOV      #4,-(SP)
      (3) 026054 010600      MOV      SP,R0
      (4) 026056 104017      EMT      C$PNTF
      (4) 026060 062706 000012      ADD      #12,SP
3979
3980 026064 005722      31$:  TST      (R2)+      ;NEXT
3981 026066 005237 002732      INC      SECWRD      ;NEXT
3982 026072 023727 002732 000200  CMP      SECWRD,#128. ;DONE WITH SECTOR?
3983 026100 001347      BNE      3$         ;NO GO BACK
3984
3985 026102      PRINTF #FRM9,WCOUNT ;PRINT SUMMARY
      (8) 026102 013746 002730      MOV      WCOUNT,-(SP)
      (7) 026106 012746 021216      MOV      #FRM9,-(SP)
      (6) 026112 012746 000002      MOV      #2,-(SP)
      (3) 026116 010600      MOV      SP,R0
      (4) 026120 104017      EMT      C$PNTF
      (4) 026122 062706 000006      ADD      #6,SP
3986
3987 026126 013700 002720      4$:  MOV      SECT,R0      ;SET SECTOR IN LIST TO THE
3988 026132 004537 025344      JSR      R5,SETLST   ;CREDIT OF THIS DRIVE
3989
3990 026136 005237 002720      5$:  INC      SECT        ;NEXT SECTOR
3991 026142 023727 002720 000050  CMP      SECT,#40.
3992 026150 001003      BNE      6$
3993 026152 162737 000050 002720  SUB      #40.,SECT
3994 026160 000241      6$:  CLC          ;CLEAR CARRY
3995 026162 006001      ROR      R1          ;NEXT BIT
3996 026164 103225      BCC      1$         ;IF CLEAR NEXT
3997
3998 026166 012602      MOV      (SP)+,R2    ;RESTORE R2-R0, EXIT
3999 026170 012601      MOV      (SP)+,R1
4000 026172 012600      MOV      (SP)+,R0
4001 026174 000205      RTS      R5

```

:ROUTINE TO VERIFY THAT A DRIVE CAN RECOVER ANOTHER DRIVE'S DATA.

4004

```

4005 ;CALL: JSR R5,VEROD      USES R3 AS BIT MAP OF SECTORS TO
4006 ;                                     CHECK. R3 IS LOAD BY WRSEC (WE
4007 ;                                     USE R3 COMPLIMENTED.
4008 ;
4009 ;
4010 VEROD: MOV RO,-(SP)      ;SAVE RO-R2
4011 MOV R1,-(SP)
4012 MOV R2,-(SP)
4013 MOV #100000,R1          ;BIT MASK FOR SECTORS
4014 MOV #28,SECT           ;START WITH SECTOR 28
4015 TST F0WR              ;CHECK FOR FIRST OVERWRITE
4016 BNE 6$
4017
4018 MOV #-128,BMP          ;SET UP READ (ONE SECTOR)
4019 MOV #BUF,BBA           ;BUS ADDRESS
4020 BIC #77,BDA           ;CLEAR SECTOR BITS
4021 BIS SECT,BDA          ;SET IN SECTOR BITS
4022 BIT R1,R3             ;CHECK THIS SECTOR?
4023 BNE 5$               ;NO BRANCH
4024
4025 MOV SECT,RO            ;FIND DRIVE THAT WROTE
4026 JSR R5,FNDDR          ;SECTOR LAST
4027 MOV CSR(RO),LSTCLR    ;GET CSR OF DRIVE
4028 MOVB DSB+1(RO),LSTDRV ;GET DRIVE
4029 MOV PAT(RO),GDATA     ;GET PATTERN
4030
4031 JSR R5,LDFUNC          ;READ
4032 READ
4033
4034 TST E.CS               ;ERROR?
4035 BPL 5$                ;NO, NEXT SECTOR
4036 MOV #RECMS,REASON     ;SET READ RECOVERY MESSAGE
4037 ERRDF 14,RECER,ERR4  ;REPORT ERROR
(3) TRAP T$ERCODE
(5) .WORD 14
(5) .WORD RECER
(5) .WORD ERR4
4038
4039 CLR WCOUNT           ;CLEAR BAD WORD COUNT
4040 CLR SECWRD            ;CLEAR WORD W/I SECTOR
4041 MOV #BUF,R2           ;START OF BUFFER
4042 CMP GDATA,(R2)       ;DATA COMPARE
4043 BEQ 4$               ;YES, CHECK NEXT
4044
4045 INC WCOUNT           ;ACCOUNT FOR ERROR
4046 PRINTF #FRM8,SECWRD,GDATA,(R2) ;PRINT ERROR
(10) MOV (R2),-(SP)
(9) MOV GDATA,-(SP)
(8) MOV SECWRD,-(SP)
(7) MOV #FRM8,-(SP)
(6) MOV #4,-(SP)
(3) MOV SP,RO
(4) EMT C$PNTF
(4) ADD #12,SP
4047
4048 TST (R2)+             ;NEXT

```

OUTERR MACY11 30(1046) 06-DEC-77 18:18 PAGE 84-3
DZPLFA P11 23-NOV-77 13:47

H 4

GLOBAL SUBROUTINES SECTION

SEQ 0046

4049	026426	005237	002732		INC	SECWRD	;NEXT WORD IN SECTOR
4050	026432	023727	002732	000200	CMP	SECWRD, #128.	; DONE?
4051	026440	001347			BNE	3\$; NO

```

4053
4054 026442          PRINTF #FRM9,WCOUNT ;PRINT SUMMARY
      (8) 026442 013746 002730  MOV   WCOUNT,-(SP)
      (7) 026446 012746 021216  MOV   #FRM9,-(SP)
      (6) 026452 012746 000002  MOV   #2,-(SP)
      (3) 026456 010600          MOV   SP,R0
      (4) 026460 104017          EMT   C$PNTF
      (4) 026462 062706 000006  ADD   #6,SP
4055
4056 026466 005237 002720      5$:  INC   SECT          ;NEXT SECTOR
4057 026472 023727 002720 000050  CMP   SECT,#40.
4058 026500 001002          BNE   7$
4059 026502 005037 002720      CLR   SECT
4060 026506 000241          7$:  CLC
4061 026510 006001          ROR   R1          ;NEXT BIT MAP
4062 026512 103244          BCC   1$
4063
4064 026514 012602          6$:  MOV   (SP)+,R2      ;RESTORE R2-R0, EXIT
4065 026516 012601          MOV   (SP)+,R1
4066 026520 012600          MOV   (SP)+,R0
4067 026522 000205          RTS   R5
4068
4069
4070          ;ROUTINE TO VERIFY THE ADJ. CYL. WRITE IS GOOD
4071          ;USES R3 AND WORD FOLLOWING CALL
4072          ;IF WRITE WAS GOOD,SECTOR WILL BE STORED IN MAP
4073          ;USING RSADJS/, WORD 1
4074
4075 026524 010046          VAJWR: MOV   R0,-(SP)      ;SAVE REGISTERS
4076 026526 010146          MOV   R1,-(SP)
4077 026530 010246          MOV   R2,-(SP)
4078 026532 012701 100000      MOV   #100000,R1  ;BIT MASK FOR CYLINDER
4079 026536 012502          MOV   (R5)+,R2    ;STARTING SECTOR
4080 026540 105000          CLRB  R0
4081 026542 153700 002710      BISB  DESCYL,R0
4082 026546 000300          SWAB  R0
4083 026550 006000          ROR   R0
4084 026552 032737 000400 002710  BIT   #400,DESCYL
4085 026560 001402          BEQ   3$
4086 026562 052700 000100      BIS   #HEAD,R0
4087 026566 050200          3$:  BIS   R2,R0
4088 026570 030103          4$:  BIT   R1,R3
4089 026572 001462          BEQ   5$
4090 026574 012737 177600 002774  MOV   #-128.,BMP
4091 026602 010037 002772      MOV   R0,BDA
4092 026606 010037 002656      MOV   R0,TEMP
4093 026612 042700 177700      BIC   #177700,R0
4094 026616 020027 000047      CMP   R0,#39.
4095 026622 003406          BLE   6$
4096 026624 162737 000050 002772  SUB   #40.,BDA
4097 026632 162737 000050 002656  SUB   #40.,TEMP
4098 026640 012737 003020 002770  6$:  MOV   #BUF,BBA
4099 026646 005037 002700      CLR   HSFLG
4100 026652 013700 002656      MOV   TEMP,R0
4101 026656 004537 030550      10$: JSR   R5,LDFUNC  ;READ FUNCTION
4102 026662 000014          READ
  
```

GLOBAL SUBROUTINES SECTION

```

4103 026664 005737 002664      TST      ERFLG
4104 026670 001416              BEQ      7$
4105 026672 005737 002700      TST      HSFLG
4106 026676 001007              BNE     11$
4107 026700              ERRSOFT 120.,READ1,ERR2
      (3) 026700 104464          TRAP    T$ERCODE
      (5) 026702 000170          .WORD  120
      (5) 026704 017672          .WORD  READ1
      (5) 026706 020004          .WORD  ERR2
4108 026710 005237 002700      INC      HSFLG
4109 026714 000760              BR      10$
4110 026716              11$:  ERRHRD 130.,READ1,ERR2
      (3) 026716 104463          TRAP    T$ERCODE
      (5) 026720 000202          .WORD  130
      (5) 026722 017672          .WORD  READ1
      (5) 026724 020004          .WORD  ERR2
4111 026726 010046              7$:  MOV    R0,-(SP)
4112 026730 004537 025404      JSR     R5,RSADJS      ;STORE ADJ. CYL. SECTOR INFO.
4113 026734 000001          .WORD  1
4114 026736 012600              MOV    (SP)+,R0      ;RESTORE R0
4115 026740 005200              5$:  INC    R0
4116 026742 000241              CLC
4117 026744 006001              ROR    R1
4118 026746 103310              BCC    4$
4119 026750 012602              MOV    (SP)+,R2
4120 026752 012601              MOV    (SP)+,R1
4121 026754 012600              MOV    (SP)+,R0
4122 026756 000205              RTS     R5

;ROUTINE TO VERIFY THAT WRITE DID NOT DISTURB ADJACENT TRACKS
;WRITTEN BY OTHER DRIVES.
;CALL JSR R5,BSVWR
;      .WORD          ;STARTING SECTOR
;
;USES "ADJLOC" TO GET +1/-1 CYLINDER OFFSET
;USES R3 FOR SECTOR MAP, USES MAP AT "SECBUF" FOR INFO
4132 026760 010046      BSVWR: MOV    R0,-(SP)      ;SAVE REGISTERS
4133 026762 010146          MOV    R1,-(SP)
4134 026764 010246          MOV    R2,-(SP)
4135 026766 013746 002710      MOV    DESCYL,-(SP)    ;SAVE PRESENT POSITION
4136 026772 012546          MOV    (R5)+,-(SP)    ;GET STARTING SECTOR
4137 026774 123727 002670 000003  CMPB   ADJLOC,#3      ;ON MIDDLE TRACK???
4138 027002 001455          BEQ    B$EXIT         ;YES, THEN NO CHECK
4139 027004 162716 000042          SUB    #34,(SP)      ;SETUP SECTOR START FOR OUTSIDE
4140 027010 100002          BPL    1$            ;IF POSITIVE OKAY ELSE FIX
4141 027012 062716 000050          ADD    #40,(SP)      ;FIX IT
4142 027016 123727 002670 000001 1$:  CMPB   ADJLOC,#1      ;ON OUTER LIMIT???
4143 027024 001412          BEQ    INAWR         ;YES, SKIP CHECK
4144 027026 105337 002670          DECB  ADJLOC        ;OUTER ADJ TRACK
4145 027032 105337 002710          DECB  DESCYL       ;CREATE CYLINDER
4146 027036 004537 027160          JSR   R5,CHECK      ;GO CHECK ADJ SECTORS
4147 027042 105237 002710          INCB  DESCYL       ;FIX BACK
4148 027046 105237 002670          INCB  ADJLOC
4149 027052 062716 000104      INAWR: ADD    #68,(SP)    ;INNER SECTOR START
4150 027056 021627 000050          CMP    (SP),#40     ;WITHIN LIMITS???
  
```

```

4151 027062 002407          BLT      1$          ;YES, OKAY
4152 027064 162716 000050    SUB      #40.,(SP)    ;FIX SECTOR
4153 027070 021627 000050    CMP      (SP),#40.
4154 027074 002402          BLT      1$
4155 027076 162716 000050    SUB      #40.,(SP)
4156 027102 123727 002670 000005 1$:  CMPB    ADJLOC,#5    ;INNER LIMIT??
4157 027110 001412          BEQ      BSEXIT      ;YES, SKIP CHECK
4158 027112 105237 002670    INCB    ADJLOC      ;FIX FOR INNER
4159 027116 105237 002710    INCB    DESCYL
4160 027122 004537 027160    JSR     R5,CHECK    ;GO CHECK ADJ SECTORS
4161 027126 105337 002670    DECB    ADJLOC      ;FIX BACK
4162 027132 105337 002710    DECB    DESCYL
4163 027136 005726          BSEXIT: TST      (SP)+ ;THROW OFF SECTOR
4164 027140 012637 002710    MOV     (SP)+,DESCYL ;GET OLD CYLINDER
4165 027144 012602          MOV     (SP)+,R2
4166 027146 012601          MOV     (SP)+,R1
4167 027150 012600          MOV     (SP)+,R0
4168 027152 004537 024706    JSR     R5,SKCYL    ;SEEK BACK
4169 027156 000205          RTS      R5         ;RETURN
4170
4171
4172          ;ROUTINE TO VERIFY AN ADJACENT SECTOR
4173          ;CALLED FROM BSVWR
4174
4175
4176 027160 012701 100000    CHECK: MOV     #100000,R1 ;SECTOR MASK
4177 027164 004537 024706    JSR     R5,SKCYL    ;GET TO DESIRED CYLINDER
4178 027170 005002          CLR     R2         ;CREATE ADDRESS
4179 027172 153702 002710    BISB   DESCYL,R2
4180 027176 000302          SWAB   R2
4181 027200 006002          ROR    R2
4182 027202 032737 000400 002710  BIT     #400,DESCYL ;HEAD SET???
4183 027210 001402          BEQ    3$         ;NO
4184 027212 052702 000100    BIS    #HEAD,R2
4185 027216 056602 000002    3$:  BIS    2(SP),R2 ;SET IN SECTOR
4186 027222 030103    4$:  BIT    R1,R3    ;THIS SECTOR IN LIST???
4187 027224 001452          BEQ    5$         ;NO, NEXT
4188 027226 010200          MOV    R2,R0     ;COPY SECTOR
4189 027230 042700 177700    BIC    #177700,R0 ;ONLY SECTOR LEFT
4190 027234 020027 000050    CMP    R0,#40.   ;SECTOR OKAY???
4191 027240 002404          BLT    6$         ;YES
4192 027242 162700 000050    SUB    #40.,R0
4193 027246 162702 000050    SUB    #40.,R2    ;FIX SECTOR
4194 027252 004537 025404    6$:  JSR    R5,RSADJS ;FIND IF SECTOR PREVIOUSLY WRITTEN
4195 027256 000000          WORD  0
4196 027260 005700          TST    R0         ;WAS IT??
4197 027262 001433          BEQ    5$         ;NO
4198 027264 010237 002772    MOV    R2,BDA    ;LOAD DISK ADDRESS
4199 027270 012737 177600 002774  MOV    #-128.,BMP ;LOAD WC
4200 027276 004537 030550    JSR    R5,LDFUNC ;LOAD
4201 027302 000014          READ
4202 027304 005737 002664    TST    ERFLG     ;WAS READ GOOD
4203 027310 001420          BEQ    5$
4204
4205 027312 010346          MOV    R3,-(SP)
4206 027314 010237 002720    MOV    R2,SECT
    
```

```

4207 027320 010003          MOV      R0,R3
4208
4209 027322 042737 177700 002720      BIC      #177700,SECT
4210 027330          ERRHRD  140.,ADJTXT,ERR3
      (3) 027330 104463          TRAP    T$ERCODE
      (5) 027332 000214          .WORD  140
      (5) 027334 017717          .WORD  ADJTXT
      (5) 027336 020044          .WORD  ERR3
4211 027340 012603          MOV      (SP)+,R3
4212 027342          ERRHRD  110.,READ1,ERR2
      (3) 027342 104463          TRAP    T$ERCODE
      (5) 027344 000156          .WORD  110
      (5) 027346 017672          .WORD  READ1
      (5) 027350 020004          .WORD  ERR2
4213
4214 027352 005202          55:     INC      R2          ;NEXT SECTOR
4215 027354 000241          CLC
4216 027356 006001          ROR      R1          ;SHIFT MASK
4217 027360 103320          BCC     45
4218 027362 000205          RTS     R5
4219
4220
4221          ;ROUTINE TO MERGE BAD SECTOR FILES
4222          ;ENTRY INTO THIS ROUTINE WILL OCCUR AFTER THE "SERNUM" ROUTINE
4223          ;IS PERFORMED. THE FACTORY BAD SECTOR FILE WILL BE LOCATED IN
4224          ;FIRST 400(8) LOCATIONS.
4225          ;THIS ROUTINE WILL STORE THE FIELD BAD SECTORS INTO THE NEXT
4226          ;400 LOCATIONS AND THEN MERGE THE FACTORY BAD FILE
4227          ;WITH THE FIELD BAD FILE.
4228
4229          ;FACTORY BAD AT BUF
4230          ;FIELD BAD AT BUF + 512.
4231
4232 027364 010146          MERGE:  MOV      R1,-(SP)      ;SAVE P1, R2, R3
4233 027366 010246          MOV      R2,-(SP)
4234 027370 010346          MOV      R3,-(SP)
4235 027372 012737 003420 002770      MOV      #BUF+400,BBA      ;BUFFER START FOR FIELD BAD
4236 027400 012737 077724 002772      MOV      #77724,BDA       ;DA OF FIELD BAD SEC. START
4237 027406 012737 177400 002774      MOV      #-256.,BMP       ;SETUP TO READ TWO SECTORS
4238 027414 004537 030550          975:   JSR      R5,LDFUNC        ;LOAD READ FUNCTION
4239 027420 000014          READ
4240 027422 005737 002664          TST     ERFLG            ;TEST ERROR FLAG
4241 027426 001421          BEQ     985             ;YES;MERGE BAD SECTOR FILES
4242 027430 062737 000004 002772      ADD     #4,BDA          ;TRY NEXT FIELD BAD SECTOR FILE
4243 027436 022737 077750 002772      CMP     #77750,BDA      ;COMPLETED FIELD BAD SECTORS?
4244 027444 001363          BNE     975             ;NO,DO NEXT FIELD BAD SECTOR
4245 027446          PRINTF #7RM15
      (7) 027446 012746 021552      MOV     #7RM15,-(SP)
      (6) 027452 012746 000001      MOV     #1,-(SP)
      (3) 027456 010600          MOV     SP,R0
      (4) 027460 104017          EMT    C$PNTF
      (4) 027462 062706 000004      ADD     #4,SP
4246 027466          9995:  BREAK
      (3) 027466 104022          EMT    C$BRK
4247 027470 000776          BR     9995
4248 027472 012701 003030          985:   MOV     #BUF+10,R1      ;GET PAST ID ETC.

```

```
4249 027476 012702 000176      MOV      #126.,R2      ;MAX = 126
4250 027502 005721      15:     TST      (R1)+      ;SECTOR OR END
4251 027504 100404      BMI      25          ;END, GO GET FIELD
4252 027506 005721      TST      (R1)+      ;REST OF SECTOR
4253 027510 005302      DEC      R2          ;MAX REACHED
4254 027512 001373      BNE      15          ;NO, KEEP GOING
4255 027514 000401      BR       35          ;YES, SKIP BACK UP
4256 027516 005741      25:     TST      -(R1)      ;BACK UP PAST TERMINATOR
4257 027520 012703 000176      35:     MOV      #126.,R3      ;SET 126 MAX
4258 027524 012702 003430      MOV      #BUF+410,R2    ;GET FIELD SECTORS
4259 027530 012221      45:     MOV      (R2)+,(R1)+    ;MERGE AT END OF FACTORY
4260 027532 100403      BMI      55          ;DONE?
4261 027534 012221      MOV      (R2)+,(R1)+    ;NO, MERGE REST OF SECTOR
4262 027536 005303      DEC      R3          ;DONE
4263 027540 001373      BNE      45          ;NO, GO BACK
4264 027542 012603      55:     MOV      (SP)+,R3      ;RESTORE R3, R2, R1
4265 027544 012602      MOV      (SP)+,R2
4266 027546 012601      MOV      (SP)+,R1
4267 027550 000205      RTS      R5          ;EXIT
4268
4269
4270 027552 012537 002734      FNDTRK: MOV      (R5)+,OFFSET ;GET INCREMENT/DECREMENT
4271 027556 012537 002744      MOV      (R5)+,SURFACE  ;GET HEAD (SURFACE)
```

```

4273 027562 112537 002740      MOVB      (R5)+,FRTRK      ;BEGINNING TRACK
4274 027566 112537 002736      MOVB      (R5)+,LSTTRK     ;ENDING TRACK
4275 027572 005037 002746      CLR       TRKFND          ;CLEAR OUT FLAG FOUND
4276 027576 005037 002750      CLR       TRKCNT          ;CLEAR OUT TRACK COUNT
4277 027602 013737 002740 002742  MOV       FRTRK,PRSTRK    ;GET FIRST TRACK
4278 027610                                15:
4279 027610 004537 027674      JSR       R5,FNDBSC       ;IS TRACK IN BAD SECTOR FILE
4280 027614 005737 002112      TST      HDRFND          ;WAS IT?
4281 027620 001003                                BNE      25              ;YES, CLEAR TRKCNT
4282 027622 005237 002750      INC      TRKCNT          ;NO, INDICATE GOOD TRACK
4283 027626 000402                                BR       35              ;CONTINUE
4284 027630 005037 002750      CLR      TRKCNT          ;START COUNT OVER
4285 027634 023727 002750 000005 35:  CMP      TRKCNT,#5        ;FIND 5 TRACKS YET?
4286 027642 001003                                BNE      45              ;NO, CONTINUE
4287 027644 005237 002746      INC      TRKFND          ;YES, EXIT WITH GOOD FLAG
4288 027650 000205                                RTS      R5              ;EXIT
4289 027652 023737 002742 002736 45:  CMP      PRSTRK,LSTTRK    ;ARE WE DONE?
4290 027660 001001                                BNE      55              ;NO, KEEP LOOKING
4291 027662 000205                                RTS      R5              ;EXIT WITH NOT FOUND
4292 027664 063737 002734 002742 55:  ADD      OFFSET,PRSTRK    ;NEXT TRACK
4293 027672 000746      BR       15
4294
4295                                ;ROUTINE TO FIND BAD TRACK IN FILE
4296
4297 027674 005037 002112      FNDBSC: CLR      HDRFND      ;INITIALIZE FLAG
4298 027700 010146      MOV      R1,-(SP)        ;SAVE R1, R2
4299 027702 010246      MOV      R2,-(SP)
4300 027704 012701 003030      MOV      #BUF+10,R1      ;SETUP FOR BEGINNING OF FILE
4301 027710 005711                                15:  TST      (R1)            ;END?
4302 027712 100421                                BMI      25              ;IF MINUS AT END, EXIT
4303 027714 023721 002742      CMP      PRSTRK,(R1)+    ;CYLINDER CORRECT?
4304 027720 001011                                BNE      35              ;NO, NEXT
4305 027722 105724      TSTB    (R4)+            ;UPPER HALF OF WORD
4306 027724 123711 002744      CMPB    SURFACE,(R1)    ;CORRECT SURFACT
4307 027730 001402                                BEQ      45
4308 027732 105744      TSTB    -(R4)
4309 027734 000403                                BR       35
4310 027736 005237 002112      45:  INC      HDRFND          ;SET FOUND
4311 027742 000405      BR       25
4312
4313 027744 005721                                35:  TST      (R1)+            ;NEXT WORD
4314 027746 005202                                INC      R2              ;ACCOUNT FOR IT
4315 027750 020227 000374      CMP      R2,#252        ;DONE?
4316 027754 001355                                BNE      15              ;NO, KEEP CHECKING
4317 027756 012601                                25:  MOV      (SP)+,R1        ;RESTORE R2, R1, EXIT
4318 027760 012602      MOV      (SP)+,R2
4319 027762 000205      RTS      R5
4320
4321 027764 013701 002742      FIXCYL: MOV      PRSTRK,R1    ;GET TRACK WHICH IS GOOD
4322 027770 005737 002734      TST      OFFSET        ;WHICH WAY WERE WE LOOKING
4323 027774 100402                                BMI      15              ;IN WORD, BRANCH
4324 027776 162701 000004      SUB      #4,R1          ;BACK IT UP BY FOUR
4325 030002 012702 000005      15:  MOV      #5,R2          ;GOING STORE AWAY 5 TRACKS
4326 030006 110120      25:  MOVB    R1,(R0)+        ;STORE THEM
4327 030010 005201      INC      R1
4328 030012 005302      DEC      R2
  
```

```

4329 030014 001374          BNE 2$
4330 030016 000205          RTS  R5
4331
4332
4333          ;ROUTINE TO GET SERIAL NUMBER
4334
4335          ;CALL JSR R5,SERNUM
4336
4337 030020 012737 000013 002772 SERNUM: MOV #13,BDA
4338 030026 004537 030550          JSR R5,LDFUNC ;GET STATUS
4339 030032 000004          GSTAT
4340 030034 004537 030550          JSR R5,LDFUNC ;READ HEADER
4341 030040 000010          RDHDR
4342 030042 013700 002760          MOV E.MP,RO ;GET THE HEADER
4343 030046 042700 000077 1$: BIC #77,RO ;CLEAR SECTOR BITS
4344 030052 020027 077700          CMP RO,#77700 ;ON LAST TRACK
4345 030056 001425          BEQ 2$ ;IF SO, DON'T SEEK
4346 030060 042700 000100          BIC #100,RO ;CLEAR HEAD
4347 030064 012701 077600          MOV #77600,R1 ;LAST CYLINDER
4348 030070 160001          SUB RO,R1 ;CALCULATE DIF OF SEEK
4349 030072 010137 002772          MOV R1,BDA ;SET UP DIF WORD
4350 030076 052737 000025 002772 BIS #25,BDA ;SEEK IN, HEAD 1
4351 030104 004537 030550          JSR R5,LDFUNC ;SEEK
4352 030110 000006          SEEK
4353 030112 004537 030550          JSR R5,LDFUNC ;VERIFY POSITION
4354 030116 000010          RDHDR
4355 030120 013700 002760          MOV E.MP,RO ;GET HEADER
4356 030124 022700 077700          CMP #77700,RO ;COMPARE AGAINST LAST
4357 030130 003346          BGT 1$ ;IF WRONG, RE-SEEK
4358 030132 012737 077700 002772 2$: MOV #77700,BDA ;BAD SECTOR DA START
4359 030140 012737 003020 002770 97$: MOV #BUF,BBA ;READ IN BAD SECTOR FILE
4360 030146 012737 177400 002774          MOV #-256.,BMP
4361 030154 004537 030550          JSR R5,LDFUNC ;READ
4362 030160 000014          READ
4363 030162 005737 002664          TST ERFLG ;TEST ERROR FLAG
4364 030166 001410          BEQ 98$ ;YES,COMPARE SERIAL NUMBERS
4365 030170 062737 000004 002772 ADD #4,BDA ;NO,SETUP FOR NEXT FACTORY BAD SECTOR
4366 030176 022737 077724 002772 CMP #77724,BDA ;END OF FACTORY BAD FILE?
4367 030204 001355          BNE 97$ ;GET NEXT FACTORY BAD SECTOR
4368 030206 000445          BR 99$ ;REPORT ERROR
4369 030210 012701 003020 98$: MOV #BUF,R1 ;COMPARE SERIAL NUMBERS
4370 030214 005737 002776          TST SERNM1 ;HAVE WE GOT ONE TO COMPARE
4371 030220 100005          BPL 3$ ;YES, BRANCH
4372 030222 011137 002776          MOV (R1),SERNM1 ;NO, CALL THIS ONE IT
4373 030226 016137 000002 003000          MOV 2(R1),SERNM2 ;
4374 030234 021137 002776 3$: CMP (R1),SERNM1 ;SERNUM OKAY
4375 030240 001004          BNE 4$ ;NO, PRINT ERROR
4376 030242 026137 000002 003000 CMP 2(R1),SERNM2 ;OTHER HALF OKAY
4377 030250 001436          BEQ 5$ ;YES, EXIT
4378 030252 4$: PRINTF #FRM3,2(R1),(R1),SERNM2,SERNM1
(11) 030252 013746 002776          MOV SERNM1,-(SP)
(10) 030256 013746 003000          MOV SERNM2,-(SP)
(9) 030262 011146          MOV (R1),-(SP)
(8) 030264 016146 000002          MOV 2(R1),-(SP)
(7) 030270 012746 020702          MOV #FRM3,-(SP)
(6) 030274 012746 000005          MOV #5,-(SP)
  
```

(3)	030300	010600		MOV	SP, R0	
(4)	030302	104017		EMT	C\$PNTF	
(4)	030304	062706	000014	ADD	#14, SP	
4379	030310	004537	030350	JSR	R5, UNLOAD	; LET OPERATOR CHANGE
4380	030314	004537	030450	JSR	R5, LOAD	; PACK
4381	030320	000637		BR	SERNUM	; GO CHECK IT AGAIN.
4382	030322			99\$: PRINTF	#FRM15	; MESSAGE
(7)	030322	012746	021552	MOV	#FRM15, -(SP)	
(6)	030326	012746	000001	MOV	#1, -(SP)	
(3)	030332	010600		MOV	SP, R0	
(4)	030334	104017		EMT	C\$PNTF	
(4)	030336	062706	000004	ADD	#4, SP	
4383	030342			999\$: BREAK		
(3)	030342	104022		EMT	C\$BRK	
4384	030344	000776		BR	999\$	
4385	030346	000205		5\$: RTS	R5	
4386						
4387						

GLOBAL SUBROUTINES SECTION

```

4389 ;ROUTINE UNLOAD
4390
4391 ;CALL JSR R5,UNLOAD
4392
4393 UNLOAD: PRINTF #FRM1,<B,DSB+1(R4)>,CSR(R4)
(9) 030350 016446 000000 MOV CSR(R4),-(SP)
(8) 030354 005046 CLR -(SP)
(8) 030356 156416 000005 BISB DSB+1(R4),(SP)
(7) 030362 012746 020510 MOV #FRM1,-(SP)
(6) 030366 012746 000003 MOV #3,-(SP)
(3) 030372 010600 MOV SP,R0
(4) 030374 104017 EMT C$PNTF
(4) 030376 062706 000010 ADD #10,SP
4394 030402 012701 000074 MOV #60,R1 ;SETUP 60 SECOND TIMER
4395 030406 012700 000200 MOV #200,R0
4396 030412 056400 000004 BIS DSB(R4),R0
4397 030416 010074 000000 MOV R0,@CSR(R4)
4398 030422 032774 000001 000000 2$: BIT #DRDY,@CSR(R4) ;CHECK DRDY FOR ZERO
4399 030430 001406 BEQ 3$ ;PACK UNLOADED
4400 030432 WAITMS #10. ;WAIT 1 SECOND
(3) 030432 012700 000012 MOV #10.,R0
(3) 030436 104026 EMT C$WTM
4401 030440 005301 DEC R1 ;HAS 60 SEC PASSED?
4402 030442 001367 BNE 2$ ;NO, RETEST DRDY, CONTINUE WAIT
4403 030444 000741 BR UNLOAD ;YES, REPEAT MESSAGE CONTINUE WAIT
4404 030446 000205 3$: RTS R5 ;RETURN WITH PACK UNLOADED
4405
4406 ;ROUTINE LOAD
4407
4408 ;CALL JSR R5,LOAD
4409
4410 LOAD: PRINTF #FRM2,<B,DSB+1(R4)>,CSR(R4)
(9) 030450 016446 000000 MOV CSR(R4),-(SP)
(8) 030454 005046 CLR -(SP)
(8) 030456 156416 000005 BISB DSB+1(R4),(SP)
(7) 030462 012746 020605 MOV #FRM2,-(SP)
(6) 030466 012746 000003 MOV #3,-(SP)
(3) 030472 010600 MOV SP,R0
(4) 030474 104017 EMT C$PNTF
(4) 030476 062706 000010 ADD #10,SP
4411 030502 012701 000170 MOV #120,R1 ;SETUP 120 SEC TIMER
4412 030506 012700 000200 MOV #200,R0 ;SETUP CONTROLLER READY BIT
4413 030512 056400 000004 BIS DSB(R4),R0 ;SELECT DRIVE
4414 030516 010074 000000 MOV R0,@CSR(R4)
4415 030522 032774 000001 000000 2$: BIT #DRDY,@CSR(R4)
4416 030530 001006 BNE 3$
4417 030532 WAITMS #10.
(3) 030532 012700 000012 MOV #10.,R0
(3) 030536 104026 EMT C$WTM
4418 030540 005301 DEC R1
4419 030542 001367 BNE 2$
4420 030544 000741 BR LOAD

```

GLOBAL SUBROUTINES SECTION

```

4422 030546 000205      3$:   RTS      R5
4423
4424                   ;ROUTINE LDFUNC
4425                   ;CALL   JSR      R5, LDFUNC
4426
4427 030550 010046      LDFUNC: MOV     RD, -(SP)
4428 030552 010346      MOV     R3, -(SP)
4429 030554 010146      MOV     R1, -(SP)
4430 030556 005037 002664 CLR     ERFLG      ;CLEAR ERROR FLAG
4431 030562 016403 000000 MOV     CSR(R4), R3 ;GET CSR
4432 030566 013763 002774 000006 MOV     BMP, MP(R3) ;LOAD MULTIPURPOSE
4433 030574 013763 002772 000004 MOV     BDA, DA(R3) ;LOAD DISK ADDRESS
4434 030602 013763 002770 000002 MOV     BBA, BA(R3) ;LOAD BUS ADDRESS
4435 030610 011537 002766      MOV     (R5), BCS   ;GET FUNCTION TO LOAD
4436 030614 056437 000004 002766 BIS     DSB(R4), BCS ;SELECT BITS
4437 030622 012701 000031      MOV     #25, R1    ;SET WATCHDOG TO 250MS
4438 030626 052737 000200 002766 BIS     #200, BCS
4439 030634 013763 002766 000000 MOV     BCS, CS(R3) ;LOAD FUNCTION
4440 030642 016337 000000 002766 MOV     CS(R3), BCS
4441 030650 042763 000200 000000 BIC     #200, CS(R3)
4442 030656 032763 000200 000000 1$: BIT     #200, CS(R3) ;CNTRLR READY?
4443 030664 001034      BNE     2$        ;YES, GO
4444 030666      WAITUS #100.      ;WAIT 10 MILLISECONDS
(3) 030666 012700 000144      MOV     #100, R0
(3) 030672 104027      EMT     CSWTU
4445 030674 005301      DEC     R1
4446 030676 001367      BNE     1$
4447
4448 030700 016337 000000 002752 MOV     CS(R3), E. CS ;READ ALL REGISTERS
4449 030706 016337 000002 002754 MOV     BA(R3), E. BA
4450 030714 016337 000004 002756 MOV     DA(R3), E. DA
4451 030722 016337 000006 002760 MOV     MP(R3), E. MP
4452 030730 016337 000006 002762 MOV     MP(R3), E. MP1
4453 030736 016337 000006 002764 MOV     MP(R3), E. MP2
4454 030744      ERRDF 210, CNTTOT, ERR5; CNTRLR TIMEOUT
(3) 030744 104462      TRAP  TSERCODE
(5) 030746 000322      .WORD 210
(5) 030750 017062      .WORD CNTTOT
(5) 030752 020362      .WORD ERR5
4455 030754 000425      BR     4$
4456
4457 030756 016337 000000 002752 2$: MOV     CS(R3), E. CS ;READ ALL REGISTERS
4458 030764 016337 000002 002754 MOV     BA(R3), E. BA
4459 030772 016337 000004 002756 MOV     DA(R3), E. DA
4460 031000 016337 000006 002760 MOV     MP(R3), E. MP
4461 031006 016337 000006 002762 MOV     MP(R3), E. MP1
4462 031014 016337 000006 002764 MOV     MP(R3), E. MP2
4463
4464 031022 005737 002752      TST     E. CS      ;ANY ERRORS?
4465 031026 100002      BPL     3$        ;YES, GO SERVICE
4466 031030 005237 002664      4$: INC     ERFLG
4467 031034 005725      3$: TST     (R5)+
4468 031036 012601      MOV     (SP)+, R1
4469 031040 012603      MOV     (SP)+, R3
4470 031042 012600      MOV     (SP)+, R0
4471 031044 000205      RTS      R5
  
```

```

4472
4473
4474
4475 031046          ENDMOD
4476
4477
4478
4479
4480
4481 031046          BGNTST
4482
4483
4484                ; CONTROL SECTION COMPATABILITY PROGRAM
4485                ; PRINT UNLOAD AND LOAD DRIVE MESSAGES
4486                ; PERFORM SERIAL CHECK ROUTINE
4487                ; PERFORM READ/WRITE CHECKS ON DRIVES
4488
4489 031046 012701 002236  COMPAT: MOV      #SECBUF,R1      ; ADJ. CYLINDER BUFFER
4490 031052 012700 000170      MOV      #120.,R0      ; ADJ. CYLINDER BUFFER COUNT
4491 031056 005021          45:  CLR      (R1)+      ; CLEAR ADJ. CYL. BUFFER AT STARTUP
4492 031060 005300          DEC      R0          ; BUFFER CLEARED?
4493 031062 001375          BNE     45          ; CLEAR NEXT BUFFER WORD
4494 031064 005237 002652      INC     FOUR        ; SET FIRST OVERWRITE FLAG
4495 031070 004537 023324      JSR     R5,OVWPER    ; PERFORM OVERWRITE ON FIRST DRIVE
4496 031074 177400          177400
4497 031076 000377          377
4498 031100 005037 002652      CLR     FOUR        ; CLEAR FIRST OVERWRITE
4499 031104 005237 002654      INC     FADJ        ; SET FIRST ADJ. FLAG
4500 031110 005237 002674      INC     ADJDIR      ; UP = 1
4501 031114 004537 024006      JSR     R5,ADJCYL
4502 031120 003 377          .BYTE  3,377        ; TRACK AND SECTORS FOR
4503 031122 170000          .WORD  170000       ; INWARD SEEK
4504 031124 003 000          .BYTE  3,0          ; TRACK AND SECTORS FOR
4505 031126 007777          .WORD  7777        ; OUTWARD SEEK
4506 031130 000000          .WORD  0           ; TERMINATOR
4507 031132 004537 030350      JSR     R5,UNLOAD   ; UNLOAD PACK FROM DRIVE UNIT
4508 031136 062704 000010      ADD     #PAT+2,R4   ; UPDATE POINTER FOR NEXT DRIVE
4509 031142 004537 030450      JSR     R5,LOAD     ; LOAD INTO SECOND DRIVE UNIT
4510 031146 004537 030020      JSR     R5,SERNUM   ; CHECK PACK SERIAL NUMBER
4511 031152 004537 023324      JSR     R5,OVWPER   ; PERFORM R/W OVERWRITE
4512 031156 000360          360
4513 031160 000017          17
4514 031162 005237 002674      INC     ADJDIR
4515 031166 004537 024006      JSR     R5,ADJCYL
4516 031172 002 360          .BYTE  2,360        ; IN 1/0 OUTSIDE
4517 031174 000000          .WORD  0
4518 031176 002 017          .BYTE  2,17        ; OUT 1/0 OUTSIDE
4519 031200 000000          .WORD  0
4520 031202 004 360          .BYTE  4,360       ; IN 1/0 INSIDE
4521 031204 000000          .WORD  0
4522 031206 004 017          .BYTE  4,17        ; OUT 1/0 INSIDE
4523 031210 000000          .WORD  0
4524 031212 000000          .WORD  0
4525 031214 004537 030350      JSR     R5,UNLOAD   ; UNLOAD PACK FROM DRIVE UNIT
4526 031220 023727 002716 000002  CMP     UUT,#2      ; CHECK FOR > 2 DRIVES
4527 031226 001002          BNE     10$        ; YES, GO TO NEXT DRIVE

```

GLOBAL SUBROUTINES SECTION

4528	031230	000137	031644		JMP	25		; GO TO FIRST DRIVE
4529	031234	062704	000010	105:	ADD	#PAT+2, R4		; UPDATE DRIVE BUFFER FOR THIRD DRIVE
4530	031240	004537	030450		JSR	R5, LOAD		; LOAD PACK FOR THIRD DRIVE
4531	031244	004537	030020		JSR	R5, SERNUM		; CHECK SERIAL NUMBERS
4532	031250	004537	023324		JSR	R5, OVWPER		; PERFORM R/W OVERWRITE ON THIRD DRIVE
4533	031254	006014				6014		
4534	031256	001403				1403		
4535	031260	005237	002674		INC	ADJDIR		
4536	031264	004537	024006		JSR	R5, ADJCYL		
4537	031270	002	000		. BYTE	2, 0		; IN 2/0 OUTSIDE
4538	031272	170000			. WORD	170000		
4539	031274	002	000		. BYTE	2, 0		; OUT 2/0 OUTSIDE
4540	031276	007400			. WORD	7400		
4541	031300	004	000		. BYTE	4, 0		; IN 2/0 INSIDE
4542	031302	170000			. WORD	170000		
4543	031304	004	000		. BYTE	4, 0		; OUT 2/0 INSIDE
4544	031306	007400			. WORD	7400		
4545	031310	001	200		. BYTE	1, 200		; IN 2/1 OUTSIDE
4546	031312	000000			. WORD	0		
4547	031314	001	100		. BYTE	1, 100		; OUT 2/1 OUTSIDE
4548	031316	000000			. WORD	0		
4549	031320	005	200		. BYTE	5, 200		; IN 2/1 INSIDE
4550	031322	000000			. WORD	0		
4551	031324	005	100		. BYTE	5, 100		; OUT 2/1 INSIDE
4552	031326	000000			. WORD	0		
4553	031330	000000			. WORD	0		; TERMINATOR
4554	031332	004537	030350		JSR	R5, UNLOAD		; UNLOAD PACK ON THIRD DRIVE
4555	031336	023727	002716	000003	CMP	UUT, #3		; CHECK FOR > 3 DRIVES
4556	031344	001500			BEQ	15		; NO. GO TO 2ND DRIVE
4557	031346	062704	000010		ADD	#PAT+2, R4		; UPDATE DRIVE BUFFER FOR 4TH DRIVE
4558	031352	004537	030450		JSR	R5, LOAD		; LOAD PACK ON 4TH DRIVE
4559	031356	004537	030020		JSR	R5, SERNUM		; CHECK PACK ON FOURTH DRIVE
4560	031362	004537	023324		JSR	R5, OVWPER		; PERFORM R/W OVERWRITE
4561	031366	021040				21040		
4562	031370	010420				10420		
4563	031372	005237	002674		INC	ADJDIR		
4564	031376	004537	024006		JSR	R5, ADJCYL		
4565	031402	002	000		. BYTE	2, 0		; IN 3/0 OUTSIDE
4566	031404	000360			. WORD	360		
4567	031406	002	000		. BYTE	2, 0		; OUT 3/0 OUTSIDE
4568	031410	000017			. WORD	17		
4569	031412	004	000		. BYTE	4, 0		; IN 3/0 INSIDE
4570	031414	000360			. WORD	360		
4571	031416	004	000		. BYTE	4, 0		; OUT 3/0 INSIDE
4572	031420	000017			. WORD	17		
4573	031422	001	040		. BYTE	1, 40		; IN 3/1 OUTSIDE
4574	031424	000000			. WORD	0		
4575	031426	001	020		. BYTE	1, 20		; OUT 3/1 OUTSIDE
4576	031430	000000			. WORD	0		
4577	031432	005	040		. BYTE	5, 40		; IN 3/1 INSIDE
4578	031434	000000			. WORD	0		
4579	031436	005	020		. BYTE	5, 20		; OUT 3/1 INSIDE
4580	031440	000000			. WORD	0		
4581	031442	001	000		. BYTE	1, 0		; IN 3/2 OUTSIDE
4582	031444	100000			. WORD	100000		
4583	031446	001	000		. BYTE	1, 0		; OUT 3/2 OUTSIDE

4584	031450	040000			.WORD	40000	
4585	031452	005	000		.BYTE	5.0	; IN 3/2 INSIDE
4586	031454	100000			.WORD	100000	
4587	031456	005	000		.BYTE	5.0	; OUT 3/2 INSIDE
4588	031460	040000			.WORD	40000	
4589	031462	000000			.WORD	0	; TERMINATOR
4590	031464	004537	030350		JSR	R5, UNLOAD	; UNLOAD PACK FROM 4TH DRIVE
4591	031470	162704	000010		SUB	#PAT+2, R4	; SET DRIVE BUFFER FOR 3RD DRIVE
4592	031474	004537	030450		JSR	R5, LOAD	; LOAD PACK ON 3RD DRIVE
4593	031500	004537	030020		JSR	R5, SERNUM	; CHECK FOR PACK SERIAL NUMBER
4594	031504	004537	023324		JSR	R5, OVWPER	; PERFORM R/W OVERWRITE ON 3RD DRIVE
4595	031510	020000				20000	
4596	031512	010000				10000	
4597	031514	004537	024006		JSR	R5, ADJCYL	
4598	031520	001	000		.BYTE	1.0	; IN 2/3 OUTSIDE
4599	031522	000200			.WORD	200	
4600	031524	001	000		.BYTE	1.0	; OUT 2/3 OUTSIDE
4601	031526	000100			.WORD	100	
4602	031530	005	000		.BYTE	5.0	; IN 2/3 INSIDE
4603	031532	000200			.WORD	200	
4604	031534	005	000		.BYTE	5.0	; OUT 2/3 INSIDE
4605	031536	000100			.WORD	100	
4606	031540	000000			.WORD	0	; TERMINATOR
4607	031542	004537	030350		JSR	R5, UNLOAD	; UNLOAD PACK FROM 3RD DRIVE
4608	031546	162704	000010	15:	SUB	#PAT+2, R4	; SET DRIVE BUFFER FOR 2ND DRIVE
4609	031552	004537	030450		JSR	R5, LOAD	; LOAD PACK ON THIRD DRIVE
4610	031556	004537	030020		JSR	R5, SERNUM	; CHECK PACK SERIAL NUMBER
4611	031562	004537	023324		JSR	R5, OVWPER	; PERFORM R/W OVERWRITE ON 2ND DRIVE
4612	031566	004040				4040	
4613	031570	002020				2020	
4614	031572	004537	024006		JSR	R5, ADJCYL	
4615	031576	001	000		.BYTE	1.0	; IN 1/2 OUTSIDE
4616	031600	020000			.WORD	20000	
4617	031602	001	000		.BYTE	1.0	; OUT 1/2 OUTSIDE
4618	031604	010000			.WORD	10000	
4619	031606	005	000		.BYTE	5.0	; IN 1/2 INSIDE
4620	031610	020000			.WORD	20000	
4621	031612	005	000		.BYTE	5.0	; OUT 1/2 INSIDE
4622	031614	010000			.WORD	10000	
4623	031616	001	000		.BYTE	1.0	; IN 1/3 OUTSIDE
4624	031620	000040			.WORD	40	
4625	031622	001	000		.BYTE	1.0	; OUT 1/3 OUTSIDE
4626	031624	000020			.WORD	20	
4627	031626	005	000		.BYTE	5.0	; IN 1/3 INSIDE
4628	031630	000040			.WORD	40	
4629	031632	005	000		.BYTE	5.0	; OUT 1/3 INSIDE
4630	031634	000020			.WORD	20	
4631	031636	000000			.WORD	0	; TERMINATOR
4632	031640	004537	030350		JSR	R5, UNLOAD	; UNLOAD PACK FROM 2ND DRIVE
4633	031644	162704	000010	25:	SUB	#PAT+2, R4	; SET DRIVE BUFFER FOR 1ST DRIVE
4634	031650	004537	030450		JSR	R5, LOAD	; LOAD PACK INTO FIRST DRIVE UNIT
4635	031654	004537	030020		JSR	R5, SERNUM	; CHECK SERIAL NUMBER
4636	031660	004537	023324		JSR	R5, OVWPER	; PERFORM R/W OVERWRITE
4637	031664	021040				21040	
4638	031666	010420				10420	
4639	031670	004537	024006		JSR	R5, ADJCYL	

GLOBAL SUBROUTINES SECTION

4640	031674	001	010	. BYTE	1,10	; IN 0/1 OUTSIDE
4641	031676	000000		. WORD	0	
4642	031700	001	004	. BYTE	1,4	; OUT 0/1 OUTSIDE
4643	031702	000000		. WORD	0	
4644	031704	005	010	. BYTE	5,10	; IN 0/1 INSIDE
4645	031706	000000		. WORD	0	
4646	031710	005	004	. BYTE	5,4	; OUT 0/1 INSIDE
4647	031712	000000		. WORD	0	
4648	031714	001	000	. BYTE	1,0	; IN 0/2 OUTSIDE
4649	031716	004000		. WORD	4000	
4650	031720	001	000	. BYTE	1,0	; OUT 0/2 OUTSIDE
4651	031722	002000		. WORD	2000	
4652	031724	005	000	. BYTE	5,0	; IN 0/2 INSIDE
4653	031726	004000		. WORD	4000	
4654	031730	005	000	. BYTE	5,0	; OUT 0/2 INSIDE
4655	031732	002000		. WORD	2000	
4656	031734	001	000	. BYTE	1,0	; IN 0/3 OUTSIDE
4657	031736	000010		. WORD	10	
4658	031740	001	000	. BYTE	1,0	; OUT 0/3 OUTSIDE
4659	031742	000004		. WORD	4	
4660	031744	005	000	. BYTE	5,0	; IN 0/3 INSIDE
4661	031746	000010		. WORD	10	
4662	031750	005	000	. BYTE	5,0	; OUT 0/3 INSIDE
4663	031752	000004		. WORD	4	
4664	031754	000000		. WORD	0	; TERMINATOR
4665	031756	004537	030350	JSR	R5, UNLOAD	; UNLOAD PACK
4666	031762			PRINTF	#ENDPAS	; END OF PASS
(7)	031762	012746	021775	MOV	#ENDPAS, -(SP)	
(6)	031766	012746	000001	MOV	#1, -(SP)	
(3)	031772	010600		MOV	SP, R0	
(4)	031774	104017		EMT	CSPNTF	
(4)	031776	062706	000004	ADD	#4, SP	
4667	032002			BREAK		
(3)	032002	104022		EMT	C\$BRK	
4668	032004	000776		BR	3\$	
4669						
4670						
4671						
4672	032006			ENDTST		
(3)	032006			L10012:		
(3)	032006	104001		EMT	C\$ETST	
4673						
4674	032010			BGNMOD	HRDPRM	
4675						
4676	032010			BGNHRD		
(3)	032010	000025		. WORD	L10013-L\$HARD/2	
4677						
4678	032012			GPRML	RLMSG, RLCNT, 1, YES	
(4)	032012	004130		. WORD	T\$CODE	
(4)	032014	032064		. WORD	RLMSG	
(4)	032016	000001		. WORD	1	
4679	032020			GPRMA	CSRMSG, CSR, 0, 160000, 177776, YES	
(4)	032020	000031		. WORD	T\$CODE	
(4)	032022	032071		. WORD	CSRMSG	
(4)	032024	160000		. WORD	T\$LOLIM	
(4)	032026	177776		. WORD	T\$HILIM	

4680	032030				GPRMA	VECMMSG, VECT, 0, 0, 776, YES
(4)	032030	001031			. WORD	TSCODE
(4)	032032	032116			. WORD	VECMMSG
(4)	032034	000000			. WORD	TSL0LIM
(4)	032036	000776			. WORD	TSHILIM
4681	032040				GPRMD	BRMSG, PRIOR, 0, 340, 0, 7, YES
(4)	032040	002032			. WORD	TSCODE
(4)	032042	032105			. WORD	BRMSG
(4)	032044	000340			. WORD	340
(4)	032046	000000			. WORD	TSL0LIM
(4)	032050	000007			. WORD	TSHILIM
4682	032052				GPRMD	DRMSG, DRBT, 0, 03400, 0, 7, YES
(4)	032052	003032			. WORD	TSCODE
(4)	032054	032125			. WORD	DRMSG
(4)	032056	003400			. WORD	03400
(4)	032060	000000			. WORD	TSL0LIM
(4)	032062	000007			. WORD	TSHILIM
4683						
4684	032064				ENDHRD	
(2)					. EVEN	
(3)	032064				L10013:	
4685						
4686	032064	046122	030461	000	RLMSG:	. ASCIZ /RL11/
4687	032071	102	051525	040440	CSRMSG:	. ASCIZ /BUS ADDRESS/
	032076	042104	042522	051523		
	032104	000				
4688	032105	102	020122	042514	BRMSG:	. ASCIZ /BR LEVEL/
	032112	042526	000114			
4689	032116	042526	052103	051117	VECMMSG:	. ASCIZ /VECTOR/
	032124	000				

OUTERR MACY11 30(1046) 06-DEC-77 18:18 PAGE 89
DZPLFA P11 23-NOV-77 13:47 GLOBAL SUBROUTINES SECTION

SEQ 0062

4691	032125	104	044522	042526	DRMSG: .ASCIZ /DRIVE/
	032132	000			
4692		032134			. EVEN
4693					
4694	032134				ENDMOD
4695					
4696					
4697	032134				LASTAD
(2)					. EVEN
4698					
4699					

PDP-11 DIAGNOSTIC SUPERVISOR
DZRLFA.SUP 11-OCT-77 15:41

L 5
MACY11 30(1046) 06-DEC-77 18:18 PAGE 90
GLOBAL SUBROUTINES SECTION

SEQ 0063

4700
15996 063034
15998 071776
15999 071776 000000
16001 072000
16002 000200

.TITLE PDP-11 DIAGNOSTIC SUPERVISOR
.END.SUPV=. +2
.=71776
.WORD 0
X1X1=
.END 200

ABOFLA 032406 G	BIT9 = 001000 G	CSAAL 045630	CSUNBU= 000031	EMT. TR 032412 G
ABOPAS 032320 G	BLD. HW 040052	CSABRT= 000021	CSWTM = 000026	END 022206
ABO. FM 035262	BLOCK 055206	CSADR = 000020	CSWTU = 000027	ENDBUF 017060
ADJCYL 024006	BMP 002774	CSAU = 000054	DA = 000004	ENDPAS 021775
ADJDIR 002674	BRMSG 032105	CSBRK = 000022	DCKER 017164	END. OF 040740
ADJFLG 002672	BSEXIT 027136	CSBSEG= 000004	DCRC = 004000	END. SU= 063034
ADJLC2 003006	BSVWR 026760	CSBSUB= 000002	DECMSG 051376	EOP. CH 061512 G
ADJLC3 003010	BUF 003020	CSBUFF= 000030	DERR = 040000	EOP. FM 035276
ADJLC4 003012	BSAAB 041454	CSCEFG= 000046	DESCYL 002710	EOP. IN 037350
ADJLOC 002670	BSAAF 041366	CSCLEA= 000012	DIAG. T 032414 G	ERFLG 002664
ADJTRK 003002	CALLPC= 000022	CSCLP1= 000006	DIRC 002706	ERR = 100000
ADJTXT 017717	CALLPS= 000024	CSCVEC= 000036	DLT = 010000	ERRFND 017452
ADJUUT 003004	CALLSP= 000026	CSDCLN= 000044	DPDVD 062200 G	ERRFOR 045706
ALLOC 053052	CALLTC= 000030	CSDODU= 000053	DPMUL 062066 G	ERRHAN 044722
ASAAW 037172	CAL. CL 057570	CSDRPT= 000024	DRBT = 000006	ERR. HR 045500
ASAAZ 037206	CAL. TI 057626 G	CSDU = 000055	DRBUF 017020	ERR. SF 045504
ASAAZ 037214	CHECK 027160	CSEDIT= 000006	DRDY = 000001	ERR1 017746 G
ASAAZ 037230	CHKLUP 041470	CSERDF= 000002	DRMSG 032125	ERR1FO 045772
ASABA 037240	CHKSTR 053414	CSERHR= 000003	DRPCOD 023320 G	ERR2 020004 G
BA = 000002	CHKTTY 051502	CSERSF= 000001	DRST = 000013	ERR3 020044 G
BA16 = 000020	CHK. FO 033700	CSERSO= 000004	DRSTAT 002676	ERR4 020176 G
BA17 = 000040	CHK. MA 037630	CSESCA= 000010	DSB = 000004	ERR5 020362 G
BBA 002770	CHK. PC 044716	CSSEEG= 000005	DSPCOD 022036 G	ERR6 020422 G
BCS 002766	CHK. SW 033422	CSesub= 000003	DUNIT. 032324 G	ESC. PC 044714
BDA 002772	CHRCNT 052734	CSSETST= 000001	DVC. FT 045470	EXIT 023312
BDATA 002726	CH. FLA 037332	CSEXIT= 000032	DSAAG 046340	E. BA 002754
BGN. SU= 032134	CH. PAS 037354	CSGMAN= 000043	DSAAH 046356	E. CS 002752
BINMSG 051362	CLEAR. 040752	CSGPHR= 000042	DSAAI 051130	E. DA 002756
BIT0 = 000001 G	CLKACC 032316 G	CSGPRI= 000040	DSAAJ 051134	E. MP 002760
BIT00 = 000001 G	CLKBFR 057572	CSGTIM= 000052	DSAAK 051152	E. MP1 002762
BIT01 = 000002 G	CLKCNT 032314 G	CSINIT= 000011	DSAAL 051170	E. MP2 002764
BIT02 = 000004 G	CLKRES 061170 G	CSINLP= 000020	DSAAW 051200	FADJ 002654
BIT03 = 000010 G	CLKSER 061470 G	CSKWF= 000035	EF. CON= 000036 G	FEW 017202
BIT04 = 000020 G	CLKSON 032360 G	CSKWON= 000034	EF. NEW= 000035 G	FILL 052232
BIT05 = 000040 G	CLK. SE 037432	CSLOOP= 000100	EF. PWR= 000034 G	FILL. C 000204 G
BIT06 = 000100 G	CLNCOD 023314 G	CSMANI= 000051	EF. RES= 000037 G	FIXCYL 027764
BIT07 = 000200 G	CLR. MA 037706	CSMSG = 000023	EF. STA= 000040 G	FLAGS 032356 G
BIT08 = 000400 G	COMPENA 023270	CSPNTB= 000014	EFO1 = 000001 G	FLAGTA 055404
BIT09 = 001000 G	CNTTOT 017062	CSPNTF= 000017	EFO2 = 000002 G	FLAG. I 037416
BIT1 = 000002 G	CNVT 055646	CSPNTS= 000016	EFO3 = 000003 G	FLA. SE 055352
BIT10 = 002000 G	COMMTA 055466	CSPNTX= 000015	EFO4 = 000004 G	FLG. MA 037356
BIT11 = 004000 G	COMPAT 031046	CSPOIN= 000040	EFO5 = 000005 G	FNDBSC 027674
BIT12 = 010000 G	CONTCL 061250 G	CSQIO = 000377	EFO6 = 000006 G	FNDDRV 025552
BIT13 = 020000 G	CRDY = 000200	CSRDBU= 000007	EFO7 = 000007 G	FNDTRK 027552
BIT14 = 040000 G	CRLF 051564	CSREFG= 000050	EFO8 = 000010 G	FORM. T 046002
BIT15 = 100000 G	CRSET = 000002	CSREQT= 000045	EFO9 = 000011 G	FORSK 002714
BIT2 = 000004 G	CS = 000000	CSRESE= 000033	EFO10 = 000012 G	FOWR 002652
BIT3 = 000010 G	CSR = 000000	CSREVI= 000001	EFO11 = 000013 G	FREE 053310
BIT4 = 000020 G	CSRMSG 032071	CSRPT = 000025	EFO12 = 000014 G	FRM1 020510
BIT5 = 000040 G	CURR. T 032334 G	CSSEFG= 000047	EFO13 = 000015 G	FRM10 021260
BIT6 = 000100 G	CSAAD 044670	CSSPRI= 000041	EFO14 = 000016 G	FRM11 021324
BIT7 = 000200 G	CSAAE 044702	CSVVEC= 000037	EFO15 = 000017 G	FRM12 021361
BITS = 000400 G	CSAAK 045520	CSTPRI= 000013	EFO16 = 000020 G	FRM13 021440

FRM14	021523	G\$OFFS=	000400	ISCLN =	000041	LSINIT	022042 G	MODR	062000 G
FRM15	021552	G\$OFSI=	000376	ISDU =	000041	LSLADP	002076 G	MP =	000006
FRM16	021611	G\$PRMA=	000001	ISHRD =	000041	LSLAST	032134 G	MUL	061734 G
FRM17	021676	G\$PRMD=	000002	ISINIT=	000041	LSMREV	002012 G	NEWPRI	061460 G
FRM18	021732	G\$PRML=	000000	ISMOD =	000041	LSNAME	002000 G	NEXTAR	055570
FRM2	020605	G\$RADA=	000140	ISMSG =	000041	LSREPP	002054 G	NONE	017300
FRM3	020702	G\$RADB=	000000	ISPR =	000041	LSREV	002010 G	NO. CLK	034472
FRM4	020761	G\$RADD=	000040	ISRPT =	000041	LSRPC	002030 G	NO. FLA	055364
FRM5	021022	G\$RADF=	000200	ISSEG =	000041	LSRPCP	002050 G	NO. LPT	052702
FRM6	021071	G\$RADL=	000120	ISSRV =	000041	LSSTP	002060 G	NO. PTA	037162
FRM7	021112	G\$RADO=	000020	ISSUB =	000041	LSSTA	002066 G	NR =	000000
FRM8	021152	G\$RADT=	000100	ISTST =	000041	LSTIML	002022 G	NUMBIN	046026
FRM9	021216	G\$XFER=	000004	JSJMP =	000167	LSTIMU	002020 G	NUM. LA	046174
FRTTRK	002740	G\$YES =	000010	KBPTR	032166 G	LSTIM1	002016 G	NUM. UN	032532
FUNERR	017556	HCORED	037114	KBUF	032170 G	LSTSTI	002074 G	NUNITS	041442
FWD	017625	HCOREQ	036774	LDFUNC	030550	LSUNIT	002014 G	NXM =	020000
F\$AU =	000015	HCORET	032346 G	LINE. F	032410 G	L. CLK.	036720	NXTFOR	055640
F\$BGN =	000040	HCRC =	004000	LOAD	030450	L10000	020002	OCTMSG	051370
F\$CLEA=	000007	HDRFND	002112	LOAD. F	037352	L10001	020042	OFFSET	002734
F\$DU =	000016	HEAD =	000100	LOGMSG	051404	L10002	020174	OP1 =	002000
F\$END =	000041	HEAD01	002704	LPBFR	032164 G	L10003	020360	OQU10	002126
F\$HARD=	000004	HERTZ.	036734	LPCNTR	032162 G	L10004	020420	OQU11	002133
F\$HW =	000013	HNF =	010000	LPT. AD	036752	L10005	020506	OQU20	002127
F\$INIT=	000006	HOLDSP=	000020	LPT. RE	036746	L10006	022036	OQU21	002134
F\$JMP =	000050	HPTCOD	022022 G	LSI. RE	036742	L10007	023312	OQU30	002130
F\$MOD =	000000	HRDPRM	032010 G	LSTCLR	002660	L10010	023316	OQU31	002135
F\$MSG =	000011	HSFLG	002700	LSTDRV	002722	L10011	023322	OQU40	002131
F\$PWR =	000017	HSAB	056174	LSTTRK	002736	L10012	032006	OQU41	002136
F\$RPT =	000012	INAWR	027052	LUP	057474	L10013	032064	OQU50	002132
F\$SEG =	000003	ININIT	032336 G	LUP. AD	044720	MAJ. IN	032140 G	OQU51	002137
F\$SOFT=	000005	INITCO	022042 G	LSAUT	002070 G	MAJ. LO	057574	OSECT	002702
F\$SRV =	000010	INITIA	051412	LSCCP	002044 G	MAJ. US	032142 G	OUT10	002114
F\$SUB =	000002	INITWR	017102	LSCLEA	023314 G	MANY	017241	OUT11	002121
F\$SW =	000014	INIT. M	037754	LSDEPO	002011 G	MAN. TI	034442	OUT20	002115
F\$TEST=	000001	INIT. R	032150 G	LSDEVP	002052 G	MAP16	062436 G	OUT21	002122
GARBAG	052736	INN10	002164	LSDISP	022040 G	MASK. B	041466	OUT30	002116
GDATA	002724	INN11	002171	LSDR	002102 G	MASK. W	041464	OUT31	002123
GETCHR	051442	INN20	002165	LSDRCT	002062 G	MDHEDR	002000 G	OUT40	002117
GETCMN	055026	INN21	002172	LSDRS	002064 G	MEN. SI	036762	OUT41	002124
GETPAR	046520	INN30	002166	LSDRST	002102 G	MERGE	027364	OUT50	002120
GETSWI	054022	INN31	002173	LSDTP	002040 G	MID10	002140	OUT51	002125
GET. TW	053572	INN40	002167	LSDU	023320 G	MID11	002145	OVMES	017351
GLBDAT	002112 G	INN41	002174	LSDUT	002072 G	MID20	002141	OVWER	017512
GLBEQA	002112 G	INN50	002170	LSDVTY	002104 G	MID21	002146	OVWPER	023324
GLBERR	017746 G	INN51	002175	LSEF	002024 G	MID30	002142	OVWTRK	002616
GLBSUB	023324 G	INPUTA	052340	LSEXP1	002032 G	MID31	002147	OSAPTS=	000000
GLBTXT	017062 G	INTEN =	000100	LSEXP2	002034 G	MID40	002143	OSAU =	000000
G\$BIT =	000003	INTFOR	045636	LSEXP3	002036 G	MID41	002150	OSBGNR=	000000
G\$STAT =	000004	INVAL.	037062	LSHARD	032012 G	MID50	002144	OSBGNS=	000000
G\$EXCP=	000400	INVINT	045530	LSHPCP	002046 G	MID51	002151	OSDU =	000000
G\$HILI=	000002	INV. SW	033356	LSHPTP	002056 G	MIN. IN	032134 G	OSGNSW=	000000
G\$LOLI=	000001	IN. SUF	040724	LSHW	022024 G	MIN. US	032136 G	OSPOIN=	000001
G\$NO =	000000	ISAU =	000041	LSICP	002042 G	MK =	000001	OSPWR =	000000

PAR. JS 055100	RLCNT = 000010	SUP. PR 033346	TYPLIN 051626	VAL. LA 033332
PAR. LA 051072	RLMSG 032064	SURFAC 002744	TYPNUM 051214	VAL. SW 037370
PAT = 000006	RSADJS 025404	SVC CNT= 177777	TYPSTR 051646	VEC = 000002
PATLST 002642	RSTACK 061662 G	SVCGBL= 177777	TYP. ER 045510	VECMG 032116
PRINTC 052712	RSX. FL 037346	SVCHAN 041646	TY. UNI 040744	VECT = 000002
PRINTF 056214	SEARCH 053540	SVCINS= 000000	TSARGC= 000001	VEROD 026176
PRIOR = 000004	SECBUF 002236	SVCSTK= 177777	TS CODE= 003032	VEROW 025612
PRIO0 = 000000 G	SECLST 002176	SVCSUB= 177777	TSERCO= 000062	WCOUNT 002730
PRIO1 = 000040 G	SECT 002720	SVCTAG= 000000	TSERRN= 000322	WIDTH 046374
PRIO2 = 000100 G	SECWRD 002732	SVCTST= 177777	TSXCP= 000000	WRITE = 000012
PRIO3 = 000140 G	SEEK = 000006	SWCHAN 037154	TSHILI= 000007	WRIT1 017645
PRIO4 = 000200 G	SEGSTA 032362 G	SWITCH 055544	TSLOLI= 000000	WRSEC 023556
PRIO5 = 000240 G	SERNM1 002776	SW. PTA 037140	TSLSYM= 010000	XEQDIA 061546 G
PRIO6 = 000300 G	SERNM2 003000	SYS. FT 045460	TSMCAL= 177777	XEQSUB 061534 G
PRIO7 = 000340 G	SERNUM 030020	S%LSYM= 010000	TSNEST= 177777	XEQ. CL 041404
PRNTST 052602	SETLST 025344	TEMP 002656	TSNSKO= 000000	XEQ. CM 036712
PRO. CM 037324	SETUP 022260	TERMI 057564	TSNSK1= 000004	XEQ. IN 041066
PRSTRK 002742	SET. MA 037542	TERMLI 055372	TS\$AVL= 177777	XEQ. LA 035250
PTAB. S 032344 G	SHIFT 062520 G	TERMTA 051354	TS\$EGL= 177777	XEQ. OP 041160
PUTCHR 051416	SIGN = 000004	TEST. M 037264	TS\$SUBN= 000000	XEQ. PR 034502
PWR. FA 062672 G	SIZE. C 061376 G	TIMFLG 032312 G	TS\$TAGL= 177777	XEQ. TE 041224
PWR. FL 032146 G	SIZE. M 061314 G	TIM. CO 032144 G	TS\$TAGN= 010014	XTIME 060254 G
PWR. MS 063020	SIZ. TR 061454	TIM. OP 046000	TS\$TEMP= 000000	XTIMEN 061100
PWR. SA 063014	SKCYL 024706	TGO. MA 051334	TS\$TEST= 000001	XTIMST 060276
PWR. UP 063016	SKER 017606	TQU10 002152	TS\$TSTM= 177777	XXDP. D 037122
P. CLK. 036726	SKHS = 000020	TQU11 002157	TS\$TSTS= 000001	X\$ALWA= 000000
RDHDR = 000010	SPEC. U 037252	TQU20 002153	TS\$CLE= 010010	X\$FALS= 000040
READ = 000014	SPV. SE 033754	TQU21 002160	TS\$SDU = 010011	X\$OFFS= 000400
READ. P 057576 G	STARTC 061244 G	TQU30 002154	TS\$SHAR= 010013	X\$TRUE= 000020
READ1 017672	STFLG 002666	TQU31 002161	TS\$SHW = 010006	X1X1 = 072000
REASON 002662	STRCHR 052272	TQU40 002155	TS\$INI= 010007	\$BREG 037430
RECER 017532	STREQ. 037074	TQU41 002162	TS\$MSG= 010005	\$ENDAD 061520 G
RECMS 017404	STRT. T 037330	TQU50 002156	TS\$TES= 010012	\$SAV2 062564 G
REGBAC 062422 G	STSEC 003016	TQU51 002163	T1 031046 G	\$SAV3 062600 G
REGDMP 025134	STSEC1 003014	TRKCNT 002750	UNI. MA 037254	\$SAV4 062616 G
REGSAV 062406 G	ST. REQ 037250	TRKFND 002746	UNLOAD 030350	\$SAV5 062636 G
REQN. P 037334	ST. SET 033574	TST. AB 041600	USER. P 032340 G	= 072000
REQN. T 037326	SUNIT. 037336	TST. TO 033404	USER. T 032342 G	
REV 017635	SUPERV 035314	TYPEC 051730	UUT 002716	
REVSK 002712	SUPFLA 032322 G	TYPEPC 045624	VAJWR 026524	
RE. SET 033524	SUPV. T 032500 G	TYPFLA 055246	VAL ID. 032602	

ABS. 072000 000

ERRORS DETECTED: 0

DSKZ: DZRLFA, DSKZ: DZRLFA=DZRLFA. SML, DZRLFA. P11, DZRLFA. SUP
 RUN-TIME: 24 26 .8 SECONDS
 RUN-TIME RATIO: 174/51=3.4
 CORE USED: 17K (33 PAGES)