

# RL11/RLV11,RL01

CONTROLLER TEST PART 2  
MD-11-DZRLB-A

EP-DZRLB-A-DL  
COPYRIGHT © 1977  
FICHE 1 OF 1

DEC 1977  
**digital**  
MADE IN USA

This microfiche card contains a grid of frames, each representing a page of test data. The frames are arranged in approximately 10 rows and 10 columns. Each frame contains a table with multiple columns and rows of data, likely representing test results for different parameters or components. The data is printed in a small font, typical of microfiche technology. The overall layout is a dense grid of information, with some frames appearing to have headers or sub-headers. The data within the frames is too small to be legible in this image, but the structure suggests a systematic collection of test results.

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZRLB-A-D  
PRODUCT NAME: RL11/RLV11 RLO1 CONTROLLER TEST (PART 2)  
DATE CREATED: 28 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 RESPONSIBILITY 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 RL11/RLV11 CONTROLLER TEST (PART 2) IS A PDP-11 (LSI-11) BASED PROGRAM THAT WILL TEST THE CONTROLLER. IT COMPLIMENTS PART 1 BY EXTENDING THE TEST COVERAGE TO INCLUDE WRITE DATA, READ DATA, WRITE CHECK AND READ DATA WITHOUT HEADER COMPARE. IT IS AIMED AT FULLY TESTING THE CONTROLLER IN THESE AREAS, BUT BY DEFAULT ALSO EXERCISES THE DRIVE. THE TEST COVERAGE OF THE PROGRAM IS EXTREMELY HIGH.

## 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 - 8 RLO1 DRIVES  
 1 - 8 RLO1K CARTRIDGES WITH BAD SECTOR FILE  
 KW11P, KW11L (OPTIONAL)  
 LINEPRINTER(OPTIONAL)

### 1.2.2 SOFTWARE REQUIREMENTS

---

MAINDEC-11-DZRLB-A

### 1.3 RELATED DOCUMENTS AND STANDARDS

-----

RLO1 USERS MANUAL (EK-RLO1-UG-PRE)  
XXDP USERS MANUAL  
DIAGNOSTIC SUPERVISOR PROGRAM LISTING

### 1.4 DIAGNOSTIC HIERARCHY PREREQUISITES

-----

THE RLO1 SUBSYSTEM SHOULD HAVE SUCCESSFULLY RUN THE FOLLOWING PROGRAMS:

MD-11-DZRLA	RL11/RLV11 RLO1 CONTROLLER TEST (PART 1)
MD-11-DVRLA	RLV11 RLO1 DISKLESS TEST (RLV11 ONLY)

### 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 AND SOFTWARE 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, XXDP CHAIN,  
ACT, SLIDE AND APT.

## 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
PMT	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

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.

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 DIALOGUE. 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

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

2. 3. 4 PROCEED COMMAND

-----

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

\*\*\*\*\*

## 2. 3. 4. 1 FLAGS SWITCH (/FLAGS: &lt;FLAG-LIST&gt;)

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

## 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

-----

\*\*\*\*\*  
 CCI/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

THE PURPOSE OF THIS COMMAND IS TO CREATE A BIC FILE SUITABLE FOR CHAIN MODE EXECUTION. THE XXDP PROCEDURE IS AS FOLLOWS:

```

  INVOKE THE XXDP UTILITY UPD1
  LOAD XXN: FILE. BIN
  START 200
  <QUESTIONS AND ANSWERS>
  RESTART UPD1 USING RESTART ADDRESS
  MICORE ADDRESS (IF "PASSED 14.5K" MESSAGE CAME)
  DUMP XXN: FILE. BIC
  
```

THE OPERATOR DIALOGUE (HARDWARE AND SOFTWARE) WILL BE EXECUTED AS IN THE START COMMAND, BUT AT THE END OF THE QUESTIONS THE HALT STATE WILL BE ENTERED, AT WHICH TIME THE OPERATOR SHOULD DUMP THE PROGRAM TO THE XXDP LIBRARY USING A BIC EXTENSION TO INDICATE THAT THIS FILE IS CHAINABLE. HE SHOULD USE THE XXDP UTILITY "UPD1" TO DO THIS. IF THE P-TABLES EXTEND BEYOND 14.5K, A MESSAGE WILL BE ISSUED GIVING THE NEW UPPER CORE LIMIT, TO WHICH THE OPERATOR MUST ADJUST BEFORE DUMPING. HE MAY NOW DELETE THE NON-CHAINABLE BIN FILE IF DESIRED, SINCE THE BIC FILE HAS ALL THE CAPABILITIES OF IT.

WHEN THIS BIC FILE IS SUBSEQUENTLY EXECUTED IN CHAIN MODE, THE OPERATOR DIALOGUES WILL BE BYPASSED. HOWEVER, IF IT IS EXECUTED STANDALONE, THE DIALOGUE WILL BE REISSUED.

NOTE THAT IF THE MESSAGE "TOO MANY UNITS" IS ISSUED, TWO OR MORE CORE IMAGES MUST BE CREATED (WITH DIFFERENT NAMES) TO TEST ALL UNITS.

## 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

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  
-----

\*\*\*\*\*  
PR!(NT)  
\*\*\*\*\*

2. 3. 8. 1 EFFECT OF COMMAND

ALL STATISTICS TABLES ACCUMULATED BY THE DIAGNOSTIC ARE PRINTED. THE ISR (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

J 1

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.

SEQ 0009

## 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 CONTROL CHARACTERS

-----

A CONTROL C ( C ) ENTERED VIA THE CONSOLE DEVICE DURING THE  
EXECUTION OF A DIAGNOSTIC CAUSES A RETURN TO THE DIAGNOSTIC  
SUPERVISOR COMMAND MODE.

A CONTROL Z ( Z ) ENTERED WITHIN ON OF THE THREE OPERATOR DIALOGUES  
(HARDCORE, HARDWARE, OR SOFTWARE QUESTIONS) CAUSES THE DEFAULT  
VALUES TO BE TAKEN FOR THE REMAINDER OF THAT DIALOGUE.

A CONTROL O ( O ) ENTERED DURING THE EXECUTION OF A DIAGNOSTIC CAUSES  
ALL CONSOLE DEVICE OUTPUT TO BE SUPPRESSED FOR THE REMAINDER OF  
THE DIAGNOSTIC OR UNTIL ANOTHER CONTROL O IS TYPED.

## 2. 3. 13 HARDWARE PARAMETERS

-----

THE FOLLOWING QUESTIONS WILL BE ASKED ON A START COMMAND. THE  
VALUE LOCATED TO THE LEFT OF THE QUESTION MARK IS THE DEFAULT  
VALUE THAT WILL BE TAKEN ON A CARRIAGE RETURN RESPONSE.

RL11 (L) Y?

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

BUS ADDRESS (O) 174400?

ANSWER WITH THE BUS ADDRESS OF THE CONTROLLER.

VECTOR (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. 14 SOFTWARE PARAMETERS

-----

THE FOLLOWING QUESTIONS ARE ASKED ON A START, RESTART OR CONTINUE  
IF THE QUESTION:

CHANGE SW?

IS ANSWERED YES(Y). THE QUESTIONS ARE:

DROP ON ERROR LIMIT (L) Y?

TO ALLOW THE UNIT TO BE DROPPED ONCE A PREDETERMINED NUMBER OF  
ERRORS ARE ENCOUNTERED.

ANSWER Y OR N

ERROR LIMIT (D) 10?

NUMBER OF ERRORS ALLOWED BEFORE DROPPING UNIT.

ANSWER 1 TO 65K

AUTOSIZE (L) N?

TO CHECK TO SEE IF UNIT SPECIFIED ACTUALLY EXISTS BEFORE TESTING  
IT (VIA DRIVE READY), IF NOT UNIT WILL NOT BE TESTED.

ANSWER Y OR N

COMPARE DATA ON DCK (L) N?

WHEN A DATA CHECK IS ENCOUNTERED AND DATA IS KNOWN, ALLOW AN  
INCORE COMPARISON OF DATA.

ANSWER Y OR N

# OF WORDS IN ERROR REPORTED (D) 3?

NUMBER OF MISCOMPARES TO BE PRINTED ON CONSOLE DEVICE.

ANSWER 0 - 128

#### 2. 3. 15 EXTENDED DISCUSSION OF P-TABLE DIALOGUE

-----

L 1

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

SEQ 0011

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

-----

ONE PASS OF THE PROGRAM TAKES APPROXIMATELY 90 SECONDS.

## 3.0 ERROR INFORMATION

-----

### 3.1 ERROR REPORTING

-----

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  
 RRRRRR 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 COMMAND: CS: XXXXXX BA: XXXXXX DA: XXXXXX MP: XXXXXX  
 TIME OF ERROR: CS: XXXXXX BA: XXXXXX DA: XXXXXX MP: XXXXXX XXXXXX XXXXXX

REGISTER DESCRIPTIONS CAN BE FOUND IN SECTION 5.0.

### 3.2 ERROR HALTS

-----

N 1

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

SEQ 0013

#### 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 (XXXXXD)

-----

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

BIT 0 - DRIVE READY

RLBA - BUS ADDRESS REGISTER (XXXXX2)

-----

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

RLDA - DISK ADDRESS REGISTER (XXXXX4)  
-----

FOR READ/WRITE FUNCTIONS  
-----

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

FOR SEEK FUNCTION  
-----

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

FOR GET STATUS FUNCTION  
-----

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

RLMP - MULTIPURPOSE REGISTER  
-----

FOR READ/WRITE FUNCTION  
-----

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

FOR READ HEADER FUNCTION  
-----

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

FOR GET STATUS FUNCTION  
-----

HAS DRIVE STATUS

BIT 15 - WRITE DATA ERROR  
BIT 14 - CURRENT HEAD ERROR(CHE)  
BIT 13 - WRITE LOCK STATUS(WL)  
BIT 12 - SEEK TIME OUT(SKTO)  
BIT 11 - SPIN ERROR(SPE)  
BIT 10 - WRITE GATE ERROR(WGE)  
BIT 9 - VOLUME CHECK(VC)

BIT 8 - DRIVE SELECT ERROR(DSE)  
BIT 7 - RESERVED(0)  
BIT 6 - SURFACE  
BIT 5 - COVER OPEN  
BIT 4 - HEADS HOME  
BIT 3 - BRUSHES HOME  
BIT 2-0 - STATE BITS  
    0 - LOAD STATE  
    1 - SPIN UP  
    2 - BRUSH CYCLE  
    3 - LOAD HEADS  
    4 - SEEK - TRACK COUNTING  
    5 - SEEK - LINEAR MODE  
    6 - UNLOAD HEADS  
    7 - SPIN DOWN

## 6.0 TEST SUMMARIES

-----

### TEST 01 - WRITE NPR INTEGRITY

THIS TEST WILL VERIFY THAT THE WRITE FUNCTION WILL NOT CAUSE A BUS TRAP THEREFORE VERIFYING THE NPR LOGIC BETWEEN THE CONTROLLER AND PROCESSOR.

### TEST 02 - WRITE FUNCTION

THIS TEST WILL VERIFY THAT THE WRITE FUNCTION WILL RESET CONTROLLER READY AND POST NO ERRORS.

### TEST 03 - WRITE FUNCTION INTERRUPT

THIS TEST WILL VERIFY THAT THE WRITE FUNCTION WILL GENERATE AN INTERRUPT ON COMPLETION.

### TEST 04 - PROPER INCREMENT OF RLBA ON WRITE

THIS TEST WILL VERIFY THAT THE BUS ADDRESS REGISTER INCREMENTS PROPERLY ON A WRITE FUNCTION.

### TEST 05 - PROPER INCREMENT OF RLDA ON WRITE

THIS TEST WILL VERIFY THAT THE DISK ADDRESS REGISTER INCREMENTS PROPERLY ON A WRITE FUNCTION.

### TEST 06 - FORCE HEADER NOT FOUND WITH WRITE

THIS TEST WILL FORCE A HEADER NOT FOUND ERROR ON A WRITE. THE RLDA IS SET UP TO LOOK FOR SECTOR 40, A WRITE IS THEN ISSUED. THE HEADER NOT FOUND ERROR SHOULD THEN SET.



**TEST 07 - FORCE INTERRUPT WITH HNF**

THIS TEST WILL FORCE A HEADER NOT FOUND ERROR UNDER INTERRUPT CONTROL.

**TEST 08 - CHECK OPI TIME WITH HNF**

THIS TEST WILL TIME THE SETTING OF HNF (OPI) FROM ISSUANCE. THIS IS DONE BY ISSUING A WRITE TO SECTOR 40. THE TIME OF OPI SHOULD BE AROUND 200 MILLISECONDS.

**TEST 09 - MULTIPLE SECTOR TRANSFER ON WRITE**

THIS TEST THE ABILITY FOR THE WRITE FUNCTION TO WRITE MORE THAN ONE SECTOR. WE SET UP FOR A TWO SECTOR WRITE.

**TEST 10 - CHECK DIRECTION OF WRITE NPR**

THIS TEST WILL VERIFY THAT THE NPR DIRECTION OF A WRITE FUNCTION IS FROM MEMORY TO THE CONTROLLER. THIS IS DONE BY WRITING A PATTERN IN MEMORY AND ISSUING A WRITE, THEN CHECKING MEMORY TO VERIFY THAT IT DID NOT GET DISTURBED.

**TEST 11 - CHECK FULL INCREMENT OF RLBA**

THIS TEST WILL CHECK THAT THE RLBA CAN INCREMENT OF THE FULL 16 BIT RANGE. THIS IS DONE BY ISSUING A ONE WORD WRITE TO CHECK EACH BIT TOGGLE FROM 1-0 AND 0-1. THIS IS DONE FROM 0 TO 177776 REGARDLESS OF MEMORY SIZE.

**TEST 12 - BA BIT 16 INCREMENT**

THIS TEST WILL CHECK THAT BUS ADDRESS BIT 16 WILL SET WHEN THE RLBA IS 177776. AND THAT THE RLBA GOES TO 0.

**TEST 13 - BA BIT 17 INCREMENT**

THIS TEST WILL CHECK THAT BUS ADDRESS BIT 17 WILL SET WHEN BIT 16 AND THE RLBA ARE SET. THE RLBA AND BIT 16 ARE CHECKED TO GO TO ZERO.

**TEST 14 - READ NPR INTEGRITY**

THIS TEST WILL VERIFY THAT THE READ FUNCTION WILL NOT CAUSE A BUS TRAP THEREFORE VERIFYING THE NPR LOGIC BETWEEN THE CONTROLLER AND PROCESSOR.

**TEST 15 - READ FUNCTION**

THIS TEST WILL VERIFY THAT THE READ FUNCTION WILL RESET

CONTROLLER READY AND POST NO ERRORS.

E 2

SEQ 0017

TEST 16 - READ FUNCTION INTERRUPT

THIS TEST WILL VERIFY THAT THE READ FUNCTION WILL GENERATE AN INTERRUPT ON COMPLETION.

TEST 17 - CHECK DIRECTION OF READ NPR

THIS TEST WILL VERIFY THAT THE NPR DIRECTION OF A READ FUNCTION IS FROM CONTROLLER TO THE MEMORY. THIS IS DONE BY WRITING A PATTERN IN MEMORY AND ISSUING A READ, THEN CHECKING MEMORY TO VERIFY THAT IT DID NOT GET DISTURBED.

TEST 18 - PROPER INCREMENT OF RLBA ON READ

THIS TEST WILL VERIFY THAT THE BUS ADDRESS REGISTER INCREMENTS PROPERLY ON A READ FUNCTION.

TEST 19 - PROPER INCREMENT OF RLDA ON READ

THIS TEST WILL VERIFY THAT THE DISK ADDRESS REGISTER INCREMENTS PROPERLY ON A READ FUNCTION.

TEST 20 - FORCE HEADER NOT FOUND WITH READ

THIS TEST WILL FORCE A HEADER NOT FOUND ERROR ON A READ. THE RLDA IS SET UP TO LOOK FOR SECTOR 40, A READ IS THEN ISSUED. THE HEADER NOT FOUND ERROR SHOULD THEN SET.

TEST 21 - FORCE INTERRUPT WITH HNF

THIS TEST WILL FORCE A HEADER NOT FOUND ERROR UNDER INTERRUPT CONTROL.

TEST 22 - CHECK HEADER COMPARE LOGIC

THIS TEST WILL EXTENSIVELY CHECK THE CYLINDER AND HEAD BITS OF THE HEADER WORD TO COMPARE CORRECTLY. THIS IS DONE BY WALKING AND GROWING 0'S AND 1'S THRU THE PROPER RLDA BITS AND ISSUING READ TO SEE IF ALL BIT POSITIONS CAN COMPARE.

TEST 23 - MULTIPLE SECTOR TRANSFER ON READ

THIS TEST THE ABILITY FOR THE READ FUNCTION TO WRITE MORE THAN ONE SECTOR. WE SET UP FOR A TWO SECTOR READ.

TEST 24 - FORCE HNF AT END OF TRACK

THIS TEST WILL CHECK THE ABILITY TO DETECT HEADER NOT FOUND AT THE END OF A TRACK. THIS DONE BY SETTING UP FOR A TWO

TEST 25 - FORCE NON-EXISTANT MEMORY ERROR

THIS TEST WILL CHECK THAT THE NON-EXISTANT MEMORY ERROR (NXM) CAN SET. WE WILL ISSUE A READ TO THE MAXIMUM ADDRESS AND EXPECT A NXM ERROR. (THIS TEST WILL NOT BE DONE ON A 128K MACHINE.)

TEST 26 - FORCE NXM UNDER INTERRUPT

THIS TEST WILL ATTEMPT TO FORCE AN INTERRUPT VIA NXM. (THIS TEST WILL NOT BE DONE ON A 128K MACHINE.)

TEST 27 - CHECK READ WRITE LOOP

THIS TEST WILL WRITE A PATTERN TO SECTOR 0 AND TRY TO RECOVER IT WITH A WRITE.

TEST 28 - CHECK OF SILO LINES

THIS TEST WILL CHECK THAT WE CAN WRITE AND READ UNIQUE BIT PATTERNS VERIFY THAT THE LINES ON THE SILO ARE NOT STUCK OR TIED TOGETHER. THIS IS DONE WITH WALKING AND GROWING 0'S AND 1'S.

TEST 29 - CHECK THROUGHPUT OF SILO

THIS TEST WILL ATTEMPT TO CHECK THAT THE FALL THROUGH OF THE SILO IS WORKING CORRECTLY. WE WRITE A SECTOR OF 128 UNIQUE PATTERNS AND READ IT BACK CHECKING THAT EACH LOCATION IS UNIQUE AND CORRECT.

TEST 30 - CHECK ZERO FILL ON WRITE

THIS TEST WILL CHECK THE ABILITY OF THE CONTROLLER TO FILL THE REMAINING SECTOR WITH ZEROS ON A WRITE. WE WRITE A SECTOR WITH FROM 1 TO 127 WORDS, READ IT BACK AND VERIFY THAT THE NON WRITTEN WORDS ARE ZERO.

TEST 31 - CHECK SECTOR BITS ON HEADER COMPARE

THIS TEST WILL CHECK THAT THE SECTOR BITS CAN COMPARE CORRECTLY. THIS IS DONE BY WRITING THE SECTORS ADDRESS INTO THE SECTOR FOR A FULL TRACK. EACH SECTOR IS READ TO VERIFY THE SECTOR HAS THE CORRECT DATA, IF NOT THEN THE SECTOR BITS ARE NOT COMPARING CORRECTLY.

TEST 32 - WRITE CHECK NPR INTEGRITY

THIS TEST WILL CHECK THAT THE WRITE CHECK WILL FUNCTION WITHOUT CAUSING A BUS TRAP. TEST IS SET UP TO HANDLE BUS TRAPS.

## TEST 33 - WRITE CHECK FUNCTION

THIS TEST WILL CHECK THAT A WRITE CHECK FUNCTION WILL COMPLETE WITH THE SPECIFIED TIME WITHOUT POSTING ERRORS.

## TEST 34 - WRITE CHECK FUNCTION INTERRUPT

THIS TEST WILL CHECK THAT AN INTERRUPT CAN BE GENERATED FROM ISSUING A WRITE CHECK.

## TEST 35 - PROPER INCREMENT OF RLBA ON WRITE CHECK

THIS TEST WILL CHECK THAT THE RLBA INCREMENTS PROPERLY DURING A WRITE CHECK.

## TEST 36 - PROPER INCREMENT OF RLDA ON WRITE CHECK

THIS TEST WILL CHECK THAT THE RLDA INCREMENTS PROPERLY DURING A WRITE CHECK.

## TEST 37 - MULTIPLE SECTOR WRITE CHECK

THIS TEST WILL CHECK THAT WE CAN WRITE CHECK MORE THAN ONE SECTOR AT A TIME.

## TEST 38 - FORCE DCK WITH WRITE CHECK

THIS TEST WILL CHECK THAT WE CAN DETECT A DCK DURING A WRITE CHECK. THIS IS DONE BY MODIFYING MEMORY BETWEEN A WRITE AND A WRITE CHECK.

## TEST 39 - FORCE DCK WITH WRITE CHECK INTERRUPT

THIS TEST WILL CHECK THAT A DCK DURING A WRITE CHECK WILL CAUSE AN INTERRUPT TO OCCUR.

## TEST 40 - CHECK ZERO FILL ON WRITE WITH WRITE CHECK

THIS TEST WILL VERIFY THAT WE CAN SUCCESSFULLY WRITE CHECK ALL WORD COUNTS FROM 1 - 127.

## TEST 41 - 42 - EXTENDED CHECK OF WRITE CHECK

THESE TESTS VERIFY THAT WE CAN WRITE CHECK SUCCESSFULLY ALL PATTERNS. PATTERNS USED ARE WALKING 1'S, 0'S, GROWING 1'S, 0'S.

## TEST 43 - READ WITHOUT HEADER COMPARE

THIS TEST VERIFIES THAT THE FUNCTION READ WITHOUT HEADER COMPARE (7) RESETS THE CONTROLLER READY AND POSTS NO ERRORS. THE DISK ADDRESS IS SET TO ALL ONES.

## TEST 44 - READ WITHOUT HEADER COMPARE INTERRUPT

THIS TEST WILL VERIFY THAT THE FUNCTION READ WITHOUT HEADER COMPARE (7) CAN GENERATE AN INTERRUPT ON COMPLETION.

## TEST 45 - CHECK RD W/O HDR CMP READS

THIS TEST CHECKS THAT THE FUNCTION CAN ACTUALLY RECOVER DATA. WE WRITE A PATTERN IN MEMORY AND CHECK THAT THE FUNCTION CAN OVERLAY IT WITH DATA.

## TEST 46 - CHECK RLBA INCREMENT WITH RD W/O HDR CMP

THIS TEST CHECKS THAT THE RLBA CAN INCREMENT PROPERLY ON THE FUNCTION.

## TEST 47 - CHECK RLDA DOES INCREMENT

THIS TEST CHECKS THAT THE RLDA DOES INCREMENT WITH THE FUNCTION READ WITHOUT HEADER COMPARE.

-----

[The remainder of the page is blank, containing only faint smudges and artifacts.]

2893	GLOBAL DATA
2958	LIST TO CHECK HEADER COMPARE LOGIC
3025	BUFFER FOR READ/WRITE
3031	GLOBAL TEXT
3138	GLOBAL ERRORS
3384	INITIALIZATION CODE
3525	GLOBAL SUBROUTINES
3559	ROUTINE TO CHECK FOR CONTROLLER ERRORS
3621	LOAD RLCS
3858	**TEST 1** - WRITE NPR INTEGRITY
3909	**TEST 2** - WRITE FUNCTION
3965	**TEST 3** - WRITE FUNCTION INTERRUPT
4007	**TEST 4** - PROPER INCREMENT OF RLBA ON WRITE
4050	**TEST 5** - PROPER INCREMENT OF RLDA ON WRITE
4093	**TEST 6** - FORCE HEADER NOT FOUND WITH WRITE
4136	**TEST 7** - FORCE HEADER NOT FOUND WITH WRITE INTERRUPT
4192	**TEST 8** - CHECK OPI TIME WITH HDR NT FND
4255	**TEST 9** - MULTIPLE SECTOR TRANSFER ON WRITE
4308	**TEST 10** - CHECK DIRECTION OF WRITE NPR
4366	**TEST 11** - CHECK FULL RLBA INCREMENT
4416	**TEST 12** - BA BIT 16 INCREMENT
4472	**TEST 13** - BA BIT 17 INCREMENT
4528	**TEST 14** - TEST READ NPR INTEGRITY
4571	**TEST 15** - READ FUNCTION
4605	**TEST 16** - READ FUNCTION INTERRUPT
4645	**TEST 17** - CHECK READ NPR DIRECTION
4707	**TEST 18** - PROPER INCREMENT OF RLBA ON READ
4747	**TEST 19** - PROPER INCREMENT OF RLDA ON READ
4789	**TEST 20** - FORCE HEADER NOT FOUND WITH READ
4828	**TEST 21** - FORCE HEADER NOT FOUND WITH READ INTERRUPT
4877	**TEST 22** - CHECK HEADER COMPARE LOGIC
5008	**TEST 23** - CHECK MULTIPLE SECTORS ON READ
5067	**TEST 24** - FORCE HDR NT FND AT END OF TRACK
5103	**TEST 25** - FORCE NON-EXISTANT MEMORY ERROR
5146	**TEST 26** - FORCE NON-EXISTANT MEMORY ERROR INTERRUPT
5193	**TEST 27** - CHECK READ WRITE LOOP
5279	**TEST 28** - CHECK SILO LINES
5376	**TEST 29** - CHECK THROUGHPUT OF SILO
5472	**TEST 30** - CHECK ZERO FILL ON WRITE
5577	**TEST 31** - CHECK SECTOR BITS OF HEADER COMPARE
5689	**TEST 32** - WRITE CHECK NPR INTEGRITY
5772	**TEST 33** - WRITE CHECK FUNCTION
5837	**TEST 34** - WRITE CHECK FUNCTION INTERRUPT
5908	**TEST 35** - PROPER INCREMENT OF RLBA ON WRITE CHECK
5981	**TEST 36** - PROPER INCREMENT OF RLDA ON WRITE CHECK
6054	**TEST 37** - MULTIPLE SECTOR WRITE CHECK
6140	**TEST 38** - FORCE DCK WITH WRITE CHECK
6213	**TEST 39** - FORCE DCK WITH WRITE CHECK INTERRUPT
6297	**TEST 40** - CHECK ZERO FILL ON WRITE WITH WRITE CHECK
6374	**TEST 41** - EXTENDED CHECK OF WRITE CHECK FUNCTION
6455	**TEST 42** - EXTENDED CHECK OF WRITE CHECK FUNCTION
6536	**TEST 43** - READ WITHOUT HEADER COMPARE FUNCTION
6566	**TEST 44** - READ WITHOUT HEADER COMPARE FUNCTION INTERRUPT
6602	**TEST 45** - CHECK RD W/O HDR CMP ACTUALLY READS
6664	**TEST 46** - CHECK RLBA INCREMENT WITH RD W/O HDR CMP
6710	**TEST 47** - CHECK RLDA DOES INCREMENT WITH RD W/O HDR CMP

2806			. ENABLE AMA
2807			. ENABLE ABS
2808			. NLIST ME, CND, MD
2809			
2810			
2812			
2823			
2824			
2825	002000		. =2000
2826			
2827	002000		SVC
2828		000000	SVC INS=0
2829		000000	SVCTAG=0
2830			
2831	002000		POINTER BGNSW, BGNSFT, BGNDU
2832			
2833	002000		BGNMOD MDHEDR
2834			
2835	002000		HEADER DZRLB, A, 0
(5)	002000	104	. ASCII @D@
(5)	002001	132	. ASCII @Z@
(5)	002002	122	. ASCII @R@
(5)	002003	114	. ASCII @L@
(5)	002004	102	. ASCII @B@
(6)	002005	000	. BYTE 0
(6)	002006	000	. BYTE 0
(5)	002007	000	. BYTE 0
(4)	002010	101	. ASCII @A@
(4)	002011	060	. ASCII @O@
(4)	002012	001	. BYTE CSREVISION
(3)	002013	006	. BYTE CSREDIT
(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	016602	. WORD LSDISPATCH
(4)	002042	016740	. WORD LSINIT
(4)	002044	017606	. WORD LSCLEAN
(4)	002046	037404	. WORD LSHARD
(4)	002050	037530	. WORD LSSOFT
(4)	002052	002104	. WORD LSDVTYP
(4)	002054	000000	. WORD 0
(4)	002056	016552	. WORD LSHW
(4)	002060	016566	. WORD LSSW
(4)	002062	002102	. WORD LSDR
(4)	002064	002102	. WORD LSDRST
(4)	002066	000000	. WORD 0
(4)	002070	000000	. WORD 0
(4)	002072	017702	. WORD LSDU
(4)	002074	000000	. WORD 0



(4) 002076 037734 .WORD L\$LAST  
2836  
2837 002100 ENDMOD  
2838  
2839  
2840 002100 DEVREG  
(5) 002100 000000 .WORD 0  
(2) 002102 000001 .BLKW  
2841  
2842 002104 DEVTYP <RLO1>  
(3) 002104 046122 030460 000 .ASCIZ @RLO1@  
(2) 002112 .EVEN  
2843 002112 BGNMOD GLBEQAT  
2844 002112 EQUALS

(1) ;  
(1) ; BIT DIFINITIONS  
(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) BIT9== BIT09  
(1) BIT8== BIT08  
(1) BIT7== BIT07  
(1) BIT6== BIT06  
(1) BIT5== BIT05  
(1) BIT4== BIT04  
(1) BIT3== BIT03  
(1) BIT2== BIT02  
(1) BIT1== BIT01  
(1) 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.



```

2874 ; OFFSET FOR HARDWARE P-TABLE
2875
2876 000000 CSR=0
2877 000002 VECT=2
2878 000004 PRIOR=4
2879 000006 DRBT=6
2880 000010 CNT=10
2881
2882 ; OFFSET FOR SOFTWARE P-TABLE
2883
2884 000000 DLT=0
2885 000002 ELT=2
2886 000004 SIZE=4
2887 000006 DMPCK=6
2888 000010 DLMT=10
2889
2890 002112 ENDMOD
2891 002112 BGNMOD GLBDAT
2892
2893 .SBTTL GLOBAL DATA
2894
2895 002112 000000 CHECK: .WORD 0
2896 002114 000000 T. CRC: .WORD 0
2897 002116 000000 WHY: .WORD 0
2898 002120 000000 CDCNT: .WORD 0
2899 002122 000004 ERRVEC: .WORD 4
2900 002124 000000 DRIVE: .WORD 0
2901 002126 000000 UUT: .WORD 0
2902 002130 000000 UNITST: .WORD 0
2903 002132 000000 TRPFLG: .WORD 0
2904 002134 000000 INTFLG: .WORD 0 ; INTERRUPT OCCURANCE FLAG
2905 002136 000000 LDCSR: .WORD 0 ; LOCATION TO FORM RLCS
2906 002140 000077 SECMSK: .WORD 77 ; MASK OUT SECTOR
2907 002142 120001 XPOLY: .WORD 120001 ; POLYNOMIAL FOR CRC 16
2908 002144 000000 BCCFBK: .WORD 0 ; LOCATION USED BY "SIMBCC"
2909 002146 000000 CALBCC: .WORD 0 ; LOCATION USED BY "SIMBCC"
2910 002150 000000 TMPO: .WORD 0
2911 002152 000000 TMP1: .WORD 0
2912 002154 000000 TMP2: .WORD 0
2913 002156 000000 GDDAT: .WORD 0
2914 002160 000000 BDDAT: .WORD 0
2915 002162 000000 TEMP2: .WORD 0 ; LOCATION USED BY "SIMBCC"
2916 002164 000000 TEMP3: .WORD 0 ; LOCATION USED BY "SIMBCC"
2917 002166 000000 TEMP4: .WORD 0 ; LOCATION USED BY "SIMBCC"
2918 002170 000000 FIRST: .WORD 0 ; FIRST SECTOR READ
2919 002172 177700 CYLSK: .WORD 177700 ; MASK CYLINDER AND HEAD SELECT
2920 002174 000050 MXSEC1: .WORD 40 ; MAX SECTOR ADDRESS +1
2921 002176 000047 MAXSEC: .WORD 39 ; MAX SECTOR ADDRESS
2922 002200 000000 DWORD: .WORD 0 ; DIFFERENCE WORD (SEEK)
2923 002202 077600 MAXCYL: .WORD 77600 ; MAXIMUM CYLINDER ADDRESS
2924 002204 000000 SVHD: .WORD 0 ; SAVE CURRENT HEAD SELECT
2925 002206 000000 B. CS: .WORD 0 ; CS - BEFORE OPERATION
2926 002210 000000 B. BA: .WORD 0 ; BA - BEFORE OPERATION
2927 002212 000000 B. DA: .WORD 0 ; DA - BEFORE OPERATION
2928 002214 000000 B. MP: .WORD 0 ; MP - BEFORE OPERATION
2929 002216 000000 E. CS: .WORD 0 ; CS - AT OCCURANCE OF EPROR
  
```

2930	002220	000000	E. BA:	. WORD	0		; BA - AT OCCURANCE OF ERROR
2931	002222	000000	E. DA:	. WORD	0		; DA - AT OCCURANCE OF ERROR
2932	002224	000000	E. MP:	. WORD	0		; MP - AT OCCURANCE OF ERROR
2933	002226	000000	E. MP1:	. WORD	0		
2934	002230	000000	E. MP2:	. WORD	0		
2935	002232	000000	RLCS:	. WORD	0		
2936	002234	000000	RLBA:	. WORD	0		
2937	002236	000000	RLDA:	. WORD	0		
2938	002240	000000	RLMP:	. WORD	0		
2939	002242	000000	BCSR:	. WORD	0		; CSR FROM P TABLE
2940	002244	000000	BVEC:	. WORD	0		; VECTOR FROM P TABLE
2941	002246	000000	BPRIOR:	. WORD	0		; BR LEVEL FROM P TABLE
2942	002250	000000	FNDFNC:	. WORD	0		
2943	002252	000000	XMEM:	. WORD	0		
2944	002254	000000	TRYFNC:	. WORD	0		
2945	002256	000000	ERFLG:	. WORD	0		
2946	002260	000000	ERRLMT:	. WORD	0		; PRESENT ERROR COUNT
2947	002262	001212	LOPIMX:	. WORD	650.		
2948	002264	000233	LOPIMN:	. WORD	155.		
2949	002266	000620	UOPIMX:	. WORD	400.		
2950	002270	000240	UOPIMN:	. WORD	160.		
2951	002272	000000	OPIMN:	. WORD	0		
2952	002274	000000	OPIMX:	. WORD	0		
2953	002276	000000	PWRFLG:	. WORD	0		
2954	002300	000000	T. CNTLR:	. WORD	0		
2955	002302	000000	DERFLG:	. WORD	0		

2956							
2957							
2958			. SBTTL	LIST TO CHECK HEADER COMPARE LOGIC			
2959	002304	000000	HDRTAB.	. WORD	0		; WALK 1
2960	002306	000001		. WORD	BIT0		
2961	002310	000002		. WORD	BIT1		
2962	002312	000004		. WORD	BIT2		
2963	002314	000010		. WORD	BIT3		
2964	002316	000020		. WORD	BIT4		
2965	002320	000040		. WORD	BIT5		
2966	002322	000100		. WORD	BIT6		
2967	002324	000200		. WORD	BIT7		
2968	002326	000400		. WORD	BIT8		
2969	002330	001000		. WORD	BIT9		
2970	002332	002000		. WORD	BIT10		
2971	002334	004000		. WORD	BIT11		
2972	002336	010000		. WORD	BIT12		
2973	002340	020000		. WORD	BIT13		
2974	002342	040000		. WORD	BIT14		
2975	002344	000003		. WORD	3		; GROW 1
2976	002346	000007		. WORD	7		
2977	002350	000017		. WORD	17		
2978	002352	000037		. WORD	37		
2979	002354	000137		. WORD	137		
2980	002356	000337		. WORD	337		
2981	002360	000737		. WORD	737		
2982	002362	001737		. WORD	1737		
2983	002364	003737		. WORD	3737		
2984	002366	007737		. WORD	7737		
2985	002370	017737		. WORD	17737		

2986	002372	037737			. WORD	37737	
2987	002374	077737			. WORD	77737	
2988	002376	077736			. WORD	77736	;GROW 0
2989	002400	077734			. WORD	77734	
2990	002402	077730			. WORD	77730	
2991	002404	077720			. WORD	77720	
2992	002406	077700			. WORD	77700	
2993	002410	077600			. WORD	77600	
2994	002412	077400			. WORD	77400	
2995	002414	077000			. WORD	77000	
2996	002416	076000			. WORD	76000	
2997	002420	074000			. WORD	74000	
2998	002422	070000			. WORD	70000	
2999	002424	060000			. WORD	60000	
3000	002426	077735			. WORD	77735	;WALK 0
3001	002430	077733			. WORD	77733	
3002	002432	077727			. WORD	77727	
3003	002434	077717			. WORD	77717	
3004	002436	077637			. WORD	77637	
3005	002440	077537			. WORD	77537	
3006	002442	077337			. WORD	77337	
3007	002444	076737			. WORD	76737	
3008	002446	075737			. WORD	75737	
3009	002450	073737			. WORD	73737	
3010	002452	067737			. WORD	67737	
3011	002454	057737			. WORD	57737	
3012	002456	037737			. WORD	37737	
3013	002460	000000			.HDREND: . WORD	0	
3014							
3015							
3016	002462	000001	000002	000004	DATPAT: . WORD	1, 2, 4, 10, 20, 40, 100, 200, 400, 1000, 2000, 4000, 10000, 20000, 40000, 100000	
	002470	000010	000020	000040			
	002476	000100	000200	000400			
	002504	001000	002000	004000			
	002512	010000	020000	040000			
	002520	100000					
3017	002522	177777	177776	177775	. WORD	177777, 177776, 177775, 177773, 177767, 177757, 177737, 177677	
	002530	177773	177767	177757			
	002536	177737	177677				
3018	002542	177577	177377	176777	. WORD	177577, 177377, 176777, 175777, 173777, 167777, 157777, 137777	
	002550	175777	173777	167777			
	002556	157777	137777				
3019	002562	077777	177774	177770	. WORD	77777, 177774, 177770, 177760, 177740, 177700, 177600, 177400	
	002570	177760	177740	177700			
	002576	177600	177400				
3020	002602	177000	176000	174000	. WORD	177000, 176000, 174000, 170000, 160000, 140000, 3, 7, 17, 37, 77	
	002610	170000	160000	140000			
	002616	000003	000007	000017			
	002624	000037	000077				
3021	002630	000177	000377	000777	. WORD	177, 377, 777, 1777, 3777, 7777, 17777, 37777, 0	
	002636	001777	003777	007777			
	002644	017777	037777	000000			

3022  
3023  
3024  
3025

.SBTTL BUFFER FOR READ/WRITE

```

3026 002652 002000          BUF:   .BLKW  1024.
3027
3028
3029 006652          ENDMOD
3030
3031          .SBTTL  GLOBAL TEXT
3032 006652          BGNMOD  GLBTXT
3036 006652 047516 041440 047117 NORES: .ASCIZ  /NO CONTROLLER/
3037 006670 047516 042040 044522 NORDY: .ASCIZ  /NO DRIVE/
3038 006701      103 035123 000040 ARLCS: .ASCIZ  /CS: /
3039 006706 041040 035101 000040 ARLBA: .ASCIZ  / BA: /
3040 006714 042040 035101 000040 ARLDA: .ASCIZ  / DA: /
3041 006722 046440 035120 000040 ARLMP: .ASCIZ  / MP: /
3042 006730 042502 047506 042522 BEREG: .ASCIZ  /BEFORE COMMAND: /
3043 006751      124 046511 020105 AFREG: .ASCIZ  /TIME OF ERROR: /
3044 006772 047503 052116 047522 CRTIM: .ASCIZ  /CONTROLLER TIMED OUT/
3045 007017      104 044522 042526 DRTIM: .ASCIZ  /DRIVE READY TIMED OUT/
3046 007045      040 051104 000126 DEMES: .ASCIZ  / DRV/
3047 007052 047040 046530      000 NXMMES: .ASCIZ  / NXM/
3048 007057      040 050117 000111 OPIMES: .ASCIZ  / OPI/
3049 007064 044040 051103 000103 HRCMES: .ASCIZ  / HCRC/
3050 007072 044040 043116      000 HNFMES: .ASCIZ  / HNF/
3051 007077      040 041504 000113 DCKMES: .ASCIZ  / DCK/
3052 007104 042040 052114      000 DLTMES: .ASCIZ  / DLT/
3053 007111      015      000 LF: .ASCIZ  <15>
3054 007113      015 000012 MSCRLF: .ASCIZ  <15><12>
3055 007116 041440 046517 000120 COMP: .ASCIZ  / COMP/
3056 007124 047506 041522 042105 OPIERR: .ASCIZ  /FORCED OPI(GET STATUS) CAUSED OTHER ERRORS/
3057 007177      116 047517 020120 NOPMES: .ASCIZ  /NOOP OPERATION-FLAG MODE/
3058 007230 047516 050117 047440 NOPINT: .ASCIZ  /NOOP OPERATION-INTR. MODE/
3059 007262 051127 052111 020105 WCKMES: .ASCIZ  /WRITE CHECK OPERATION-FLAG MODE/
3060 007322 051127 052111 020105 WCKINT: .ASCIZ  /WRITE CHECK OPERATION-INTR. MODE/
3061 007363      122 040505 020104 RDMES: .ASCIZ  /READ HEADER OPERATION-FLAG MODE/
3062 007423      122 040505 020104 RHDINT: .ASCIZ  /READ HEADER OPERATION-INTR. MODE/
3063 007464 042523 045505 047440 SEKMES: .ASCIZ  /SEEK OPERATION-FLAG MODE/
3064 007515      123 042505 020113 SEKINT: .ASCIZ  /SEEK OPERATION-INTR. MODE/
3065 007547      107 052105 051440 GSTMES: .ASCIZ  /GET STATUS OPERATION-FLAG MODE/
3066 007606 042507 020124 052123 GSTINT: .ASCIZ  /GET STATUS OPERATION-INTR MODE/
3067 007645      122 040505 020104 RDDMES: .ASCIZ  /READ OPERATION-FLAG MODE/
3068 007676 042522 042101 047440 RDDINT: .ASCIZ  /READ OPERATION-INTR MODE/
3069 007727      127 044522 042524 WRTMES: .ASCIZ  /WRITE OPERATION-FLAG MODE/
3070 007761      127 044522 042524 WRTINT: .ASCIZ  /WRITE OPERATION-INTR MODE/
3071 010013      122 040505 020104 RDNMES: .ASCIZ  %READ W/O HEADER - FLAG MODE%
3072 010047      122 040505 020104 RDNINT: .ASCIZ  %READ W/O HEADER - INTR MODE%
3073 010103      103 047101 052047 SKHOME: .ASCIZ  /CAN'T SEEK TO TRACK 0/
3074 010131      127 044522 042524 WRLOCK: .ASCIZ  /WRITE LOCK ERROR/
3075 010152 046122 051503 041440 EM1: .ASCIZ  /RLCS CONTAINED FOLLOWING ERROR(S): /
3076 010217 000170 EM100: .BLKB  120.
3077 010407      116 020117 047111 EM4: .ASCIZ  /NO INTERRUPT ON READ OPERATION/
3078 010446 042522 042101 047440 EM5: .ASCIZ  /READ OPERATION DID NOT WRITE MEMORY/
3079 010512 046122 040502 042040 EM6: .ASCIZ  /RLBA DID NOT INCREMENT PROPERLY DURING READ/
3080 010566 042523 052103 051117 EM7: .ASCIZ  /SECTOR DID NOT INCREMENT PROPERLY AFTER READ/
3081 010643      110 040505 042504 EM10: .ASCIZ  /HEADER NOT FOUND COULD NOT BE FORCED/
3082 010710 051127 047117 020107 EM11: .ASCIZ  /WRONG CYLINDER ON SEEK/
3083 010737      110 040505 042504 EM12: .ASCIZ  /HEADER NOT FOUND WOULD NOT SET/
3084 010776 051104 053111 020105 EM13: .ASCIZ  /DRIVE READY WOULD NOT SET/

```

3085	011030	044504	045523	040440	EM14:	.ASCIZ	/DISK ADDRESS INCORRECT AFTER MULTIPLE SECTOR READ/
3086	011112	051104	053111	020105	EM16:	.ASCIZ	/DRIVE ERROR ON WRITE OPERATION/
3087	011151	122	020117	047111	EM17:	.ASCIZ	/NO INTERRUPT ON WRITE OPERATION/
3088	011211	122	041114	020101	EM20:	.ASCIZ	/RLBA DID NOT INCREMENT PROPERLY DURING WRITE/
3089	011266	042523	052103	051117	EM21:	.ASCIZ	/SECTOR DID NOT INCREMENT PROPERLY AFTER WRITE/
3090	011344	044504	045523	040440	EM22:	.ASCIZ	/DISK ADDRESS (RLDA) INCORRECT AFTER MULTIPLE SECTOR WRITE/
3091	011436	042110	020122	047516	EM23:	.ASCIZ	/HDR NOT FND COULD NOT BE FORCED AT END OF TRACK/
3092	011516	047516	026516	054105	EM24:	.ASCIZ	/NON-EXISTANT MEMORY ERROR COULD NOT BE FORCED/
3093	011574	040504	040524	041440	EM25:	.ASCIZ	%DATA COMPARISON ERROR - READ/WRITE ERROR%
3094							
3095	011645	127	044522	042524	EM26:	.ASCIZ	/WRITE OPERATION MODIFIED MEMORY/
3096	011705	105	051122	051117	EM27:	.ASCIZ	/ERROR ON PARTIAL SECTOR WRITE - ZERO FILL CHECK/
3097	011765	122	041114	020101	EM30:	.ASCIZ	/RLBA DID NOT INCREMENT PROPERLY/
3098	012025	102	020101	044502	EM31:	.ASCIZ	/BA BIT 16 DID NOT SET ON INCREMENT/
3099	012070	040502	041040	052111	EM32:	.ASCIZ	/BA BIT 17 SET ON BA16 INCREMENT TEST/
3100	012135	122	041114	020101	EM33:	.ASCIZ	/RLBA DID NOT INCREMENT WITH BA16/
3101	012176	040502	041040	052111	EM34:	.ASCIZ	/BA BIT 17 DID NOT SET ON INCREMENT/
3102	012241	102	020101	044502	EM35:	.ASCIZ	/BA BIT 16 DID NOT CLEAR ON INCREMENT/
3103	012306	046122	040502	042040	EM36:	.ASCIZ	/RLBA DID NOT INCREMENT WITH BA17/
3104	012347	122	040505	024104	EM40:	.ASCIZ	/READ(FUNCTION ?) DID NOT INTERRUPT/
3105	012412	042522	042101	043050	EM41:	.ASCIZ	/READ(FUNCTION ?) ERROR - BAD DATA/
3106	012454	042522	042101	043050	EM42:	.ASCIZ	/READ(FUNCTION ?) ERROR AT END OF TRACK/
3107	012523	116	020117	047111	EM43:	.ASCIZ	/NO INTERRUPT WITH HDR NT FND FORCED/
3108	012567	116	020117	047111	EM44:	.ASCIZ	/NO INTERRUPT WITH NXM FORCED/
3109	012624	051105	047522	020122	EM45:	.ASCIZ	%ERROR ON BIT BANG OF SILO%
3110	012656	044523	047514	047440	EM47:	.ASCIZ	/SILO OPERATION FAILURE/
3111	012705	110	040505	042504	EM50:	.ASCIZ	/HEADER COMPARE FAILURE - SECTOR/
3112	012745	127	044522	042524	EM51:	.ASCIZ	/WRITE NPR CAUSED BUS TRAP/
3113	012777	122	040505	020104	EM52:	.ASCIZ	/READ NPR CAUSED BUS TRAP/
3114	013030	042522	042101	053440	EM55:	.ASCIZ	?READ W/O HDR CMP OPERATION DID NOT WRITE MEMORY?
3115	013110	046122	040502	042040	EM53:	.ASCIZ	?RLBA DID NOT INCREMENT PROPERLY DURING READ W/O HDR CMP?
3116	013200	046122	040504	042040	EM54:	.ASCIZ	?RLDA DID NOT INCREMENT AFTER READ W/O HDR CMP?
3117	013256	050117	020111	044524	EM56:	.ASCIZ	/OPI TIMING ERROR/
3118	013277	127	044522	042524	EM57:	.ASCIZ	/WRITE CHECK NPR CAUSED BUS TRAP/
3119	013337	127	044522	042524	EM60:	.ASCIZ	/WRITE CHECK DID NOT INTERRUPT/
3120	013375	122	041114	020101	EM61:	.ASCIZ	/RLBA DID NOT INCREMENT PROPERLY DURING WRCHK/
3121	013452	046122	040504	042040	EM62:	.ASCIZ	/RLDA DID NOT INCREMENT PROPERLY DURING WRCHK/
3122	013527	122	042114	020101	EM63:	.ASCIZ	/RLDA DID NOT INCREMENT PROPERLY AFTER A MULTIPLE SECTOR WRITE CHK/
3123	013631	127	044522	042524	EM64:	.ASCIZ	/WRITE CHECK OF PARTIAL SECTOR WRITE FAILURE/
3124	013705	103	047101	047040	EM65:	.ASCIZ	/CAN NOT FORCE DCK ON WRITE CHECK/
3125	013746	040503	020116	047516	EM66:	.ASCIZ	/CAN NOT FORCE INTERRUPT WITH DCK ON WRCHK/
3126	014020	051127	052111	020105	EM70:	.ASCIZ	/WRITE CHECK FAILURE/
3127							
3128						.EVEN	
3129							
3130							
3134	014044					ENDMOD	
3135							
3136	014044					BGNMOD	GLBERR
3137							
3138						.SBTTL	GLOBAL ERRORS
3139	014044					BGNMSG	ERRO
3140							
3141	014044	004737	015056			JSR	PC,LINE1
3142	014050	004737	015112			JSR	PC,LINE2
3143							

```

3144
3145 014054 004537 017724          JSR    R5,CKERLT          ; INCREMENT ERROR AND CHECK LIMIT
3146
3147 014060          ENDMSG
(3) 014060          L10000:
(3) 014060 104023          EMT    C$MSG
3148
3149 014062          BGNMSG  ERR1
3150
3151 014062 004737 015056          JSR    PC,LINE1
3152
3153
3154 014066 004537 017724          JSR    R5,CKERLT          ; INCREMENT ERROR AND CHECK LIMIT
3155
3156 014072          ENDMSG
(3) 014072          L10001:
(3) 014072 104023          EMT    C$MSG
3157
3158 014074          BGNMSG  ERR2
3159
3160 014074 004737 015056          JSR    PC,LINE1
3161 014100          PRINTB #FRMT4,GDDAT,BDDAT
(9) 014100 013746 002160          MOV    BDDAT,-(SP)
(8) 014104 013746 002156          MOV    GDDAT,-(SP)
(7) 014110 012746 015533          MOV    #FRMT4,-(SP)
(6) 014114 012746 000003          MOV    #3,-(SP)
(3) 014120 010600          MOV    SP,RO
(4) 014122 104014          EMT    C$PNTB
(4) 014124 062706 000010          ADD    #10,SP
3162
3163
3164 014130 004537 017724          JSR    R5,CKERLT          ; INCREMENT ERROR AND CHECK LIMIT
3165
3166 014134          ENDMSG
(3) 014134          L10002:
(3) 014134 104023          EMT    C$MSG
3167
3168 014136          BGNMSG  ERR3
3169
3170 014136 004737 015056          JSR    PC,LINE1
3171 014142 004737 015112          JSR    PC,LINE2
3172 014146          PRINTB #FRMT5,TMPO,BDDAT,GDDAT
(10) 014146 013746 002156          MOV    GDDAT,-(SP)
(9) 014152 013746 002160          MOV    BDDAT,-(SP)
(8) 014156 013746 002150          MOV    TMPO,-(SP)
(7) 014162 012746 015571          MOV    #FRMT5,-(SP)
(6) 014166 012746 000004          MOV    #4,-(SP)
(3) 014172 010600          MOV    SP,RO
(4) 014174 104014          EMT    C$PNTB
(4) 014176 062706 000012          ADD    #12,SP
3173
3174
3175 014202 004537 017724          JSR    R5,CKERLT          ; INCREMENT ERROR AND CHECK LIMIT
3176
3177 014206          ENDMSG
(3) 014206          L10003:
  
```



```
(3) 014206 104023          EMT      C$MSG
3178
3179 014210          BGNMSG  ERR4
3180
3181 014210 004737 015056      JSR      PC,LINE1
3182 014214 004737 015112      JSR      PC,LINE2
3183 014220          PRINTB  #FRMT4,GDDAT,BDDAT
(9) 014220 013746 002160      MOV      BDDAT,-(SP)
(8) 014224 013746 002156      MOV      GDDAT,-(SP)
(7) 014230 012746 015533      MOV      #FRMT4,-(SP)
(6) 014234 012746 000003      MOV      #3,-(SP)
(3) 014240 010600          MOV      SP,RO
(4) 014242 104014          EMT      C$PNTB
(4) 014244 062706 000010      ADD      #10,SP
3184
3185
3186 014250 004537 017724      JSR      R5,CKERLT          ; INCREMENT ERROR AND CHECK LIMIT
3187
3188 014254          ENDMSG
(3) 014254          L10004:
(3) 014254 104023          EMT      C$MSG
3189
3190 014256          BGNMSG  ERR5
3191
3192 014256 004737 015056      JSR      PC,LINE1
3193 014262          PRINTB  #FRMT3,RESTMS
(8) 014262 013746 020236      MOV      RESTMS,-(SP)
(7) 014266 012746 015526      MOV      #FRMT3,-(SP)
(6) 014272 012746 000002      MOV      #2,-(SP)
(3) 014276 010600          MOV      SP,RO
(4) 014300 104014          EMT      C$PNTB
(4) 014302 062706 000006      ADD      #6,SP
3194
3195
3196 014306 004537 017724      JSR      R5,CKERLT          ; INCREMENT ERROR AND CHECK LIMIT
3197
3198 014312          ENDMSG
(3) 014312          L10005:
(3) 014312 104023          EMT      C$MSG
3199
3200 014314          BGNMSG  ERR6
3201
3202 014314 004737 015056      JSR      PC,LINE1
3203 014320 004737 015334      JSR      PC,LINE3
3204 014324 004737 015112      JSR      PC,LINE2
3205
3206
3207 014330          PRINTB  #FRMT99
(7) 014330 012746 016467      MOV      #FRMT99,-(SP)
(6) 014334 012746 000001      MOV      #1,-(SP)
(3) 014340 010600          MOV      SP,RO
(4) 014342 104014          EMT      C$PNTB
(4) 014344 062706 000004      ADD      #4,SP
3208 014350 004537 017724      JSR      R5,CKERLT          ; INCREMENT ERROR AND CHECK LIMIT
3209
3210 014354          ENDMSG
```

```

(3) 014354          L10006:
(3) 014354 104023   EMT      C$MSG
3211
3212 014356          BGNMSG  ERR7
3213
3214
3215
3216 014356 004537 017724      JSR      R5,CKERLT      ; INCREMENT ERROR AND CHECK LIMIT
3217
3218 014362          ENDMSG
(3) 014362          L10007:
(3) 014362 104023   EMT      C$MSG
3219
3220
3221
3222 014364          BGNMSG  ERR8
3223
3224 014364 004737 015056      JSR      PC,LINE1
3225 014370 004737 015112      JSR      PC,LINE2
3226 014374          PRINTB  #FRMT6,TMP1,GDDAT,BDDAT
(10) 014374 013746 002160      MOV      BDDAT,-(SP)
(9) 014400 013746 002156      MOV      GDDAT,-(SP)
(8) 014404 013746 002152      MOV      TMP1,-(SP)
(7) 014410 012746 015642      MOV      #FRMT6,-(SP)
(6) 014414 012746 000004      MOV      #4,-(SP)
(3) 014420 010600          MOV      SP,RO
(4) 014422 104014          EMT      C$PNTB
(4) 014424 062706 000012      ADD      #12,SP
3227
3228
3229 014430 004537 017724      JSR      R5,CKERLT      ; INCREMENT ERROR AND CHECK LIMIT
3230
3231 014434          ENDMSG
(3) 014434          L10010:
(3) 014434 104023   EMT      C$MSG
3232
3233 014436          BGNMSG  ERR9
3234
3235 014436 004737 015056      JSR      PC,LINE1
3236 014442 004737 015112      JSR      PC,LINE2
3237 014446          PRINTB  #FRMT4,TMPD,R2
(9) 014446 010246          MOV      R2,-(SP)
(8) 014450 013746 002150      MOV      TMPD,-(SP)
(7) 014454 012746 015533      MOV      #FRMT4,-(SP)
(6) 014460 012746 000003      MOV      #3,-(SP)
(3) 014464 010600          MOV      SP,RO
(4) 014466 104014          EMT      C$PNTB
(4) 014470 062706 000010      ADD      #10,SP
3238
3239
3240 014474 004537 017724      JSR      R5,CKERLT      ; INCREMENT ERROR AND CHECK LIMIT
3241
3242 014500          ENDMSG
(3) 014500          L10011:
(3) 014500 104023   EMT      C$MSG
3243
  
```

3244	014502			BGNMSG	ERR10		
3245							
3246	014502	004737	015056		JSR	PC, LINE1	
3247	014506	004737	015112		JSR	PC, LINE2	
3248	014512				PRINTB	#FRMT7, TMP1, GDDAT, BDDAT	
(10)	014512	013746	002160		MOV	BDDAT, -(SP)	
(9)	014516	013746	002156		MOV	GDDAT, -(SP)	
(8)	014522	013746	002152		MOV	TMP1, -(SP)	
(7)	014526	012746	015717		MOV	#FRMT7, -(SP)	
(6)	014532	012746	000004		MOV	#4, -(SP)	
(3)	014536	010600			MOV	SP, R0	
(4)	014540	104014			EMT	C\$PNTB	
(4)	014542	062706	000012		ADD	#12, SP	
3249							
3250							
3251	014546	004537	017724		JSR	R5, CKERLT	; INCREMENT ERROR AND CHECK LIMIT
3252							
3253	014552				ENDMSG		
(3)	014552			L10012:			
(3)	014552	104023			EMT	C\$MSG	
3254							
3255	014554			BGNMSG	ERR11		
3256							
3257	014554	004737	015056		JSR	PC, LINE1	
3258	014560	004737	015112		JSR	PC, LINE2	
3259	014564				PRINTB	#FRMT8, TMP0, GDDAT, BDDAT	
(10)	014564	013746	002160		MOV	BDDAT, -(SP)	
(9)	014570	013746	002156		MOV	GDDAT, -(SP)	
(8)	014574	013746	002150		MOV	TMP0, -(SP)	
(7)	014600	012746	015771		MOV	#FRMT8, -(SP)	
(6)	014604	012746	000004		MOV	#4, -(SP)	
(3)	014610	010600			MOV	SP, R0	
(4)	014612	104014			EMT	C\$PNTB	
(4)	014614	062706	000012		ADD	#12, SP	
3260							
3261							
3262	014620	004537	017724		JSR	R5, CKERLT	; INCREMENT ERROR AND CHECK LIMIT
3263							
3264	014624				ENDMSG		
(3)	014624			L10013:			
(3)	014624	104023			EMT	C\$MSG	
3265							
3266	014626			BGNMSG	ERR12		
3267							
3268	014626	004737	015056		JSR	PC, LINE1	
3269	014632	004737	015112		JSR	PC, LINE2	
3270	014636				PRINTB	#FRMT9, TMP1, R3, GDDAT, BDDAT	
(11)	014636	013746	002160		MOV	BDDAT, -(SP)	
(10)	014642	013746	002156		MOV	GDDAT, -(SP)	
(9)	014646	010346			MOV	R3, -(SP)	
(8)	014650	013746	002152		MOV	TMP1, -(SP)	
(7)	014654	012746	016112		MOV	#FRMT9, -(SP)	
(6)	014660	012746	000005		MOV	#5, -(SP)	
(3)	014664	010600			MOV	SP, R0	
(4)	014666	104014			EMT	C\$PNTB	
(4)	014670	062706	000014		ADD	#14, SP	

3271						
3272						
3273	014674	004537	017724	JSR	R5, CKERLT	; INCREMENT ERROR AND CHECK LIMIT
3274						
3275	014700			ENDMSG		
(3)	014700			L10014:		
(3)	014700	104023		EMT	C\$MSG	
3276						
3277	014702			BGNMSG	ERR13	
3278						
3279	014702	004737	015056	JSR	PC, LINE1	
3280	014706			PRINTB	#FRMT10, OPIMN, OPIMX, BDDAT	
(10)	014706	013746	002160	MOV	BDDAT, -(SP)	
(9)	014712	013746	002274	MOV	OPIMX, -(SP)	
(8)	014716	013746	002272	MOV	OPIMN, -(SP)	
(7)	014722	012746	016215	MOV	#FRMT10, -(SP)	
(6)	014726	012746	000004	MOV	#4, -(SP)	
(3)	014732	010600		MOV	SP, R0	
(4)	014734	104014		EMT	C\$PNTB	
(4)	014736	062706	000012	ADD	#12, SP	
3281						
3282						
3283	014742	004537	017724	JSR	R5, CKERLT	; INCREMENT ERROR AND CHECK LIMIT
3284						
3285	014746			ENDMSG		
(3)	014746			L10015:		
(3)	014746	104023		EMT	C\$MSG	
3286						
3287	014750			BGNMSG	ERR14	
3288						
3289	014750	004737	015056	JSR	PC, LINE1	
3290	014754	004737	015112	JSR	PC, LINE2	
3291	014760			PRINTB	#FRMT14, TMP1, #BUF	
(9)	014760	012746	002652	MOV	#BUF, -(SP)	
(8)	014764	013746	002152	MOV	TMP1, -(SP)	
(7)	014770	012746	016041	MOV	#FRMT14, -(SP)	
(6)	014774	012746	000003	MOV	#3, -(SP)	
(3)	015000	010600		MOV	SP, R0	
(4)	015002	104014		EMT	C\$PNTB	
(4)	015004	062706	000010	ADD	#10, SP	
3292						
3293						
3294	015010	004537	017724	JSR	R5, CKERLT	; INCREMENT ERROR AND CHECK LIMIT
3295						
3296	015014			ENDMSG		
(3)	015014			L10016:		
(3)	015014	104023		EMT	C\$MSG	
3297						
3298	015016			BGNMSG	ERR15	
3299						
3300	015016	004737	015056	JSR	PC, LINE1	
3301	015022	004737	015112	JSR	PC, LINE2	
3302	015026			PRINTB	#FRMT15, R2	
(8)	015026	010246		MOV	R2, -(SP)	
(7)	015030	012746	016523	MOV	#FRMT15, -(SP)	
(6)	015034	012746	000002	MOV	#2, -(SP)	

(3)	015040	010600		MOV	SP,RO
(4)	015042	104014		EMT	C\$PNTB
(4)	015044	062706	000006	ADD	#6,SP
3303	015050	004537	017724	JSR	R5,CKERLT
3304					
3305	015054			ENDMSG	
(3)	015054			L10017:	
(3)	015054	104023		EMT	C\$MSG
3306					
3307	015056			LINE1:	PRINTB #FRMT1,RLCS,<B,DRIVE+1>
(9)	015056	005046		CLR	-(SP)
(9)	015060	153716	002125	BISB	DRIVE+1,(SP)
(8)	015064	013746	002232	MOV	RLCS, -(SP)
(7)	015070	012746	015406	MOV	#FRMT1, -(SP)
(6)	015074	012746	000003	MOV	#3, -(SP)
(3)	015100	010600		MOV	SP,RO
(4)	015102	104014		EMT	C\$PNTB
(4)	015104	062706	000010	ADD	#10,SP
3308	015110	000207		RTS	PC
3309					
3310	015112			LINE2:	PRINTB #FRMT2,#BEREG,#ARLCS,B.CS,#ARLBA,B.BA
(12)	015112	013746	002210	MOV	B.BA, -(SP)
(11)	015116	012746	006706	MOV	#ARLBA, -(SP)
(10)	015122	013746	002206	MOV	B.CS, -(SP)
(9)	015126	012746	006701	MOV	#ARLCS, -(SP)
(8)	015132	012746	006730	MOV	#BEREG, -(SP)
(7)	015136	012746	015445	MOV	#FRMT2, -(SP)
(6)	015142	012746	000006	MOV	#6, -(SP)
(3)	015146	010600		MOV	SP,RO
(4)	015150	104014		EMT	C\$PNTB
(4)	015152	062706	000016	ADD	#16,SP
3311	015156			PRINTB	#FRMT2A,#ARLDA,B.DA,#ARLMP,B.MP
(11)	015156	013746	002214	MOV	B.MP, -(SP)
(10)	015162	012746	006722	MOV	#ARLMP, -(SP)
(9)	015166	013746	002212	MOV	B.DA, -(SP)
(8)	015172	012746	006714	MOV	#ARLDA, -(SP)
(7)	015176	012746	015464	MOV	#FRMT2A, -(SP)
(6)	015202	012746	000005	MOV	#5, -(SP)
(3)	015206	010600		MOV	SP,RO
(4)	015210	104014		EMT	C\$PNTB
(4)	015212	062706	000014	ADD	#14,SP
3312	015216			PRINTB	#FRMT2,#AFREG,#ARLCS,E.CS,#ARLBA,E.BA
(12)	015216	013746	002220	MOV	E.BA, -(SP)
(11)	015222	012746	006706	MOV	#ARLBA, -(SP)
(10)	015226	013746	002216	MOV	E.CS, -(SP)
(9)	015232	012746	006701	MOV	#ARLCS, -(SP)
(8)	015236	012746	006751	MOV	#AFREG, -(SP)
(7)	015242	012746	015445	MOV	#FRMT2, -(SP)
(6)	015246	012746	000006	MOV	#6, -(SP)
(3)	015252	010600		MOV	SP,RO
(4)	015254	104014		EMT	C\$PNTB
(4)	015256	062706	000016	ADD	#16,SP
3313	015262			PRINTB	#FRMT2B,#ARLDA,E.DA,#ARLMP,E.MP,E.MP1,E.MP2
(13)	015262	013746	002230	MOV	E.MP2, -(SP)
(12)	015266	013746	002226	MOV	E.MP1, -(SP)
(11)	015272	013746	002224	MOV	E.MP, -(SP)

```

(10) 015276 012746 006722      MOV      #ARLMP, -(SP)
(9)  015302 013746 002222      MOV      E. DA, -(SP)
(8)  015306 012746 006714      MOV      #ARLDA, -(SP)
(7)  015312 012746 015477      MOV      #FRMT2B, -(SP)
(6)  015316 012746 000007      MOV      #7, -(SP)
(3)  015322 010600                MOV      SP, RO
(4)  015324 104014                EMT      C$PNTB
(4)  015326 062706 000020      ADD      #20, SP
3314 015332 000207                RTS      PC

```

```

3315
3316 015334                LINE3: PRINTB #FRMT3, #EM1
(8)  015334 012746 010152      MOV      #EM1, -(SP)
(7)  015340 012746 015526      MOV      #FRMT3, -(SP)
(6)  015344 012746 000002      MOV      #2, -(SP)
(3)  015350 010600                MOV      SP, RO
(4)  015352 104014                EMT      C$PNTB
(4)  015354 062706 000006      ADD      #6, SP
3317 015360                PRINTB #FRMT3, #EM100
(8)  015360 012746 010217      MOV      #EM100, -(SP)
(7)  015364 012746 015526      MOV      #FRMT3, -(SP)
(6)  015370 012746 000002      MOV      #2, -(SP)
(3)  015374 010600                MOV      SP, RO
(4)  015376 104014                EMT      C$PNTB
(4)  015400 062706 000006      ADD      #6, SP
3318 015404 000207                RTS      PC

```

```

3319
3320
3324
3325 015406 040445 047503 052116 FRMT1: .ASCIZ  /%ACONTROLLER: %06%A DRIVE: %01/
3326 015445      045 022516 022524 FRMT2: .ASCIZ  /%N%T%T%06%T%06/
3327 015464 052045 047445 022466 FRMT2A: .ASCIZ  /%T%06%T%06/
3328 015477      045 022524 033117 FRMT2B: .ASCIZ  /%T%06%T%06%A %06%A %06/
3329 015526 047045 052045      000 FRMT3: .ASCIZ  /%N%T/
3330 015533      045 022516 042501 FRMT4: .ASCIZ  /%N%AE%P'D: %06%A REC'D: %06%N/
3331 015571      045 022516 046101 FRMT5: .ASCIZ  /%N%ALAST: %06%A PRES: %06%A EXP'D: %06%N/
3332 015642 047045 040445 052502 FRMT6: .ASCIZ  /%N%ABUS ADR: %06%A EXP'D: %06%A REC'D: %06%N/
3333 015717      045 022516 053501 FRMT7: .ASCIZ  /%N%AWORD: %D3%A EXP'D: %06%A REC'D: %06%N/
3334 015771      045 022516 042101 FRMT8: .ASCIZ  /%N%ADH: %06%A REC'D: %06%A EXP'D: %06%N/
3335 016041      045 022516 053501 FRMT14: .ASCIZ  /%N%AWORDS WRITTEN: %D3%A BUS ADDR: %06%N/
3336 016112 047045 040445 047527 FRMT9: .ASCIZ  /%N%AWORDS WRITTEN: %D3%A BUS ADDR: %06%A EXP'D: %06%A REC'D: %06%N/
3337 016215      045 022516 051101 FRMT10: .ASCII  /%N%ARANGE %D3%A - %D3%A MILLISECONDS WAS %D6%N/
3338 016274 040445 040515 044530      .ASCIZ  /%AMAXIMUM TIMEOUT OF PROGRAM IS 3 SECONDS%N/
3339 016350 047045 040445 051105 FRMT11: .ASCIZ  /%N%AERROR LIMIT EXCEEDED - DROPPED%N/
3340 016415      045 042101 044522 FRMT98: .ASCII  /%ADRIVE DID NOT RECOVER FROM POWER FAILURE/
3341 016467      045 000116      FRMT99: .ASCIZ  /%N/
3342 016472 047045 052045 040445 FRMT13: .ASCIZ  /%N%T%A - WILL NOT TEST%N/
3343 016523      045 022516 050101 FRMT15: .ASCIZ  /%N%APATTERN WAS: %06/

```

```

3344
3345                . EVEN
3346
3350
3351 016550                ENDMOD
3352 016550                BGNMOD HPTCODE
3353
3354 016550                BGNHW
(3)  016550 000005                . WORD  L10020-L$HW/2

```

3355	016552	174400	. WORD	174400	; CSR
3356	016554	000330	. WORD	330	; VECTOR
3357	016556	000240	. WORD	240	; PRIORITY
3358	016560	000000	. WORD	0	; DRIVE (BITS 8,9,10)
3359	016562	000001	. WORD	1	; RL11=1 RLV11=0
3360					
3361	016564		ENDHW		
(3)	016564		L10020:		
3362					
3363	016564		ENDMOD		
3364					
3365	016564		BGNMOD	SPTCODE	
3366					
3367	016564		BGNSW		
(3)	016564	000005	. WORD	L10021-L\$SW/2	
3368					
3369	016566	000000	DROP:	. WORD	0
3370	016570	000012	MERLMT:	. WORD	10.
3371	016572	000000	T. SIZE:	. WORD	0
3372	016574	000000	T. DMP:	. WORD	0
3373	016576	000000	T. LMT:	. WORD	0
3374					
3375	016600		ENDSW		
(3)	016600		L10021:		
3376					
3377	016600		ENDMOD		
3378					
3379	016600		BGNMOD	DSPCODE	
3380					
3381	016600		DISPATCH	47	
(4)	016600	000057	. WORD	47	
(6)	016602	021430	. WORD	T1	
(6)	016604	021700	. WORD	T2	
(6)	016606	022044	. WORD	T3	
(6)	016610	022174	. WORD	T4	
(6)	016612	022330	. WORD	T5	
(6)	016614	022462	. WORD	T6	
(6)	016616	022620	. WORD	T7	
(6)	016620	023016	. WORD	T8	
(6)	016622	023320	. WORD	T9	
(6)	016624	023510	. WORD	T10	
(6)	016626	023706	. WORD	T11	
(6)	016630	024060	. WORD	T12	
(6)	016632	024256	. WORD	T13	
(6)	016634	024456	. WORD	T14	
(6)	016636	024626	. WORD	T15	
(6)	016640	024730	. WORD	T16	
(6)	016642	025054	. WORD	T17	
(6)	016644	025250	. WORD	T18	
(6)	016646	025404	. WORD	T19	
(6)	016650	025536	. WORD	T20	
(6)	016652	025656	. WORD	T21	
(6)	016654	026036	. WORD	T22	
(6)	016656	026614	. WORD	T23	
(6)	016660	027010	. WORD	T24	
(6)	016662	027154	. WORD	T25	

(6)	016664	027270			.WORD	T26
(6)	016666	027446			.WORD	T27
(6)	016670	030046			.WORD	T28
(6)	016672	030470			.WORD	T29
(6)	016674	031116			.WORD	T30
(6)	016676	031576			.WORD	T31
(6)	016700	032230			.WORD	T32
(6)	016702	032640			.WORD	T33
(6)	016704	033072			.WORD	T34
(6)	016706	033362			.WORD	T35
(6)	016710	033656			.WORD	T36
(6)	016712	034150			.WORD	T37
(6)	016714	034542			.WORD	T38
(6)	016716	035042			.WORD	T39
(6)	016720	035402			.WORD	T40
(6)	016722	035714			.WORD	T41
(6)	016724	036200			.WORD	T42
(6)	016726	036470			.WORD	T43
(6)	016730	036560			.WORD	T44
(6)	016732	036712			.WORD	T45
(6)	016734	037110			.WORD	T46
(6)	016736	037246			.WORD	T47
3382	016740				ENDMOD	
3383						
3384				.SBTTL	INITIALIZATION CODE	
3385						
3386	016740			BGNMOD	INITCODE	
3387						
3388	016740				BGNINIT	
3389						
3390	016740			SETPRI	#PRI07	
(3)	016740	012700	000340	MOV	#PRI07,RO	
(3)	016744	104041		EMT	CSSPRI	
3391						
3392	016746			READEF	#EF.PWR	
(3)	016746	012700	000034	MOV	#EF.PWR,RO	
(3)	016752	104050		EMT	CSREFG	
3393	016754			BNCOMPLETE	NOPWR	
(2)	016754	103004		BCC	NOPWR	
3394	016756	013737	002014 002276	MOV	LSUNIT,PWRFLG	
3395	016764	000472		BR	CONT	
3396	016766			NOPWR: READEF	#EF.RESTART	
(3)	016766	012700	000037	MOV	#EF.RESTART,RO	
(3)	016772	104050		EMT	CSREFG	
3397	016774			BCOMPLETE	START	
(2)	016774	103410		BCS	START	
3398	016776			READEF	#EF.NEW	
(3)	016776	012700	000035	MOV	#EF.NEW,RO	
(3)	017002	104050		EMT	CSREFG	
3399	017004			BCOMPLETE	START	
(2)	017004	103404		BCS	START	
3400	017006			READEF	#EF.START	
(3)	017006	012700	000040	MOV	#EF.START,RO	
(3)	017012	104050		EMT	CSREFG	
3401	017014			BNCOMPLETE	CONTINUET	
(2)	017014	103007		BCC	CONTINUET	



3402	017016	013737	002014	002126	START:	MOV	LSUNIT, UUT		; GET NUMBER OF UNITS
3403	017024	012737	177777	002130		MOV	#-1, UNITST		
3404	017032	000404				BR	NXT		
3405									
3406									
3407	017034				CONTINUE:	REDEF	#EF, CONTINUE		
(3)	017034	012700	000036			MOV	#EF, CONTINUE, RO		
(3)	017040	104050				EMT	C\$REFG		
3408	017042					BCOMPLETE	CONT		
(2)	017042	103443				BCS	CONT		
3409	017044	005737	002126		NXT:	TST	UUT		; DONE WITH ALL UNITS
3410	017050	001006				BNE	1\$		; NO
3411	017052	012737	177777	002130		MOV	#-1, UNITST		
3412	017060	013737	002014	002126		MOV	LSUNIT, UUT		
3413									
3414	017066	005237	002130		1\$:	INC	UNITST		
3415	017072	005337	002126			DEC	UUT		
3416	017076				REST:	GPHARD	UNITST, RO		
(3)	017076	013700	002130			MOV	UNITST, RO		
(3)	017102	104042				EMT	C\$GPHRD		
3417	017104					BCOMPLETE	2\$		
(2)	017104	103406				BCS	2\$		
3418	017106	005737	002276			TST	PWRFLG		
3419	017112	001754				BEQ	NXT		
3420	017114	005337	002276			DEC	PWRFLG		
3421	017120	000751				BR	NXT		
3422	017122	012037	002242		2\$:	MOV	(RO)+, BCSR		; GET BUS ADDRESS
3423	017126	012037	002244			MOV	(RO)+, BVEC		; GET VECTOR
3424	017132	012037	002246			MOV	(RO)+, BPRIOR		; GET PRIORITY
3425	017136	012037	002124			MOV	(RO)+, DRIVE		; GET DRIVE
3426	017142	012037	002300			MOV	(RO)+, T. CNTLR		; GET CONTROLLER TYPE
3427									
3428	017146	005037	002260			CLR	ERRLMT		
3429	017152	013700	002242		CONT:	MOV	BCSR, RO		; CREATE REGISTERS
3430	017156	010037	002232			MOV	RO, RLCS		
3431	017162	062700	000002			ADD	#2, RO		
3432	017166	010037	002234			MOV	RO, RLBA		
3433	017172	062700	000002			ADD	#2, RO		
3434	017176	010037	002236			MOV	RO, RLDA		
3435	017202	062700	000002			ADD	#2, RO		
3436	017206	010037	002240			MOV	RO, RLMP		
3437									
3438	017212	005737	002276			TST	PWRFLG		
3439	017216	001064				BNE	5\$		
3440	017220	005737	016572			TST	T. SIZE		; DO WE WANT TO CHECK UNITS??
3441	017224	001461				BEQ	5\$		; NO
3442	017226	005037	002132			CLR	TRPFLG		; CLEAR OUT TRAP INDICATOR
3443	017232					SETVEC	ERRVEC, #TRPHAN, #340		; SETUP TO CATCH TIMEOUT
(7)	017232	012746	000340			MOV	#340, -(SP)		
(6)	017236	012746	021146			MOV	#TRPHAN, -(SP)		
(5)	017242	013746	002122			MOV	ERRVEC, -(SP)		
(4)	017246	012746	000003			MOV	#3, -(SP)		
(3)	017252	104037				EMT	C\$SVEC		
(2)	017254	062706	000010			ADD	#10, SP		
3444	017260	005777	162746			TST	@RLCS		; ACCESS CONTROLLER
3445	017264					CLRVEC	ERRVEC		

(3)	017264	013700	002122			MOV	ERRVEC,RO	
(3)	017270	104036				EMT	CSCVEC	
3446	017272	005737	002132			TST	TRPFLG	; DID TRAP OCCUR??
3447	017276	001404				BEQ	7\$	; NO, CHECK DRIVE
3448	017300	012737	006652	002116		MOV	#NORES,WHY	
3449	017306	000415				BR	8\$	
3450								
3451	017310	012777	000200	162714	7\$:	MOV	#200, @RLCS	; NOW CHECK DRIVE FOR READY
3452	017316	053777	002124	162706		BIS	DRIVE, @RLCS	
3453	017324	032777	000001	162700		BIT	#1, @RLCS	
3454	017332	001016				BNE	5\$	
3455	017334	012737	006670	002116		MOV	#NORDY,WHY	
3456	017342				8\$:	PRINTB	#FRMT13,WHY	
(8)	017342	013746	002116			MOV	WHY, -(SP)	
(7)	017346	012746	016472			MOV	#FRMT13, -(SP)	
(6)	017352	012746	000002			MOV	#2, -(SP)	
(3)	017356	010600				MOV	SP,RO	
(4)	017360	104014				EMT	C\$PNTB	
(4)	017362	062706	000006			ADD	#6, SP	
3457	017366	000444				BR	6\$	
3458								
3459	017370	005737	002276		5\$:	TST	PWRFLG	; POWER UP
3460	017374	001451				BEQ	END	; NO
3461	017376	012777	000200	162626		MOV	#200, @RLCS	
3462	017404	053777	002124	162620		BIS	DRIVE, @RLCS	
3463	017412	012701	000074			MOV	#60, R1	
3464	017416				3\$:	WAITMS	#10,	
(3)	017416	012700	000012			MOV	#10, RO	
(3)	017422	104026				EMT	C\$WTH	
3465	017424	032777	000001	162600		BIT	#1, @RLCS	
3466	017432	001032				BNE	END	
3467	017434	005301				DEC	R1	
3468	017436	001367				BNE	3\$	
3469								
3470	017440					PRINTF	#FRMT99	
(7)	017440	012746	016467			MOV	#FRMT99, -(SP)	
(6)	017444	012746	000001			MOV	#1, -(SP)	
(3)	017450	010600				MOV	SP,RO	
(4)	017452	104017				EMT	C\$PNTF	
(4)	017454	062706	000004			ADD	#4, SP	
3471	017460					PRINTF	#FRMT98	
(7)	017460	012746	016415			MOV	#FRMT98, -(SP)	
(6)	017464	012746	000001			MOV	#1, -(SP)	
(3)	017470	010600				MOV	SP,RO	
(4)	017472	104017				EMT	C\$PNTF	
(4)	017474	062706	000004			ADD	#4, SP	
3472	017500	004737	015056		6\$:	JSR	PC, LINE1	
3473	017504					DODU	UNITST	
(3)	017504	013700	002130			MOV	UNITST,RO	
(3)	017510	104053				EMT	C\$DODU	
3474	017512					DOCLN		
(3)	017512	104044				EMT	C\$DCLN	
3475	017514	000137	017044			JMP	NXT	
3476								
3477	017520	013737	002270	002272	END:	MOV	UOPIMN,OPIMN	
3478	017526	013737	002266	002274		MOV	UOPIMX,OPIMX	

```

3479 017534 005737 002300          TST      T.CNTRL          ;RL11??
3480 017540 001006                   BNE      1$              ;YES, THEN KEEP LIMITS SET
3481 017542 013737 002264 002272    MOV      LOPIMN,OPIMN
3482 017550 013737 002262 002274    MOV      LOPIMX,OPIMX
3483 017556                   1$:
3484 017556                   SETVEC   BVEC,#INTSRV,#340
(7) 017556 012746 000340          MOV      #340,-(SP)
(6) 017562 012746 017706          MOV      #INTSRV,-(SP)
(5) 017566 013746 002244          MOV      BVEC,-(SP)
(4) 017572 012746 000003          MOV      #3,-(SP)
(3) 017576 104037                   EMT      C$SVEC
(2) 017600 062706 000010          ADD      #10,SP
3485
3486
3487
3488 017604                   ENDINIT
(3) 017604                   L10022:
(3) 017604 104011                   EMT      C$INIT
3489
3490 017606                   ENDMOD
3491
3492 017606                   BGNMOD  CLNCODE
3493
3494 017606                   BGNCLN
3495
3496
3497 017606                   SETVEC   ERRVEC,#TRPHAN,#340
(7) 017606 012746 000340          MOV      #340,-(SP)
(6) 017612 012746 021146          MOV      #TRPHAN,-(SP)
(5) 017616 013746 002122          MOV      ERRVEC,-(SP)
(4) 017622 012746 000003          MOV      #3,-(SP)
(3) 017626 104037                   EMT      C$SVEC
(2) 017630 062706 000010          ADD      #10,SP
3498 017634 032777 000200 162370 1$:
3499 017642 001774                   BEQ      1$
3500
3501 017644 042777 000100 162360    BIC      #INTEN,@RLCS
3502
3503 017652                   CLRVEC   BVEC
(3) 017652 013700 002244          MOV      BVEC,R0
(3) 017656 104036                   EMT      C$CVEC
3504 017660 005737 002276          TST      PWRFLG
3505 017664 001402                   BEQ      2$
3506 017666 005337 002276          DEC      PWRFLG
3507 017672                   2$:
(3) 017672 013700 002122          CLRVEC   ERRVEC
(3) 017676 104036                   MOV      ERRVEC,R0
(3) 017676 104036                   EMT      C$CVEC
3508
3509
3510
3511 017700                   ENDCLN
(3) 017700                   L10023:
(3) 017700 104012                   EMT      C$CLEAN
3512
3513 017702                   ENDMOD
3514

```

```

3515 017702          BGNMOD DRPCODE
3516
3517 017702          BGNDU
3518
3519 017702 000240   NOP
3520
3521 017704          ENDDU
(3) 017704          L10024:
(3) 017704 104055   EMT      C$DU
3522
3523 017706          ENDMOD
3524
3525          .SBTTL GLOBAL SUBROUTINES
3526
3527 017706          BGNMOD GLBSUB
3528
3529 017706          BGNSRV
3530 017706 005237 002134 INTSRV: INC      INTFLG      ;SET INTERRUPT OCCURANCE FLAG
3531
3532 017712          ENDSRV
(3) 017712          L10C25:
(2) 017712 000002   RTI
3533
3534          ;ROUTINE USED IN TIMING OPI
3535
3536 017714 005237 002134 TIMSRV: INC      INTFLG
3537 017720          ABORTWAIT
(3) 017720 104021   EMT      C$ABRT
3538 017722 000002   RTI
3539
3540 017724 000240   CKERLT: NOP
3541 017726          INLOOP
(3) 017726 104020   EMT      C$INLP
3542 017730          BCOMPLETE 99%
(2) 017730 103427   BCS      99%
3543
3544 017732 005737 016566          TST      DROP
3545 017736 001424          BEQ      99%
3546 017740 005237 002260          INC      ERRLMT
3547 017744 023737 002260 016570          CMP      ERRLMT, MERLMT
3548 017752 002416          BLT      99%
3549
3550 017754          PRINTF #FRMT11
(7) 017754 012746 016350          MOV      #FRMT11, -(SP)
(6) 017760 012746 000001          MOV      #1, -(SP)
(3) 017764 010600          MOV      SP, R0
(4) 017766 104017          EMT      C$PNTF
(4) 017770 062706 000004          ADD      #4, SP
3551 017774 004737 015056          JSR      PC, LINE1
3552 020000          DODU UNITST ;DROP THIS UNIT
(3) 020000 013700 002130          MOV      UNITST, R0
(3) 020004 104053          EMT      C$DODU
3553 020006          DOCLN
(3) 020006 104044          EMT      C$DCLN
3554
3555 020010          99%:
  
```

```

3556 020010 000205          RTS    R5
3557
3558
3559          .SBTTL  ROUTINE TO CHECK FOR CONTROLLER ERRORS
3560
3561          ;*****
3562          ;*THIS ROUTINE WILL CHECK RLCS FOR ERRORS AND PRINT THEM
3563          ;*ACCORDINGLY. IT WILL MERGE THE ERROR PRINTOUT WITH THE TEST
3564          ;*ERROR MESSAGE.
3565          ;*
3566          ;*ROUTINE USES R0,R1 AND PICKS HEADER FROM R3
3567          ;*
3568          ;*      CALL    JSR      R5,CHERR          ;CHECK CNTLR FOR ERRORS
3569          ;*
3570          ;*
3571          ;*
3572
3573 020012 005037 002114      CHERR: CLR    T.CRC
3574 020016 032737 176000 002216  BIT    #176000,E.CS ;ANY ERROR BITS SET?
3575 020024 001001          BNE    2$      ;YES,FIND OUT WHICH
3576 020026 000205          RTS    R5      ;NO EXIT
3577 020030 012701 010217      2$:  MOV    #EM100,R1 ;GET START OF STRING
3578 020034 005737 002216      TST    E.CS    ;IS COMPOSITE ERROR SET?(BETTER BE)
3579 020040 100003          BPL    99$    ;IT'S NOT SOMETHING IS WRONG
3580 020042 004537 020550      JSR    R5,FIX ;YES, PUT "COMP" IN STRING
3581 020046 007116          COMP  ;"COMP"
3582 020050 032737 040000 002216  99$: BIT    #DERR,E.CS ;DRIVE ERROR SET?
3583 020056 001405          BEQ    3$      ;NO, CONTINUE
3584 020060 005237 002302      INC    DERFLG
3585 020064 004537 020550      JSR    R5,FIX ;YES, PUT "DRV" INTO STRING
3586 020070 007045          DEMES ;"DRV"
3587 020072 032737 020000 002216  3$: BIT    #NXM,E.CS ;NON-EXISTENT MEMORY ERROR?
3588 020100 001403          BEQ    4$      ;NO, CONTINUE
3589 020102 004537 020550      JSR    R5,FIX ;YES, PUT "NXM" INTO STRING
3590 020106 007052          NXMES ;"NXM"
3591 020110 032737 002000 002216  4$: BIT    #OPI,E.CS ;IS OPI SET?
3592 020116 001422          BEQ    6$      ;NO, GO CHECK BITS 11 & 12
3593 020120 004537 020550      JSR    R5,FIX ;PUT "OPI" INTO STRING
3594 020124 007057          OPIMES ;"OPI"
3595 020126 032737 004000 002216      BIT    #BIT11,E.CS ;HEADERCRC ERROR?
3596 020134 001403          BEQ    5$      ;NO, GO CHECK HEADER NOT FOUND
3597 020136 004537 020550      JSR    R5,FIX ;GO PUT "HCRC" IN STRING
3598 020142 007064          HRCMES ;"HCRC"
3599 020144 032737 010000 002216  5$: BIT    #BIT12,E.CS ;HEADER NOT FOUND?
3600 020152 001424          BEQ    8$      ;NO, GO PUT "CRLF" IN STRING
3601 020154 004537 020550      JSR    R5,FIX ;PUT "HNF" IN STRING
3602 020160 007072          HNFMES ;"HNF"
3603 020162 000420          BR     8$      ;PUT "CRLF" IN STRING
3604 020164 032737 004000 002216  6$: BIT    #BIT11,E.CS ;DATA CPC ERROR?
3605 020172 001405          BEQ    7$      ;NO, GO CHECK DATA LATE
3606 020174 005237 002114      INC    T.CRC
3607 020200 004537 020550      JSR    R5,FIX ;PUT "DCK" IN STRING
3608 020204 007077          DCKMES ;"DCK"
3609 020206 032737 010000 002216  7$: BIT    #BIT12,E.CS ;DATA LATE ERROR?
3610 020214 001403          BEQ    8$      ;NO, GO PUT IN "CRLF"
3611 020216 004537 020550      JSR    R5,FIX ;PUT "DLT" IN STRING

```

```
3612 020222 007104          DLTMS          ;"DLT"  
3613 020224 004537 020550 8$: JSR      R5, FIX    ;PUT "CRLF" INTO STRING  
3614 020230 007113          MSCRLF         ;"CRLF"  
3615 020232 004537 020550  RESTMS: JSR      R5, FIX    ;MOVE HEADER  
3616 020236 000000          .WORD      0      ;HEADER FROM TEST  
3617 020240 105011          CLR      (R1)    ;PUT TERMINATOR IN  
3618 020242          ERRDF      300., LF, ERR6  
   (3) 020242 104462          TRAP     T$ERCODE  
   (5) 020244 000454          .WORD     300  
   (5) 020246 007111          .WORD     LF  
   (5) 020250 014314          .WORD     ERR6  
3619 020252 000205          RTS      R5      ;EXIT ROUTINE  
3620  
3621          .SBTTL  LOAD RLCS  
3622          ;*****  
3623          ;* ROUTINE TO LOAD RLCS WITH FUNCTION TO BE PERFORMED  
3624          ;*      CALL:  JSR      R5, LDFUNC    ;LOAD THE FUNCTION IN NEXT WORD  
3625          ;*          .WORD          ;BITS TO BE LOADED, FUNCTION  
3626          ;*          ;AND INTR ENABLE ONLY  
3627          ;*  
3628          ;  
3629  
3630 020254 032777 040000 161750 LDFUNC: BIT      #BIT14, @RLCS ;DRIVE ERROR SET  
3631 020262 001426          BEQ      5$  
3632 020264 017737 161746 002212  MOV     @RLDA, B, DA  
3633 020272 012777 000013 161736  MOV     #13, @RLDA  
3634 020300 012737 000200 002206  MOV     #200, B, CS  
3635 020306 053737 002124 002206  BIS     DRIVE, B, CS  
3636 020314 013777 002206 161710  MOV     B, CS, @RLCS  
3637 020322 032777 000200 161702 6$: BIT      #200, @RLCS  
3638 020330 001774          BEQ      6$  
3639 020332 013777 002212 161676  MOV     B, DA, @RLDA  
3640 020340 012537 002136 5$: MOV     (R5)+, LDCSR ;GET BITS TO LOAD  
3641 020344 010346          MOV     R3, -(SP) ;SAVE R3  
3642 020346 042737 177661 002136  BIC     #177661, LDCSR ;CLEAR ALL BUT FUNC & INTR EN  
3643 020354 013737 002136 002250  MOV     LDCSR, FNDFNC ;SAVE FUNCTION  
3644 020362 042737 000100 002250  BIC     #INTEN, FNDFNC ;ONLY FUNCTION  
3645 020370 012703 020510          MOV     #HDRLIST, R3 ;GET HEADER LIST  
3646 020374 006237 002250          ASR     FNDFNC ;ALIGN TO LEFT  
3647 020400 001404          BEQ     2$ ;IF EQUAL TO ZERO, SET R3  
3648 020402 022323          1$: CMP     (R3)+, (R3)+ ;BUMP R3 BY 4  
3649 020404 005337 002250          DEC     FNDFNC ;DEC FUNCTION  
3650 020410 001374          BNE     1$ ;FOUND IT? NO-GO BACK  
3651 020412 032737 000100 002136 2$: BIT      #INTEN, LDCSR ;YES, DO WE WANT FLAG OR INTR?  
3652 020420 001401          BEQ     3$ ;FLAG BRANCH  
3653 020422 005723          TST     (R3)+ ;INTR POINT TO THAT ONE  
3654 020424 011303          3$: MOV     (R3), R3 ;SET HEADER  
3655 020426 010337 020236          MOV     R3, RESTMS ;SET UP HEADER  
3656 020432 010337 002254          MOV     R3, TRYFNC ;SAVE HEADER FOR LATER  
3657 020436 053737 002252 002136  BIS     XMEM, LDCSR ;LOAD E. A. BITS  
3658 020444 005037 002252          CLR     XMEM ;CLEAR OUT THE BITS  
3659 020450 053737 002124 002136  BIS     DRIVE, LDCSR ;SELECT DRIVE  
3660 020456 052737 000200 002136  BIS     #200, LDCSR  
3661 020464 013777 002136 161540  MOV     LDCSR, @RLCS ;LOAD FUNCTION  
3662 020472 004537 020562          JSR     R5, BEFORE ;READ REGISTERS  
3663 020476 042777 000200 161526 4$: BIC     #200, @RLCS ;ISSUE COMMAND
```

```

3664 020504 012603          MOV    (SP)+,R3      ;RESTORE R3
3665 020506 000205          RTS      R5          ;EXIT
3666
3667
3668
3669 020510 007177          HDRLST: NOPMES
3670 020512 007230          NOPINT
3671 020514 007262          WCKMES
3672 020516 007322          WCKINT
3673 020520 007547          GSTMES
3674 020522 007606          GSTINT
3675 020524 007464          SEKMES
3676 020526 007515          SEKINT
3677 020530 007363          RHDMES
3678 020532 007423          RHDINT
3679 020534 007727          WRTMES
3680 020536 007761          WRTINT
3681 020540 007645          RDDMES
3682 020542 007676          RDDINT
3683 020544 010013          RDNMES
3684 020546 010047          RDNINT
3685
3686          ;*****
3687          ;*ROUTINE TO MOVE ASCII STRINGS
3688          ;*USES REGISTERS R1 - WHERE STRING IS BEING BUILT
3689          ;*
3690          ;*      CALL    JSR      R5, FIX
3691          ;*      .WORD    .        ;ADDRESS OF STRING TO MOVE
3692
3693 020550 012504          FIX:  MOV    (R5)+,R4      ;GET ADDRESS AND MOVE RETURN
3694 020552 112421          15:  MOVB   (R4)+,(R1)+    ;GET BYTE AND UPDATE
3695 020554 001376          BNE    15              ;WATCH 0 BYTE TERMINATOR
3696 020556 105741          TSTB   -(R1)          ;BACK UP OVER ZERO BYTE
3697 020560 000205          RTS      R5          ;EXIT
3698
3699
3700          ;ROUTINE TO READ REGISTERS PRIOR TO OPERATION
3701          ;CALL:  JSR R5,BEFORE
3702
3703 020562 017737 161444 002206  BEFORE: MOV    @RLCS,B. CS      ;READ CS
3704 020570 017737 161440 002210          MOV    @RLBA,B. BA      ;   BA
3705 020576 017737 161434 002212          MOV    @RLDA,B. DA      ;   DA
3706 020604 017737 161430 002214          MOV    @RLMP,B. MP      ;   MP
3707 020612 000205          RTS      R5
3708
3709          ;ROUTINE TO READ REGISTERS AT TIME OF ERROR
3710          ;CALL:  JSR R5,AFTER
3711
3712 020614 017737 161412 002216  AFTER: MOV    @RLCS,E. CS    ;READ CS
3713 020622 017737 161406 002220          MOV    @RLBA,E. BA      ;   BA
3714 020630 017737 161402 002222          MOV    @RLDA,E. DA      ;   DA
3715 020636 017737 161376 002224          MOV    @RLMP,E. MP      ;   MP
3716 020644 017737 161370 002226          MOV    @RLMP,E. MP1     ;   MP
3717 020652 017737 161362 002230          MOV    @RLMP,E. MP2     ;   MP
3718 020660 000205          RTS      R5
3719

```

```

3720
3721 020662 010046          SIMBCC: MOV    RO, -(SP)      ;SAVE R0
3722 020664 010146          MOV    R1, -(SP)      ;SAVE R1
3723 020666 010246          MOV    R2, -(SP)      ;SAVE R2
3724 020670 012537 002162    MOV    (R5)+, TEMP2   ;GET NUMBER OF BITS
3725 020674 012537 002164    MOV    (R5)+, TEMP3   ;GET DATA FOR CRC CALCULATION
3726 020700 012537 002166    MOV    (R5)+, TEMP4   ;GET STARTING CRC
3727 020704 005037 002144    15:   CLR    BCCFBK      ;
3728 020710 013700 002166    MOV    TEMP4, RO      ;GET PRESENT CRC
3729 020714 006037 002164    ROR    TEMP3          ;ROTATE NEW DATA
3730 020720 005500          ADC    RO              ;MERGE NEW WITH OLD
3731 020722 032700 000001    BIT    #1, RO         ;BIT 0 SET
3732 020726 001402          BEQ    25             ;IF NOT CONTINUE
3733 020730 005137 002144    COM    BCCFBK        ;
3734 020734 013700 002142    25:   MOV    XPOLY, RO   ;GET CRC POLYNOMIAL (CRC-16)
3735 020740 005100          COM    RO             ;COMPLIMENT POLYNOMIAL
3736 020742 040037 002144    BIC    RO, BCCFBK
3737 020746 000241          CLC                    ;CLEAR CARRY
3738 020750 006037 002166    ROR    TEMP4
3739 020754 013700 002144    MOV    BCCFBK, RO
3740 020760 013701 002166    MOV    TEMP4, R1
3741 020764 010102          MOV    R1, R2
3742 020766 040100          BIC    R1, RO
3743 020770 043702 002144    BIC    BCCFBK, R2
3744 020774 050200          BIS    R2, RO
3745 020776 043737 002142 002166    BIC    XPOLY, TEMP4
3746 021004 050037 002166    BIS    RO, TEMP4
3747 021010 005337 002162    DEC    TEMP2
3748 021014 001333          BNE    15
3749
3750 021016 013737 002166 002146    MOV    TEMP4, CALBCC
3751 021024 012602          MOV    (SP)+, R2
3752 021026 012601          MOV    (SP)+, R1
3753 021030 012600          MOV    (SP)+, RO
3754 021032 000205          RTS    R5              ;RETURN
3755
3756
3757          ;ROUTINE TO WAIT FOR DRIVE READY
3758
3759
3760
3761
3762 021034 012701 000144          WTDROY: MOV    #100., R1
3763 021040 032777 000001 161164 15:   BIT    #DRDY, @RLCS
3764 021046 001011          BNE    25
3765
3766 021050          WAITUS #20.
(3) 021050 012700 000024    MOV    #20., RO
(3) 021054 104027          EMT    CSWTU
3767 021056 005301          DEC    R1
3768 021060 001367          BNE    15
3769
3770 021062          ERRDF 200., DRTIM, ERR5
(3) 021062 104462          TRAP  TSERCODE
(5) 021064 000310          .WORD 200
(5) 021066 007017          .WORD  DRTIM
  
```



```

(5) 021070 014256 . WORD ERR5
3771
3772 021072 000205 2$: RTS R5
3773
3774 ;ROUTINE TO WAIT FOR CONTROLLER
3775
3776 021074 012701 000310 WTCRDY: MOV #200.,R1
3777 021100 032777 000200 161124 1$: BIT #CRDY,@RLCS
3778 021106 001014 BNE 2$
3779
3780 WAITUS #20.
(3) 021110 012700 000024 MOV #20.,R0
(3) 021114 104027 EMT C$WTU
3781 021116 005301 DEC R1
3782 021120 001367 BNE 1$
3783 021122 004537 020614 JSR R5,AFTER
3784
3785 021126 ERDF 100.,CRTIM,ERR5
(3) 021126 104462 TRAP T$ERCODE
(5) 021130 000144 . WORD 100
(5) 021132 006772 . WORD CRTIM
(5) 021134 014256 . WORD ERR5
3786 021136 000205 RTS R5
3787
3788 021140 004537 020614 2$: JSR R5,AFTER
3789 021144 000205 RTS R5
3790
3791
3792 021146 005237 002132 TRPHAN: INC TRPFLG
3793 021152 000002 RTI
3794
3795 HDHOME:
3796
3797 021154 BGNSEG ;%%START OF SEGMENT%%
(3) 021154 104004 EMT C$BSEG
3798 ;ISSUE DRIVE RESET
3799
3800 021156 012737 000001 002256 MOV #1,ERFLG ;SET ERROR FLAG
3801 021164 012777 000013 161044 MOV #DRST!MK!GSBIT,@RLDA
3802 021172 004537 020254 JSR R5,LDFUNC ;LOAD THE FUNCTION IN NEXT WORD
3803 021176 000004 GSTAT
3804 021200 004537 021074 JSR R5,WTCRDY
3805 021204 ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 021204 104010 EMT C$ESCAPE
(3) 021206 000216 . WORD 10000$-.
3806 021210 004537 020012 JSR R5,CHERR ;CHECK CNTLR FOR ERRORS
3807 021214 ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 021214 104010 EMT C$ESCAPE
(3) 021216 000206 . WORD 10000$-.
3808
3809
3810 021220 004537 020254 JSR R5,LDFUNC ;LOAD THE FUNCTION IN NEXT WORD
3811 021224 000010 RDHDR
3812 021226 ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 021226 104010 EMT C$ESCAPE
(3) 021230 000174 . WORD 10000$-.
  
```

3813	021232	004537	021074		JSR	R5, WTCRDY	
3814	021236				ESCAPE	SEG	; CHECK FOR FL: LOE, ELSE EXIT SEG
(3)	021236	104010			EMT	C\$ESCAPE	
(3)	021240	000164			. WORD	10000\$-	
3815							
3816	021242	004537	020012		JSR	R5, CHERR	; CHECK CNTLR FOR ERRORS
3817	021246				ESCAPE	SEG	; CHECK FOR FL: LOE, ELSE EXIT SEG
(3)	021246	104010			EMT	C\$ESCAPE	
(3)	021250	000154			. WORD	10000\$-	
3818							
3819	021252	013737	002224	002150	MOV	E. MP, TMPO	; GET HEADER
3820	021260	042737	000077	002150	BIC	#77, TMPO	
3821	021266	001424			BEQ	99\$	; SEEK IS NOT NECESSARY
3822	021270	042737	000100	002150	BIC	#100, TMPO	
3823	021276	012777	000001	160732	MOV	#MK, @RLDA	; SET TO SEEK
3824	021304	053777	002150	160724	BIS	TMPO, @RLDA	; SET IN DIFFERENCE
3825							
3826	021312	004537	020254		JSR	R5, LDFUNC	; LOAD THE FUNCTION IN NEXT WORD
3827	021316	000006			SEEK		
3828	021320	004537	021074		JSR	R5, WTCRDY	
3829	021324				ESCAPE	SEG	; CHECK FOR FL: LOE, ELSE EXIT SEG
(3)	021324	104010			EMT	C\$ESCAPE	
(3)	021326	000076			. WORD	10000\$-	
3830							
3831	021330	004537	020012		JSR	R5, CHERR	; CHECK CNTLR FOR ERRORS
3832	021334				ESCAPE	SEG	; CHECK FOR FL: LOE, ELSE EXIT SEG
(3)	021334	104010			EMT	C\$ESCAPE	
(3)	021336	000066			. WORD	10000\$-	
3833							
3834	021340	004537	020254	99\$:	JSR	R5, LDFUNC	; LOAD THE FUNCTION IN NEXT WORD
3835	021344	000010			RDHDR		
3836	021346	004537	021074		JSR	R5, WTCRDY	
3837	021352				ESCAPE	SEG	; CHECK FOR FL: LOE, ELSE EXIT SEG
(3)	021352	104010			EMT	C\$ESCAPE	
(3)	021354	000050			. WORD	10000\$-	
3838	021356	004537	020012		JSR	R5, CHERR	
3839	021362				ESCAPE	SEG	
(3)	021362	104010			EMT	C\$ESCAPE	
(3)	021364	000040			. WORD	10000\$-	
3840							
3841	021366	013737	002224	002150	MOV	E. MP, TMPO	; GET HEADER
3842	021374	043737	002140	002150	BIC	SECMSK, TMPO	; IGNORE SECTOR
3843	021402	001404			BEQ	1\$	; ON ZERO
3844							
3845	021404				ERRDF	400., SKHOME, ERRO	; CAN'T SEEK TO TRACK 0
(3)	021404	104462			TRAP	T\$ERCODE	
(5)	021406	000620			. WORD	400	
(5)	021410	010103			. WORD	SKHOME	
(5)	021412	014044			. WORD	ERRO	
3846							
3847	021414			1\$:	ESCAPE	SEG	; CHECK FOR FL: LOE, ELSE EXIT SEG
(3)	021414	104010			EMT	C\$ESCAPE	
(3)	021416	000006			. WORD	10000\$-	
3848							
3849	021420	005037	002256		CLR	ERFLG	; INDICATE SUCCESS BACK TO MAIN PROGRAM
3850							

```
3851
3852 021424          ENDSEG          ;%%END OF SEGMENT%%
(3) 021424          10000$:
(3) 021424 104005    EMT          C$ESEG
3853
3854 021426 000207    RTS          PC
3855
3856 021430          ENDMOD
3857
3858          .SBTTL  **TEST 1** - WRITE NPR INTEGRITY
3859
3860 021430          BGNST          ;**START OF TEST**
3861
3862 021430          STARS
(2)          ; ;*****
3863          ;CHECK THAT NPR WILL NOT INTERFERE WITH THE OPERATION OF THE
3864          ;UNIBUS. WE SET UP LOCATION 4 TO HANDLE THE TRAP IF IT HAPPENS.
3865 021430          STARS
(2)          ; ;*****
3866
3867
3868 021430 004737 021154    JSR          PC, HDHOME          ;HEADS OVER TRACK 0
3869 021434          CKERFG          ;HEADS GO HOME OKAY
(4) 021442 104032    EMT          C$EXIT
(4) 021444 000232    .WORD        L10026-.
3870
3871 021446          BGNSEG          ;%%START OF SEGMENT%%
(3) 021446 104004    EMT          C$BSEG
3872
3873 021450          1$:          SETVEC  ERRVEC, #TRPHAN, #340          ;SET UP FOR TRAP
(7) 021450 012746 000340    MOV          #340, -(SP)
(6) 021454 012746 021146    MOV          #TRPHAN, -(SP)
(5) 021460 013746 002122    MOV          ERRVEC, -(SP)
(4) 021464 012746 000003    MOV          #3, -(SP)
(3) 021470 104037    EMT          C$SVEC
(2) 021472 062706 000010    ADD          #10, SP
3874 021476 005037 002132    CLR          TRPFLG          ;CLEAR TRAP OCCURANCE
3875 021502 012777 002652 160524    MOV          #BUF, @RLBA          ;BUS ADDRESS
3876 021510 005077 160522    CLR          @RLDA          ;LOAD DISK ADDRESS
3877 021514 012777 177777 160516    MOV          #-1, @RLMP          ;WORD COUNT OF 1
3878 021522 005037 002156    CLR          GDDAT          ;SET UP CSR TO LOAD
3879 021526 013737 002124 002156    MOV          DRIVE, GDDAT          ;SET IN DRIVE
3880 021534 052737 000012 002156    BIS          #WRITE, GDDAT          ;SET IN FUNCTION
3881 021542 004537 020562    JSR          R5, BEFORE          ;LOAD FOR ERROR PRINTOUT
3882 021546 013737 002156 002206    MOV          GDDAT, B. CS          ;SET IN COMMAND
3883 021554 052737 000201 002206    BIS          #201, B. CS          ;LOAD CRDY
3884 021562 042737 002000 002206    BIC          #OPI, B. CS          ;CLEAR (BIT 10)
3885 021570 013777 002156 160434    MOV          GDDAT, @RLCS          ;ISSUE WRITE
3886 021576 012701 000144    MOV          #100, R1          ;WAIT FOR CRDY
3887 021602 032777 000200 160422 5$:    BIT          #CRDY, @RLCS          ;NPR DONE
3888 021610 001013    BNE          6$          ;YES, 6$
3889 021612          WAITUS  #20.          ;WAIT A WHILE
(3) 021612 012700 000024    MOV          #20, R0
(3) 021616 104027    EMT          C$WTU
3890 021620 005301    DEC          R1          ;A WHILE UP
3891 021622 001367    BNE          5$          ;NO, GO BACK
```

```

3892
3893 021624 004537 020614      JSR      R5, AFTER
3894 021630                      ERRDF    0., CRTIM, ERRS      ;CONTROLLER TIMED OUT
(3) 021630 104462      TRAP     T$ERCODE
(5) 021632 000000      .WORD   0
(5) 021634 006772      .WORD   CRTIM
(5) 021636 014256      .WORD   ERR5
3895 021640                      65:     CLRVEC  ERRVEC      ;CLEAR VECTOR
(3) 021640 013700 002122      MOV     ERRVEC, R0
(3) 021644 104036      EMT     C$CVEC
3896 021646                      ESCAPE  SEG          ;CHECK FOR FL: LOE, ELSE EXIT SEG
(3) 021646 104010      EMT     C$ESCAPE
(3) 021650 000024      .WORD   100015-.
3897
3898 021652 005737 002132      TST     TRPFLG      ;DID TRAP OCCUR?
3899 021656 001406      BEQ     7$          ;NO
3900 021660 004537 020614      JSR     R5, AFTER
3901 021664                      ERRSF   1., EM51, ERRO    ;TRAP ON WRITE
(3) 021664 104461      TRAP     T$ERCODE
(5) 021666 000001      .WORD   1
(5) 021670 012745      .WORD   EM51
(5) 021672 014044      .WORD   ERRO
3902 021674                      7$:
3903
3904
3905 021674                      ENDSEG          ;%%END OF SEGMENT%%
(3) 021674                      100015:
(3) 021674 104005      EMT     C$ESEG
3906
3907 021676                      ENDTST
(3) 021676                      L10026:        ;**END OF TEST**
(3) 021676 104001      EMT     C$ETST
3908
3909 .SBTTL **TEST 2** - WRITE FUNCTION
3910
3911 021700                      BGNTST        ;**START OF TEST**
3912
3913
3914
3915 021700                      STARS
(2) ;*****
3916 ;CHECK OF WRITE LOGIC UNDER FLAG MODE, WE WILL FIRST ISSUE A
3917 ;READ HEADER SO THAT WE DON'T WRITE ON THE BAD SECTOR
3918 ;FILE TRACK. WE WILL WRITE A FULL SECTOR (128 WORDS) FROM
3919 ;MEMORY (BUF). WE CHECK THAT NO ERRORS OCCUR. IF WE
3920 ;HAVE A DRIVE ERROR WE WILL DO A "GET STATUS" TO SEE
3921 ;IF WRITE PROTECT IS SET IF IT IS WE WILL ABORT THE
3922 ;TEST. AN ERROR ON THE WRITE WILL LOOP ON JUST THE
3923 ;WRITE PORTION. LOOP ON TEST WILL READ HEADER, SEEK (IF
3924 ;NECESSARY) AND WRITE.
3925 021700                      STARS
(2) ;*****
3926
3927
3928 021700 004737 021154      JSR     PC, HDHOME    ;HEADS OVER TRACK 0
3929 021704                      CKERFG          ;HEADS GO HOME OKAY
  
```

```
(4) 021712 104032      EMT      C$EXIT
(4) 021714 000126      .WORD   L10027-.
3930
3931 021716            BGNSEG                      ;%%START OF SEGMENT%%
(3) 021716 104004      EMT      C$BSEG
3932
3933 021720            3$:
3934 021720 005077 160312      CLR      @RLDA              ;SET DISK ADDRESS
3935 021724 012777 177600 160306      MOV      #-128.,@RLMP      ;WORD COUNT
3936 021732 012777 002652 160274      MOV      #BUF,@RLBA       ;BUS ADDRESS
3937 021740 004537 020254      JSR      R5,LDFUNC        ;LOAD THE FUNCTION IN NEXT WORD
3938 021744 000012      WRITE                      ;WRITE
3939
3940 021746 004537 021074      JSR      R5,WTCRDY        ;WAIT FOR CONTROLLER READY
3941 021752            ESCAPE   SEG          ;CHECK FOR FL: LOE, ELSE EXIT SEG
(3) 021752 104010      EMT      C$ESCAPE
(3) 021754 000064      .WORD   10000$-.
3942
3943
3944 021756 032777 040000 160246      BIT      #DERR,@RLCS      ;DRIVE ERROR SET?
3945 021764 001425      BEQ                      ;BRANCH IF NOT
3946
3947 021766 012777 000003 160242      MOV      #MK!GSBIT,@RLDA ;SET GET STATUS OF DRIVE
3948 021774 004537 020254      JSR      R5,LDFUNC        ;LOAD THE FUNCTION IN NEXT WORD
3949 022000 000004      GSTAT                      ;GET STATUS
3950 022002 004537 021074      JSR      R5,WTCRDY        ;WAIT FOR CONTROLLER READY
3951 022006            ESCAPE   SEG          ;CHECK FOR FL: LOE, ELSE EXIT SEG
(3) 022006 104010      EMT      C$ESCAPE
(3) 022010 000030      .WORD   10000$-.
3952
3953 022012 013737 002224 002156      MOV      E.MP,GDDAT       ;READ DRIVE STATUS
3954 022020 032737 020000 002156      BIT      #BIT13,GDDAT     ;WRITE LOCK ERROR?
3955 022026 001404      BEQ      4$              ;NO, BRANCH
3956
3957
3958 022030            ERRSF   3.,WRLOCK,ERRO      ;WRITE LOCK ERROR
(3) 022030 104461      TRAP    T$ERCODE
(5) 022032 000003      .WORD   3
(5) 022034 010131      .WORD   WRLOCK
(5) 022036 014044      .WORD   ERRO
3959 022040            4$:
3960
3961
3962 022040            ENDSEG                      ;%%END OF SEGMENT%%
(3) 022040            10000$:
(3) 022040 104005      EMT      C$ESEG
3963 022042            ENDTST
(3) 022042            L10027:
(3) 022042 104001      EMT      C$ETST
3964
3965            .SBTTL  **TEST 3** - WRITE FUNCTION INTERRUPT
3966
3967 022044            BGNTST                      ;**START OF TEST**
3968
3969 022044            STARS
(2) ; ;*****
```

```
3970 ;CHECK OF WRITE LOGIC UNDER INTERRUPT MODE, WE WILL ISSUE A
3971 ;READ HEADER SO THAT WE DON'T WRITE ON THE BAD SECTOR FILE
3972 ;TRACK. WE WILL WRITE A FULL SECTOR (128 WORDS) FROM MEMORY (BUF).
3973 ;WE CHECK THAT NO ERRORS OCCUR. WE DO NOT CHECK RLDA OR RLBA
3974 ;INCREMENT AT THIS TIME.
3975 022044 STARS
(2) ;*****
3976
3977
3978 022044 004737 021154 JSR PC,HDHOME ;HEADS OVER TRACK 0
3979 022050 CKERFG ;HEADS GO HOME OKAY
(4) 022056 104032 EMT C$EXIT
(4) 022060 000112 .WORD L10030-.
3980
3981 022062 BGNSEG ;%%START OF SEGMENT%%
(3) 022062 104004 EMT C$BSEG
3982
3983
3984 022064 005037 002134 CLR INTFLG ;CLEAR INTERRUPT OCCURANCE FLAG
3985 022070 005077 160142 CLR @RLDA
3986 022074 012777 177600 160136 MOV #-128.,@RLMP ;SET UP WORD COUNT
3987 022102 012777 002652 160124 MOV #BUF,@RLBA ;SET UP BUS ADDRESS
3988
3989 022110 SETPRI #PR100 ;PRIORITY TO 0
(3) 022110 012700 000000 MOV #PR100,RO
(3) 022114 104041 EMT C$SPRI
3990 022116 004537 020254 JSR R5,LDFUNC ;LOAD THE FUNCTION IN NEXT WORD
3991 022122 000112 WRITE!INTEN ;WRITE UNDER INTERRUPT
3992 022124 004537 021074 JSR R5,WTCRDY ;WAIT FOR INTERRUPT
3993 022130 ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 022130 104010 EMT C$ESCAPE
(3) 022132 000036 .WORD 10000$-.
3994
3995 022134 SETPRI #PR107 ;SET PRIORITY TO 7
(3) 022134 012700 000340 MOV #PR107,RO
(3) 022140 104041 EMT C$SPRI
3996 022142 005737 002134 TST INTFLG ;DID INTERRUPT OCCUR?
3997 022146 001004 BNE ZS ;YES-BRANCH NO-REPORT
3998
3999 022150 ERRDF 4.,EM17,ERRO ;WRITE DID NOT INTERRUPT
(3) 022150 104462 TRAP T$ERCODE
(5) 022152 000004 .WORD 4
(5) 022154 011151 .WORD EM17
(5) 022156 014044 .WORD ERRO
4000 022160 25: ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 022160 104010 EMT C$ESCAPE
(3) 022162 000006 .WORD 10000$-.
4001
4002 022164 004537 020012 JSR R5,CHERR ;CHECK CNTLR FOR ERRORS
4003
4004 022170 ENDSEG ;%%END OF SEGMENT%%
(3) 022170 10000$: EMT C$ESEG
(3) 022170 104005
4005 022172 ENDTST ;**END OF TEST**
(3) 022172 L10030: EMT C$SETST
(3) 022172 104001
```

```

4006
4007          .SBTTL **TEST 4** - PROPER INCREMENT OF RLBA ON WRITE
4008
4009 022174    BGNTST          ;**START OF TEST**
4010
4011
4012 022174    STARS
(2)          ;*****
4013          ;CHECK THAT THE RLBA WILL INCREMENT PROPERLY AFTER THE
4014          ;WRITE WAS FINISHED THE RLBA SHOULD BE 128 WORDS (256 BYTES)
4015          ;CREATER. STARTING RLBA IS "BUF", ENDING SHOULD BE "BUF + 256."
4016          ;WE WILL MONITOR ALL ERRORS AND REPORT THEM ACCORDINGLY
4017 022174    STARS
(2)          ;*****
4018
4019
4020 022174 004737 021154    JSR      PC,HDHOME      ;HEADS OVER TRACK 0
4021 022200    CKERFG          ;HEADS GO HOME OKAY
(4) 022206 104032    EMT      C$EXIT
(4) 022210 000116    .WORD   L10031-.
4022
4023 022212    BGNSEG          ;%%START OF SEGMENT%%
(3) 022212 104004    EMT      C$BSEG
4024
4025 022214    3$:
4026 022214 005077 160016    CLR      @RLDA
4027 022220 012777 002652 160006    MOV      #BUF,@RLBA      ;SET UP BUS ADDRESS
4028 022226 012777 177600 160004    MOV      #-128,@RLMP     ;WORD COUNT
4029 022234 012737 002652 002156    MOV      #BUF,GDDAT      ;FORM EXPECTED BUS ADDRESS
4030 022242 062737 000400 002156    ADD      #256,GDDAT      ;AFTER WRITE
4031
4032 022250 004537 020254    JSR      R5,LDFUNC        ;LOAD THE FUNCTION IN NEXT WORD
4033 022254 000012    WRITE          ;WRITE
4034 022256 004537 021074    JSR      R5,WTCRDY        ;WAIT FOR CONTROLLER READY
4035 022262    ESCAPE SEG      ;CHECK FOR FL: LOE, ELSE EXIT SEG
(3) 022262 104010    EMT      C$ESCAPE
(3) 022264 000040    .WORD   10000$-.
4036
4037 022266 004537 020012    JSR      R5,CHERR          ;CHECK CNTLR FOR ERRORS
4038 022272    ESCAPE SEG      ;CHECK FOR FL: LOE, ELSE EXIT SEG
(3) 022272 104010    EMT      C$ESCAPE
(3) 022274 000030    .WORD   10000$-.
4039 022276 017737 157732 002160    MOV      @RLBA,BDDAT      ;READ 'RLBA' FOR PRESENT ADDRESS
4040 022304 023737 002160 002156    CMP      BDDAT,GDDAT      ;DID 'BA' INCREMENT PROPERLY?
4041 022312 001404    BEQ      2$              ;YES, CONTINUE
4042
4043 022314    ERRDF 5.,EM20,ERR4      ;BA DID NOT INCREMENT
(3) 022314 104462    TRAP    T$ERCODE
(5) 022316 000005    .WORD   5
(5) 022320 011211    .WORD   EM20
(5) 022322 014210    .WORD   ERR4
4044
4045 022324    2$:
4046
4047 022324    ENDSEG          ;%%END OF SEGMENT%%
(3) 022324    10000$:

```

```

(3) 022324 104005          EMT      C$ESEG
4048 022326                ENDTST          ;**END OF TEST**
(3) 022326                L10031:
(3) 022326 104001          EMT      C$ETST
4049
4050          .SBTTL  **TEST 5** - PROPER INCREMENT OF RLDA ON WRITE
4051
4052 022330                BGNTST          ;**START OF TEST**
4053
4054 022330                STARS
(2)          ;*****
4055          ;CHECK THAT THE SECTOR INCREMENTS AFTER THE WRITE WAS FINISHED.
4056          ;WE RANDOMLY PICK A SECTOR (OTHER THAN LAST TRACK) AND ISSUE
4057          ;A FULL SECTOR WRITE THE RLDA SHOULD REFLECT AN INCREMENT
4058          ;OF THE SECOTR. "GDDAT" WAS THE EXPECTED RLDA.
4059 022330                STARS
(2)          ;*****
4060
4061
4062 022330 004737 021154    JSR      PC, HDHOME      ;HEADS OVER TRACK 0
4063 022334                CKERFG          ;HEADS GO HOME OKAY
(4) 022342 104032          EMT      C$EXIT
(4) 022344 000114          .WORD    L10032-.
4064
4065 022346                BGNSEG          ;%%START OF SEGMENT%%
(3) 022346 104004          EMT      C$BSEG
4066
4067 022350                35:
4068 022350 005037 002156    CLR      GDDAT
4069 022354 013777 002156 157654  MOV      GDDAT, @RLDA    ;SETUP DISK ADDRESS
4070 022362 005237 002156    INC      GDDAT          ;CREATE EXPECTED SECTOR
4071 022366 012777 177600 157644  MOV      #-128., @RLMP   ;WORD COUNT
4072 022374 012777 002652 157632  MOV      #BUF, @RLBA    ;SETUP BUS ADDRESS
4073
4074 022402 004537 020254    JSR      R5, LDFUNC      ;LOAD THE FUNCTION IN NEXT WORD
4075 022406 000012          WRITE          ;WRITE
4076 022410 004537 021074    JSR      R5, WTCRDY      ;WAIT FOR CONTROLLER READY
4077 022414                ESCAPE SEG          ;CHECK FOR FL: LOE, ELSE EXIT SEG
(3) 022414 104010          EMT      C$ESCAPE
(3) 022416 000040          .WORD    10000$-.
4078
4079 022420 004537 020012    JSR      R5, CHERR          ;CHECK CNTLR FOR ERRORS
4080 022424                ESCAPE SEG          ;CHECK FOR FL: LOE, ELSE EXIT SEG
(3) 022424 104010          EMT      C$ESCAPE
(3) 022426 000030          .WORD    10000$-.
4081
4082 022430 013737 002222 002160  MOV      E. DA, BDDAT    ;READ DISK ADDRESS
4083 022436 023737 002156 002160  CMP      GDDAT, BDDAT    ;DID SECTOR INCREMENT PROPERLY
4084 022444 001404          BEQ      25             ;YES, BRANCH NO, REPORT ERROR
4085
4086 022446                ERRDF 6., EM21, ERR4    ;DA DID NOT INCREMENT
(3) 022446 104462          TRAP      T$ERCODE
(5) 022450 000006          .WORD    6
(5) 022452 011266          .WORD    EM21
(5) 022454 014210          .WORD    ERR4
4087

```



```
4088 022456          25:
4089
4090 022456          ENDSEG                ;%%END OF SEGMENT%%
   (3) 022456          10000$:
   (3) 022456 104005  EMT      C$ESEG
4091 022460          ENDTST                ;**END OF TEST**
   (3) 022460          L10032:
   (3) 022460 104001  EMT      C$ETST
4092
4093          .SBTTL  **TEST 6** - FORCE HEADER NOT FOUND WITH WRITE
4094
4095 022462          BGNTST                ;**START OF TEST**
4096
4097 022462          STARS
   (2)          ; ;*****
4098          ; FORCE HEADER NOT FOUND ERROR TO OCCUR. THIS IS DONE
4099          ; BY SETTING SECTOR 40 OF THE RLDA AND ISSUING A
4100          ; WRITE. SECTOR 40 DOES NOT EXIST ON THE RLO1 PACK
4101          ; THEREFORE HDR NT FOUND SHOULD SET.
4102 022462          STARS
   (2)          ; ;*****
4103
4104 022462 004737 021154  JSR      PC, HDHOME      ; HEADS OVER TRACK 0
4105 022466          CKERFG          ; HEADS GO HOME OKAY
   (4) 022474 104032  EMT      C$EXIT
   (4) 022476 000120  .WORD    L10033-.
4106
4107 022500          BGNSEG                ;%%START OF SEGMENT%%
   (3) 022500 104004  EMT      C$BSEG
4108
4109
4110 022502 012777 000050 157526  MOV     #40, @RLDA      ; INSURE NOT TO FIND HEADER BY
4111 022510 012777 002652 157516  MOV     #BUF, @RLBA    ; SETTING SECTOR 40 OF CYL. ADDR.
4112 022516 012777 177777 157514  MOV     #-1, @RLMP    ; WORD COUNT
4113
4114 022524 004537 020254  JSR     R5, LDFUNC      ; LOAD THE FUNCTION IN NEXT WORD
4115 022530 000012          WRITE          ; WRITE
4116 022532 004537 021074  JSR     R5, WTCRDY     ; WAIT FOR CONTROLLER READY
4117 022536          ESCAPE          SEG      ; CHECK FOR FL: LOE, ELSE EXIT SEG
   (3) 022536 104010  EMT      C$ESCAPE
   (3) 022540 000054  .WORD    10000$-.
4118
4119 022542 013737 002216 002150  MOV     E, CS, TMPO    ; GET RLCS
4120 022550 042737 001777 002150  BIC     #1777, TMPO    ; SAVE ERROR BITS
4121 022556 022737 112000 002150  CMP     #BIT15!BIT12!BIT10, TMPO ; HDR NOT FOUND SET.
4122 022564 001402          BEQ     15          ; YES, CONTINUE
4123
4124 022566 004537 020012          JSR     R5, CHERR
4125 022572          CKLOOP
   (3) 022572 104006  EMT      C$CLP1
4126
4127 022574 022737 112000 002150  CMP     #BIT15!BIT12!BIT10, TMPO
4128 022602 001404          BEQ     25
4129 022604          ERRDF          23, EN10, ERRO
   (3) 022604 104462  TRAP     T$ERCODE
   (5) 022606 000027  .WORD    23
```

```

(5) 022610 010643 . WORD EM10
(5) 022612 014044 . WORD ERRO
4130 ; WHEN FORCED
4131 022614 25:
4132
4133 022614 ENDSEG ; %%END OF SEGMENT%%
(3) 022614 100005: EMT C$ESEG
(3) 022614 104005 ENDTST ; **END OF TEST**
4134 022616 L10033: EMT C$ETST
(3) 022616 104001
4135 . SBTTL **TEST 7** - FORCE HEADER NOT FOUND WITH WRITE INTERRUPT
4136
4137 BGNTST ; **START OF TEST**
4138 022620
4139
4140
4141 022620 STARS
(2) ; ; *****
4142 ; TEST THAT HEADER NOT FOUND ERROR WILL GENERATE AN INTERRUPT
4143 ; ON OCCURANCE. HEADER NOT FOUND WILL BE FORCED BY SETTING
4144 ; SECTOR 40 OF RLDA AND ISSUING A WRITE
4145 022620 STARS
(2) ; ; *****
4146
4147
4148 022620 004737 021154 JSR PC, HDHOME ; HEADS OVER TRACK 0
4149 022624 CKERFG ; HEADS GO HOME OKAY
(4) 022632 104032 EMT C$EXIT
(4) 022634 000160 . WORD L10034-.
4150
4151 022636 BGNSEG ; %%START OF SEGMENT%%
(3) 022636 104004 EMT C$BSEG
4152
4153 022640 SETPRI #PR100
(3) 022640 012700 000000 MOV #PR100, RO
(3) 022644 104041 EMT C$SPRI
4154 022646 005037 002134 CLR INTFLG ; CLEAR INTERRUPT OCCURANCE FLAG
4155 022652 012777 000050 157356 MOV #40, @RLDA ; INSURE NOT TO FIND HEADER BY
4156 022660 012777 002652 157346 MOV #BUF, @RLBA ; SETTING SECTOR 40 OF CYL. ADDR.
4157 022666 012777 177777 157344 MOV #-1, @RLMP ; WORD COUNT
4158
4159 022674 004537 020254 JSR R5, LDFUNC ; LOAD THE FUNCTION IN NEXT WORD
4160 022700 000112 WRITE! INTEN ; WRITE
4161 022702 004537 021074 JSR R5, WTCRDY ; WAIT FOR CONTROLLER READY
4162 022706 CKLOOP
(3) 022706 104006 EMT C$CLP1
4163 022710 SETPRI #PR107
(3) 022710 012700 000340 MOV #PR107, RO
(3) 022714 104041 EMT C$SPRI
4164
4165 022716 005737 002134 TST INTFLG ; DID INTERRUPT OCCUR
4166 022722 001004 BNE 25 ; YES OKAY
4167
4168 022724 ERRDF 24, EM43, ERRO ; NO INTERRUPT FROM OPI
(3) 022724 104462 TRAP T$ERCODE

```

```

OUTERR MACY11 30(1046) 03-NOV-77 10:02 PAGE 83-35          G 5
DZRLER P11 03-NOV-77 09:54          **TEST 7** - FORCE HEADER NOT FOUND WITH WRITE INTERRUPT          SEQ 0058

(5) 022726 000030          .WORD 24
(5) 022730 012523          .WORD EM43
(5) 022732 014044          .WORD ERRO
4169
4170 022734          2$: ESCAPE SEG          ;CHECK FOR FL: LOE, ELSE EXIT SEG
(3) 022734 104010          EMT C$ESCAPE
(3) 022736 000054          .WORD 10000$-.
4171
4172
4173 022740 013737 002216 002150          MOV E. CS, TMPO          ;GET RLCS
4174 022746 042737 001777 002150          BIC #1777, TMPO          ;SAVE ERROR BITS
4175 022754 022737 112000 002150          CMP #BIT15!BIT12!BIT10, TMPO ;WDR NOT FOUND SET.
4176 022762 001402          BEQ 1$          ;YES, CONTINUE
4177
4178 022764 004537 020012          JSR R5, CHERR
4179 022770          1$: CKLOOP
(3) 022770 104006          EMT C$CLP1
4180
4181 022772 022737 112000 002150          CMP #BIT15!BIT12!BIT10, TMPO
4182 023000 001404          BEQ 3$
4183 023002          ERRDF 25, EM10, ERRO
(3) 023002 104462          TRAP T$ERCODE
(5) 023004 000031          .WORD 25
(5) 023006 010643          .WORD EM10
(5) 023010 014044          .WORD ERRO
4184
4185 023012          3$:          ;WHEN FORCED
4186
4187 023012          ENDSEG          ;%%END OF SEGMENT%%
(3) 023012          10000$.
(3) 023012 104005          EMT C$ESEG
4188 023014          ENDTST          ;**END OF TEST**
(3) 023014          L10034:
(3) 023014 104001          EMT C$ETST
4189
4190
4191
4192          .SBTTL **TEST 8** - CHECK OPI TIME WITH HDR NT FND
4193
4194 023016          BGNTST          ;**START OF TEST**
4195
4196 023016          STARS
(2)          ;*****
4197          ;CHECK OPI TIME IT SHOULD BE AROUND 200 MILLISECONDS (ON UNIBUS)
4198          ;CHECK THIS BY TIMING OPI ON A FORCED HEADER NOT FOUND
4199          ;ISSUE WRITE WITH SECTOR 40 SET IN THE DISK ADDRESS
4200 023016          STARS
(2)          ;*****
4201 023016 004737 021154          JSR PC, HDHOME          ;HEADS OVER TRACK 0
4202 023022          CKERFG          ;HEADS GO HOME OKAY
(4) 023030 104032          EMT C$EXIT
(4) 023032 000264          .WORD L10035-.
4204
4205 023034          BGNSEG          ;%%START OF SEGMENT%%
(3) 023034 104004          EMT C$BSEG

```

```
4206
4207 023036          CLRVEC  BVEC          ;CLEAR PRESENT INTERRUPT VECTOR
      (3) 023036 013700 002244  MOV     BVEC,RO
      (3) 023042 104036          EMT     C$CVEC
4208 023044          SETVEC  BVEC,#TIMSRV,#340 ;SET INTR. VEC. WITH ABORT WAIT
      (7) 023044 012746 000340  MOV     #340,-(SP)
      (6) 023050 012746 017714  MOV     #TIMSRV,-(SP)
      (5) 023054 013746 002244  MOV     BVEC,-(SP)
      (4) 023060 012746 000003  MOV     #3,-(SP)
      (3) 023064 104037          EMT     C$SVEC
      (2) 023066 062706 000010  ADD     #10,SP
4209 023072          SETPRI  #PR100
      (3) 023072 012700 000000  MOV     #PR100,RO
      (3) 023076 104041          EMT     C$SPRI
4210 023100 005037 002134          CLR     INTFLG          ;CLEAR INTERRUPT FLAG
4211 023104 012777 000050 157124  MOV     #40, @RLDA      ;SET UP FOR HDR NT FND
4212 023112 012777 002652 157114  MOV     #BUF, @RLBA     ;BUS ADDRESS
4213 023120 012777 177777 157112  MOV     #-1, @RLMP      ;WORD COUNT
4214
4215 023126 004537 020254          JSR     R5, LDFUNC      ;LOAD THE FUNCTION IN NEXT WORD
4216 023132 000112          WRITE! INTEN
4217
4218 023134 013700 002274          MOV     OPIMX,RO
4219 023140 006300          ASL     RO
4220 023142 006300          ASL     RO
4221 023144 006300          ASL     RO
4222 023146 063700 002274          ADD     OPIMX,RO
4223 023152 063700 002274          ADD     OPIMX,RO
4224 023156          WAITUS  RO          ;WAIT MAX MILLISECONDS
      (3) 023156 104027          EMT     C$WTU
4225 023160 010037 002160          MOV     RO, BDDAT      ;SETUP FOR WORST CASE
4226 023164 005737 002134          TST     INTFLG        ;DID INTERRUPT OCCUR
4227 023170 001427          BEQ     4$            ;NO, REPORT ERROR
4228
4229 023172          GETTIM  BDDAT          ;GET TIME EXPIRED
      (3) 023172 104052          EMT     C$GTIM
      (3) 023174 010037 002160          MOV     RO, BDDAT
4230 023200 005000          CLR     RO          ;DIVIDE
4231 023202 162737 000012 002160 15:  SUB     #10, BDDAT     ;ANSWER
4232 023210 100402          BMI     3$            ;BY 10 TO GET
4233 023212 005200          INC     RO          ;RIGHT ANSWER
4234 023214 000772          BR      1$
4235 023216 010037 002160          3$:  MOV     RO, BDDAT
4236
4237          ;CHECK THAT OPI TIME IS WITHIN LIMITS
4238
4239 023222          2$:  SETPRI  #PR107
      (3) 023222 012700 000340  MOV     #PR107,RO
      (3) 023226 104041          EMT     C$SPRI
4240 023230 023737 002274 002160          CMP     OPIMX, BDDAT   ;IS IT WITHIN LIMITS
4241 023236 002404          BLT     4$            ;NO, REPORT ERROR
4242
4243 023240 023737 002272 002160          CMP     OPIMN, BDDAT   ;WITHIN LIMITS
4244 023246 003404          BLE     5$            ;YES
4245
4246          4$:  ERRDF  974.,EM56,ERR13 ;OPI TIMING INCORRECT
```

```
(3) 023250 104462          TRAP      TSERCODE
(5) 023252 001716          .WORD    974
(5) 023254 013256          .WORD    EM56
(5) 023256 014702          .WORD    ERR13
4247
4248 023260                55:      CLRVEC   BVEC          ;CLEAR PRESENT VECTOR
(3) 023260 013700 002244    MOV      BVEC,RO
(3) 023264 104036          EMT      C$CVEC
4249 023266                SETVEC   BVEC,#INTSRV,#340 ;SET IN OLD VECTOR
(7) 023266 012746 000340    MOV      #340,-(SP)
(6) 023272 012746 017706    MOV      #INTSRV,-(SP)
(5) 023276 013746 002244    MOV      BVEC,-(SP)
(4) 023302 012746 000003    MOV      #3,-(SP)
(3) 023306 104037          EMT      C$SVEC
(2) 023310 062706 000010    ADD      #10,SP
4250
4251 023314                10000$:  ENDSEG          ;%%END OF SEGMENT%%
(3) 023314
(3) 023314 104005          EMT      C$ESEG
4252
4253 023316                ENDTST          ;**END OF TEST**
(3) 023316                L10035:
(3) 023316 104001          EMT      C$ETST
4254
4255          .SBTTL  **TEST 9** - MULTIPLE SECTOR TRANSFER ON WRITE
4256
4257 023320                BGNTST          ;**START OF TEST**
4258
4259 023320                STARS
(2)          ; ;*****
4260          ;CHECK FOR MULTIPLE SECTOR TRANSFER ON WRITE. THIS TEST CHECKS
4261          ;THAT TWO SECTORS CAN BE SUCCESSFULLY WRITTEN. WE LOAD
4262          ;A WORD COUNT OF 129 WORDS (ONE SECTOR + 1 WORD) STARTING AT
4263          ;SECTOR 0 THRU SECTOR 37 AND VERIFY THAT THE RLDA DOES
4264          ;A DOUBLE INCREMENT EACH TIME.
4265 023320                STARS
(2)          ; ;*****
4266
4267
4268
4269 023320 004737 021154        JSR      PC,HDHOME ;HEADS OVER TRACK 0
4270 023324                CKERFG          ;HEADS GO HOME OKAY
(4) 023332 104032          EMT      C$EXIT
(4) 023334 000152          .WORD    L10036-.
4271
4272 023336 005037 002150        CLR      TMP0          ;CLEAR TEMP LOCATIONS
4273 023342 005037 002152        CLR      TMP1
4274
4275 023346                BGNSEG          ;%%START OF SEGMENT%%
(3) 023346 104004          EMT      C$BSEG
4276
4277
4278 023350 013737 002152 002156 15:  MOV      TMP1,GDDAT ;GET CYLINDER
4279 023356 053737 002150 002156    BIS      TMP0,GDDAT ;GET SECTOR
4280 023364 013777 002156 156644    MOV      GDDAT,@RLDA ;SET DISK ADDRESS-SECTOR 0
4281 023372 062737 000002 002156    ADD      #2,GDDAT ;SET EXPECTED + 2
```

```

4282 023400 012777 002652 156626      MOV      #BUF, @RLBA      ;SET BUS ADDRESS
4283 023406 012777 177577 156624      MOV      #-129., @RLMP   ;WORD COUNT-SECTOR+1 WORD
4284
4285 023414 004537 020254      JSR      R5, LDFUNC      ;LOAD THE FUNCTION IN NEXT WORD
4286 023420 000012      WRITE
4287 023422 004537 021074      JSR      R5, WTCRDY      ;WRITE
                               ;WAIT FOR CONTROLLER READY?
4288 023426      ESCAPE  SEG              ;CHECK FOR FL: LOE, ELSE EXIT SEG
(3) 023426 104010      EMT      C$ESCAPE
(3) 023430 000054      .WORD   10000$-.
4289
4290 023432 004537 020012      JSR      R5, CHERR      ;CHECK CNTLR FOR ERRORS
4291 023436      ESCAPE  SEG              ;CHECK FOR FL: LOE, ELSE EXIT SEG
(3) 023436 104010      EMT      C$ESCAPE
(3) 023440 000044      .WORD   10000$-.
4292
4293 023442 013737 002222 002160      MOV      E. DA, BDDAT    ;READ DISK ADDRESS
4294 023450 023737 002160 002156      CMP      BDDAT, GDDAT    ;IS DISK ADDRESS CORRECT
4295 023456 001404      BEQ      25              ;YES, BRANCH NO, REPORT ERROR
4296
4297 023460      ERRDF  7., EM22, ERR4   ;DISK ADDRESS NOT CORRECT
(3) 023460 104462      TRAP    T$ERCODE
(5) 023462 000007      .WORD   7
(5) 023464 011344      .WORD   EM22
(5) 023466 014210      .WORD   ERR4
4298
4299 023470      25:
4300
4301 023470 005237 002150      INC      TMO
4302 023474 022737 000046 002150      CMP      #46, TMO        ;NEXT SECTOR
4303 023502 001322      BNE      15              ;AT END?
4304
4305 023504      ENDSEG                  ;%%END OF SEGMENT%%
(3) 023504      10000$:
(3) 023504 104005      EMT      C$ESEG
4306 023506      ENDTST                  ;**END OF TEST**
(3) 023506      L10036:
(3) 023506 104001      EMT      C$ETST
4307
4308      .SBTTL **TEST 10** - CHECK DIRECTION OF WRITE NPR
4309
4310 023510      BGNTST                  ;**START OF TEST**
4311
4312 023510      STARS
(2) ;*****
4313 ;VERIFY THAT A WRITE IS WRITING NOT READING. WE WRITE A
4314 ;KNOWN PATTERN IN "BUF" (128 WORD), WE THEN ISSUE A WRITE.
4315 ;ONCE THE WRITE IS FINISHED WE CHECK THAT "BUF" IS INTACT.
4316 ;THIS IS DONE TO PROVE THAT THE NPR IS GOING THE RIGHT
4317 ;WAY.
4318 023510      STARS
(2) ;*****
4319
4320
4321 023510 004737 021154      JSR      PC, HDHOME     ;HEADS OVER TRACK 0
4322 023514      CKERFG
(4) 023522 104032      EMT      C$EXIT         ;HEADS GO HOME OKAY

```

K 5

OUTERR MACY11 30(1046) 03-NOV-77 10:02 PAGE 83-39  
 DZRLER P11 03-NOV-77 09:54 \*\*TEST 10\*\* - CHECK DIRECTION OF WRITE NPR SEQ 0062

(4)	023524	000160			. WORD	L10037-.	
4323							
4324	023526				BGNSEG	;%%START OF SEGMENT%%	
(3)	023526	104004			EMT	C\$BSEG	
4325							
4326	023530			2\$:			
4327	023530	012702	002652		MOV	#BUF, R2 ;WRITE BUFFER FOR WRITE OPERATION	
4328	023534	012701	000200		MOV	#128., R1 ;ONE SECTOR'S WORTH	
4329	023540	012722	125252	3\$:	MOV	#125252, (R2)+ ;WRITE BUFFER	
4330	023544	005301			DEC	R1 ;DONE?	
4331	023546	001374			BNE	3\$ ;NO, GO BACK	
4332							
4333	023550	005077	156462		CLR	@RLDA ;LOAD DISK ADDRESS	
4334	023554	012777	177600	156456	MOV	#-128., @RLMP ;WORD COUNT	
4335	023562	012777	002652	156444	MOV	#BUF, @RLBA ;BUS ADDRESS	
4336	023570	004537	020254		JSR	R5, LDFUNC ;LOAD THE FUNCTION IN NEXT WORD	
4337	023574	000012			WRITE	;WRITE SOME DATA	
4338	023576	004537	021074		JSR	R5, WTCRDY ;WAIT FOR IT TO FINISH	
4339	023602				ESCAPE	SEG ;CHECK FOR FL: LOE, ELSE EXIT SEG	
(3)	023602	104010			EMT	C\$ESCAPE	
(3)	023604	000076			. WORD	10000\$-	
4340							
4341	023606	004537	020012		JSR	R5, CHERR ;CHECK CNTLR FOR ERRORS	
4342	023612				ESCAPE	SEG ;CHECK FOR FL: LOE, ELSE EXIT SEG	
(3)	023612	104010			EMT	C\$ESCAPE	
(3)	023614	000066			. WORD	10000\$-	
4343							
4344	023616	012702	002652		MOV	#BUF, R2 ;SET UP TO CHECK BUFFER	
4345	023622	012701	000200		MOV	#128., R1 ;CHECK 128 WORDS	
4346							
4347	023626				BGNSEG	;%%START OF SEGMENT%%	
(3)	023626	104004			EMT	C\$BSEG	
4348							
4349	023630	012737	125252	002156	MOV	#125252, GDDAT ;DATA SHOULD BE 125252	
4350	023636	011237	002160		MOV	(R2), BDDAT ;LOAD DATA INTO BDDAT	
4351	023642	023737	002156	002160	4\$:	CMP	GDDAT, BDDAT ;IS IT OKAY?
4352	023650	001406			BEQ	5\$ ;YES, CONTINUE	
4353							
4354	023652	010237	002152		MOV	R2, TMP1 ;LOAD MEMORY LOCATION OF FAILURE	
4355	023656				ERRDF	8., EM26, ERR8	
(3)	023656	104462			TRAP	T\$ERCODE	
(5)	023660	000010			. WORD	8	
(5)	023662	011645			. WORD	EM26	
(5)	023664	014364			. WORD	ERR8	
4356							
4357	023666			5\$:	ESCAPE	SEG ;CHECK FOR FL: LOE, ELSE EXIT SEG	
(3)	023666	104010			EMT	C\$ESCAPE	
(3)	023670	000010			. WORD	10001\$-	
4358	023672	005722		6\$:	TST	(R2)+ ;NEXT!	
4359	023674	005301			DEC	R1 ;DONE?	
4360	023676	001357			BNE	4\$ ;NO, GO BACK	
4361							
4362	023700				ENDSEG	;%%END OF SEGMENT%%	
(3)	023700			10001\$:	EMT	C\$ESEG	
(3)	023700	104005			ENDSEG	;%%END OF SEGMENT%%	
4363	023702						

```

(3) 023702 10000$:
(3) 023702 104005 EMT C$ESEG
4364 023704 ENDTST ;**END OF TEST**
(3) 023704 L10037:
(3) 023704 104001 EMT C$ETST
4365
4366 .SBTTL **TEST 11** - CHECK FULL RLBA INCREMENT
4367
4368 023706 BGNTST ;**START OF TEST**
4369
4370 023706 STARS
(2) ;*****
4371 ;TEST THAT THE RLBA WILL INCREMENT, WE DO NOT DO A FULL 16
4372 ;BIT INCREMENT WE CHECK THAT EACH BIT WILL TOGGLE 0 TO 1
4373 ;AND 1 TO 0. WE DO CHECK ALL BITS EVEN IF ALL MEMORY
4374 ;IS NOT AVAILABLE. (WE IGNORE NON-EXISTANT MEMORY ERRORS).
4375 ;WE USE THE SAME DISK ADDRESS (RANDOM) AND A 1 WORD TRANSFER.
4376 023706 STARS
(2) ;*****
4377
4378
4379 023706 004737 021154 JSR PC,HDHOME ;HEADS OVER TRACK 0
4380 023712 CKERFG ;HEADS GO HOME OKAY
(4) 023720 104032 EMT C$EXIT
(4) 023722 000134 .WORD L10040-.
4381
4382
4383 023724 005037 002152 CLR TMP1 ;CLEAR LOCATION
4384
4385 023730 BGNSEG ;%%START OF SEGMENT%%
(3) 023730 104004 EMT C$BSEG
4386
4387 023732 3$: MOV #-1, @RLMP ;ONLY ONE (1) WORD
4388 023732 012777 177777 156300 CLR @RLDA ;LOAD DISK ADDRESS
4389 023740 005077 156272 MOV TMP1, @RLBA ;BUS ADDRESS
4390 023744 013777 002152 156262
4391
4392 023752 004537 020254 JSR R5, LDFUNC ;LOAD THE FUNCTION IN NEXT WORD
4393 023756 000012 WRITE
4394 023760 004537 021074 JSR R5, WTCRDY ;WAIT FOR WRITE TO FINISH
4395 023764 ESCAPE SEG ;CHECK FOR FL: LOE, ELSE EXIT SEG
(3) 023764 104010 EMT C$ESCAPE
(3) 023766 000066 .WORD 10000$-.
4396
4397 023770 013737 002152 002156 4$: MOV TMP1, GDDAT ;SET UP EXPECTED RLBA
4398 023776 062737 000002 002156 ADD #2, GDDAT ;PREVIOUS RLBA+2
4399 024004 013737 002220 002160 MOV E. BA, BDDAT ;READ RLBA
4400 024012 023737 002156 002160 CMP GDDAT, BDDAT ;WAS IT UPDATED PROPERLY?
4401 024020 001404 BEQ 5$ ;YES, CONTINUE
4402
4403 024022 ERRDF 9. , EM30, ERR4 ;BA INCREMENT ERROR
(3) 024022 104462 TRAP T$ERCODE
(5) 024024 000011 .WORD 9
(5) 024026 011765 .WORD EM30
(5) 024030 014210 .WORD ERR4
4404 024032 5$: ESCAPE SEG ;CHECK FOR FL: LOE, ELSE EXIT SEG

```



```
(3) 024032 104010 EMT C$ESCAPE
(3) 024034 000020 .WORD 10000$-.
4405
4406 024036 006337 002152 ASL TMP1 ;NEXT PATTERN TO TEST RLBA
4407 024042 103404 BCS 6$ ;DONE?
4408 024044 052737 000002 002152 BIS #BIT1,TMP1 ;NO, SET IN BIT 1
4409 024052 000727 BR 3$ ;GO CHECK NEXT.
4410
4411 024054 6$: ;END TEST
4412
4413 024054 ENDSEG ;%%END OF SEGMENT%%
(3) 024054 10000$:
(3) 024054 104005 EMT C$ESEG
4414 024056 ENDTST ;**END OF TEST**
(3) 024056 L10040:
(3) 024056 104001 EMT C$ETST
4415
4416 .SBTTL **TEST 12** - BA BIT 16 INCREMENT
4417
4418 024060 BGNTST ;**START OF TEST**
4419
4420 024060 STARS
(2) ;*****
4421 ;CHECK THAT BA BIT 16 WILL INCREMENT. WE WILL LOAD THE
4422 ;RLBA WITH 177776 AND ISSUE A ONE WORD WRITE WE THEN
4423 ;CHECK BA BIT 16 TO SET, BA 17 TO STAY A 0 AND THE RLBA
4424 ;TO GO TO ZERO
4425 024060 STARS
(2) ;*****
4426
4427
4428 024060 004737 021154 JSR PC,HDHOME ;HEADS OVER TRACK 0
4429 024064 CKERFG ;HEADS GO HOME OKAY
(4) 024072 104032 EMT C$EXIT
(4) 024074 000160 .WORD L10041-.
4430
4431 024076 BGNSEG ;%%START OF SEGMENT%%
(3) 024076 104004 EMT C$BSEG
4432
4433 024100 2$:
4434 024100 012777 177776 156126 MOV #177776,@RLBA ;SET MAX BA TO INC. BA16
4435 024106 005037 002252 CLR XMEM ;WE DON'T WANT TO LOAD ANY EA
4436 024112 012777 177777 156120 MOV #-1,@RLMP ;ONE WORD TRANSFER
4437 024120 005077 156112 CLR @RLDA
4438 024124 004537 020254 JSR R5,LDFUNC ;LOAD THE FUNCTION IN NEXT WORD
4439 024130 000012 WRITE
4440 024132 004537 021074 JSR R5,WTCRDY ;WAIT FOR WRITE TO FINISH
4441 024136 ESCAPE SEG ;CHECK FOR FL: LOE, ELSE EXIT SEG
(3) 024136 104010 EMT C$ESCAPE
(3) 024140 000112 .WORD 10000$-.
4442 024142 032737 020000 002216 BIT #NXM,E.CS ;NON-EXISTANT MEMORY ERROR?
4443 024150 001002 BNE 3$ ;YES, CONTINUE
4444
4445 024152 004537 020012 JSR R5,CHERR ;CHECK CNTLR FOR ERRORS
4446 024156 3$: ESCAPE SEG ;CHECK FOR FL: LOE, ELSE EXIT SEG
(3) 024156 104010 EMT C$ESCAPE
```

```

(3) 024160 000072 .WORD 10000$.
4447
4448 024162 032737 000020 002216 BIT #BA16,E.CS ;DID BA16 SET?
4449 024170 001004 BNE 4$ ;YES, CONTINUE
4450
4451 024172 ERRDF 10.,EM31,ERRO ;BA 16 DID NOT INCREMENT
(3) 024172 104462 TRAP TSERCODE
(5) 024174 000012 .WORD 10
(5) 024176 012025 .WORD EM31
(5) 024200 014044 .WORD ERRO
4452
4453 024202 4$: CKLOOP
(3) 024202 104006 EMT C$CLP1
4454
4455 024204 032737 000040 002216 BIT #BA17,E.CS ;DID BA17 SET ALSO?
4456 024212 001404 BEQ 5$ ;NO, GOOD CONTINUE
4457
4458 024214 ERRDF 11.,EM32,ERRO ;BA 17 GOT CARRIED AWAY
(3) 024214 104462 TRAP TSERCODE
(5) 024216 000013 .WORD 11
(5) 024220 012070 .WORD EM32
(5) 024222 014044 .WORD ERRO
4459
4460 024224 5$: CKLOOP
(3) 024224 104006 EMT C$CLP1
4461
4462 024226 005037 002156 CLR GDDAT ;CHECK THAT BA15-BA0 IS CLEAR
4463 024232 013737 002220 002160 MOV E.BA,BDDAT ;READ BA
4464 024240 001404 BEQ 6$ ;IS BA ZERO?
4465 024242 ERRDF 12.,EM33,ERR4 ;BA SHOULD BE ZERO
(3) 024242 104462 TRAP TSERCODE
(5) 024244 000014 .WORD 12
(5) 024246 012135 .WORD EM33
(5) 024250 014210 .WORD ERR4
4466
4467 024252 6$:
4468
4469 024252 ENDSEG ;%%END OF SEGMENT%%
(3) 024252 10000$:
(3) 024252 104005 EMT C$ESEG
4470 024254 ENDTST ;**END OF TEST**
(3) 024254 L10041:
(3) 024254 104001 EMT C$ETST
4471
4472 .SBTTL **TEST 13** - BA BIT 17 INCREMENT
4473
4474 024256 BGNTST ;**START OF TEST**
4475
4476 024256 STARS
(2) ;*****
4477 ;CHECK THAT BA BIT 17 WILL INCREMENT. WE WILL LOAD THE
4478 ;RLBA WITH 177776 AND BA 16 SET, WE WILL ISSUE A ONE WORD
4479 ;WRITE. WE THEN CHECK BA17 TO SET, BA16 TO CLEAR AND
4480 ;BA15 - BA0 TO CLEAR.
4481 024256 STARS
(2) ;*****

```

4482									
4483									
4484									
4485	024256	004737	021154			JSR	PC, HDHOME		; HEADS OVER TRACK 0
4486	024262					CKERFG			; HEADS GO HOME OKAY
(4)	024270	104032				EMT	CSEXIT		
(4)	024272	000162				.WORD	L10042-		
4487									
4488	024274					BGNSEG			; %%START OF SEGMENT%%
(3)	024274	104004				EMT	C\$BSEG		
4489									
4490	024276				25:				
4491	024276	012777	177776	155730		MOV	#177776, @RLBA		; SET MAX BA TO INC. BA16
4492	024304	012737	000020	002252		MOV	#BA16, XMEM		; SET BA16 IN RLCS
4493	024312	012777	177777	155720		MOV	#-1, @RLMP		; ONE WORD TRANSFER
4494	024320	005077	155712			CLR	@RLDA		
4495	024324	004537	020254			JSR	R5, LDFUNC		; LOAD THE FUNCTION IN NEXT WORD
4496	024330	000012				WRITE			
4497	024332	004537	021074			JSR	R5, WTCRDY		; WAIT FOR WRITE TO FINISH
4498	024336					ESCAPE	SEG		; CHECK FOR FL: LOE, ELSE EXIT SEG
(3)	024336	104010				EMT	C\$ESCAPE		
(3)	024340	000112				.WORD	10000\$-		
4499	024342	032737	020000	002216		BIT	#NXM, E. CS		; NON-EXISTANT MEMORY ERROR?
4500	024350	001002				BNE	3\$		; YES, CONTINUE
4501									
4502	024352	004537	020012			JSR	R5, CHERR		; CHECK CNTLR FOR ERRORS
4503	024356				35:	ESCAPE	SEG		; CHECK FOR FL: LOE, ELSE EXIT SEG
(3)	024356	104010				EMT	C\$ESCAPE		
(3)	024360	000072				.WORD	10000\$-		
4504									
4505	024362	032737	000040	002216		BIT	#BA17, E. CS		; DID BA17SET?
4506	024370	001004				BNE	4\$		; YES, CONTINUE
4507									
4508	024372					ERRDF	13, EM34, ERRO		; BA 17 DID NOT SET
(3)	024372	104462				TRAP	T\$ERCODE		
(5)	024374	000015				.WORD	13		
(5)	024376	012176				.WORD	EM34		
(5)	024400	014044				.WORD	ERRO		
4509									
4510	024402				45:	CKLOOP			
(3)	024402	104006				EMT	C\$CLP1		
4511									
4512	024404	032737	000020	002216		BIT	#BA16, E. CS		; DID BA16 SET ALSO?
4513	024412	001404				BEQ	5\$		; NO, GOOD CONTINUE
4514									
4515	024414					ERRDF	14, EM35, ERRO		; BA 16 DIDN'T KNOW WHEN TO QUIT.
(3)	024414	104462				TRAP	T\$ERCODE		
(5)	024416	000016				.WORD	14		
(5)	024420	012241				.WORD	EM35		
(5)	024422	014044				.WORD	ERRO		
4516	024424				55:	CKLOOP			
(3)	024424	104006				EMT	C\$CLP1		
4517									
4518	024426	005037	002156			CLR	GDDAT		; CHECK THAT BA15-BA0 IS CLEAR
4519	024432	013737	002220	002160		MOV	E. BA, BDDAT		; READ BA
4520	024440	001404				BEQ	6\$		; IS BA ZERO?

```

4521 024442          ERRDF 15, EM36, ERR4 ; BA SHOULD BE ZERO
      (3) 024442 104462 TRAP  T$ERCODE
      (5) 024444 000017 .WORD 15
      (5) 024446 012306 .WORD EM36
      (5) 024450 014210 .WORD ERR4
4522
4523 024452          65: ;
4524
4525 024452          ENDSEG ; %%END OF SEGMENT%%
      (3) 024452          10000$:
      (3) 024452 104005 EMT  C$ESEG
4526 024454          ENDTST ; **END OF TEST**
      (3) 024454          L10042:
      (3) 024454 104001 EMT  C$ETST
4527
4528 .SBTTL **TEST 14** - TEST READ NPR INTEGRITY
4529
4530 024456          BGNTST ; **START OF TEST**
4531
4532
4533
4534
4535 024456          STARS
      (2) ; *****
4536 ; CHECK THAT NPR WILL NOT INTERFERE WITH THE OPERATION OF THE UNIBUS
4537 ; WE SETUP LOCATION 4 TO HANDLE THE TRAP IF IT HAPPENS
4538 024456          STARS
      (2) ; *****
4539
4540
4541 024456 004737 021154 JSR  PC, HDHOME ; HEADS OVER TRACK 0
4542 024462          CKERFG ; HEADS GO HOME OKAY
      (4) 024470 104032 EMT  C$EXIT
      (4) 024472 000132 .WORD L10043-
4543
4544 024474          BGNSEG ; %%START OF SEGMENT%%
      (3) 024474 104004 EMT  C$BSEG
4545
4546
4547 024476          15: SETVEC ERRVEC, #TRPHAN, #340 ; SET UP VECTOR
      (7) 024476 012746 000340 MOV  #340, -(SP)
      (6) 024502 012746 021146 MOV  #TRPHAN, -(SP)
      (5) 024506 013746 002122 MOV  ERRVEC, -(SP)
      (4) 024512 012746 000003 MOV  #3, -(SP)
      (3) 024516 104037          EMT  C$SVEC
      (2) 024520 062706 000010 ADD  #10, SP
4548 024524 005037 002132 CLR  TRPFLG ; CLEAR TRAP PLAY
4549 024530 012777 002652 155476 MOV  #BUF, @RLBA ; LOAD BA
4550 024536 005077 155474 CLR  @RLDA ; LOAD DA
4551 024542 012777 177777 155470 MOV  #-1, @RLMP ; LOAD WC
4552 024550 004537 020254 JSR  R5, LDFUNC ; LOAD THE FUNCTION IN NEXT WORD
4553 024554 000014          READ
4554 024556 004537 021074 JSR  R5, WTCRDY ;
4555 024562          CLRVEC ERRVEC ; CLEAR OUT VECTOR
      (3) 024562 013700 002122 MOV  ERRVEC, R0
      (3) 024566 104036          EMT  C$CVEC
    
```

```

4556 024570          ESCAPE SEG          ;CHECK FOR FL: LOE, ELSE EXIT SEG
      (3) 024570 104010 EMT C$ESCAPE
      (3) 024572 000030 .WORD 10000$-
4557 024574 004537 020012 JSR R5, CHERR          ;CHECK CNTLR FOR ERRORS
4558 024600          ESCAPE SEG          ;CHECK FOR FL: LOE, ELSE EXIT SEG
      (3) 024600 104010 EMT C$ESCAPE
      (3) 024602 000020 .WORD 10000$-
4559
4560 024604 005737 002132 TST TRPFLG          ;DID TRAP OCCUR?
4561 024610 001404 BEQ 7$             ;NO, OKAY
4562 024612          ERRDF 17, EM52, ERRO ;YES, PRINT ERROR
      (3) 024612 104462 TRAP T$ERCODE
      (5) 024614 000021 .WORD 17
      (5) 024616 012777 .WORD EM52
      (5) 024620 014044 .WORD ERRO
4563 024622          7$:
4564
4565
4566 024622          ENDSEG          ;%%END OF SEGMENT%%
      (3) 024622 10000$: EMT C$ESEG
      (3) 024622 104005
4567
4568
4569 024624          ENDTST          ;**END OF TEST**
      (3) 024624 L10043: EMT C$ETST
      (3) 024624 104001
4570
4571          .SBTTL **TEST 15** - READ FUNCTION
4572
4573 024626          BGNTST          ;**START OF TEST**
4574
4575 024626          STARS
      (2)          ; ;*****
4576          ;CHECK OF THE READ FUNCTION. WE WILL FIRST DO A READ
4577          ;HEADER TO FIND OUT WHERE WE ARE AND THEN ISSUE
4578          ;A FULL SECTOR READ, WAIT FOR READY AND CHECK FOR
4579          ;ANY ERRORS
4580 024626          STARS
      (2)          ; ;*****
4581
4582
4583 024626 004737 021154 JSR PC, HDHOME      ;HEADS OVER TRACK 0
4584 024632          CKERFG          ;HEADS GO HOME OKAY
      (4) 024640 104032 EMT C$EXIT
      (4) 024642 000064 .WORD L10044-
4585
4586 024644          BGNSEG          ;%%START OF SEGMENT%%
      (3) 024644 104004 EMT C$BSEG
4587
4588 024646 012737 001750 002150 MOV #1000, T$MPO
4589 024654 005077 155356 15: CLR @RLDA          ;LOAD DISK ADDRESS
4590 024660 012777 177600 155352 MOV #-128, @RLMP   ;SET WORD LENGTH
4591 024666 012777 002652 155340 MOV #BUF, @RLBA    ;SET BUS ADDRESS
4592
4593 024674 004537 020254 JSR R5, LDFUNC      ;LOAD THE FUNCTION IN NEXT WORD
4594 024700 000014 READ          ;READ
  
```

```

                                E 6
OUTERR MACY11 30(1046) 03-NOV-77 10:02 PAGE 83-46
DZRLBA.P11 03-NOV-77 09:54 **TEST 15** - READ FUNCTION                               SEQ 0069

4595 024702 004537 021074      JSR    R5,WTCRDY      ;WAIT FOR CONTROLLER READY
4596 024706                      ESCAPE  SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
   (3) 024706 104010          EMT    C$ESCAPE
   (3) 024710 000014          .WORD  10000$-.

4597
4598 024712 004537 020012      JSR    R5,CHERR      ;CHECK CNTLR FOR ERRORS
4599
4600 024716 005337 002150      DEC    TMPO
4601 024722 001354          BNE    1$
4602 024724                      ENDSEG          ;%%END OF SEGMENT%%
   (3) 024724 10000$:
   (3) 024724 104005          EMT    C$ESEG
4603 024726                      ENDTST          ;**END OF TEST**
   (3) 024726 L10044:
   (3) 024726 104001          EMT    C$ETST

4604
4605      .SBTTL **TEST 16** - READ FUNCTION INTERRUPT
4606
4607 024730      BGNTST          ;**START OF TEST**
4608
4609 024730      STARS
   (2) ;*****
4610 ;CHECK OF THE READ FUNCTION UNDER INTERRUPT CONTROL, WE WILL
4611 ;ISSUE A READ HEADER TO GET POSITION AND THEN READ
4612 ;A FULL SECTOR WAITING FOR THE INTERRUPT. CHECK FOR
4613 ;ERRORS ON INTERRUPT.
4614 024730      STARS
   (2) ;*****

4615
4616
4617 024730 004737 021154      JSR    PC,HDHOME    ;HEADS OVER TRACK 0
4618 024734                      CKERFG          ;HEADS GO HOME OKAY
   (4) 024742 104032          EMT    C$EXIT
   (4) 024744 000106          .WORD  L10045-.

4619
4620 024746                      BGNSEG          ;%%START OF SEGMENT%%
   (3) 024746 104004          EMT    C$BSEG

4621
4622 024750 005037 002134      CLR    INTFLG      ;CLEAR INTERRUPT INDICATOR
4623 024754 005077 155256      CLR    @RLDA      ;SET DISK ADDRESS
4624 024760 012777 177600 155252  MOV    #-128.,@RLMP ;SET UP WORD COUNT
4625 024766 012777 002652 155240  MOV    #BUF,@RLBA  ;SET UP BUS ADDRESS

4626
4627 024774                      SETPRI #PR100      ;PRIORITY TO 0
   (3) 024774 012700 000000      MOV    #PR100,RO
   (3) 025000 104041          EMT    C$SPRI

4628 025002 004537 020254      JSR    R5,LDFUNC   ;LOAD THE FUNCTION IN NEXT WORD
4629 025006 000114          READ!INTEN      ;READ UNDER INTERRUPT
4630 025010 004537 021074      JSR    R5,WTCRDY  ;WAIT FOR INTERRUPT
4631 025014                      CKLOOP
   (3) 025014 104006          EMT    C$CLP1
4632 025016                      SETPRI #PR107      ;PRIORITY TO 7
   (3) 025016 012700 000340      MOV    #PR107,RO
   (3) 025022 104041          EMT    C$SPRI

4633
4634 025024 005737 002134      TST    INTFLG     ;DID INTERRUPT OCCUR?

```

```
4635 025030 001004          BNE      15          ;YES-BRANCH NO-REPORT
4636
4637 025032          ERRDF   19.,EM4,ERRO ;READ DID NOT INTERRUPT
(3) 025032 104462          TRAP   TSERCODE
(5) 025034 000023          .WORD  19
(5) 025036 010407          .WORD  EM4
(5) 025040 014044          .WORD  ERRO
4638 025042          15:      CKLOOP          ;CHECK FOR LOOP
(3) 025042 104006          EMT    C$CLP1
4639
4640 025044 004537 020012          JSR    R5,CHERR          ;CHECK CNTLR FOR ERRORS
4641
4642 025050          ENDSEG          ;%%END OF SEGMENT%%
(3) 025050          10000$:
(3) 025050 104005          EMT    C$ESEG
4643 025052          ENDTST          ;**END OF TEST**
(3) 025052          L10045:
(3) 025052 104001          EMT    C$ETST
4644
4645          .SBTTL  **TEST 17** - CHECK READ NPR DIRECTION
4646
4647 025054          BGNTST          ;**START OF TEST**
4648
4649 025054          STARS
(2)          ;*****
4650          ;CHECK THAT THE READ FUNCTION ACTUALLY READS (INTO MEMORY)
4651          ;WE WILL WRITE A PATTERN INTO MEMORY AND THEN ISSUE
4652          ;A READ TO OVERLAY THAT PATTERN. AFTER THE READ
4653          ;WE CHECK TO SEE IF THE WRITTEN PATTERN HAS CHANGED.
4654          ;IF NOT WE ISSUE IT AGAIN AT THE SAME SECTION AFTER
4655          ;HAVING MODIFIED OUR PATTERN IN MEMORY (SINCE THERE IS
4656          ;ONE CHANCE THAT THE DISK COULD HAVE OUR PATTERN). AFTER
4657          ;THE SECOND READ WE CHECK THE BUFFER AGAIN. IF IT'S
4658          ;NO CHANGED WE REPORT AN ERROR
4659 025054          STARS
(2)          ;*****
4660
4661
4662 025054 004737 021154          JSR    PC,HDHOME        ;HEADS OVER TRACK 0
4663 025060          CKERFG          ;HEADS GO HOME OKAY
(4) 025066 104032          EMT    C$EXIT
(4) 025070 000156          .WORD  L10046-.
4664
4665 025072          BGNSEG          ;%%START OF SEGMENT%%
(3) 025072 104004          EMT    C$BSEG
4666
4667 025074 012737 123456 002150          MOV    #123456,TMPO     ;SET PATTERN TO WRITE
4668 025102 005037 002152          CLR    TMP1            ;CLEAR PASS INDICATOR
4669 025106 012700 002652          15:    MOV    #BUF,RO    ;SET UP BUFFER BEGINNING
4670 025112 012701 000200          MOV    #128.,R1
4671 025116 013720 002150          25:    MOV    TMPO,(RO)+ ;WRITE BUFFER
4672 025122 005301          DEC    R1              ;DONE??
4673 025124 001374          BNE    25              ;NO, GO BACK
4674 025126 005077 155104          CLR    @RLDA           ;LOAD DISK ADDRESS
4675 025132 012777 177600 155100          MOV    #-128.,@RLMP    ;SET WORD COUNT
4676 025140 012777 002652 155066          MOV    #BUF,@RLBA     ;LOAD BUS ADDRESS
```

```

4677 025146 012737 002652 002156      MOV      #BUF,GDDAT      ;FOR ERROR PRINT
4678
4679 025154 004537 020254              JSR      R5,LDFUNC      ;LOAD THE FUNCTION IN NEXT WORD
4680 025160 000014                      READ                     ;READ
4681 025162 004537 021074              JSR      R5,WTCRDY     ;WAIT FOR CONTROLLER READY
4682 025166                      ESCAPE SEG              ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 025166 104010                      EMT      C$ESCAPE
(3) 025170 000054                      .WORD   100005-
4683
4684 025172 004537 020012              JSR      R5,CHERR      ;CHECK CNTLR FOR ERRORS
4685 025176                      ESCAPE SEG              ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 025176 104010                      EMT      C$ESCAPE
(3) 025200 000044                      .WORD   100005-
4686
4687 025202 012702 002652      MOV      #BUF,R2      ;SET TO START COMPARING DATA
4688 025206 022237 002150      45:    CMP      (R2)+,TMPO ;DID DATA CHANGE?
4689 025212 001014                      BNE      65            ;YES, CHECK FOR END
4690
4691
4692
4693 025214 005737 002152      TST      TMP1          ;DATA DIDN'T CHANGE, CHECK
4694 025220 001005                      BNE      55            ;IF 1ST OR 2ND TIME?
4695
4696 025222 005237 002152      INC      TMP1          ;INC PASS COUNT
4697 025226 005137 002150      COM      TMPO          ;COMPLIMENT PATTERN
4698 025232 000725                      BR       15            ;GO DO IT AGAIN
4699
4700 025234      55:    ERRDF  20.,EM5,ERR9 ;READ DID NOT MODIFY MEMORY
(3) 025234 104462                      TRAP    T$ERCODE
(5) 025236 000024                      .WORD   20
(5) 025240 010446                      .WORD   EM5
(5) 025242 014436                      .WORD   ERR9
4701
4702 025244      65:
4703
4704 025244                      ENDSEG                  ;%%END OF SEGMENT%%
(3) 025244      100005:
(3) 025244 104005                      EMT      C$ESEG
4705 025246                      ENDTST                  ;**END OF TEST**
(3) 025246      L10046:
(3) 025246 104001                      EMT      C$ETST
4706
4707      .SBTTL  **TEST 18** - PROPER INCREMENT OF RLBA ON READ
4708
4709 025250      BGNTST                  ;**START OF TEST**
4710
4711 025250      STARS
(2) ;*****
4712 ;CHECK THAT THE RLBA WILL INCREMENT WITH THE READ
4713 ;THE RLBA SHOULD CONTAIN "BUF +256." AFTER A FULL SECTOR
4714 ;READ.
4715 025250      STARS
(2) ;*****
4716
4717
4718 025250 004737 021154      JSR      PC,HDHOME     ;HEADS OVER TRACK 0

```



```
4719 025254          CKERFG          ;HEADS GO HOME OKAY
      (4) 025262 104032 EMT          CSEXIT
      (4) 025264 000116 . WORD      L10047-
4720
4721 025266          BGNSEG          ;%%START OF SEGMENT%%
      (3) 025266 104004 EMT          C$BSEG
4722
4723 025270 005077 154742 CLR          @RLDA          ;SET UP DISK ADDRESS
4724 025274 012777 002652 154732 MOV          #BUF,@RLBA      ;SET UP BUS ADDRESS
4725 025302 012777 177600 154730 MOV          #-128,@RLMP     ;WORD COUNT
4726 025310 012737 002652 002156 MOV          #BUF,GDDAT     ;FORM EXPECTED BUS ADDRESS
4727 025316 062737 000400 002156 ADD          #256,GDDAT     ;AFTER READ
4728
4729 025324 004537 020254 JSR          R5,LDFUNC      ;LOAD THE FUNCTION IN NEXT WORD
4730 025330 000014 READ          ;READ
4731 025332 004537 021074 JSR          R5,WTCRDY     ;WAIT FOR CONTROLLER READY
4732 025336          ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
      (3) 025336 104010 EMT          C$ESCAPE
      (3) 025340 000040 . WORD      10000$-
4733
4734 025342 004537 020012 JSR          R5,CHERR      ;CHECK CNTLR FOR ERRORS
4735 025346          ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
      (3) 025346 104010 EMT          C$ESCAPE
      (3) 025350 000030 . WORD      10000$-
4736 025352 013737 002220 002160 MOV          E.BA,BDDAT     ;READ 'RLBA' FOR PRESENT ADDRESS
4737 025360 023737 002160 002156 CMP          BDDAT,GDDAT    ;DID 'BA' INCREMENT PROPERLY?
4738 025366 001404 BEQ          1$          ;YES, CONTINUE
4739
4740 025370          ERRDF 21,EM6,ERR4 ;BA DID NOT INCREMENT PROPERLY
      (3) 025370 104462 TRAP         TSERCODE
      (5) 025372 000025 . WORD      21
      (5) 025374 010512 . WORD      EM6
      (5) 025376 014210 . WORD      ERR4
4741
4742 025400          1$:
4743
4744 025400          ENDSEG          ;%%END OF SEGMENT%%
      (3) 025400          10000$:
      (3) 025400 104005 EMT          C$ESEG
4745 025402          ENDTST          ;**END OF TEST**
      (3) 025402          L10047:
      (3) 025402 104001 EMT          C$ETST
4746
4747          .SBTTL **TEST 19** - PROPER INCREMENT OF RLDA ON READ
4748
4749 025404          BGNTST          ;**START OF TEST**
4750
4751 025404          STARS
      (2)          ;*****
4752          ;CHECK THAT THE RLDA INCREMENTS BY ONE AFTER A
4753          ;FULL SECTOR READ, WE FIRST READ A HEADER TO FIND
4754          ;OUT WHERE WE ARE, THEN ISSUE A READ AFTER
4755          ;THE READ THE RLDA SHOULD BE RLDA (START) + 1
4756 025404          STARS
      (2)          ;*****
4757
```

```
4758 025404 004737 021154 JSR PC,HDHOME ;HEADS OVER TRACK 0
4759 025410 CKERFG ;HEADS GO HOME OKAY
(4) 025416 104032 EMT C$EXIT
(4) 025420 000114 .WORD L10050-.
4760
4761 025422 BGNSEG ;%%START OF SEGMENT%%
(3) 025422 104004 EMT C$BSEG
4762
4763
4764 025424 005037 002156 CLR GDDAT
4765 025430 013777 002156 154600 MOV GDDAT,@RLDA ;SETUP DISK ADDRESS
4766 025436 005237 002156 INC GDDAT ;CREATE EXPECTED SECTOR
4767 025442 012777 177600 154570 MOV #-128.,@RLMP ;WORD COUNT
4768 025450 012777 002652 154556 MOV #BUF,@RLBA ;SETUP BUS ADDRESS
4769
4770 025456 004537 020254 JSR R5,LDFUNC ;LOAD THE FUNCTION IN NEXT WORD
4771 025462 000014 READ ;READ
4772 025464 004537 021074 JSR R5,WTCRDY ;WAIT FOR CONTROLLER READY
4773 025470 ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 025470 104010 EMT C$ESCAPE
(3) 025472 000040 .WORD 10000$-.
4774
4775 025474 004537 020012 JSR R5,CHERR ;CHECK CNTLR FOR ERRORS
4776 025500 ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 025500 104010 EMT C$ESCAPE
(3) 025502 000030 .WORD 10000$-.
4777
4778 025504 013737 002222 002160 MOV E.DA,BDDAT ;READ DISK ADDRESS
4779 025512 023737 002156 002160 CMP GDDAT,BDDAT ;DID SECTOR INCREMENT PROPERLY
4780 025520 001404 BEQ 15 ;YES, BRANCH NO, REPORT ERROR
4781
4782 025522 ERRDF 22.,EM7,ERP4 ;DISK ADDRESS DID NOT INCREMENT
(3) 025522 104462 TRAP T$ERCODE
(5) 025524 000026 .WORD 22
(5) 025526 010566 .WORD EM7
(5) 025530 014210 .WORD ERR4
4783
4784 025532 15:
4785
4786 025532 ENDSEG ;%%END OF SEGMENT%%
(3) 025532 10000$: EMT C$ESEG
(3) 025532 104005 ENDTST ;**END OF TEST**
(3) 025534 L10050: EMT C$ETST
(3) 025534 104001
4788
4789 .SBTTL **TEST 20** - FORCE HEADER NOT FOUND WITH READ
4790
4791 025536 BGNTST ;**START OF TEST**
4792
4793 025536 STARS
(2) ;*****
4794 ;FORCE HEADER NOT FOUND ERROR TO OCCUR. THIS IS DONE
4795 ;BY SETTING SECTOR 40 OF THE RLDA AND ISSUING A
4796 ;READ. SECTOR 40 DOES NOT EXIST ON THE RLO1 PACK
4797 ;THEREFORE HDR NT FOUND SHOULD SET.
```

4798 025536

(2)

4799

4800 025536 004737 021154

4801 025542

(4) 025550 104032

(4) 025552 000102

4802

4803 025554

(3) 025554 104004

4804

4805

4806 025556 012777 000050 154452

4807 025564 012777 002652 154442

4808 025572 012777 177777 154440

4809

4810 025600 004537 020254

4811 025604 000014

4812 025606 004537 021074

4813 025612

(3) 025612 104010

(3) 025614 000036

4814

4815 025616 013737 002216 002150

4816 025624 042737 001777 002150

4817 025632 022737 112000 002150

4818 025640 001404

4819

4820 025642

(3) 025642 104462

(5) 025644 000027

(5) 025646 010643

(5) 025650 014044

4821

4822 025652

4823

4824

4825 025652

(3) 025652

(3) 025652 104005

4826 025654

(3) 025654

(3) 025654 104001

4827

4828

4829

4830 025656

4831

4832

4833 025656

(2)

4834

4835

4836

4837 025656

(2)

STARS

;;\*\*\*\*\*

JSR PC,HDHOME ;HEADS OVER TRACK 0

CKERFG ;HEADS GO HOME OKAY

EMT C\$EXIT

.WORD L10051-

BGNSEG ;%%START OF SEGMENT%%

EMT C\$BSEG

MOV #40, @RLDA ;INSURE NOT TO FIND HEADER BY

MOV #BUF, @RLBA ;SETTING SECTOR 40 OF CYL. ADDR.

MOV #-1, @RLMP ;WORD COUNT

JSR R5, LDFUNC ;LOAD THE FUNCTION IN NEXT WORD

READ ;READ

JSR R5, WTCRDY ;WAIT FOR CONTROLLER READY

ESCAPE SEG ;CHECK FOR FL: LOE, ELSE EXIT SEG

EMT C\$ESCAPE

.WORD 10000\$-

MOV E. CS, TMPO ;GET RLCS

BIC #1777, TMPO ;SAVE ERROR BITS

CMP #BIT15!BIT12!BIT10, TMPO ;HDR NOT FOUND SET.

BEQ 15 ;YES, CONTINUE

ERRDF 23, EM10, ERRO ;HEADER NOT FOUND WOULD NOT SET

TRAP T\$ERCODE

.WORD 23

.WORD EM10

.WORD ERRO

15:

;

ENDSEG ;%%END OF SEGMENT%%

10000\$:

EMT C\$ESEG

ENDTST ;\*\*END OF TEST\*\*

L10051:

EMT C\$ETST

.SBTTL \*\*TEST 21\*\* - FORCE HEADER NOT FOUND WITH READ INTERRUPT

BGNTST ;\*\*START OF TEST\*\*

STARS

;;\*\*\*\*\*

;TEST THAT HEADER NOT FOUND ERROR WILL GENERATE AN INTERRUPT

;ON OCCURANCE. HEADER NOT FOUND WILL BE FORCED BY SETTING

;SECTOR 40 OF RLDA AND ISSUING A READ

STARS

;;\*\*\*\*\*

```

4838
4839
4840 025656 004737 021154      JSR    PC,HDHOME      ;HEADS OVER TRACK 0
4841 025662                                CKERFG                ;HEADS GO HOME OKAY
(4) 025670 104032      EMT    C$EXIT
(4) 025672 000142      .WORD  L10052-.
4842
4843 025674                                BGNSEG                ;%%START OF SEGMENT%%
(3) 025674 104004      EMT    C$BSEG
4844
4845 025676                                SETPRI #PR100
(3) 025676 012700 000000      MOV    #PR100,RO
(3) 025702 104041      EMT    C$SPRI
4846 025704 005037 002134      CLR    INTFLG        ;CLEAR INTERRUPT OCCURANCE FLAG
4847 025710 012777 000050 154320      MOV    #40.,@RLDA    ;INSURE NOT TO FIND HEADER BY
4848 025716 012777 002652 154310      MOV    #BUF,@RLBA    ;SETTING SECTOR 40 OF CYL. ADDR.
4849 025724 012777 177777 154306      MOV    #-1,@RLMP     ;WORD COUNT
4850
4851 025732 004537 020254      JSR    R5,LDFUNC      ;LOAD THE FUNCTION IN NEXT WORD
4852 025736 000114                                READ!INTEN           ;READ
4853 025740 004537 021074      JSR    R5,WTCRDY     ;WAIT FOR CONTROLLER READY
4854 025744                                CKLOOP
(3) 025744 104006      EMT    C$CLP1
4855 025746                                SETPRI #PR107
(3) 025746 012700 000340      MOV    #PR107,RO
(3) 025752 104041      EMT    C$SPRI
4856
4857 025754 005737 002134      TST    INTFLG        ;DID INTERRUPT OCCUR
4858 025760 001004      BNE    25            ;YES
4859
4860 025762                                ERRDF 24.,EM43,ERRO ;HNF DID NOT INTERRUPT
(3) 025762 104462      TRAP  T$ERCODE
(5) 025764 000030      .WORD 24
(5) 025766 012523      .WORD EM43
(5) 025770 014044      .WORD ERRO
4861
4862 025772                                25: ESCAPE SEG      ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 025772 104010      EMT    C$ESCAPE
(3) 025774 000036      .WORD 100005-.
4863
4864
4865 025776 013737 002216 002150      MOV    E.CS, TMPO    ;GET RLCS
4866 026004 042737 001777 002150      BIC    #1777, TMPO   ;SAVE ERROR BITS
4867 026012 022737 112000 002150      CMP    #BIT15!BIT12!BIT10, TMPO ;WDR NOT FOUND SET.
4868 026020 001404      BEQ    15            ;YES, CONTINUE
4869
4870 026022                                ERRDF 25.,EM10,ERRO
(3) 026022 104462      TRAP  T$ERCODE
(5) 026024 000031      .WORD 25
(5) 026026 010643      .WORD EM10
(5) 026030 014044      .WORD ERRO
4871
4872 026032                                15: ;WHEN FORCED
4873
4874 026032                                100005: ENDSEG      ;%%END OF SEGMENT%%
(3) 026032

```

(3) 026032 104005  
4875 026034  
(3) 026034  
(3) 026034 104001  
4876  
4877  
4878  
4879 026036  
4880  
4881 026036  
(2)  
4882  
4883  
4884  
4885  
4886  
4887  
4888  
4889  
4890  
4891 026036  
(2)  
4892  
4893  
4894 026036 004737 021154  
4895 026042  
(4) 026050 104032  
(4) 026052 000540  
4896  
4897 026054  
(3) 026054 104004  
4898  
4899 026056  
(3) 026056 012700 000340  
(3) 026062 104041  
4900 026064 012703 002304  
4901  
4902 026070  
(3) 026070 104004  
4903 026072  
4904 026072 004537 020254  
4905 026076 000010  
4906 026100 004537 021074  
4907 026104  
(3) 026104 104010  
(3) 026106 000500  
4908  
4909 026110 004537 020012  
4910 026114  
(3) 026114 104010  
(3) 026116 000470  
4911 026120 013737 002224 002152  
4912  
4913 026126 042737 000177 002152  
4914 026134 012777 000001 154074  
4915 026142 011337 002154

```
EMT C$ESEG ;**END OF TEST**
ENDTST
L10052:
EMT C$ETST
.SBTTL **TEST 22** - CHECK HEADER COMPARE LOGIC
BGNTST ;**START OF TEST**
STARS
;*****
;CHECK THE HEADER COMPARE LOGIC WORKS. UP TO THIS POINT WE
;KNOW THAT THE LOGIC FUNCTIONS PROPERLY BUT NOW WE WILL
;CHECK ALL THE BITS IN THE HEADER WORD. FOUR PATTERNS
;ARE USED A WALKING 1, GROWING 1, WALKING 0, GROWING 0. A SEEK
;IS ISSUED BEFORE EACH READ TO INSURE WE ARE ON THE PROPER
;TRACK. ONCE WE ARE ON THE RIGHT TRACKWE LOAD THE RLDA
;AND ISSUE THE READ. UPON COMPLETION WE WILLCHECK FOR ERRORS
;WE THEN LOAD THE COMPLIMENT PATTERN INTO THE RLDA
;EXPECTING A HEADER NOT FOUND TO SET
STARS
;*****
JSR PC, HDHOME ;HEADS OVER TRACK 0
CKERFG ;HEADS GO HOME OKAY
EMT C$EXIT
.WORD L10053-.
BGNSEG ;%%START OF SEGMENT%%
EMT C$BSEG
SETPRI #PRI07 ;PRIORITY TO 7
MOV #PRI07, R0
EMT C$SPRI
MOV #HDRTAB, R3 ;GET LIST START
BGNSEG ;%%START OF SEGMENT%%
EMT C$BSEG
15: JSR R5, LDFUNC ;LOAD THE FUNCTION IN NEXT WORD
RDHDR ;READ HEADER
JSR R5, WTCRDY ;WAIT FOR CONTROLLRE READY
ESCAPE SEG ;CHECK FOR FL: LOE, ELSE EXIT SEG
EMT C$ESCAPE
.WORD 100015-.
JSR R5, CHERR ;CHECK CNTLR FOR ERRORS
ESCAPE SEG ;CHECK FOR FL: LOE, ELSE EXIT SEG
EMT C$ESCAPE
.WORD 100015-.
MOV E. MP, TMP1 ;READ AND SAVE HEADER
BIC #177, TMP1 ;CLEAR OUT SECTOR AND H. S.
MOV #1, @RLDA ;SETUP MARKER FOR SEEK
MOV (R3), TMP2 ;GET HEADER PATTERN
```

4916	026146	042737	000177	002154		BIC	#177, TMP2	; CLEAR OUT SECTOR AND H. S.
4917	026154	163737	002152	002154		SUB	TMP1, TMP2	; CALCULATE DIFFERENCE TO SEEK
4918	026162	103404				BCS	25	; BRANCH FOR SEEK OUT
4919	026164	052777	000004	154044		BIS	#SIGN, @RLDA	; SEEK TOWARDS SPINDLE
4920	026172	000402				BR	35	; GO PUT IN DIFFERENCE WORD
4921	026174	005437	002154		25:	NEG	TMP2	; WE HAVE TO NEGATE DIFFERENCE
4922	026200	053777	002154	154030	35:	BIS	TMP2, @RLDA	; SET IN DIFFERENCE WORD
4923	026206	032713	000100			BIT	#RHHS, (R3)	; DO WE WANT HEAD SELECT AS 0?
4924	026212	001403				BEQ	45	; YES, SKIP OVER SETTING H. S.
4925	026214	052777	000020	154014		BIS	#DAHS, @RLDA	; SET HEAD SELECT TO ONE
4926	026222	004537	020254		45:	JSR	R5, LDFUNC	; LOAD THE FUNCTION IN NEXT WORD
4927	026226	000006				SEEK		; SEEK
4928								
4929								
4930	026230	004537	021074			JSR	R5, WTCRDY	; WAIT FOR CONTROLLER READY
4931	026234					ESCAPE	SEG	; CHECK FOR FL: LOE, ELSE EXIT SEG
(3)	026234	104010				EMT	C\$ESCAPE	
(3)	026236	000350				. WORD	100015-	
4932								
4933	026240	004537	020012			JSR	R5, CHERR	; CHECK CNTLR FOR ERRORS
4934	026244					ESCAPE	SEG	; CHECK FOR FL: LOE, ELSE EXIT SEG
(3)	026244	104010				EMT	C\$ESCAPE	
(3)	026246	000340				. WORD	100015-	
4935								
4936	026250	004537	021034			JSR	R5, WTD RDY	; WAIT FOR DRIVE PEADY
4937	026254					ESCAPE	SEG	; CHECK FOR FL: LOE, ELSE EXIT SEG
(3)	026254	104010				EMT	C\$ESCAPE	
(3)	026256	000330				. WORD	100015-	
4938	026260	004537	020254			JSR	R5, LDFUNC	; LOAD THE FUNCTION IN NEXT WORD
4939	026264	000010				RDHDR		; READ HEADER (VERIFY SEEK)
4940	026266	004537	021074			JSR	R5, WTCRDY	; WAIT FOR CONTROLLER READY
4941	026272					ESCAPE	SEG	; CHECK FOR FL: LOE, ELSE EXIT SEG
(3)	026272	104010				EMT	C\$ESCAPE	
(3)	026274	000312				. WORD	100015-	
4942								
4943	026276	004537	020012			JSR	R5, CHERR	; CHECK CNTLR FOR ERRORS
4944	026302					ESCAPE	SEG	; CHECK FOR FL: LOE, ELSE EXIT SEG
(3)	026302	104010				EMT	C\$ESCAPE	
(3)	026304	000302				. WORD	100015-	
4945								
4946	026306	013737	002224	002160		MOV	E. MP, BDDAT	; READ HEADER
4947	026314	043737	002140	002160		BIC	SECMSK, BDDAT	; SAVE CYLINDER FOR COMPARE
4948	026322	011337	002156			MOV	(R3), GDDAT	; GET EXPECTED HEADER
4949	026326	043737	002140	002156		BIC	SECMSK, GDDAT	; SAVE CYLINDER FOR COMPARE
4950	026334	023737	002156	002160		CMP	GDDAT, BDDAT	; SEEK END UP OKAY
4951	026342	001404				BEQ	55	; YES, CONTINUE
4952								
4953	026344					ERRDF	27, EM11, ERR4	; SEEK INCORRECT
(3)	026344	104462				TRAP	T\$ERCODE	
(5)	026346	000033				. WORD	27	
(5)	026350	010710				. WORD	EM11	
(5)	026352	014210				. WORD	ERR4	
4954								
4955	026354				55:	ESCAPE	SEG	; CHECK FOR FL: LOE, ELSE EXIT SEG
(3)	026354	104010				EMT	C\$ESCAPE	
(3)	026356	000230				. WORD	100015-	

4956									
4957	026360	011377	153652			MOV	(R3), @RLDA		; SET UP DISK ADDRESS
4958	026364	042777	000077	153644		BIC	#77, @RLDA		
4959	026372	012777	177777	153640		MOV	#-1, @RLMP		; WORD COUNT
4960	026400	012777	002652	153626		MOV	#BUF, @RLBA		; BUS ADDRESS
4961									
4962	026406	004537	020254			JSR	R5, LDFUNC		; LOAD THE FUNCTION IN NEXT WORD
4963	026412	000014				READ			; READ
4964	026414	004537	021074			JSR	R5, WTCRDY		; WAIT FOR CONTROLLER READY
4965	026420					ESCAPE	SEG		; CHECK FOR FL: LOE, ELSE EXIT SEG
(3)	026420	104010				EMT	C\$ESCAPE		
(3)	026422	000164				. WORD	10001\$-		
4966									
4967	026424	004537	020012			JSR	R5, CHERR		; CHECK CNTLR FOR ERRORS
4968	026430					ESCAPE	SEG		; CHECK FOR FL: LOE, ELSE EXIT SEG
(3)	026430	104010				EMT	C\$ESCAPE		
(3)	026432	000154				. WORD	10001\$-		
4969									
4970	026434	011377	153576			MOV	(R3), @RLDA		; SET UP DISK ADDRESS AS
4971	026440	005177	153572			COM	@RLDA		; COMPLIMENT TO CAUSE HDR NT FND
4972	026444	012777	177777	153566		MOV	#-1, @RLMP		; WORD COUNT
4973	026452	012777	002652	153554		MOV	#BUF, @RLBA		; BUS ADDRESS
4974									
4975	026460	004537	020254			JSR	R5, LDFUNC		; LOAD THE FUNCTION IN NEXT WORD
4976	026464	000014				READ			; READ
4977	026466	004537	021074			JSR	R5, WTCRDY		; WAIT FOR CONTROLLER READY
4978	026472					ESCAPE	SEG		; CHECK FOR FL: LOE, ELSE EXIT SEG
(3)	026472	104010				EMT	C\$ESCAPE		
(3)	026474	000112				. WORD	10001\$-		
4979	026476	013737	002216	002150		MOV	E. CS, TMPO		; GET CS
4980	026504	042737	001777	002150		BIC	#1777, TMPO		; SAVE ERROR BITS
4981	026512	022737	112000	002150		CMP	#BIT15!BIT12!BIT10, TMPO		; DID HEADER NOT FOUND SET
4982	026520	001402				BEQ	8\$		; YES, CONTINUE
4983	026522	004537	020012			JSR	R5, CHERR		
4984	026526				8\$:	CKLOOP			
(3)	026526	104006				EMT	C\$CLP1		
4985									
4986	026530	022737	112000	002150		CMP	#BIT15!BIT12!BIT10, TMPO		
4987	026536	001413				BEQ	6\$		
4988									
4989	026540	011337	002156			MOV	(R3), GDDAT		; SET UP DATA FOR ERROR
4990	026544	013737	002156	002160		MOV	GDDAT, BDDAT		; PRINT OUT
4991	026552	005137	002160			COM	BDDAT		
4992									
4993	026556					ERRDF	28, EM12, ERR4		; HDR NOT FOUND WOULD NOT SET
(3)	026556	104462				TRAP	T\$ERCODE		
(5)	026560	000034				. WORD	28		
(5)	026562	010737				. WORD	EM12		
(5)	026564	014210				. WORD	ERR4		
4994									
4995	026566				6\$:	ESCAPE	SEG		; CHECK FOR FL: LOE, ELSE EXIT SEG
(3)	026566	104010				EMT	C\$ESCAPE		
(3)	026570	000016				. WORD	10001\$-		
4996									
4997	026572	005723				TST	(R3)+		; GET NEXT PATTERN
4998	026574	020327	002460			CMP	R3, #HOREND		; AT END?

\*\*TEST 22\*\* - CHECK HEADER COMPARE LOGIC

SEQ 0079

```
4999 026600 001402          BEQ      7$          ;YES, EXIT TEST
5000 026602 000137 026072    JMP      1$          ;NO, GO BACK
5001
5002 026606          7$:
5003 026606          ENDSEG          ;%%END OF SEGMENT%%
   (3) 026606          10001$:
   (3) 026606 104005    EMT      C$ESEG
5004
5005 026610          ENDSEG          ;%%END OF SEGMENT%%
   (3) 026610          10000$:
   (3) 026610 104005    EMT      C$ESEG
5006 026612          ENDTST          ;**END OF TEST**
   (3) 026612          L10053:
   (3) 026612 104001    EMT      C$ETST
5007
5008          .SBTTL  **TEST 23** - CHECK MULTIPLE SECTORS ON READ
5009
5010 026614          BGNTST          ;**START OF TEST**
5011
5012 026614          STARS
   (2)          ;*****
5013          ;VERIFY THAT MULTIPLE SECTORS CAN BE READ, WE WILL CHECK
5014          ;THAT THE RLDA INCREMENTS PROPERLY.
5015 026614          STARS
   (2)          ;*****
5016
5017
5018 026614 004737 021154    JSR      PC,HDHOME    ;HEADS OVER TRACK 0
5019 026620          CKERFG          ;HEADS GO HOME JKAY
   (4) 026626 104032    EMT      C$EXIT
   (4) 026630 000156    .WORD   L10054-
5020
5021
5022 026632 005037 002150    CLR      TMP0          ;CLEAR LOCATIONS
5023 026636 005037 002152    CLR      TMP1
5024
5025 026642          BGNSEG          ;%%START OF SEGMENT%%
   (3) 026642 104004    EMT      C$BSEG
5026
5027 026644          1$:
5028 026644 013737 002152 002156  MOV      TMP1,GDDAT    ;GET CYLINDER
5029 026652 053737 002150 002156  BIS      TMP0,GDDAT    ;GET SECTOR
5030 026660 013777 002156 153350  MOV      GDDAT,@RLDA   ;SET DISK ADDRESS-SECTOR 0
5031 026666 062737 000002 002156  ADD      #2,GDDAT       ;SET EXPECTED + 2
5032 026674 012777 002652 153332  MOV      #BUF,@RLBA    ;SET BUS ADDRESS
5033 026702 012777 177577 153330  MOV      #-129,@RLMP   ;WORD COUNT-SECTOR+1 WORD
5034
5035 026710 004537 020254    JSR      R5,LDFUNC     ;LOAD THE FUNCTION IN NEXT WORD
5036 026714 000014          READ          ;READ
5037 026716 004537 021074    JSR      R5,WTCRDY    ;WAIT FOR CONTROLLER READY?
5038 026722          ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
   (3) 026722 104010    EMT      C$ESCAPE
   (3) 026724 000060    .WORD   10000$-
5039
5040 026726 004537 020012    JSR      R5,CHERR     ;CHECK CNTLR FOR ERRORS
5041 026732          ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
```



```

(3) 026732 104010      EMT      C$ESCAPE
(3) 026734 000050      .WORD   100005-.
5042
5043 026736 013737 002222 002160      MOV      E,DA,BDDAT      ;READ DISK ADDRESS
5044 026744 023737 002160 002156      CMP      BDDAT,GDDAT    ;IS DISK ADDRESS CORRECT
5045 026752 001404      BEQ      2$             ;YES, BRANCH NO, REPORT ERROR
5046
5047 026754      ERRDF   29.,EM14,ERR4   ;DA DID NOT INCREMENT
(3) 026754 104462      TRAP    T$ERCODE
(5) 026756 000035      .WORD   29
(5) 026760 011030      .WORD   EM14
(5) 026762 014210      .WORD   ERR4
5048
5049 026764      2$:    ESCAPE  SEG      ;CHECK FOR FL: LOE, ELSE EXIT SEG
(3) 026764 104010      EMT      C$ESCAPE
(3) 026766 000016      .WORD   100005-.
5050
5051 026770 005237 002150      INC      TMPO           ;NEXT SECTOR?
5052 026774 022737 000046 002150      CMP      #46, TMPO     ;DONE?
5053 027002 001320      BNE     1$             ;NO, GO BACK
5054
5055
5056 027004      ENDSEG                    ;%%END OF SEGMENT%%
(3) 027004      100005:
(3) 027004 104005      EMT      C$ESEG
5057 027006      ENDTST                    ;**END OF TEST**
(3) 027006      L10054:
(3) 027006 104001      EMT      C$ETST
5058 027010      STARS
(2) ;*****
5059 ;CHECK THAT WE CAN FORCE A HEADER NOT FOUND AT THE
5060 ;END OF A TRACK DOING A MULTIPLE SECTOR READ. WE
5061 ;SET UP TO READ TWO SECTORS STARTING AT SECTOR 39
5062 ;WE SHOULD TRANSFER 128 WORDS THEN ABORT WITH A
5063 ;HEADER NOT FOUND FOR SECTOR 40
5064 027010      STARS
(2) ;*****
5065
5066
5067 .SBTTL **TEST 24** - FORCE HDR NT FND AT END OF TRACK
5068
5069 027010      BGNTST                    ;**START OF TEST**
5070
5071
5072 027010 004737 021154      JSR      PC,HDHOME     ;HEADS OVER TRACK 0
5073 027014      CKERFG                    ;HEADS GO HOME OKAY
(4) 027022 104032      EMT      C$EXIT
(4) 027024 000126      .WORD   L10055-.
5074
5075 027026      BGNSEG                    ;%%START OF SEGMENT%%
(3) 027026 104004      EMT      C$BSEG
5076
5077 027030 012737 000047 002156      MOV      #39.,GDDAT    ;CREATE LAST SECTOR
5078 027036 013777 002156 153172      MOV      GDDAT,@RLDA   ;LOAD DISK ADDRESS
5079 027044 012777 177577 153166      MOV      #-129.,@RLMP  ;WORD COUNT
5080 027052 012777 002652 153154      MOV      #BUF,@RLBA    ;BUS ADDRESS
  
```

```

5081 027060 004537 020254 JSR R5, LDFUNC ;LOAD THE FUNCTION IN NEXT WORD
5082 027064 000014 READ ;READ
5083 027066 004537 021074 JSR R5, WTCRDY ;WAIT FOR CONTROLLER READY
5084 027072 ESCAPE SEG ;CHECK FOR FL: LOE, ELSE EXIT SEG
(3) 027072 104010 EMT C$ESCAPE
(3) 027074 000054 .WORD 10000$.
5085
5086 027076 013737 002216 002160 MOV E, CS, BDDAT ;READ CS
5087 027104 042737 001777 002160 BIC #1777, BDDAT ;SAVE ERROR BITS
5088 027112 022737 112000 002160 CMP #112000, BDDAT ;HDR NOT FOUND SET?
5089 027120 001402 BEQ 4$ ;YES, CONTINUE
5090 027122 004537 020012 JSR R5, CHERR
5091 027126 4$: CKLOOP
(3) 027126 104006 EMT C$CLP1
5092
5093 027130 022737 112000 002160 CMP #112000, BDDAT
5094 027136 001404 BEQ 1$
5095
5096 027140 ERRDF 30, EM23, ERRO ;HEADER NOT FOUND DID NOT SET
(3) 027140 104462 TRAP T$ERCODE
(5) 027142 000036 .WORD 30
(5) 027144 011436 .WORD EM23
(5) 027146 014044 .WORD ERRO
5097
5098 027150 1$:
5099
5100 027150 ENDSEG ;%%END OF SEGMENT%%
(3) 027150 10000$:
(3) 027150 104005 EMT C$ESEG
5101 027152 ENDTST ;**END OF TEST**
(3) 027152 L10055:
(3) 027152 104001 EMT C$ETST
5102
5103 .SBTTL **TEST 25** - FORCE NON-EXISTANT MEMORY ERROR
5104
5105 027154 BGNTST ;**START OF TEST**
5106
5107
5108 027154 STARS
(2) ; ;*****
5109 ; ;FORCE A NON-EXISTANT MEMORY ERROR,
5110 ; ;WE SET THE RLBA TO EQUAL THE
5111 ; ;LAST ADDRESS IN MEMORY AND ISSUE A READ. THE
5112 ; ;READ SHOULD ABORT AFTER ONE WORD TRANSFERRED
5113 027154 STARS
(2) ; ;*****
5114
5115
5116 027154 004737 021154 JSR PC, HDHOME ;HEADS OVER TRACK 0
5117 027160 CKERFG ;HEADS GO HOME OKAY
(4) 027166 104032 EMT C$EXIT
(4) 027170 000076 .WORD L10056-.
5118
5119 027172 BGNSEG ;%%START OF SEGMENT%%
(3) 027172 104004 EMT C$BSEG
5120
  
```

```
5121
5122
5123
5124 027174 012777 177774 153032      MOV    #177774, @RLBA    ; LEAD BA
5125 027202 012737 000060 002252      MOV    #BA16!BA17, XMEM ; SET EA BIT
5126 027210 005077 153022                CLR    @RLDA            ; LOAD DISK AVAILABLE
5127 027214 012777 177600 153016      MOV    #-128., @RLMP    ; WORD COUNT
5128 027222 004537 020254                JSR    R5, LDFUNC       ; LOAD THE FUNCTION IN NEXT WORD
5129 027226 000014                READ                   ; READ
5130 027230 004537 021074                JSR    R5, WTCRDY       ; WAIT FOR CONTROLLER
5131 027234                ESCAPE  SEG            ; CHECK FOR FL: LOE, ELSE EXIT SEG
(3) 027234 104010                EMT    C$ESCAPE
(3) 027236 000026                .WORD 10000$-.
5132
5133 027240 032737 020000 002216      BIT    #NXM, E. CS      ; DID NXM SET?
5134 027246 001004                BNE    3$              ; YES, CONTINUE
5135
5136 027250                ERRDF  31., EM24, ERRO  ; NXM DID NOT SET
(3) 027250 104462                TRAP  T$ERCODE
(5) 027252 000037                .WORD 31
(5) 027254 011516                .WORD EM24
(5) 027256 014044                .WORD ERRO
5137
5138 027260                3$:  ESCAPE  SEG            ; CHECK FOR FL: LOE, ELSE EXIT. SEG
(3) 027260 104010                EMT    C$ESCAPE
(3) 027262 000002                .WORD 10000$-.
5139
5140
5141
5142
5143 027264                ENDSEG                  ; %%END OF SEGMENT%%
(3) 027264                10000$:
(3) 027264 104005                EMT    C$ESEG
5144 027266                ENDTST                  ; **END OF TEST**
(3) 027266                L10056:
(3) 027266 104001                EMT    C$ETST
5145
5146                .SBTTL  **TEST 26** - FORCE NON-EXISTANT MEMORY ERROR INTERRUPT
5147
5148 027270                BGNTST                  ; **START OF TEST**
5149 027270                STARS
(2)                ; *****
5150                ; CHECK THAT WE CAN FORCE AN INTERRUPT WITH A
5151                ; NON-EXISTANT MEMORY ERROR.
5152 027270                STARS
(2)                ; *****
5153
5154
5155 027270 004737 021154                JSR    PC, HDHOME       ; HEADS OVER TRACK 0
5156 027274                CKERFG                  ; HEADS GO HOME OKAY
(4) 027302 104032                EMT    C$EXIT
(4) 027304 000140                .WORD L10057-.
5157
5158 027306                BGNSEG                  ; %%START OF SEGMENT%%
(3) 027306 104004                EMT    C$BSEG
5159
```

```

5160 027310 005037 002134          CLR      INTFLG          ;CLEAR INTERRUPT OCCURANCE FLAG
5161
5162
5163
5164 027314          SETPRI  #PR100
(3) 027314 012700 000000          MOV      #PR100,RO
(3) 027320 104041          EMT      C$SPRI
5165 027322 012777 177774 152704          MOV      #177774, @RLBA ;PRELOAD BA
5166 027330 012737 000060 002252          MOV      #BA16!BA17,XMEM ;SET EA BITS
5167 027336 005077 152674          CLR      @RLDA          ;LOAD DA
5168 027342 012777 177777 152670          MOV      #-1, @RLMP     ;WORD COUNT
5169 027350 004537 020254          JSR      R5, LDFUNC     ;LOAD THE FUNCTION IN NEXT WORD
5170 027354 000114          READ! INTEN           ;READ
5171 027356 004537 021074          JSR      R5, WTCRDY    ;WAIT FOR CONTROLLER
5172 027362          SETPRI  #PR107        ;PRIORITY TO 7
(3) 027362 012700 000340          MOV      #PR107,RO
(3) 027366 104041          EMT      C$SPRI
5173 027370          ESCAPE  SEG           ;CHECK FOR FL: LOE, ELSE EXIT SEG
(3) 027370 104010          EMT      C$ESCAPE
(3) 027372 000050          .WORD   10000$-
5174
5175 027374 005737 002134          TST      INTFLG        ; INTERRUPT OCCUR?
5176 027400 001004          BNE     4$             ; YES OKAY
5177
5178 027402          ERRDF  32, EM44, ERRO ; NO INTERRUPT W/NXM
(3) 027402 104462          TRAP   T$ERCODE
(5) 027404 000040          .WORD  32
(5) 027406 012567          .WORD  EM44
(5) 027410 014044          .WORD  ERRO
5179
5180 027412          4$: ESCAPE  SEG           ;CHECK FOR FL: LOE, ELSE EXIT SEG
(3) 027412 104010          EMT      C$ESCAPE
(3) 027414 000026          .WORD   10000$-
5181
5182 027416 032737 020000 002216          BIT      #NXM, E. CS   ; DID NXM SET?
5183 027424 001004          BNE     3$             ; YES, CONTINUE
5184
5185 027426          ERRDF  33, EM24, ERRO ; NO NXM
(3) 027426 104462          TRAP   T$ERCODE
(5) 027430 000041          .WORD  33
(5) 027432 011516          .WORD  EM24
(5) 027434 014044          .WORD  ERRO
5186
5187 027436          3$: ESCAPE  SEG           ;CHECK FOR FL: LOE, ELSE EXIT SEG
(3) 027436 104010          EMT      C$ESCAPE
(3) 027440 000002          .WORD   10000$-
5188
5189
5190 027442          ENDSEG                ;%%END OF SEGMENT%%
(3) 027442          10000$: EMT      C$ESEG
(3) 027442 104005          .WORD
5191 027444          ENDTST                ;**END OF TEST**
(3) 027444          L10057: EMT      C$SETST
(3) 027444 104001          .WORD
5192
5193          .SBTTL **TEST 27** - CHECK READ WRITE LOOP

```

```
5194
5195 027446          BGNTST                      ;**START OF TEST**
5196
5197 027446          STARS
(2)                ;*****
5198                ;VERIFY THAT THE WRITE ACTUALLY WRITES.  AT THIS
5199                ;TIME WE KNOW THAT THE WRITE FUNCTION GOES THRU
5200                ;THE MOTIONS BUT WE DON'T KNOW THAT THE DATA
5201                ;ACTUALLY GETS RECORDED ON THE PLATTER.
5202 027446          STARS
(2)                ;*****
5203
5204
5205 027446 004737 021154      JSR      PC,HDHOME      ;HEADS OVER TRACK 0
5206 027452          CKERFG          ;HEADS GO HOME OKAY
(4) 027460 104032          EMT      C$EXIT
(4) 027462 000362          .WORD   L10060-.
5207
5208 027464          BGNSEG          ;%%START OF SEGMENT%%
(3) 027464 104004          EMT      C$BSEG
5209
5210 027466 012700 002652      MOV      #BUF,RO      ;SET UP WRITE BUFFER
5211 027472 012701 000200      MOV      #128.,R1     ;128 WORDS/ONE SECTOR
5212 027476 012720 125252      3$:    MOV      #125252,(RO)+ ;WRITE PATTERN TO BUFFER
5213 027502 005301          DEC      R1           ;DONE?
5214 027504 001374          BNE     3$           ;NO, BRANCH BACK
5215 027506 005077 152524      CLR     @RLDA        ;DISK ADDRESS
5216 027512 012777 177600 152520  MOV     #-128.,@RLMP  ;WORD COUNT
5217 027520 012777 002652 152506  MOV     #BUF,@RLBA   ;BUS ADDRESS
5218 027526 004537 020254      JSR     R5,LDFUNC   ;LOAD THE FUNCTION IN NEXT WORD
5219 027532 000012          WRITE          ;WRITE THE PATTERN
5220 027534 004537 021074      JSR     R5,WTCRDY   ;WAIT FOR CONTROLLER READY
5221 027540          ESCAPE          ;CHECK FOR FL: LOE, ELSE EXIT SEG
(3) 027540 104010          EMT      C$ESCAPE
(3) 027542 000300          .WORD   10000$-.
5222
5223 027544 004537 020012      JSR     R5,CHERR    ;CHECK CNTLR FOR ERRORS
5224 027550          ESCAPE          ;CHECK FOR FL: LOE, ELSE EXIT SEG
(3) 027550 104010          EMT      C$ESCAPE
(3) 027552 000270          .WORD   10000$-.
5225 027554          BGNSEG          ;%%START OF SEGMENT%%
(3) 027554 104004          EMT      C$BSEG
5226 027556 012700 002652      MOV     #BUF,RO     ;CLEAR OUT BUFFER BEFORE
5227 027562 012701 000200      MOV     #128.,R1    ;READING
5228 027566 005020          4$:    CLR     (RO)+      ;CLEAR BUFFER
5229 027570 005301          DEC     R1          ;DONE?
5230 027572 001375          BNE     4$         ;NO, BRANCH BACK
5231
5232 027574 005077 152436      CLR     @RLDA        ;LOAD DISK ADDRESS
5233 027600 012777 177600 152432  MOV     #-128.,@RLMP ;WORD COUNT/ONE SECTION
5234 027606 012777 002652 152420  MOV     #BUF,@RLBA  ;LOAD BUS ADDRESS
5235 027614 004537 020254      JSR     R5,LDFUNC   ;LOAD THE FUNCTION IN NEXT WORD
5236 027620 000014          READ          ;GO READ
5237 027622 004537 021074      JSR     R5,WTCRDY   ;WAIT FOR CONTROLLER READY
5238 027626          ESCAPE          ;CHECK FOR FL: LOE, ELSE EXIT SEG
(3) 027626 104010          EMT      C$ESCAPE
```

```
(3) 027630 000210 . WORD 100015-.  
5239  
5240 027632 004537 020012 JSR R5,CHERR ;CHECK CNTLR FOR ERRORS  
5241 027636 005737 002114 TST T.CRC ;WAS ERROR A DCK??  
5242 027642 001003 BNE 8$ ;YES,SEE IF WE A DUMP  
5243 027644 105: ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG  
(3) 027644 104010 EMT C$ESCAPE  
(3) 027646 000172 . WORD 100015-.  
5244 027650 000404 BR 99$ ;SKIP AROUND  
5245 027652 005737 016574 8$: TST T.DMP ;DO WE STILL WANT TO CHECK IT  
5246 027656 001772 BEQ 10$ ;NO  
5247 027660 CKLOOP ;YES, CHECK FOR LOOP FIRST  
(3) 027660 104006 EMT C$CLP1  
5248  
5249 027662 005037 002120 99$: CLR CDCNT ;CLEAR NUMBER WE'RE TO PRINT  
5250 027666 005037 002112 CLR CHECK ;ALLOW HEADER ON FIRST PRINT  
5251 027672 012702 002652 MOV #BUF,R2 ;COMPARE BUFFER TO CHECK WRITE  
5252 027676 012701 000200 MOV #128.,R1 ;128 WORDS  
5253 027702 012737 125252 002156 MOV #125252,GDDAT ;SET UP EXPECTED  
5254 027710 011237 002160 5$: MOV (R2),BDDAT ;GET DATA  
5255 027714 023737 002156 002160 CMP GDDAT,BDDAT ;IS DATA OKAY  
5256 027722 001442 BEQ 6$ ;YES, CONTINUE  
5257 027724 010237 002152 MOV R2,TMP1 ;LOAD BAD MEM LOCATION  
5258 027730 023737 002120 016576 CMP CDCNT,T.LMT ;CHECKED ENOUGH??  
5259 027736 001002 BNE 333$ ;NO  
5260 027740 ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG  
(3) 027740 104010 EMT C$ESCAPE  
(3) 027742 000076 . WORD 100015-.  
5261 027744 005237 002120 333$: INC CDCNT ;ACCOUNT FOR IT  
5262  
5263 027750 005737 002112 TST CHECK ;HEADER OR JUST DATA  
5264 027754 001007 BNE 9$ ;JUST DATA  
5265 027756 ERRDF 34.,EM25,ERR8 ;BAD DATA  
(3) 027756 104462 TRAP T$ERCODE  
(5) 027760 000042 . WORD 34  
(5) 027762 011574 . WORD EM25  
(5) 027764 014364 . WORD ERR8  
5266 027766 005237 002112 INC CHECK ;ACCOUNT FOR PRINT OF HEADER  
5267 027772 000416 BR 6$  
5268  
5269 027774 9$: PRINTB #FRMT6,TMP1,GDDAT,BDDAT  
(10) 027774 013746 002160 MOV BDDAT,-(SP)  
(9) 030000 013746 002156 MOV GDDAT,-(SP)  
(8) 030004 013746 002152 MOV TMP1,-(SP)  
(7) 030010 012746 015642 MOV #FRMT6,-(SP)  
(6) 030014 012746 000004 MOV #4,-(SP)  
(3) 030020 010600 MOV SP,R0  
(4) 030022 104014 EMT C$PNTB  
(4) 030024 062706 000012 ADD #12,SP  
5270  
5271 030030 6$: CKLOOP  
(3) 030030 104006 EMT C$CLP1  
5272 030032 005722 7$: TST (R2)+ ;BUMP BUFFER POINTER  
5273 030034 005301 DEC R1 ;DONE?  
5274 030036 001324 BNE 5$ ;NO, GO BACK  
5275 030040 ENDSEG ;%%END OF SEGMENT%%
```

```
(3) 030040          10001$:          EMT      C$ESEG          ;%%END OF SEGMENT%%
(3) 030040 104005          ENDSEG
5276 030042          10000$:          EMT      C$ESEG          ;**END OF TEST**
(3) 030042 104005          ENDTST
(3) 030044          L10060:          EMT      C$ETST
5277 030044 104001          .SBTTL  **TEST 28** - CHECK SILO LINES
(3) 030044          BGNTST          ;**START OF TEST**
5278
5279
5280
5281 030046          STARS
5282          ;*****
5283          ;TEST THAT LINES IN / TO SILO ARE GOOD, THAT IS THAT EACH LINE IS
5284          ;GOOD AND CAN BE AT EITHER A 1 OR A 0 STATE INDEPENTENTLY OF EACH
5285          ;OTHER BIT POSITION THIS IS DONE BY WRITING PATTERNS OF FLOATING 1,
5286          ;FLOATING 0, WALKING 0, WALKING 1
5287          STARS
5288          ;*****
5289          (2)
5290 030046          JSR      PC,HDHOME          ;HEADS OVER TRACK 0
5291          CKERFG          ;HEADS GO HOME OKAY
5292          (4) 030060 104032          EMT      C$EXIT
5293 030046 004737 021154          .WORD   L10061-
5294 030062 000404
5295          (4) 030062 000404
5296 030064 012703 002462          MOV      #DATPAT,R3
5297
5298          BGNSEG          ;%%START OF SEGMENT%%
5299 030070          EMT      C$BSEG
5300 030072 012700 002652          65:    MOV      #BUF,R0          ;WRITE PATTERN INTO MEMORY
5301 030076 012701 000200          MOV      #128,R1          ;128 WORDS
5302 030102 011320          25:    MOV      (R3),(R0)+        ;WRITE THE PATTERN
5303 030104 005301          DEC      R1          ;DONE?
5304 030106 001375          BNE     25          ;NO GO BACK
5305
5306 030110 012777 002652 152116          MOV     #BUF,@RLBA        ;SETUP TO WRITE PATTERN ONTO DISK
5307 030116 005077 152114          CLR     @RLDA          ;LOAD DA
5308 030122 012777 177600 152110          MOV     #-128,@RLMP      ;WORD COUNT
5309 030130 004537 020254          JSR     R5,LDFUNC        ;LOAD THE FUNCTION IN NEXT WORD
5310 030134 000012          WRITE
5311 030136 004537 021074          JSR     R5,WTCRDY
5312 030142          ESCAPE SEG          ;CHECK FOR FL: LOE, ELSE EXIT SEG
5313 030142 104010          EMT      C$ESCAPE
5314 030144 000320          .WORD   10000$-
5315 030146 004537 020012          JSR     R5,CHERR
5316 030152          ESCAPE SEG          ;CHECK CNTLR FOR ERRORS
5317 030152 104010          EMT      C$ESCAPE
5318 030154 000310          .WORD   10000$-
5319 030156          BGNSEG          ;%%START OF SEGMENT%%
(3) 030156 104004          EMT      C$BSEG
```

5316	030160	012700	002652		MOV	#BUF, R0	; CLEAR MEMORY BEFORE READING IT BACK
5317	030164	012701	000200		MOV	#128, R1	; 128 WORDS
5318	030170	005020		35:	CLR	(R0)+	; CLEAR
5319	030172	005301			DEC	R1	; EONE
5320	030174	001375			BNE	35	; NO
5321							
5322	030176	012777	002652	152030	MOV	#BUF, @RLBA	; SETUP TO READ IT BACK
5323	030204	012777	177600	152026	MOV	#-128, @RLMP	; 128 WORDS
5324	030212	005077	152020		CLR	@RLDA	; SECTOR ZERO
5325	030216	004537	020254		JSR	R5, LDFUNC	; LOAD THE FUNCTION IN NEXT WORD
5326	030222	000014			READ		
5327	030224	004537	021074		JSR	R5, WTCRDY	
5328	030230				ESCAPE	SEG	; CHECK FOR FL: LOE, ELSE EXIT SEG
(3)	030230	104010			EMT	C\$ESCAPE	
(3)	030232	000224			.WORD	100015-	
5329	030234	004537	020012		JSR	R5, CHERR	; CHECK CNTLR FOR ERRORS
5330	030240	005737	002114		TST	T. CRC	; WAS ERROR A DCK??
5331	030244	001003			BNE	85	; YES, SEE IF WE A DUMP
5332	030246			105:	ESCAPE	SEG	; CHECK FOR FL: LOE, ELSE EXIT SEG
(3)	030246	104010			EMT	C\$ESCAPE	
(3)	030250	000206			.WORD	100015-	
5333	030252	000404			BR	995	; SKIP AROUND
5334	030254	005737	016574	85:	TST	T. DMP	; DO WE STILL WANT TO CHECK IT
5335	030260	001772			BEQ	105	; NO
5336	030262				CKLOOP		; YES, CHECK FOR LOOP FIRST
(3)	030262	104006			EMT	C\$CLP1	
5337							
5338	030264	005037	002120	995:	CLR	CDCNT	; CLEAR NUMBER WE'RE TO PRINT
5339	030270	005037	002112		CLR	CHECK	; ALLOW HEADER ON FIRST PRINT
5340	030274	011337	002156		MOV	(R3), GDDAT	; COMPARE WHAT WE READ BACK
5341	030300	012737	002652	002154	MOV	#BUF, TMP2	; BUFFER START
5342	030306	012737	000001	002152	MOV	#1, TMP1	; START WITH FIRST
5343							
5344	030314	017737	151634	002160	55:	MOV	@TMP2, BDDAT
5345	030322	023737	002156	002160	CMF	GDDAT, BDDAT	; GOOD?
5346	030330	001440			BEQ	45	; YES, BRANCH
5347							
5348	030332	023737	002120	016576	CMF	CDCNT, T LMT	; CHECKED ENOUGH??
5349	030340	001002			BNE	3335	; NO
5350	030342				ESCAPE	SEG	; CHECK FOR FL: LOE, ELSE EXIT SEG
(3)	030342	104010			EMT	C\$ESCAPE	
(3)	030344	000112			.WORD	100015-	
5351	030346	005237	002120	3335:	INC	CDCNT	; ACCOUNT FOR IT
5352							
5353	030352	005737	002112		TST	CHECK	; HEADER OR JUST DATA
5354	030356	001007			BNE	95	; JUST DATA
5355	030360				ERRDF	35, EM45, ERR10	; BAD DATA BACK
(3)	030360	104462			TRAP	T\$ERCODE	
(5)	030362	000043			.WORD	35	
(5)	030364	012624			.WORD	EM45	
(5)	030366	014502			.WORD	ERR10	
5356							
5357	030370	005237	002112		INC	CHECK	; ACCOUNT FOR PRINT OF HEADER
5358	030374	000416			BR	45	
5359							
5360	030376			95:	PRINTB	#FRMT7, TMP1, GDDAT, BDDAT	



```

(10) 030376 013746 002160      MOV      BDDAT,-(SP)
(9)  030402 013746 002156      MOV      GDDAT,-(SP)
(8)  030406 013746 002152      MOV      TMP1,-(SP)
(7)  030412 012746 015717      MOV      #FRMT7,-(SP)
(6)  030416 012746 000004      MOV      #4,-(SP)
(3)  030422 010600              MOV      SP,RO
(4)  030424 104014              EMT      C$PNTB
(4)  030426 062706 000012      ADD      #12,SP
5361 030432                    4$:     CKLOOP
(3)  030432 104006              EMT      C$CLP1
5362
5363 030434 062737 000002 002154      ADD      #2,TMP2      ;NEXT LOCATION
5364 030442 005237 002152              INC      TMP1        ;NEXT WORD
5365 030446 023727 002152 000201      CMP      TMP1,#129.  ;DONE
5366 030454 001317              BNE      5$         ;NO, GO BACK
5367
5368 030456                    ENDSEG          ;%%END OF SEGMENT%%
(3)  030456                    10001$:
(3)  030456 104005              EMT      C$ESEG
5369
5370 030460 005723              TST      (R3)+      ;DONE ALL PATTERNS
5371 030462 001203              BNE      6$         ;NO, GO BACK
5372
5373 030464                    ENDSEG          ;%%END OF SEGMENT%%
(3)  030464                    10000$:
(3)  030464 104005              EMT      C$ESEG
5374 030466                    ENDTST          ;**END OF TEST**
(3)  030466                    L10061:
(3)  030466 104001              EMT      C$ETST
5375
5376 . SBTTL **TEST 29** - CHECK THROUGHPUT OF SILO
5377
5378 030470                    BGNTST          ;**START OF TEST**
5379
5380
5381
5382 030470                    STARS
(2)  ;*****
5383 ;TEST THAT THE SILO OPERATES CORRECTLY, WE WILL WRITE A PATTERN THAT CONTAINS
5384 ;A UNIQUE PATTERN IN EACH LOCATION. WE EXPECT IT BACK IN PROPER
5385 ;ORDER, WE DO A ONE SECTOR TRANSFER
5386 030470                    STARS
(2)  ;*****
5387
5388
5389 030470 004737 021154      JSR      PC,HDHOME  ;HEADS OVER TRACK 0
5390 030474                    CKERFG          ;HEADS GO HOME OKAY
(4)  030502 104032              EMT      C$EXIT
(4)  030504 000410              .WORD   L10062-.
5391
5392 030506                    BGNSEG          ;%%START OF SEGMENT%%
(3)  030506 104004              EMT      C$BSEG
5393
5394
5395 030510 012700 000001      MOV      #1,RO      ;INITIAL 1
5396 030514 012701 000200      MOV      #128.,R1   ;128 WORDS
  
```

```

5397 030520 012702 002652          MOV    #BUF,R2          ;BUFFER
5398 030524 010022          2$:   MOV    RO,(R2)+       ;WRITE A WORD
5399 030526 005200          INC    RO              ;NEXT PATTERN (1-128)
5400 030530 005301          DEC    R1              ;DONE
5401 030532 001374          BNE   2$              ;NO
5402
5403 030534 012777 002652 151472    MOV    #BUF,@RLBA      ;SETUP TO WRITE
5404 030542 012777 177600 151470    MOV    #-128,@RLMP    ;128 WORDS
5405 030550 005077 151462          CLR    @RLDA          ;DISK ADDRESS 0
5406 030554 004537 020254          JSR   R5,LDFUNC       ;LOAD THE FUNCTION IN NEXT WORD
5407 030560 000012          WRITE
5408 030562 004537 021074          JSR   R5,WTCRDY
5409 030566          ESCAPE SEG           ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 030566 104010          EMT   C$ESCAPE
(3) 030570 000322          .WORD 10000$-
5410
5411 030572 004537 020012          JSR   R5,CHERR        ;CHECK CNTLR FOR ERRORS
5412 030576          ESCAPE SEG           ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 030576 104010          EMT   C$ESCAPE
(3) 030600 000312          .WORD 10000$-
5413 030602          BGNSEG                ;%%START OF SEGMENT%%
(3) 030602 104004          EMT   C$BSEG
5414 030604 012700 002652          MOV    #BUF,RO        ;CLEAR BUFFER
5415 030610 012701 000200          MOV    #128,R1       ;128 IN LENGTH
5416 030614 005020          3$:   CLR    (RO)+         ;CLEAR
5417 030616 005301          DEC    R1              ;DOWN COUNT
5418 030620 001375          BNE   3$              ;DONE?
5419
5420 030622 012777 002652 151404    MOV    #BUF,@RLBA      ;BUS ADDRESS
5421 030630 012777 177600 151402    MOV    #-128,@RLMP    ;WORD COUNT
5422 030636 005077 151374          CLR    @RLDA          ;DISK ADDRESS
5423 030642 004537 020254          JSR   R5,LDFUNC       ;LOAD THE FUNCTION IN NEXT WORD
5424 030646 000014          READ
5425 030650 004537 021074          JSR   R5,WTCRDY
5426 030654          ESCAPE SEG           ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 030654 104010          EMT   C$ESCAPE
(3) 030656 000232          .WORD 10001$-
5427
5428 030660 004537 020012          JSR   R5,CHERR        ;CHECK CNTLR FOR ERRORS
5429 030664 005737 002114          TST   T.CRC           ;WAS ERROR A DCK??
5430 030670 001003          BNE   8$              ;YES,SEE IF WE A DUMP
5431 030672          10$:  ESCAPE SEG           ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 030672 104010          EMT   C$ESCAPE
(3) 030674 000214          .WORD 10001$-
5432 030676 000404          BR    99$            ;SKIP AROUND
5433 030700 005737 016574          8$:   TST   T.DMP         ;DO WE STILL WANT TO CHECK IT
5434 030704 001772          BEQ   10$            ;NO
5435 030706          CKLOOP
(3) 030706 104006          EMT   C$CLP1         ;YES,CHECK FOR LOOP FIRST
5436
5437 030710 005037 002120          99$:  CLR    CDCNT          ;CLEAR NUMBER WE'RE TO PRINT
5438 030714 005037 002112          CLR    CHECK          ;ALLOW HEADER ON FIRST PRINT
5439 030720 012737 000001 002156    MOV    #1,GDDAT       ;START GOOD AT 1
5440 030726 012737 002652 002154    MOV    #BUF,TMP2      ;START OF BUFFER
5441 030734 012737 000001 002152    MOV    #1,TMP1        ;FIRST WORD
5442

```

```

5443 030742 017737 151206 002160 45:  MOV  @TMP2,BDDAT ;GET WORD
5444 030750 023737 002160 002156  CMP  BDDAT,GDDAT ;CORRECT?
5445 030756 001440  BEQ  65 ;YES
5446
5447 030760 023737 002120 016576  CMP  CDCNT,T.LMT ;CHECKED ENOUGH??
5448 030766 001002  BNE  333$ ;NO
5449 030770  ESCAPE SEG ;CHECK FOR FL: LOE, ELSE EXIT SEG
(3) 030770 104010  EMT  C$ESCAPE
(3) 030772 000116  .WORD 100015-
5450 030774 005237 002120 333$: INC  CDCNT ;ACCOUNT FOR IT
5451
5452 031000 005737 002112  TST  CHECK ;HEADER OR JUST DATA
5453 031004 001007  BNE  95 ;JUST DATA
5454 031006  ERRDF 36,EM47,ERR10 ;BAD DATA
(3) 031006 104462  TRAP T$ERCODE
(5) 031010 000044  .WORD 36
(5) 031012 012656  .WORD EM47
(5) 031014 014502  .WORD ERR10
5455 031016 005237 002112  INC  CHECK ;ACCOUNT FOR PRINT OF HEADER
5456 031022 000416  BR   65
5457
5458 031024 95:  PRINTB #FRMT7,TMP1,GDDAT,BDDAT
(10) 031024 013746 002160  MOV  BDDAT,-(SP)
(9) 031030 013746 002156  MOV  GDDAT,-(SP)
(8) 031034 013746 002152  MOV  TMP1,-(SP)
(7) 031040 012746 015717  MOV  #FRMT7,-(SP)
(6) 031044 012746 000004  MOV  #4,-(SP)
(3) 031050 010600  MOV  SP,R0
(4) 031052 104014  EMT  C$PNTB
(4) 031054 062706 000012  ADD  #12,SP
5459 031060 65:  CKLOOP
(3) 031060 104006  EMT  C$CLP1
5460
5461 031062 062737 000002 002154  ADD  #2,TMP2 ;NEXT
5462 031070 005237 002152  INC  TMP1 ;NEXT
5463 031074 005237 002156  INC  GDDAT ;NEXT
5464 031100 023727 002152 000201  CMP  TMP1,#129. ;DONE?
5465 031106 001315  BNE  45
5466
5467 031110  ENDSEG ;%%END OF SEGMENT%%
(3) 031110 100015: EMT  C$ESEG
(3) 031110 104005
5468
5469 031112  ENDSEG ;%%END OF SEGMENT%%
(3) 031112 100005: EMT  C$ESEG
(3) 031112 104005
5470 031114  ENDTST ;**END OF TEST**
(3) 031114  L10062: EMT  C$ETST
(3) 031114 104001
5471
5472  .SBTTL **TEST 30** - CHECK ZERO FILL ON WRITE
5473
5474 031116  BGNST ;**START OF TEST**
5475
5476
5477

```

5478 031116

(2)

5479

5480

5481

5482

5483 031116

(2)

5484

5485 031116 004737 021154

5486 031122

(4) 031130 104032

(4) 031132 000442

5487

5488 031134

(3) 031134 104004

5489

5490 031136 012737 000001 002152

5491 031144 012700 002652

5492 031150 012701 000200

5493 031154 012720 052525

5494 031160 005301

5495 031162 001374

5496 031164 013700 002152

5497 031170 005400

5498 031172 010077 151042

5499 031176 012777 002652 151030

5500 031204 005077 151026

5501 031210 004537 020254

5502 031214 000012

5503 031216 004537 021074

5504 031222

(3) 031222 104010

(3) 031224 000346

5505

5506 031226 004537 020012

5507 031232

(3) 031232 104010

(3) 031234 000336

5508 031236

(3) 031236 104004

5509 031240 012700 002652

5510 031244 012701 000200

5511 031250 012720 125252

5512 031254 005301

5513 031256 001374

5514

5515 031260 012777 002652 150746

5516 031266 012777 177600 150744

5517 031274 005077 150736

5518 031300 004537 020254

5519 031304 000014

5520 031306 004537 021074

5521 031312

(3) 031312 104010

(3) 031314 000234

STARS

;\*\*\*\*\*

;WHEN WRITING PARTIAL SECTORS (LESS THAN 128 WORDS) THE

;CONTROLLER WILL FILL IN THE REMAINING PORTION OF

;THE SECTOR WITH ZERO WORDS. CHECK THIS FEATURE

;WITH WORD COUNTS FROM 1 TO 127

STARS

;\*\*\*\*\*

JSR PC,HDHOME ;HEADS OVER TRACK 0

CKERFG ;HEADS GO HOME OKAY

EMT C\$EXIT

.WORD L10063-

BGNSEG ;%%START OF SEGMENT%%

EMT C\$BSEG

MOV #1,TMP1 ;START WITH 1 WORD WRITE

35: MOV #BUF,RO ;WRITE BUFFER WITH 52525, WE'LL

MOV #128,R1 ;WRITE 128 WORDS ALL THOUGH WE'RE

35: MOV #52525,(RO)+ ;ONLY GOING TO TRANSFER < 128

DEC R1 ;DONE WITH BUFFER?

BNE 35 ;NO, GO BACK

33: MOV TMP1,RO ;GET TRANSFER WORD COUNT

NEG RO ;NEGATE FOR RLMP

MOV RO,@RLMP ;STORE WORD COUNT AWAY

MOV #BUF,@RLBA ;SET UP RLBA

CLR @RLDA

JSR R5,LDFUNC

;LOAD THE FUNCTION IN NEXT WORD

WRITE ;WRITE IT

JSR R5,WTCRDY ;WAIT FOR WRITE TO FINISH

ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG

EMT C\$ESCAPE

.WORD 10000-

JSR R5,CHERR ;CHECK CNTLR FOR ERRORS

ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG

EMT C\$ESCAPE

.WORD 10000-

BGNSEG ;%%START OF SEGMENT%%

EMT C\$BSEG

MOV #BUF,RO ;WE'RE GOING TO OVERLAY BUFFER BEFORE

18: MOV #128,R1 ;READING IT BACK.

MOV #125252,(RO)+ ;OVERLAY IT WITH COMPLIMENT

DEC R1 ;DONE?

BNE 18 ;NO, KEEP GOING

MOV #BUF,@RLBA ;SET UP TO READ

MOV #-128,@RLMP ;128 WORDS TO CHECK ZERO FILL

CLR @RLDA ;SECTOR

JSR R5,LDFUNC ;LOAD THE FUNCTION IN NEXT WORD

READ

JSR R5,WTCRDY ;WAIT TIL WE FINISH THE READ

ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG

EMT C\$ESCAPE

.WORD 10001-

```

5522
5523 031316 004537 020012      JSR      R5,CHERR          ;CHECK CNTLR FOR ERRORS
5524 031322 005737 002114      TST      T.CRC            ;WAS ERROR A DCK??
5525 031326 001003              BNE      8$               ;YES,SEE IF WE A DUMP
5526 031330              10$:    ESCAPE  SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
      (3) 031330 104010      EMT      C$ESCAPE
      (3) 031332 000216      .WORD   10001$-
5527 031334 000404              BR       99$             ;SKIP AROUND
5528 031336 005737 016574      8$:    TST      T.DMP          ;DO WE STILL WANT TO CHECK IT
5529 031342 001772              BEQ      10$             ;NO
5530 031344              CKLOOP
      (3) 031344 104006      EMT      C$CLP1          ;YES, CHECK FOR LOOP FIRST
5531 031346 005037 002120      99$:   CLR      CDCNT          ;CLEAR NUMBER WE'RE TO PRINT
5532 031352 005037 002112      CLR      CHECK           ;ALLOW HEADER ON FIRST PRINT
5533 031356 013702 002152      MOV      TMP1,R2         ;WORDS WRITTEN IN R2
5534 031362 012701 000200      MOV      #128,R1         ;CHECK 128 WORDS
5535
5536 031366 012703 002652      MOV      #BUF,R3         ;SET UP BUFFER BEGINNING
5537 031372 005037 002154      CLR      TMP2            ;ZERO WORD COUNT
5538 031376 012737 052525 002156  4$:    MOV      #52525,GDDAT    ;SET UP EXPECTED
5539 031404 011337 002160      MOV      (R3),BDDAT      ;GET WORD
5540 031410 023737 002160 002156  CMP      BDDAT,GDDAT     ;IS WORD CORRECT?
5541 031416 001441              BEQ      12$             ;YES, GO CHECK COUNTS AND REPEAT
5542
5543 031420 023737 002120 016576  CMP      CDCNT,T.LMT      ;CHECKED ENOUGH??
5544 031426 001002              BNE      333$           ;NO
5545 031430              ESCAPE  SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
      (3) 031430 104010      EMT      C$ESCAPE
      (3) 031432 000116      .WORD   10001$-
5546 031434 005237 002120      333$:  INC      CDCNT          ;ACCOUNT FOR IT
5547
5548 031440 005737 002112      TST      CHECK           ;HEADER OR JUST DATA
5549 031444 001007              BNE      9$              ;JUST DATA
5550 031446              ERRDF  37,EM27,ERR12
      (3) 031446 104462      TRAP    T$ERCODE
      (5) 031450 000045      .WORD   37
      (5) 031452 011705      .WORD   EM27
      (5) 031454 014626      .WORD   ERR12
5551 031456 005237 002112      INC      CHECK           ;ACCOUNT FOR PRINT OF HEADER
5552 031462 000417      BR       12$
5553
5554 031464              9$:    PRINTB #FRMT9,TMP1,R3,GDDAT,BDDAT
      (11) 031464 013746 002160      MOV      BDDAT,-(SP)
      (10) 031470 013746 002156      MOV      GDDAT,-(SP)
      (9) 031474 010346      MOV      R3,-(SP)
      (8) 031476 013746 002152      MOV      TMP1,-(SP)
      (7) 031502 012746 016112      MOV      #FRMT9,-(SP)
      (6) 031506 012746 000005      MOV      #5,-(SP)
      (3) 031512 010600      MOV      SP,R0
      (4) 031514 104014      EMT      C$PNTB
      (4) 031516 062706 000014      ADD      #14,SP
5555 031522              12$:  CKLOOP
      (3) 031522 104006      EMT      C$CLP1
5556 031524 005723      6$:    TST      (R3)+
5557 031526 005237 002154      INC      TMP2
5558 031532 005301      DEC      R1              ;DONE ALL WORDS?
  
```

```
5559 031534 001405      BEQ      7$      ;EXIT TEST
5560 031536 005302      DEC      R2      ;DONE CHECKING NON-ZERO WORDS
5561 031540 003321      BGT      4$      ;NO, BRANCH BACK
5562 031542 005037 002156  CLR      GDDAT   ;YES, SET EXP'D AS ZEPO
5563 031546 000716      BR       4$      ;BRANCH BACK
5564
5565 031550      7$:      ;EXIT TEST
5566 031550      ENDSEG      ;%%END OF SEGMENT%%
(3) 031550      10001$:
(3) 031550 104005      EMT      C$ESEG
5567
5568 031552 005237 002152      INC      TMP1
5569 031556 023727 002152 000200  CMP      TMP1,#128.
5570 031564 001402      BEQ      34$
5571 031566 000137 031144      JMP      35$
5572 031572      34$:
5573
5574 031572      ENDSEG      ;%%END OF SEGMENT%%
(3) 031572      10000$:
(3) 031572 104005      EMT      C$ESEG
5575 031574      ENDTST      ;**END OF TEST**
(3) 031574      L10063:
(3) 031574 104001      EMT      C$ETST
5576
5577      .SBTTL  **TEST 31** - CHECK SECTOR BITS OF HEADER COMPARE
5578
5579 031576      BGNTST      ;**START OF TEST**
5580
5581
5582 031576      STARS
(2) ;*****
5583 ;TEST THAT ALL SECTOR BITS OF HEADER WORD CAN COMPARE
5584 ;UNIQUELY. WE TESTED THE HEADER COMPARE LOGIC EARLIER
5585 ;BUT THAT WAS NOT AN EXTENSIVE TEST OF THE SECTOR BITS.
5586 ;THE TEST PROCEDURE IS TO WRITE EACH SECTOR OF TRACK
5587 ;0 WITH THE SECTOR ADDRESS, THEN GO BACK AND READ
5588 ;EACH SECTOR. IF ANY SECTOR HAS ANY DATA THEN THAT
5589 ;WHICH WAS EXPECTED THEN WE HAVE AN ERROR
5590 ;ERROR PRINT OUT WILL GIVE SECTOR, EXPECTED AND RECEIVED
5591 031576      STARS
(2) ;*****
5592
5593
5594
5595
5596
5597
5598
5599 031576 004737 021154      JSR      PC,HDHOME ;HEADS OVER TRACK 0
5600 031602      CKERFG      ;HEADS GO HOME OKAY
(4) 031610 104032      EMT      C$EXIT
(4) 031612 000414      .WORD   L10064-.
5601
5602 031614      BGNSEG      ;%%START OF SEGMENT%%
(3) 031614 104004      EMT      C$BSEG
5603
```

```

5604 031616 005037 002150      15:  CLR      TMPO      ;CLEAR
5605
5606 031622                      BGNSEG                      ;%%START OF SEGMENT%%
   (3) 031622 104004          EMT      C$BSEG
5607
5608 031624 012702 002652      199$: MOV      #BUF,R2      ;WRITE A PATTERN FOR THE WRITE
5609 031630 012701 000200      MOV      #128.,R1      ;ONE SECTOR'S WORTH
5610 031634 013722 002150      2$:  MOV      TMPO,(R2)+  ;WRITE IT
5611 031640 005301      DEC      R1            ;DONE,
5612 031642 001374      BNE      2$           ;IF NOT, GO BACK
5613
5614 031644 012777 177600 150366  MOV      #-128.,@RLMP  ;ONE SECTOR WORD COUNT
5615 031652 012777 002652 150354  MOV      #BUF,@RLBA  ;WRITE FROM BUF
5616 031660 013777 002150 150350  MOV      TMPO,@RLDA  ;SECTOR
5617 031666 004537 020254      JSR      R5,LDFUNC    ;LOAD THE FUNCTION IN NEXT WORD
5618 031672 000012      WRITE
5619 031674 004537 021074      JSR      R5,WTCRDY   ;WAIT FOR WRITE TO FINISH
5620 031700      ESCAPE  SEG          ;CHECK FOR FL: LOE, ELSE EXIT SEG
   (3) 031700 104010      EMT      C$ESCAPE
   (3) 031702 000320      .WORD   100015-.
5621 031704 005237 002150      INC      TMPO        ;NEXT SECTOR
5622 031710 023727 002150 000050  CMP      TMPO,#40.   ;ALL DONE?
5623 031716 001342      BNE      199$       ;NO GO BACK
5624 031720 005037 002150      CLR      TMPO        ;CLEAR
5625
5626 031724                      BGNSEG                      ;%%START OF SEGMENT%%
   (3) 031724 104004          EMT      C$BSEG
5627
5628 031726 012702 002652      98$: MOV      #BUF,R2      ;CLEAR THE BUFFER FIRST
5629 031732 012701 000200      MOV      #128.,R1    ;128 WORDS
5630 031736 005022      3$:  CLR      (R2)+
5631 031740 005301      DEC      R1
5632 031742 001375      BNE      3$
5633
5634 031744 013777 002150 150264  MOV      TMPO,@RLDA  ;GET SECTOR
5635 031752 012777 002652 150254  MOV      #BUF,@RLBA  ;SETUP BUS ADDRESS
5636
5637 031760 012777 177600 150252  MOV      #-128.,@RLMP ;READ A SECTOR
5638 031766 004537 020254      JSR      R5,LDFUNC    ;LOAD THE FUNCTION IN NEXT WORD
5639 031772 000014      READ
5640 031774 004537 021074      JSR      R5,WTCRDY   ;CHECK FOR FL: LOE, ELSE EXIT SEG
5641 032000      ESCAPE  SEG
   (3) 032000 104010      EMT      C$ESCAPE
   (3) 032002 000216      .WORD   100025-.
5642
5643 032004 004537 020012      JSR      R5,CHERR    ;CHECK CNTLR FOR ERRORS
5644 032010 005737 002114      TST     T.CRC        ;WAS ERROR A DCK??
5645 032014 001003      BNE     8$          ;YES, SEE IF WE A DUMP
5646 032016      ESCAPE  SEG          ;CHECK FOR FL: LOE, ELSE EXIT SEG
   (3) 032016 104010      EMT      C$ESCAPE
   (3) 032020 000200      .WORD   100025-.
5647 032022 000404      BR      99$         ;SKIP AROUND
5648 032024 005737 016574      8$:  TST     T.DMP        ;DO WE STILL WANT TO CHECK IT
5649 032030 001772      BEQ     10$        ;NO
5650 032032      CKLOOP 10$        ;YES, CHECK FOR LOOP FIRST
   (3) 032032 104006      EMT      C$CLP1

```

```

5651
5652           ;CHECK NOW TO SEE IF WE READ THE RIGHT SECTOR
5653
5654 032034 005037 002120          99$: CLR      CDCNT      ;CLEAR NUMBER WE'RE TO PRINT
5655 032040 005037 002112          CLR      CHECK      ;ALLOW HEADER ON FIRST PRINT
5656 032044 013737 002150 002156  MOV      TMPO,GDDAT  ;EXPECTED DATA
5657 032052 012702 002652          MOV      #BUF,R2    ;BUFFER
5658 032056 012701 000200          MOV      #128.,R1   ;WORD COUNT
5659 032062 012237 002160          5$:  MOV      (R2)+,BDDAT ;
5660 032066 023737 002160 002156  CMP      BDDAT,GDDAT
5661 032074 001440          BEQ      6$
5662
5663 032076 023737 002120 016576  CMP      CDCNT,T.LMT ;CHECKED ENOUGH??
5664 032104 001002          BNE      333$      ;NO
5665 032106          ESCAPE  SEG      ;CHECK FOR FL: LOE, ELSE EXIT SEG
(3) 032106 104010          EMT      C$ESCAPE
(3) 032110 000110          .WORD   10002$-
5666 032112 005237 002120          333$: INC      CDCNT      ;ACCOUNT FOR IT
5667
5668 032116 005737 002112          TST      CHECK      ;HEADER OR JUST DATA
5669 032122 001007          BNE      9$        ;JUST DATA
5670 032124          ERRDF  38.,EM50,ERR11 ;
(3) 032124 104462          TRAP    T$ERCODE
(5) 032126 000046          .WORD   38
(5) 032130 012705          .WORD   EM50
(5) 032132 014554          .WORD   ERR11
5671 032134 005237 002112          INC      CHECK      ;ACCOUNT FOR PRINT OF HEADER
5672 032140 000416          BR      6$
5673
5674 032142          9$:  PRINTB #FRMT8, TMPO, GDDAT, BDDAT
(10) 032142 013746 002160          MOV      BDDAT, -(SP)
(9) 032146 013746 002156          MOV      GDDAT, -(SP)
(8) 032152 013746 002150          MOV      TMPO, -(SP)
(7) 032156 012746 015771          MOV      #FRMT8, -(SP)
(6) 032162 012746 000004          MOV      #4, -(SP)
(3) 032166 010600          MOV      SP, R0
(4) 032170 104014          EMT      C$PNTB
(4) 032172 062706 000012          ADD     #12, SP
5675 032176          6$:  CKLOOP
(3) 032176 104006          EMT      C$CLP1
5676
5677 032200 005301          DEC      R1        ;ALL OF SECTOR CHECKED?
5678 032202 001327          BNE      5$        ;GO BACK IF NOT
5679 032204 005237 002150          INC      TMPO      ;NEXT SECTOR
5680 032210 023727 002150 000050  CMP      TMPO, #40. ;DONE?
5681 032216 001243          BNE      98$      ;NO, GO BACK
5682
5683 032220          ENDSEG          ;%%END OF SEGMENT%%
(3) 032220          10002$: EMT      C$ESEG
(3) 032220 104005
5684
5685 032222          ENDSEG          ;%%END OF SEGMENT%%
(3) 032222          10001$: EMT      C$ESEG
(3) 032222 104005
5686 032224          ENDSEG          ;%%END OF SEGMENT%%
(3) 032224          10000$:

```



```
(3) 032224 104005          EMT      C$ESEG
5687 032226          ENDTST          ;**END OF TEST**
(3) 032226          L10064:
(3) 032226 104001          EMT      C$ETST
5688
5689          .SBTTL  **TEST 32** - WRITE CHECK NPR INTEGRITY
5690
5691 032230          BGNTST          ;**START OF TEST**
5692
5693 032230          STARS
(2)          ;*****
5694          ;CHECK THAT NPR WILL NOT INTERFERE WITH THE OPERATION OF THE
5695          ;UNIBUS. WE SET UP LOCATION 4 TO HANDLE THE TRAP IF IT HAPPENS.
5696 032230          STARS
(2)          ;*****
5697
5698
5699 032230 004737 021154      JSR      PC,HDHOME      ;HEADS OVER TRACK 0
5700 032234          CKERFG          ;HEADS GO HOME OKAY
(4) 032242 104032          EMT      C$EXIT
(4) 032244 000372          .WORD   L10065-.
5701
5702 032246          BGNSEG          ;%%START OF SEGMENT%%
(3) 032246 104004          EMT      C$BSEG
5703
5704 032250 012700 002652      MOV      #BUF,RO        ;SETUP AND WRITE
5705 032254 012701 000200      MOV      #128.,R1       ;128 WORDS
5706 032260 012720 125252      299$:  MOV      #125252,(RO)+ ;WRITE
5707 032264 005301          DEC      R1             ;DONE??
5708 032266 001374          BNE     299$
5709
5710 032270 012777 002652 147736  MOV      #BUF,@RLBA     ;LOAD BUS ADDRESS
5711 032276 012777 177600 147734  MOV      #-128.,@RLMP   ;WORD COUNT
5712 032304 005077 147726          CLR      @RLDA         ;CLEAR DISK ADDRESS
5713 032310 004537 020254          JSR      R5,LDFUNC     ;LOAD THE FUNCTION IN NEXT WORD
5714 032314 000012          WRITE
5715 032316 004537 021074          JSR      R5,WTCRDY    ;WAIT FOR CONTROLLER READY
5716 032322          ESCAPE  SEG          ;CHECK FOR FL: LOE, ELSE EXIT SEG
(3) 032322 104010          EMT      C$ESCAPE
(3) 032324 000310          .WORD   10000$-.
5717 032326 004537 020012          JSR      R5,CHERR     ;CHECK CNTLR FOR ERRORS
5718 032332          ESCAPE  SEG          ;CHECK FOR FL: LOE, ELSE EXIT SEG
(3) 032332 104010          EMT      C$ESCAPE
(3) 032334 000300          .WORD   10000$-.
5719
5720
5721          ;VERIFY WRITE WITH READ BEFORE WRCHK
5722
5723 032336 005077 147674          CLR      @RLDA
5724 032342 012777 002652 147664  MOV      #BUF,@RLBA
5725 032350 012777 177600 147662  MOV      #-128.,@RLMP
5726 032356 004537 020254          JSR      R5,LDFUNC     ;LOAD THE FUNCTION IN NEXT WORD
5727 032362 000014          READ
5728 032364 004537 021074          JSR      R5,WTCRDY    ;CHECK FOR FL: LOE, ELSE EXIT SEG
5729 032370          ESCAPE  SEG
(3) 032370 104010          EMT      C$ESCAPE
```

```

(3) 032372 000242 .WORD 10000$-
5730 032374 004537 020012 JSR R5,CHERR ;CHECK CNTLR FOR ERRORS
5731 032400 ESCAPE SEG ;CHECK FOR FL: LOE, ELSE EXIT SEG
(3) 032400 104010 EMT C$ESCAPE
(3) 032402 000232 .WORD 10000$-
5732 BGNSEG ;%%START OF SEGMENT%%
5733 032404 EMT C$BSEG
(3) 032404 104004
5734
5735 032406 15: SETVEC ERRVEC, #TRPHAN, #340 ;SET UP FOR TRAP
(7) 032406 012746 000340 MOV #340, -(SP)
(6) 032412 012746 021146 MOV #TRPHAN, -(SP)
(5) 032416 013746 002122 MOV ERRVEC, -(SP)
(4) 032422 012746 000003 MOV #3, -(SP)
(3) 032426 104037 EMT C$SVEC
(2) 032430 062706 000010 ADD #10, SP
5736 032434 005037 002132 CLR TRPFLG ;CLEAR TRAP OCCURANCE
5737 032440 012777 002652 147566 MOV #BUF, @RLBA ;BUS ADDRESS
5738 032446 005077 147564 CLR @RLDA ;LOAD DISK ADDRESS
5739 032452 012777 177600 147560 MOV #-128, @RLMP ;WORD COUNT OF 128
5740 032460 005037 002156 CLR GDDAT ;SET UP CSR TO LOAD
5741 032464 013737 002124 002156 MOV DRIVE, GDDAT ;SET IN DRIVE
5742 032472 052737 000002 002156 BIS #WRCHK, GDDAT ;SET IN FUNCTION
5743 032500 004537 020562 JSR R5, BEFORE ;LOAD FOR ERROR PRINTOUT
5744 032504 013737 002156 002206 MOV GDDAT, B. CS ;SET IN COMMAND
5745 032512 052737 000201 002206 BIS #201, B. CS ;LOAD CRDY
5746 032520 042737 002000 002206 BIC #OPI, B. CS ;CLEAR (BIT 10)
5747 032526 013777 002156 147476 MOV GDDAT, @RLCS ;ISSUE WRITE CHECK
5748 032534 012701 000144 MOV #100, R1 ;WAIT FOR CRDY
5749 032540 032777 000200 147464 55: BIT #CRDY, @RLCS ;NPR DONE
5750 032546 001013 BNE 6$ ;YES, 6$
5751 032550 WAITUS #20. ;WAIT A WHILE
(3) 032550 012700 000024 MOV #20, R0
(3) 032554 104027 EMT C$WTU
5752 032556 005301 DEC R1 ;A WHILE UP
5753 032560 001367 BNE 5$ ;NO, GO BACK
5754
5755 032562 004537 020614 JSR R5, AFTER
5756 032566 ERRDF 0, CRTIM, ERR5 ;CONTROLLER TIMED OUT
(3) 032566 104462 TRAP T$ERCODE
(5) 032570 000000 .WORD 0
(5) 032572 006772 .WORD CRTIM
(5) 032574 014256 .WORD ERR5
5757 032576 65: CLRVEC ERRVEC ;CLEAR VECTOR
(3) 032576 013700 002122 MOV ERRVEC, R0
(3) 032602 104036 EMT C$CVEC
5758 032604 ESCAPE SEG ;CHECK FOR FL: LOE, ELSE EXIT SEG
(3) 032604 104010 EMT C$ESCAPE
(3) 032606 000024 .WORD 10001$-
5759
5760 032610 005737 002132 TST TRPFLG ;DID TRAP OCCUR?
5761 032614 001406 BEQ 7$ ;NO
5762 032616 004537 020614 JSR R5, AFTER
5763 032622 ERRSF 1, EM57, ERRO ;TRAP ON WRITE
(3) 032622 104461 TRAP T$ERCODE
(5) 032624 000001 .WORD 1

```

```

(5) 032626 013277 .WORD EM57
(5) 032630 014044 .WORD ERRO
5764 032632 75:
5765
5766
5767 032632 ENDSEG ;%%END OF SEGMENT%%
(3) 032632 100015:
(3) 032632 104005 EMT C$ESEG
5768 032634 ENDSEG ;%%END OF SEGMENT%%
(3) 032634 100005:
(3) 032634 104005 EMT C$ESEG
5769
5770 032636 ENDTST ;**END OF TEST**
(3) 032636 L10065:
(3) 032636 104001 EMT C$ETST
5771
5772 .SBTTL **TEST 33** - WRITE CHECK FUNCTION
5773
5774 032640 BGNTST ;**START OF TEST**
5775
5776 032640 STARS
(2) ;*****
5777 ;CHECK OF WRITE CHECK LOGIC UNDER FLAG MODE
5778 ; WE WILL WRITE CHECK A FULL SECTOR (128 WORDS) FROM
5779 ;MEMORY (BUF). WE CHECK THAT NO ERRORS OCCUR.
5780 032640 STARS
(2) ;*****
5781
5782
5783 032640 004737 021154 JSR PC, HDHOME ;HEADS OVER TRACK 0
5784 032644 CKERFG ;HEADS GO HOME OKAY
(4) 032652 104032 EMT C$EXIT
(4) 032654 000214 .WORD L10066-.
5785
5786 032656 BGNSEG ;%%START OF SEGMENT%%
(3) 032656 104004 EMT C$BSEG
5787
5788 032660 012700 002652 MOV #BUF, R0 ;SETUP AND WRITE
5789 032664 012701 000200 MOV #128, R1 ;128 WORDS
5790 032670 012720 125252 299$: MOV #125252, (R0)+ ;WRITE
5791 032674 005301 DEC R1 ;DONE??
5792 032676 001374 BNE 299$
5793
5794 032700 012777 002652 147326 MOV #BUF, @RLBA ;LOAD BUS ADDRESS
5795 032706 012777 177600 147324 MOV #-128, @RLMP ;WORD COUNT
5796 032714 005077 147316 CLR @RLDA ;CLEAR DISK ADDRESS
5797 032720 004537 020254 JSR R5, LDFUNC ;LOAD THE FUNCTION IN NEXT WORD
5798 032724 000012 WRITE
5799 032726 004537 021074 JSR R5, WTCRDY ;WAIT FOR CONTROLLER READY
5800 032732 ESCAPE SEG ;CHECK FOR FL: LOE, ELSE EXIT SEG
(3) 032732 104010 EMT C$ESCAPE
(3) 032734 000132 .WORD 100005-.
5801 032736 004537 020012 JSR R5, CHERR ;CHECK CNTLR FOR ERRORS
5802 032742 ESCAPE SEG ;CHECK FOR FL: LOE, ELSE EXIT SEG
(3) 032742 104010 EMT C$ESCAPE
(3) 032744 000122 .WORD 100005-.
  
```

```
5803 032746          BGNSEG          ;%%START OF SEGMENT%%
(3) 032746 104004    EMT          C$BSEG
5804
5805          ;VERIFY WRITE WITH READ BEFORE WRCHK
5806
5807 032750 005077 147262    CLR          @RLDA
5808 032754 012777 002652 147252    MOV          #BUF,@RLBA
5809 032762 012777 177600 147250    MOV          #-128.,@RLMP
5810 032770 004537 020254          JSR          R5,LDFUNC          ;LOAD THE FUNCTION IN NEXT WORD
5811 032774 000014          READ
5812 032776 004537 021074          JSR          R5,WTCRDY
5813 033002          ESCAPE        SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 033002 104010          EMT          C$ESCAPE
(3) 033004 000060          .WORD        10001$-
5814 033006 004537 020012          JSR          R5,CHERR          ;CHECK CNTLR FOR ERRORS
5815 033012          ESCAPE        SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 033012 104010          EMT          C$ESCAPE
(3) 033014 000050          .WORD        10001$-
5816
5817 033016          BGNSEG          ;%%START OF SEGMENT%%
(3) 033016 104004    EMT          C$BSEG
5818
5819 033020          35:
5820 033020 005077 147212    CLR          @RLDA
5821 033024 012777 177600 147206    MOV          #-128.,@RLMP          ;WORD COUNT
5822 033032 012777 002652 147174    MOV          #BUF,@RLBA          ;BUS ADDRESS
5823 033040 004537 020254          JSR          R5,LDFUNC          ;LOAD THE FUNCTION IN NEXT WORD
5824 033044 000002          WRCHK          ;WRITE CHECK
5825
5826 033046 004537 021074          JSR          R5,WTCRDY          ;WAIT FOR CONTROLLER READY
5827 033052          ESCAPE        SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 033052 104010          EMT          C$ESCAPE
(3) 033054 000006          .WORD        10002$-
5828
5829
5830 033056 004537 020012          JSR          R5,CHERR          ;CHECK CNTLR FOR ERRORS
5831
5832 033062          ENDSEG          ;%%END OF SEGMENT%%
(3) 033062          10002$:
(3) 033062 104005          EMT          C$ESEG
5833 033064          ENDSEG          ;%%END OF SEGMENT%%
(3) 033064          10001$:
(3) 033064 104005          EMT          C$ESEG
5834 033066          ENDSEG          ;%%END OF SEGMENT%%
(3) 033066          10000$:
(3) 033066 104005          EMT          C$ESEG
5835 033070          ENDTST          ;**END OF TEST**
(3) 033070          L10066:
(3) 033070 104001          EMT          C$ETST
5836
5837          .SBTTL **TEST 34** - WRITE CHECK FUNCTION INTERRUPT
5838
5839 033072          BGNTST          ;**START OF TEST**
5840
5841 033072          STARS
(2)          ;*****
```

```

5842 ;CHECK OF WRITE CHECK LOGIC UNDER INTERRUPT MODE
5843 ;WE WILL WRITE CHECK A FULL SECTOR (128 WORDS) FROM MEMORY (BUF).
5844 ;WE CHECK THAT NO ERRORS OCCUR. WE DO NOT CHECK RLDA OR RLBA
5845 ;INCREMENT AT THIS TIME.
5846 033072 STARS
(2) ;*****
5847
5848
5849 033072 004737 021154 JSR PC,HDHOME ;HEADS OVER TRACK 0
5850 033076 CKERFG ;HEADS GO HOME OKAY
(4) 033104 104032 EMT C$EXIT
(4) 033106 000252 .WORD L10067-.
5851
5852 033110 BGNSEG ;%%START OF SEGMENT%%
(3) 033110 104004 EMT C$BSEG
5853
5854 033112 012700 002652 MOV #BUF,RO ;SETUP AND WRITE
5855 033116 012701 000200 MOV #128.,R1 ;128 WORDS
5856 033122 012720 125252 299$: MOV #125252,(RO)+ ;WRITE
5857 033126 005301 DEC R1 ;DONE??
5858 033130 001374 BNE 299$
5859
5860 033132 012777 002652 147074 MOV #BUF,@RLBA ;LOAD BUS ADDRESS
5861 033140 012777 177600 147072 MOV #-128.,@RLMP ;WORD COUNT
5862 033146 005077 147064 CLR @RLDA ;CLEAR DISK ADDRESS
5863 033152 004537 020254 JSR R5,LDFUNC ;LOAD THE FUNCTION IN NEXT WORD
5864 033156 000012 WRITE
5865 033160 004537 021074 JSR R5,WTCRDY ;WAIT FOR CONTROLLER READY
5866 033164 ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 033164 104010 EMT C$ESCAPE
(3) 033166 000170 .WORD 10000$-.
5867 033170 004537 020012 JSR R5,CHERR ;CHECK CNTLR FOR ERRORS
5868 033174 ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 033174 104010 EMT C$ESCAPE
(3) 033176 000160 .WORD 10000$-.
5869 ;VERIFY WRITE WITH READ BEFORE WRCHK
5870
5871 033200 005077 147032 CLR @RLDA
5872 033204 012777 002652 147022 MOV #BUF,@RLBA
5873 033212 012777 177600 147020 MOV #-128.,@RLMP
5874 033220 004537 020254 JSR R5,LDFUNC ;LOAD THE FUNCTION IN NEXT WORD
5875 033224 000014 READ
5876 033226 004537 021074 JSR R5,WTCRDY
5877 033232 ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 033232 104010 EMT C$ESCAPE
(3) 033234 000122 .WORD 10000$-.
5878 033236 004537 020012 JSR R5,CHERR ;CHECK CNTLR FOR ERRORS
5879 033242 ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 033242 104010 EMT C$ESCAPE
(3) 033244 000112 .WORD 10000$-.
5880
5881 033246 BGNSEG ;%%START OF SEGMENT%%
(3) 033246 104004 EMT C$BSEG
5882
5883
5884 033250 005037 002134 CLR INTFLG ;CLEAR INTERRUPT OCCURANCE FLAG
  
```

```

5885 033254 005077 146756 CLR @RLDA
5886 033260 012777 177600 146752 MOV #-128.,@RLMP ;SET UP WORD COUNT
5887 033266 012777 002652 146740 MOV #BUF,@RLBA ;SET UP BUS ADDRESS
5888
5889 033274 SETPRI #PR100 ;PRIORITY TO 0
(3) 033274 012700 000000 MOV #PR100,RO
(3) 033300 104041 EMT C$SPRI
5890 033302 004537 020254 JSR R5,LDFUNC ;LOAD THE FUNCTION IN NEXT WORD
5891 033306 000102 WRCHK!INTEN ;WRITE CHECK UNDER INTERRUPT.
5892 033310 004537 021074 JSR R5,WTCRDY ;WAIT FOR INTERRUPT
5893 033314 ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 033314 104010 EMT C$ESCAPE
(3) 033316 000036 .WORD 100015-.
5894
5895 033320 SETPRI #PR107 ;SET PRIORITY TO 7
(3) 033320 012700 000340 MOV #PR107,RO
(3) 033324 104041 EMT C$SPRI
5896 033326 005737 002134 TST INTFLG ;DID INTERRUPT OCCUR?
5897 033332 001004 BNE 2$ ;YES-BRANCH NO-REPORT
5898
5899 033334 ERRDF 4.,EM60,ERRO ;WRITE DID NOT INTERRUPT
(3) 033334 104462 TRAP T$ERCODE
(5) 033336 000004 .WORD 4
(5) 033340 013337 .WORD EM60
(5) 033342 014044 .WORD ERRO
5900 033344 2$: ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 033344 104010 EMT C$ESCAPE
(3) 033346 000006 .WORD 100015-.
5901
5902 033350 004537 020012 JSR R5,CHERR ;CHECK CNTLR FOR ERRORS
5903
5904 033354 ENDSEG ;%%END OF SEGMENT%%
(3) 033354 100015: EMT C$ESEG
(3) 033354 104005 ENDSEG ;%%END OF SEGMENT%%
5905 033356 100005: EMT C$ESEG
(3) 033356 104005 ENDTST ;**END OF TEST**
5906 033360 L10067: EMT C$ETST
(3) 033360 104001
5907
5908 .SBTTL **TEST 35** - PROPER INCREMENT OF RLBA ON WRITE CHECK
5909
5910 033362 BGNTST ;**START OF TEST**
5911
5912
5913 033362 STARS
(2) ;*****
5914 ;CHECK THAT THE RLBA WILL INCREMENT PROPERLY AFTER THE
5915 ;WRITE CHECK WAS FINISHED THE RLBA SHOULD BE 128 WORDS (256 BYTES)
5916 ;CREATED. STARTING RLBA IS "BUF", ENDING SHOULD BE "BUF + 256."
5917 ;WE WILL MONITOR ALL ERRORS AND REPORT THEM ACCORDINGLY
5918 033362 STARS
(2) ;*****
5919
5920

```

OUTERR MACY11 30(1046) 03-NOV-77 10:02 PAGE 83-79  
 DZRLBA.P11 03-NOV-77 09:54

\*\*TEST 35\*\* - PROPER INCREMENT OF RLBA ON WRITE CHECK

SEQ 0102

5921	033362	004737	021154		JSR	PC, HDHOME		; HEADS OVER TRACK 0
5922	033366				CKERFG			; HEADS GO HOME OKAY
(4)	033374	104032			EMT	C\$EXIT		
(4)	033376	000256			.WORD	L10070-		
5923								
5924	033400				BGNSEG			; %%START OF SEGMENT%%
(3)	033400	104004			EMT	C\$BSEG		
5925								
5926	033402	012700	002652		MOV	#BUF, R0		; SETUP AND WRITE
5927	033406	012701	000200		MOV	#128., R1		; 128 WORDS
5928	033412	012720	125252		299\$: MOV	#125252, (R0)+		; WRITE
5929	033416	005301			DEC	R1		; DONE??
5930	033420	001374			BNE	299\$		
5931								
5932	033422	012777	002652	146604	MOV	#BUF, @RLBA		; LOAD BUS ADDRESS
5933	033430	012777	177600	146602	MOV	#-128., @RLMP		; WORD COUNT
5934	033436	005077	146574		CLR	@RLDA		; CLEAR DISK ADDRESS
5935	033442	004537	020254		JSR	R5, LDFUNC		; LOAD THE FUNCTION IN NEXT WORD
5936	033446	000012			WRITE			
5937	033450	004537	021074		JSR	R5, WTCRDY		; WAIT FOR CONTROLLER READY
5938	033454				ESCAPE	SEG		; CHECK FOR FL: LOE, ELSE EXIT SEG
(3)	033454	104010			EMT	C\$ESCAPE		
(3)	033456	000174			.WORD	10000\$-		
5939	033460	004537	020012		JSR	R5, CHERR		; CHECK CNTLR FOR ERRORS
5940	033464				ESCAPE	SEG		; CHECK FOR FL: LOE, ELSE EXIT SEG
(3)	033464	104010			EMT	C\$ESCAPE		
(3)	033466	000164			.WORD	10000\$-		
5941								; VERIFY WRITE WITH READ BEFORE WRCHK
5942								
5943	033470	005077	146542		CLR	@RLDA		
5944	033474	012777	002652	146532	MOV	#BUF, @RLBA		
5945	033502	012777	177600	146530	MOV	#-128., @RLMP		
5946	033510	004537	020254		JSR	R5, LDFUNC		; LOAD THE FUNCTION IN NEXT WORD
5947	033514	000014			READ			
5948	033516	004537	021074		JSR	R5, WTCRDY		
5949	033522				ESCAPE	SEG		; CHECK FOR FL: LOE, ELSE EXIT SEG
(3)	033522	104010			EMT	C\$ESCAPE		
(3)	033524	000126			.WORD	10000\$-		
5950	033526	004537	020012		JSR	R5, CHERR		; CHECK CNTLR FOR ERRORS
5951	033532				ESCAPE	SEG		; CHECK FOR FL: LOE, ELSE EXIT SEG
(3)	033532	104010			EMT	C\$ESCAPE		
(3)	033534	000116			.WORD	10000\$-		
5952								
5953	033536				BGNSEG			; %%START OF SEGMENT%%
(3)	033536	104004			EMT	C\$BSEG		
5954								
5955	033540				3\$: CLR	@RLDA		
5956	033540	005077	146472		MOV	#BUF, @RLBA		; SET UP BUS ADDRESS
5957	033544	012777	002652	146462	MOV	#-128., @RLMP		; WORD COUNT
5958	033552	012777	177600	146460	MOV	#BUF, GDDAT		; FORM EXPECTED BUS ADDRESS
5959	033560	012737	002652	002156	ADD	#256., GDDAT		; AFTER WRITE
5960	033566	062737	000400	002156				
5961								
5962	033574	004537	020254		JSR	R5, LDFUNC		; LOAD THE FUNCTION IN NEXT WORD
5963	033600	000002			WRCHK			; WRITE CHECK
5964	033602	004537	021074		JSR	R5, WTCRDY		; WAIT FOR CONTROLLER READY

```

5965 033606          ESCAPE SEG          ;CHECK FOR FL: LOE, ELSE EXIT SEG
      (3) 033606 104010 EMT          C$ESCAPE
      (3) 033610 000040 . WORD      10001$-.
5966
5967 033612 004537 020012 JSR          R5, CHERR          ;CHECK CNTLR FOR ERRORS
5968 033616          ESCAPE SEG          ;CHECK FOR FL: LOE, ELSE EXIT SEG
      (3) 033616 104010 EMT          C$ESCAPE
      (3) 033620 000030 . WORD      10001$-.
5969 033622 017737 146406 002160 MOV          @RLBA, BDDAT      ;READ 'RLBA' FOR PRESENT ADDRESS
5970 033630 023737 002160 002156 CMP          BDDAT, GDDAT     ;DID 'BA' INCREMENT PROPERLY?
5971 033636 001404 BEQ          2$              ;YES, CONTINUE
5972

```

```

5973 033640          ERRDF 5, EM61, ERR4 ;BA DID NOT INCREMENT
      (3) 033640 104462 TRAP        T$ERCODE
      (5) 033642 000005 . WORD      5
      (5) 033644 013375 . WORD      EM61
      (5) 033646 014210 . WORD      ERR4
5974

```

```

5975 033650          2$:
5976
5977 033650          ENDSEG          ;%%END OF SEGMENT%%
      (3) 033650 10001$:          10001$:
      (3) 033650 104005          EMT          C$ESEG
5978 033652          ENDSEG          ;%%END OF SEGMENT%%
      (3) 033652 10000$:          10000$:
      (3) 033652 104005          EMT          C$ESEG
5979 033654          ENDTST          ;**END OF TEST**
      (3) 033654 L10070:          L10070:
      (3) 033654 104001          EMT          C$ETST
5980

```

. SBTTL \*\*TEST 36\*\* - PROPER INCREMENT OF RLDA ON WRITE CHECK

5983 033656 BGNTST ;\*\*START OF TEST\*\*

```

5985 033656 STARS
      (2) ; ; *****
5986 ;CHECK THAT THE SECTOR INCREMENTS AFTER THE WRITE CHECK WAS FINISHED.
5987 ;A FULL SECTOR WRITE CHECK THE RLDA SHOULD REFLECT AN INCREMENT
5988 ;OF THE SECOTR. "GDDAT" WAS THE EXPECTED RLDA.
5989 033656 STARS
      (2) ; ; *****
5990

```

```

5992 033656 004737 021154 JSR          PC, HDHOME      ;HEADS OVER TRACK 0
5993 033662          CKERFG          ;HEADS GO HOME OKAY
      (4) 033670 104032          EMT          C$EXIT
      (4) 033672 000254          . WORD      L10071-.
5994

```

```

5995 033674          BGNSEG          ;%%START OF SEGMENT%%
      (3) 033674 104004          EMT          C$BSEG
5996

```

```

5997 033676 012700 002652 MOV          #BUF, R0        ;SETUP AND WRITE
5998 033702 012701 000200 MOV          #128, R1        ;128 WORDS
5999 033706 012720 125252 299$: MOV          #125252, (R0)+ ;WRITE
6000 033712 005301 DEC          R1              ;DONE??
6001 033714 001374 BNE          299$

```



6002								
6003	033716	012777	002652	146310		MOV	#BUF, @RLBA	; LOAD BUS ADDRESS
6004	033724	012777	177600	146306		MOV	#-128, @RLMP	; WORD COUNT
6005	033732	005077	146300			CLR	@RLDA	; CLEAR DISK ADDRESS
6006	033736	004537	020254			JSR	R5, LDFUNC	; LOAD THE FUNCTION IN NEXT WORD
6007	033742	000012				WRITE		
6008	033744	004537	021074			JSR	R5, WTCRDY	; WAIT FOR CONTROLLER READY
6009	033750					ESCAPE	SEG	; CHECK FOR FL: LOE, ELSE EXIT SEG
(3)	033750	104010				EMT	C\$ESCAPE	
(3)	033752	000172				. WORD	10000\$-	
6010	033754	004537	020012			JSR	R5, CHERR	; CHECK CNTLR FOR ERRORS
6011	033760					ESCAPE	SEG	; CHECK FOR FL: LOE, ELSE EXIT SEG
(3)	033760	104010				EMT	C\$ESCAPE	
(3)	033762	000162				. WORD	10000\$-	
6012								; VERIFY WRITE WITH READ BEFORE WRCHK
6013								
6014	033764	005077	146246			CLR	@RLDA	
6015	033770	012777	002652	146236		MOV	#BUF, @RLBA	
6016	033776	012777	177600	146234		MOV	#-128, @RLMP	
6017	034004	004537	020254			JSR	R5, LDFUNC	; LOAD THE FUNCTION IN NEXT WORD
6018	034010	000014				READ		
6019	034012	004537	021074			JSR	R5, WTCRDY	
6020	034016					ESCAPE	SEG	; CHECK FOR FL: LOE, ELSE EXIT SEG
(3)	034016	104010				EMT	C\$ESCAPE	
(3)	034020	000124				. WORD	10000\$-	
6021	034022	004537	020012			JSR	R5, CHERR	; CHECK CNTLR FOR ERRORS
6022	034026					ESCAPE	SEG	; CHECK FOR FL: LOE, ELSE EXIT SEG
(3)	034026	104010				EMT	C\$ESCAPE	
(3)	034030	000114				. WORD	10000\$-	
6023								
6024	034032					BGNSEG		; %%START OF SEGMENT%%
(3)	034032	104004				EMT	C\$BSEG	
6025								
6026	034034				35:			
6027	034034	005037	002156			CLR	GDDAT	
6028	034040	013777	002156	146170		MOV	GDDAT, @RLDA	; SETUP DISK ADDRESS
6029	034046	005237	002156			INC	GDDAT	; CREATE EXPECTED SECTOR
6030	034052	012777	177600	146160		MOV	#-128, @RLMP	; WORD COUNT
6031	034060	012777	002652	146146		MOV	#BUF, @RLBA	; SETUP BUS ADDRESS
6032								
6033	034066	004537	020254			JSR	R5, LDFUNC	; LOAD THE FUNCTION IN NEXT WORD
6034	034072	000002				WRCHK		; WRITE CHECK
6035	034074	004537	021074			JSR	R5, WTCRDY	; WAIT FOR CONTROLLER READY
6036	034100					ESCAPE	SEG	; CHECK FOR FL: LOE, ELSE EXIT SEG
(3)	034100	104010				EMT	C\$ESCAPE	
(3)	034102	000040				. WORD	10001\$-	
6037								
6038	034104	004537	020012			JSR	R5, CHERR	; CHECK CNTLR FOR ERRORS
6039	034110					ESCAPE	SEG	; CHECK FOR FL: LOE, ELSE EXIT SEG
(3)	034110	104010				EMT	C\$ESCAPE	
(3)	034112	000030				. WORD	10001\$-	
6040								
6041	034114	013737	002222	002160		MOV	E: DA, BDDAT	; READ DISK ADDRESS
6042	034122	023737	002156	002160		CMP	GDDAT, BDDAT	; DID SECTOR INCREMENT PROPERLY
6043	034130	001404				BEQ	25	; YES, BRANCH NO, REPORT ERROR
6044								

```

6045 034132          ERRDF 6.,EM62,ERR4 ;DA DID NOT INCREMENT
      (3) 034132 104462 TRAP  T$ERCODE
      (5) 034134 000006 .WORD 6
      (5) 034136 013452 .WORD EM62
      (5) 034140 014210 .WORD ERR4
6046
6047 034142          25:
6048
6049 034142          ENDSEG ;%%END OF SEGMENT%%
      (3) 034142          100015:
      (3) 034142 104005 EMT C$ESEG
6050 034144          ENDSEG ;%%END OF SEGMENT%%
      (3) 034144          100005:
      (3) 034144 104005 EMT C$ESEG
6051 034146          ENDTST ;**END OF TEST**
      (3) 034146          L10071:
      (3) 034146 104001 EMT C$ETST
6052
6053
6054          .SBTTL **TEST 37** - MULTIPLE SECTOR WRITE CHECK
6055
6056 034150          BGNTST ;**START OF TEST**
6057
6058 034150          STARS
      (2) ;*****
6059          ;CHECK FOR MULTIPLE SECTOR WRITE CHECK. THIS TEST CHECKS
6060          ;THAT TWO SECTORS CAN BE SUCCESSFULLY CHECKED. WE LOAD
6061          ;A WORD COUNT OF 129 WORDS (ONE SECTOR + 1 WORD) STARTING AT
6062          ;SECTOR 0 THRU SECTOR 37 AND VERIFY THAT THE RLDA DOES
6063          ;A DOUBLE INCREMENT EACH TIME.
6064 034150          STARS
      (2) ;*****
6065
6066
6067
6068 034150 004737 021154 JSR PC,HDHOME ;HEADS OVER TRACK 0
6069 034154          CKERFG ;HEADS GO HOME OKAY
      (4) 034162 104032 EMT C$EXIT
      (4) 034164 000354 .WORD L10072-.
6070
6071 034166          BGNSEG ;%%START OF SEGMENT%%
      (3) 034166 104004 EMT C$BSEG
6072
6073 034170 012737 000000 002150 MOV #0,TMP0
6074 034176 012737 000000 002152 MOV #0,TMP1
6075 034204 012700 002652 MOV #BUF,R0 ;SETUP AND WRITE
6076 034210 012701 000201 MOV #129.,R1 ;129 WORDS
6077 034214 012720 125252 2995: MOV #125252,(R0)+ ;WRITE
6078 034220 005301 DEC R1 ;DONE??
6079 034222 001374 BNE 2995
6080
6081 034224 012777 002652 146002 15: MOV #BUF,@RLBA ;LOAD BUS ADDRESS
6082 034232 012777 177577 146000 MOV #-129.,@RLMP ;WORD COUNT
6083 034240 013737 002152 002156 MOV TMP1,GDDAT
6084 034246 053737 002150 002156 BIS TMP0,GDDAT
6085 034254 013777 002156 145754 MOV GDDAT,@RLDA

```

OUTERR MACY11 30(1046) 03-NOV-77 10:02 PAGE 83-83  
 DZRLER.P11 03-NOV-77 09:54 \*\*TEST 37\*\* - MULTIPLE SECTOR WRITE CHECK

SEQ 0106

```

6086 034262 004537 020254      JSR      R5, LDFUNC          ;LOAD THE FUNCTION IN NEXT WORD
6087 034266 000012              WRITE
6088 034270 004537 021074      JSR      R5, WTCRDY        ;WAIT FOR CONTROLLER READY
6089 034274              ESCAPE  SEG                ;CHECK FOR FL: LOE, ELSE EXIT SEG
   (3) 034274 104010          EMT      C$ESCAPE
   (3) 034276 000240          .WORD   10000$-.
6090 034300 004537 020012      JSR      R5, CHERR        ;CHECK CNTLR FOR ERRORS
6091 034304              ESCAPE  SEG                ;CHECK FOR FL: LOE, ELSE EXIT SEG
   (3) 034304 104010          EMT      C$ESCAPE
   (3) 034306 000230          .WORD   10000$-.
6092
6093              ;VERIFY WRITE WITH READ BEFORE WRCHK
6094
6095 034310 013737 002152 002156  MOV      TMP1, GDDAT
6096 034316 053737 002150 002156  BIS      TMPO, GDDAT
6097 034324 013777 002156 145704  MOV      GDDAT, @RLDA
6098 034332 012777 002652 145674  MOV      #BUF, @RLBA
6099 034340 012777 177577 145672  MOV      #-129., @RLMP
6100 034346 004537 020254      JSR      R5, LDFUNC          ;LOAD THE FUNCTION IN NEXT WORD
6101 034352 000014              READ
6102 034354 004537 021074      JSR      R5, WTCRDY        ;CHECK FOR FL: LOE, ELSE EXIT SEG
6103 034360              ESCAPE  SEG
   (3) 034360 104010          EMT      C$ESCAPE
   (3) 034362 000154          .WORD   10000$-.
6104 034364 004537 020012      JSR      R5, CHERR        ;CHECK CNTLR FOR ERRORS
6105 034370              ESCAPE  SEG                ;CHECK FOR FL: LOE, ELSE EXIT SEG
   (3) 034370 104010          EMT      C$ESCAPE
   (3) 034372 000144          .WORD   10000$-.
6106
6107 034374              BGNSEG
   (3) 034374 104004          EMT      C$BSEG          ;%%START OF SEGMENT%%
6108
6109
6110 034376 013737 002152 002156  MOV      TMP1, GDDAT        ;GET CYLINDER
6111 034404 053737 002150 002156  BIS      TMPO, GDDAT        ;GET SECTOR
6112 034412 013777 002156 145616  MOV      GDDAT, @RLDA      ;SET DISK ADDRESS-SECTOR 0
6113 034420 062737 000002 002156  ADD      #2, GDDAT          ;SET EXPECTED + 2
6114 034426 012777 002652 145600  MOV      #BUF, @RLBA      ;SET BUS ADDRESS
6115 034434 012777 177577 145576  MOV      #-129., @RLMP    ;WORD COUNT-SECTOR+1 WORD
6116
6117 034442 004537 020254      JSR      R5, LDFUNC          ;LOAD THE FUNCTION IN NEXT WORD
6118 034446 000002              WRCHK
6119 034450 004537 021074      JSR      R5, WTCRDY        ;WRITE CHECK
6120 034454              ESCAPE  SEG                ;WAIT FOR CONTROLLER READY?
   (3) 034454 104010          EMT      C$ESCAPE        ;CHECK FOR FL: LOE, ELSE EXIT SEG
   (3) 034456 000042          .WORD   10001$-.
6121
6122 034460 004537 020012      JSR      R5, CHERR        ;CHECK CNTLR FOR ERRORS
6123 034464              ESCAPE  SEG                ;CHECK FOR FL: LOE, ELSE EXIT SEG
   (3) 034464 104010          EMT      C$ESCAPE
   (3) 034466 000032          .WORD   10001$-.
6124
6125 034470 013737 002222 002160  MOV      E. DA, BDDAT      ;READ DISK ADDRESS
6126 034476 023737 002160 002156  CMP      BDDAT, GDDAT      ;IS DISK ADDRESS CORRECT
6127 034504 001404              BEQ      2$                ;YES, BRANCH NO, REPORT ERROR
6128

```

```

6129 034506          ERRDF 7.,EM63,ERR4 ;DISK ADDRESS NOT CORRECT
(3) 034506 104462   TRAP  T$ERCODE
(5) 034510 000007   .WORD 7
(5) 034512 013527   .WORD EM63
(5) 034514 014210   .WORD ERR4
6130
6131 034516          25:   CKLOOP
(3) 034516 104006   EMT   C$CLP1
6132
6133 034520          100015:  ENDSEG          ;%%END OF SEGMENT%%
(3) 034520          EMT   C$ESEG
(3) 034520 104005
6134
6135 034522 005237 002150   INC   T$MPO          ;NEXT SECTOR
6136 034526 022737 000046 002150   CMP   #46,T$MPO     ;AT END?
6137 034534 001233          BNE   15             ;NO, GO BACK
6138 034536          100005:  ENDSEG          ;%%END OF SEGMENT%%
(3) 034536          EMT   C$ESEG
(3) 034536 104005   ENDTST          ;**END OF TEST**
6139 034540          L10072:
(3) 034540          EMT   C$ETST
(3) 034540 104001   .SBTTL **TEST 38** - FORCE DCK WITH WRITE CHECK
6140
6141
6142 034542          BGNTST          ;**START OF TEST**
6143
6144 034542          STARS
(2) ; ;*****
6145 ; ;FORCE A DCK WITH WRITE CHECK. THIS IS DONE BY WRITING
6146 ; ;A SECTOR AND CHANGING A WORD IN MEMORY BEFORE WRITE CHECK
6147 ; ;IS ISSUED.
6148 034542          STARS
(2) ; ;*****
6149
6150 034542 004737 021154   JSR   PC,HDHOME     ;HEADS OVER TRACK 0
6151 034546          CKERFG          ;HEADS GO HOME OKAY
(4) 034554 104032   EMT   C$EXIT
(4) 034556 000262   .WORD L10073-.
6152
6153 034560          2995:  BGNSEG          ;%%START OF SEGMENT%%
(3) 034560 104004   EMT   C$BSEG
6154
6155 034562 012700 002652   MOV   #BUF,R0        ;SETUP AND WRITE
6156 034566 012701 000200   MOV   #128.,R1       ;128 WORDS
6157 034572 012720 125252   MOV   #125252,(R0)+  ;WRITE
6158 034576 005301          DEC   R1              ;DONE??
6159 034600 001374          BNE   2995
6160
6161 034602 012777 002652 145424   MOV   #BUF,@RLBA     ;LOAD BUS ADDRESS
6162 034610 012777 177600 145422   MOV   #-128.,@RLMP   ;WORD COUNT
6163 034616 005077 145414          CLR   @RLDA          ;CLEAR DISK ADDRESS
6164 034622 004537 020254          JSR   R5,LDFUNC      ;LOAD THE FUNCTION IN NEXT WORD
6165 034626 000012          WRITE
6166 034630 004537 021074          JSR   R5,WTCRDY     ;WAIT FOR CONTROLLER READY
6167 034634          ESCAPE SEG        ;CHECK FOR FL: LOE, ELSE EXIT SEG
(3) 034634 104010          EMT   C$ESCAPE

```

```

(3) 034636 000200 . WORD 10000$-
6168 034640 004537 020012 JSR R5,CHERR ;CHECK CNTLR FOR ERRORS
6169 034644 ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 034644 104010 EMT C$ESCAPE
(3) 034646 000170 . WORD 10000$-
6170 ;VERIFY WRITE WITH READ BEFORE WRCHK
6171
6172 034650 005077 145362 CLR @RLDA
6173 034654 012777 002652 145352 MOV #BUF,@RLBA
6174 034662 012777 177600 145350 MOV #-128.,@RLMP
6175 034670 004537 020254 JSR R5,LDFUNC ;LOAD THE FUNCTION IN NEXT WORD
6176 034674 000014 READ
6177 034676 004537 021074 JSR R5,WTCRDY
6178 034702 ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 034702 104010 EMT C$ESCAPE
(3) 034704 000132 . WORD 10000$-
6179 034706 004537 020012 JSR R5,CHERR ;CHECK CNTLR FOR ERRORS
6180 034712 ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 034712 104010 EMT C$ESCAPE
(3) 034714 000122 . WORD 10000$-
6181
6182 034716 BGNSEG ;%%START OF SEGMENT%%
(3) 034716 104004 EMT C$BSEG
6183
6184
6185 034720 005037 002652 CLR BUF
6186 034724 005077 145306 CLR @RLDA
6187 034730 012777 002652 145276 MOV #BUF,@RLBA ;SETTING SECTOR 40 OF CYL. ADDR.
6188 034736 012777 177600 145274 MOV #-128.,@RLMP ;WORD COUNT
6189
6190 034744 004537 020254 JSR R5,LDFUNC ;LOAD THE FUNCTION IN NEXT WORD
6191 034750 000002 WRCHK ;WRITE CHECK
6192 034752 004537 021074 JSR R5,WTCRDY ;WAIT FOR CONTROLLER READY
6193 034756 ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 034756 104010 EMT C$ESCAPE
(3) 034760 000054 . WORD 10001$-
6194
6195 034762 013737 002216 002150 MOV E.CS, TMPO ;GET RLCS
6196 034770 042737 001777 002150 BIC #1777, TMPO ;SAVE ERROR BITS
6197 034776 022737 104000 002150 CMP #BIT15!BIT11, TMPO ;DCK SET.
6198 035004 001402 BEQ 1$ ;YES, CONTINUE
6199 035006 004537 020012 JSR R5,CHERR
6200 035012 1$: CKLOOP
(3) 035012 104006 EMT C$CLP1
6201
6202 035014 022737 104000 002150 CMP #BIT15!BIT11, TMPO
6203 035022 001404 BEQ 2$
6204
6205 035024 ERRDF 23., EM65, ERRO
(3) 035024 104462 TRAP T$ERCODE
(5) 035026 000027 . WORD 23
(5) 035030 013705 . WORD EM65
(5) 035032 014044 . WORD ERRO
6206 ;WHEN FORCED
6207 035034 2$:
6208

```

```

6209 035034          ENDSEG          ;%END OF SEGMENT%
(3) 035034          10001$: EMT C$ESEG
(3) 035034 104005   ENDSEG          ;%END OF SEGMENT%
6210 035036          10000$: EMT C$ESEG
(3) 035036 104005   ENDTST          ;**END OF TEST**
(3) 035040          L10073: EMT C$ETST
(3) 035040 104001
6212
6213 .SBTTL **TEST 39** - FORCE DCK WITH WRITE CHECK INTERRUPT
6214
6215 035042          BGNTST          ;**START OF TEST**
6216
6217
6218 035042          STARS
(2) ; ;*****
6219 ; ;FORCE A DCK IN INTERRUPT MODE
6220 035042          STARS
(2) ; ;*****
6221
6222
6223 035042 004737 021154 JSR PC, HDHOME ;HEADS OVER TRACK 0
6224 035046          CKERFG          ;HEADS GO HOME OKAY
(4) 035054 104032   EMT C$EXIT
(4) 035056 000322   .WORD L10074-.
6225
6226 035060          BGNSEG          ;%START OF SEGMENT%
(3) 035060 104004   EMT C$BSEG
6227
6228 035062 012700 002652 MOV #BUF, R0 ;SETUP AND WRITE
6229 035066 012701 000200 MOV #128, R1 ;128 WORDS
6230 035072 012720 125252 299$: MOV #125252, (R0)+ ;WRITE
6231 035076 005301 DEC R1 ;DONE??
6232 035100 001374 BNE 299$
6233
6234 035102 012777 002652 145124 MOV #BUF, @RLBA ;LOAD BUS ADDRESS
6235 035110 012777 177600 145122 MOV #-128, @RLMP ;WORD COUNT
6236 035116 005077 145114 CLR @RLDA ;CLEAR DISK ADDRESS
6237 035122 004537 020254 JSR R5, LDFUNC ;LOAD THE FUNCTION IN NEXT WORD
6238 035126 000012 WRITE
6239 035130 004537 021074 JSR R5, WTCRDY ;WAIT FOR CONTROLLER READY
6240 035134          ESCAPE SEG ;CHECK FOR FL: LOE, ELSE EXIT SEG
(3) 035134 104010 EMT C$ESCAPE
(3) 035136 000240 .WORD 10000$-.
6241 035140 004537 020012 JSR R5, CHERR ;CHECK CNTLR FOR ERRORS
6242 035144          ESCAPE SEG ;CHECK FOR FL: LOE, ELSE EXIT SEG
(3) 035144 104010 EMT C$ESCAPE
(3) 035146 000230 .WORD 10000$-.
6243 ;VERIFY WRITE WITH READ BEFORE WRCHK
6244
6245 035150 005077 145062 CLR @RLDA
6246 035154 012777 002652 145052 MOV #BUF, @RLBA
6247 035162 012777 177600 145050 MOV #-128, @RLMP
6248 035170 004537 020254 JSR R5, LDFUNC ;LOAD THE FUNCTION IN NEXT WORD
6249 035174 000014 READ

```

OUTERR MACY11 30(1046) 03-NOV-77 10:02 PAGE 83-87  
 DZRLER.P11 03-NOV-77 09:54

\*\*TEST 39\*\* - FORCE DCK WITH WRITE CHECK INTERRUPT

SEQ 0110

6250	035176	004537	021074		JSR	R5, WTCRDY	
6251	035202				ESCAPE	SEG	;CHECK FOR FL: LOE, ELSE EXIT SEG
(3)	035202	104010			EMT	C\$ESCAPE	
(3)	035204	000172			.WORD	10000\$-	
6252	035206	004537	020012		JSR	R5, CHERR	;CHECK CNTLR FOR ERRORS
6253	035212				ESCAPE	SEG	;CHECK FOR FL: LOE, ELSE EXIT SEG
(3)	035212	104010			EMT	C\$ESCAPE	
(3)	035214	000162			.WORD	10000\$-	
6254							
6255	035216				BGNSEG		;%%START OF SEGMENT%%
(3)	035216	104004			EMT	C\$BSEG	
6256							
6257	035220				SETPRI	#PR100	
(3)	035220	012700	000000		MOV	#PR100, R0	
(3)	035224	104041			EMT	C\$SPRI	
6258	035226	005037	002134		CLR	INTFLG	;CLEAR INTERRUPT OCCURANCE FLAG
6259	035232	005037	002652		CLR	BUF	
6260	035236	005077	144774		CLR	@RLDA	
6261	035242	012777	002652	144764	MOV	#BUF, @RLBA	;SETTING SECTOR 40 OF CYL. ADDR.
6262	035250	012777	177600	144762	MOV	#-128, @RLMP	;WORD COUNT
6263							
6264	035256	004537	020254		JSR	R5, LDFUNC	;LOAD THE FUNCTION IN NEXT WORD
6265	035262	000102			WRCHK!	INTEN	;WRITE CHECK
6266	035264	004537	021074		JSR	R5, WTCRDY	;WAIT FOR CONTROLLER READY
6267	035270				CKLOOP		
(3)	035270	104006			EMT	C\$CLP1	
6268	035272				SETPRI	#PR107	
(3)	035272	012700	000340		MOV	#PR107, R0	
(3)	035276	104041			EMT	C\$SPRI	
6269							
6270	035300	005737	002134		TST	INTFLG	;DID INTERRUPT OCCUR
6271	035304	001004			BNE	2\$	;YES OKAY
6272							
6273	035306				ERRDF	24, EM66, ERRO	;NO INTERRUPT FROM DCK
(3)	035306	104462			TRAP	T\$ERCODE	
(5)	035310	000030			.WORD	24	
(5)	035312	013746			.WORD	EM66	
(5)	035314	014044			.WORD	ERRO	
6274							
6275	035316			2\$:	ESCAPE	SEG	;CHECK FOR FL: LOE, ELSE EXIT SEG
(3)	035316	104010			EMT	C\$ESCAPE	
(3)	035320	000054			.WORD	10001\$-	
6276							
6277							
6278	035322	013737	002216	002150	MOV	E, CS, TMPO	;GET RLCS
6279	035330	042737	001777	002150	BIC	#1777, TMPO	;SAVE ERROR BITS
6280	035336	022737	104000	002150	CMP	#BIT15!BIT11, TMPO	;DCK SET.
6281	035344	001402			BEQ	1\$	;YES, CONTINUE
6282							
6283	035346	004537	020012		JSR	R5, CHERR	
6284	035352			1\$:	CKLOOP		
(3)	035352	104006			EMT	C\$CLP1	
6285							
6286	035354	022737	104000	002150	CMP	#BIT15!BIT11, TMPO	
6287	035362	001404			BEQ	3\$	
6288	035364				ERRDF	25, EM65, ERRO	

```

(3) 035364 104462          TRAP      T$ERCODE
(5) 035366 000031          .WORD   25
(5) 035370 013705          .WORD  EM65
(5) 035372 014044          .WORD  ERRO
6289                                     ;WHEN FORCED
6290 035374          3$:
6291
6292 035374          ENDSEG          ;%%END OF SEGMENT%%
(3) 035374          10001$:
(3) 035374 104005          EMT      C$ESEG
6293 035376          ENDSEG          ;%%END OF SEGMENT%%
(3) 035376          10000$:
(3) 035376 104005          EMT      C$ESEG
6294 035400          ENDTST          ;**END OF TEST**
(3) 035400          L10074:
(3) 035400 104001          EMT      C$ETST
6295
6296
6297          .SBTTL  **TEST 40** - CHECK ZERO FILL ON WRITE WITH WRITE CHECK
6298
6299 035402          BGNTST          ;**START OF TEST**
6300
6301
6302
6303 035402          STARS
(2)          ;*****
6304          ;WHEN WRITING PARTIAL SECTORS (LESS THAN 128 WORDS) THE
6305          ;CONTROLLER WILL FILL IN THE REMAINING PORTION OF
6306          ;THE SECTOR WITH ZERO WORDS. CHECK THIS FEATURE CAN BE WRITE CHECKED
6307          ;WITH WORD COUNTS FROM 1 TO 127
6308 035402          STARS
(2)          ;*****
6309
6310 035402 004737 021154          JSR      PC, HDHOME          ;HEADS OVER TRACK 0
6311 035406          CKERFG          ;HEADS GO HOME OKAY
(4) 035414 104032          EMT      C$EXIT
(4) 035416 000274          .WORD  L10075-.
6312
6313 035420          BGNSEG          ;%%START OF SEGMENT%%
(3) 035420 104004          EMT      C$BSEG
6314
6315 035422 012737 000001 002152          MOV      #1, TMP1          ;START WITH 1 WORD WRITE
6316 035430 012700 002652          33$:  MOV      #BUF, R0          ;WRITE BUFFER WITH 52525, WE'LL
6317 035434 012701 000200          MOV      #128, R1          ;WRITE 128 WORDS ALL THOUGH WE'RE
6318 035440 012720 052525          3$:  MOV      #52525, (R0)+      ;ONLY GOING TO TRANSFER < 128
6319 035444 005301          DEC      R1          ;DONE WITH BUFFER?
6320 035446 001374          BNE     3$          ;NO, GO BACK
6321 035450 013700 002152          MOV      TMP1, R0          ;GET TRANSFER WORD COUNT
6322 035454 005400          NEG     R0          ;NEGATE FOR RLMP
6323 035456 010077 144556          MOV     R0, @RLMP          ;STORE WORD COUNT AWAY
6324 035462 012777 002652 144544          MOV     #BUF, @RLBA          ;SET UP RLBA
6325 035470 005077 144542          CLR     @RLDA
6326 035474 004537 020254          JSR     R5, LDFUNC          ;LOAD THE FUNCTION IN NEXT WORD
6327 035500 000012          WRITE          ;WRITE IT
6328 035502 004537 021074          JSR     R5, WTCRDY          ;WAIT FOR WRITE TO FINISH
6329 035506          ESCAPE  SEG          ;CHECK FOR FL: LOE, ELSE EXIT SEG

```



```
(3) 035506 104010          EMT      C$ESCAPE
(3) 035510 000200          .WORD   10000$-.
6330
6331 035512 004537 020012    JSR      R5,CHERR          ;CHECK CNTLR FOR ERRORS
6332 035516          ESCAPE  SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 035516 104010          EMT      C$ESCAPE
(3) 035520 000170          .WORD   10000$-.
6333          ;VERIFY WRITE WITH READ BEFORE WRCHK
6334
6335 035522 005077 144510    CLR      @RLDA
6336 035526 012777 002652 144500    MOV      #BUF,@RLBA
6337 035534 013700 002152    MOV      TMP1,RO
6338 035540 005400          NEG      RO
6339 035542 010077 144472    MOV      RO,@RLMP
6340 035546 004537 020254    JSR      R5,LDFUNC          ;LOAD THE FUNCTION IN NEXT WORD
6341 035552 000014          READ
6342 035554 004537 021074    JSR      R5,WTCRDY
6343 035560          ESCAPE  SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 035560 104010          EMT      C$ESCAPE
(3) 035562 000126          .WORD   10000$-.
6344 035564 004537 020012    JSR      R5,CHERR          ;CHECK CNTLR FOR ERRORS
6345 035570          ESCAPE  SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 035570 104010          EMT      C$ESCAPE
(3) 035572 000116          .WORD   10000$-.
6346
6347 035574          BGNSEG          ;%%START OF SEGMENT%%
(3) 035574 104004          EMT      C$BSEG
6348 035576 012777 002652 144430    MOV      #BUF,@RLBA          ;SET UP TO READ
6349 035604 013700 002152    MOV      TMP1,RO
6350 035610 005400          NEG      RO
6351 035612 010077 144422    MOV      RO,@RLMP
6352 035616 005077 144414    CLR      @RLDA          ;SECTOR
6353 035622 004537 020254    JSR      R5,LDFUNC          ;LOAD THE FUNCTION IN NEXT WORD
6354 035626 000002          WRCHK
6355 035630 004537 021074    JSR      R5,WTCRDY          ;WAIT TIL WE FINISH THE WRCHK
6356 035634          ESCAPE  SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 035634 104010          EMT      C$ESCAPE
(3) 035636 000034          .WORD   10001$-.
6357
6358 035640 004537 020012    JSR      R5,CHERR          ;CHECK CNTLR FOR ERRORS
6359 035644 005737 002114    TST      T.CRC          ;WAS ERROR A DCK??
6360 035650 001003          BNE      8$          ;YES, GIVE MOR INFO
6361 035652          ESCAPE  SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 035652 104010          EMT      C$ESCAPE
(3) 035654 000016          .WORD   10001$-.
6362 035656 000405          BR       99$          ;SKIP AROUND
6363 035660          CKLOOP          ;YES, CHECK FOR LOOP FIRST
(3) 035660 104006          EMT      C$CLP1
6364 035662          ERRDF  37,EM64,ERR14
(3) 035662 104462          TRAP    T$ERCODE
(5) 035664 000045          .WORD   37
(5) 035666 013631          .WORD   EM64
(5) 035670 014750          .WORD   ERR14
6365 035672          99$:          ;EXIT TEST
6366 035672          ENDSEG          ;%%END OF SEGMENT%%
(3) 035672          10001$:
```

```

(3) 035672 104005          EMT      C$ESEG
6367
6368 035674 005237 002152      INC      TMP1
6369 035700 023727 002152 000200  CMP      TMP1,#128.
6370 035706 001250          BNE      33$
6371
6372 035710          ENDSEG          ;%%END OF SEGMENT%%
(3) 035710          10000$:
(3) 035710 104005          EMT      C$ESEG
6373 035712          ENDTST          ;**END OF TEST**
(3) 035712          L10075:
(3) 035712 104001          EMT      C$ETST
6374 .SBTTL  **TEST 41** - EXTENDED CHECK OF WRITE CHECK FUNCTION
6375
6376 035714          BGNTST          ;**START OF TEST**
6377
6378 035714          STARS
(2) ;*****
6379 ;CHECK OF WRITE CHECK LOGIC UNDER FLAG MODE
6380 ;THIS TEST IS DONE WITH ALL BIT PATTERNS
6381 ; WE WILL WRITE CHECK A FULL SECTOR (128 WORDS) FROM
6382 ;MEMORY (BUF). WE CHECK THAT NO ERRORS OCCUR.
6383 035714          STARS
(2) ;*****
6384
6385
6386 035714 004737 021154          JSR      PC,HDHOME      ;HEADS OVER TRACK 0
6387 035720          CKERFG          ;HEADS GO HOME OKAY
(4) 035726 104032          EMT      C$EXIT
(4) 035730 000246          .WORD   L10076-.
6388
6389 035732 012703 002304          MOV      #HDRTAB,R3
6390
6391 035736          BGNSEG          ;%%START OF SEGMENT%%
(3) 035736 104004          EMT      C$BSEG
6392
6393 035740 012700 002652          298$:  MOV      #BUF,R0      ;SETUP AND WRITE
6394 035744 012701 000200          MOV      #128.,R1      ;128 WORDS
6395 035750 011302          MOV      (R3),R2
6396 035752 010220          299$:  MOV      R2,(R0)+      ;WRITE
6397 035754 005301          DEC      R1              ;DONE??
6398 035756 001375          BNE      299$
6399
6400 035760 012777 002652 144246  MOV      #BUF,@RLBA      ;LOAD BUS ADDRESS
6401 035766 012777 177600 144244  MOV      #-128.,@RLMP    ;WORD COUNT
6402 035774 005077 144236          CLR      @RLDA          ;CLEAR DISK ADDRESS
6403 036000 004537 020254          JSR      R5,LDFUNC      ;LOAD THE FUNCTION IN NEXT WORD
6404 036004 000012          WRITE
6405 036006 004537 021074          JSR      R5,WTCRDY      ;WAIT FOR CONTROLLER READY
6406 036012          ESCAPE SEG          ;CHECK FOR FL: LOE, ELSE EXIT SEG
(3) 036012 104010          EMT      C$ESCAPE
(3) 036014 000160          .WORD   10000$-.
6407 036016 004537 020012          JSR      R5,CHERR
6408 036022          ESCAPE SEG          ;CHECK CNTLR FOR ERRORS
(3) 036022 104010          EMT      C$ESCAPE      ;CHECK FOR FL: LOE, ELSE EXIT SEG
(3) 036024 000150          .WORD   10000$-.

```

```

6409 036026          BGNSEG          ;%%START OF SEGMENT%%
(3) 036026 104004   EMT          CSBSEG
6410
6411                ;VERIFY WRITE WITH READ BEFORE WRCHK
6412
6413 036030 005077 144202   CLR          @RLDA
6414 036034 012777 002652 144172   MOV          #BUF, @RLBA
6415 036042 012777 177600 144170   MOV          #-128, @RLMP
6416 036050 004537 020254          JSR          R5, LDFUNC          ;LOAD THE FUNCTION IN NEXT WORD
6417 036054 000014          READ
6418 036056 004537 021074          JSR          R5, WTCRDY
6419 036062          ESCAPE          SEG          ;CHECK FOR FL: LOE, ELSE EXIT SEG
(3) 036062 104010   EMT          C$ESCAPE
(3) 036064 000076   .WORD       10001$-
6420 036066 004537 020012   JSR          R5, CHERR          ;CHECK CNTLR FOR ERRORS
6421 036072          F$ESCAPE          SEG          ;CHECK FOR FL: LOE, ELSE EXIT SEG
(3) 036072 104010   EMT          C$ESCAPE
(3) 036074 000066   .WORD       10001$-
6422
6423 036076          BGNSEG          ;%%START OF SEGMENT%%
(3) 036076 104004   EMT          CSBSEG
6424
6425 036100          3$:
6426 036100 005077 144132   CLR          @RLDA
6427 036104 012777 177600 144126   MOV          #-128, @RLMP          ;WORD COUNT
6428 036112 012777 002652 144114   MOV          #BUF, @RLBA          ;BUS ADDRESS
6429 036120 004537 020254          JSR          R5, LDFUNC          ;LOAD THE FUNCTION IN NEXT WORD
6430 036124 000002          WRCHK          ;WRITE CHECK
6431
6432 036126 004537 021074          JSR          R5, WTCRDY          ;WAIT FOR CONTROLLER READY
6433 036132          ESCAPE          SEG          ;CHECK FOR FL: LOE, ELSE EXIT SEG
(3) 036132 104010   EMT          C$ESCAPE
(3) 036134 000024   .WORD       10002$-
6434
6435
6436 036136 004537 020012   JSR          R5, CHERR          ;CHECK CNTLR FOR ERRORS
6437 036142 005737 002114   TST          T. CRC          ;WRITE CHECK ERROR??
6438 036146 001404          BEQ          4$          ;NO
6439
6440 036150          ERRHRD          410, ERR15, EM70
(3) 036150 104463   TRAP          T$ERCODE
(5) 036152 000632   .WORD       410
(5) 036154 015016   .WORD       ERR15
(5) 036156 014020   .WORD       EM70
6441
6442 036160          4$:
6443
6444 036160          ENDSEG          ;%%END OF SEGMENT%%
(3) 036160          10002$:
(3) 036160 104005   EMT          C$ESEG
6445 036162          ENDSEG          ;%%END OF SEGMENT%%
(3) 036162          10001$:
(3) 036162 104005   EMT          C$ESEG
6446
6447 036164 005723          TST          (R3)+
6448 036166 020327 002460   CMP          R3, #HDREND
  
```

```

6449 036172 001262          BNE      298$
6450
6451 036174          ENDSEG          ;%%END OF SEGMENT%%
(3) 036174          10000$:
(3) 036174 104005          EMT      C$ESEG
6452 036176          ENDTST          ;**END OF TEST**
(3) 036176          L10076:
(3) 036176 104001          EMT      C$ETST
6453
6454
6455          .SBTTL  **TEST 42** - EXTENDED CHECK OF WRITE CHECK FUNCTION
6456
6457 036200          BGNTST          ;**START OF TEST**
6458
6459 036200          STARS
(2)                ;*****
6460                ;CHECK OF WRITE CHECK LOGIC UNDER FLAG MODE
6461                ;TEST IS DONE WITH ALL BIT PATTERNS
6462                ; WE WILL WRITE CHECK A FULL SECTOR (128 WORDS) FROM
6463                ;MEMORY (BUF). WE CHECK THAT NO ERRORS OCCUR.
6464 036200          STARS
(2)                ;*****
6465
6466
6467 036200 004737 021154          JSR      PC,HDHOME      ;HEADS OVER TRACK 0
6468 036204          CKERFG          ;HEADS GO HOME OKAY
(4) 036212 104032          EMT      C$EXIT
(4) 036214 000252          .WORD   L10077-.
6469
6470 036216 012703 002304          MOV      #HRTAB,R3
6471
6472 036222          BGNSEG          ;%%START OF SEGMENT%%
(3) 036222 104004          EMT      C$BSEG
6473
6474 036224 012700 002652          298$:  MOV      #BUF,R0      ;SETUP AND WRITE
6475 036230 012701 000200          MOV      #128.,R1      ;128 WORDS
6476 036234 011302          MOV      (R3),R2      ;GET PATTERN
6477 036236 052702 100000          BIS      #BIT15,R2
6478 036242 010220          299$:  MOV      R2,(R0)+
6479 036244 005301          DEC      R1
6480 036246 001375          BNE      299$
6481
6482 036250 012777 002652 143756          MOV      #BUF,@RLBA    ;LOAD BUS ADDRESS
6483 036256 012777 177600 143754          MOV      #-128.,@RLMP ;WORD COUNT
6484 036264 005077 143746          CLR      @RLDA        ;CLEAR DISK ADDRESS
6485 036270 004537 020254          JSR      R5,LDFUNC    ;LOAD THE FUNCTION IN NEXT WORD
6486 036274 000012          WRITE
6487 036276 004537 021074          JSR      R5,WTCRDY    ;WAIT FOR CONTROLLER READY
6488 036302          ESCAPE SEG          ;CHECK FOR FL: LOE, ELSE EXIT SEG
(3) 036302 104010          EMT      C$ESCAPE
(3) 036304 000160          .WORD   10000$-.
6489 036306 004537 020012          JSR      R5,CHERR
6490 036312          ESCAPE SEG          ;CHECK CNTLR FOR ERRORS
(3) 036312 104010          EMT      C$ESCAPE    ;CHECK FOR FL: LOE, ELSE EXIT SEG
(3) 036314 000150          .WORD   10000$-.
6491 036316          BGNSEG          ;%%START OF SEGMENT%%

```

```

(3) 036316 104004          EMT      C$BSEG
6492
6493          ;VERIFY WRITE WITH READ BEFORE WRCHK
6494
6495 036320 005077 143712    CLR      @RLDA
6496 036324 012777 002652 143702  MOV      #BUF,@RLBA
6497 036332 012777 177600 143700  MOV      #-128.,@RLMP
6498 036340 004537 020254          JSR      R5, LDFUNC          ;LOAD THE FUNCTION IN NEXT WORD
6499 036344 000014
6500 036346 004537 021074    JSR      R5, WTCRDY
6501 036352          ESCAPE  SEG          ;CHECK FOR FL: LOE, ELSE EXIT SEG
(3) 036352 104010    EMT      C$ESCAPE
(3) 036354 000076    .WORD   10001$-
6502 036356 004537 020012    JSR      R5, CHERR          ;CHECK CNTLR FOR ERRORS
6503 036362          ESCAPE  SEG          ;CHECK FOR FL: LOE, ELSE EXIT SEG
(3) 036362 104010    EMT      C$ESCAPE
(3) 036364 000066    .WORD   10001$-
6504
6505 036366          BGNSEG          ;%%START OF SEGMENT%%
(3) 036366 104004    EMT      C$BSEG
6506
6507 036370          3$:
6508 036370 005077 143642    CLR      @RLDA
6509 036374 012777 177600 143636  MOV      #-128.,@RLMP    ;WORD COUNT
6510 036402 012777 002652 143624  MOV      #BUF,@RLBA    ;BUS ADDRESS
6511 036410 004537 020254          JSR      R5, LDFUNC          ;LOAD THE FUNCTION IN NEXT WORD
6512 036414 000002    WRCHK          ;WRITE CHECK
6513
6514 036416 004537 021074    JSR      R5, WTCRDY    ;WAIT FOR CONTROLLER READY
6515 036422          ESCAPE  SEG          ;CHECK FOR FL: LOE, ELSE EXIT SEG
(3) 036422 104010    EMT      C$ESCAPE
(3) 036424 000024    .WORD   10002$-
6516
6517
6518 036426 004537 020012    JSR      R5, CHERR          ;CHECK CNTLR FOR ERRORS
6519 036432 005737 002114    TST      T.CRC
6520 036436 001404    BEQ      4$
6521
6522 036440          ERRHRD  410.,ERR15,EM70
(3) 036440 104463    TRAP    T$ERCODE
(5) 036442 000632    .WORD   410
(5) 036444 015016    .WORD   ERR15
(5) 036446 014020    .WORD   EM70
6523
6524 036450          4$:
6525
6526
6527 036450          ENDSEG          ;%%END OF SEGMENT%%
(3) 036450          10002$:
(3) 036450 104005    EMT      C$ESEG
6528 036452          ENDSEG          ;%%END OF SEGMENT%%
(3) 036452          10001$:
(3) 036452 104005    EMT      C$ESEG
6529
6530 036454 005723    TST      (R3)+
6531 036456 020327 002460    CMP      R3, #HDREND
  
```

```

6532 036462 001260          BNE      298$
6533
6534 036464                ENDSEG          ;%%END OF SEGMENT%%
(3) 036464          10000$:
(3) 036464          104005      EMT      C$ESEG
6535 036466                ENDTST          ;**END OF TEST**
(3) 036466          L10077:
(3) 036466          104001      EMT      C$ETST
6536 .SBTTL  **TEST 43** - READ WITHOUT HEADER COMPARE FUNCTION
6537
6538 036470                STARS
(2) ;*****
6539 ;TEST THAT READ WITHOUT HEADER VERIFICATION WORKS. THIS FUNCTION SHOULD
6540 ;READ AT THE NEXT SECTOR ENCOUNTERED. SET THE RLDA TO 0
6541 ;AND ISSUE THE FUNCTION IN FLAG MODE. UPON COMPLETION CHECK
6542 ;FOR ERRORS
6543 036470                STARS
(2) ;*****
6544 036470                BGNTST          ;**START OF TEST**
6545
6546
6547 036470 004737 021154      JSR      PC,HDHOME      ;HEADS OVER TRACK 0
6548 036474                CKERFG          ;HEADS GO HOME OKAY
(4) 036502 104032      EMT      C$EXIT
(4) 036504 000052      .WORD   L10100-.
6549
6550 036506                BGNSEG          ;%%START OF SEGMENT%%
(3) 036506          104004      EMT      C$BSEG
6551
6552
6553 036510 012777 177600 143522  MOV     #-128.,@RLMP    ;SET UP WORD COUNT
6554 036516 012777 002652 143510  MOV     #BUF,@RLBA    ;SETUP BUS ADDRESS
6555 036524 012777 177777 143504  MOV     #-1,@RLDA    ;HEADER SHOULDN'T MATTER
6556 036532 004537 020254      JSR     R5,LDFUNC     ;LOAD THE FUNCTION IN NEXT WORD
6557 036536 000016                RDNHD          ;READ DATA WITHOUT HEADER VERIFY
6558 036540 004537 021074      JSR     R5,WTCRDY    ;WAIT FOR IT TO FINISH
6559 036544                ESCAPE   SEG      ;CHECK FOR FL: LOE, ELSE EXIT SEG
(3) 036544          104010      EMT      C$ESCAPE
(3) 036546          000006      .WORD   10000$-.
6560
6561 036550 004537 020012      JSR     R5,CHERR     ;CHECK CNTLR FOR ERRORS
6562
6563 036554                ENDSEG          ;%%END OF SEGMENT%%
(3) 036554          10000$:
(3) 036554          104005      EMT      C$ESEG
6564 036556                ENDTST          ;**END OF TEST**
(3) 036556          L10100:
(3) 036556          104001      EMT      C$ETST
6565 .SBTTL  **TEST 44** - READ WITHOUT HEADER COMPARE FUNCTION INTERRUPT
6566
6567
6568 036560                BGNTST          ;**START OF TEST**
6569
6570 036560                STARS
(2) ;*****
6571 ;TEST THAT READ WITHOUT HEADER VERIFICATION WORKS IN

```

```

6572 ; INTERRUPT MODE.
6573 036560 STARS
(2) ; *****
6574
6575 036560 004737 021154 JSR PC,HDHOME ;HEADS OVER TRACK 0
6576 036564 CKERFG ;HEADS GO HOME OKAY
(4) 036572 104032 EMT C$EXIT
(4) 036574 000114 .WORD L10101-.
6577
6578 036576 BGNSEG ;%%START OF SEGMENT%%
(3) 036576 104004 EMT C$BSEG
6579
6580 036600 005037 002134 CLR INTFLG ;CLEAR INTERRUPT OCCURANCE FLAG
6581 036604 012777 177600 143426 MOV #-128.,@RLMP ;SET UP WORD COUNT FOR ONE SECTOR
6582 036612 012777 002652 143414 MOV #BUF,@RLBA ;SETUP BUFFER ADDRESS
6583 036620 012777 177777 143410 MOV #-1,@RLDA ;DISK ADDRESS IS A DON'T CARE
6584 036626 SETPRI #PRI00
(3) 036626 012700 000000 MOV #PRI00,RO
(3) 036632 104041 EMT C$SPRI
6585 036634 004537 020254 JSR R5,LDFUNC ;LOAD THE FUNCTION IN NEXT WORD
6586 036640 000116 RDNHD!INTEN ; INTERRUPT ENABLED
6587 036642 004537 021074 JSR R5,WTCRDY ;WAIT FOR INTERRUPT
6588 036646 SETPRI #PRI07
(3) 036646 012700 000340 MOV #PRI07,RO
(3) 036652 104041 EMT C$SPRI
6589 036654 ESCAPE SEG ;CHECK FOR FL: LOE, ELSE EXIT SEG
(3) 036654 104010 EMT C$ESCAPE
(3) 036656 000030 .WORD 10000$-.
6590
6591 036660 005737 002134 TST INTFLG ;DID IT INTERRUPT
6592 036664 001004 BNE 15 ;IF INTERRUPT GO TO 15
6593
6594 036666 ERRDF 40.,EM40,ERRO ;NO INTERRUPT
(3) 036666 104462 TRAP T$ERCODE
(5) 036670 000050 .WORD 40
(5) 036672 012347 .WORD EM40
(5) 036674 014044 .WORD ERRO
6595 036676 15: ESCAPE SEG ;CHECK FOR FL: LOE, ELSE EXIT SEG
(3) 036676 104010 EMT C$ESCAPE
(3) 036700 000006 .WORD 10000$-.
6596
6597 036702 004537 020012 JSR R5,CHERR ;CHECK CNTLR FOR ERRORS
6598
6599 036706 ENDSEG ;%%END OF SEGMENT%%
(3) 036706 10000$: EMT C$ESEG
(3) 036706 104005
6600 036710 ENDTST ;**END OF TEST**
(3) 036710 L10101: EMT C$ETST
(3) 036710 104001
6601
6602 .SBTTL **TEST 45** - CHECK RD W/O HDR CMP ACTUALLY READS
6603
6604 036712 BGNTST ;**START OF TEST**
6605
6606 036712 STARS
(2) ; *****

```

```

6607 ;CHECK THAT THE READ W/O HDR CMP FUNCTION ACTUALLY READS (INTO MEMORY)
6608 ;WE WILL WRITE A PATTERN INTO MEMORY AND THEN ISSUE
6609 ;A READ TO OVERLAY THAT PATTERN. AFTER THE READ
6610 ;WE CHECK TO SEE IF THE WRITTEN PATTERN HAS CHANGED.
6611 ;IF NOT WE ISSUE IT AGAIN AT THE SAME SECTION AFTER
6612 ;HAVING MODIFIED OUR PATTERN IN MEMORY (SINCE THERE IS
6613 ;ONE CHANCE THAT THE DISK COULD HAVE OUR PATTERN). AFTER
6614 ;THE SECOND READ WE CHECK THE BUFFER AGAIN. IF IT'S
6615 ;NOT CHANGED WE REPORT AN ERROR
6616 036712 STARS
(2) ;*****
6617
6618

```

```

6619 036712 004737 021154 JSR PC,HDHOME ;HEADS OVER TRACK 0
6620 036716 CKERFG ;HEADS GO HOME OKAY
(4) 036724 104032 EMT C$EXIT
(4) 036726 000160 .WORD L10102-.
6621
6622 036730 BGNSEG ;%%START OF SEGMENT%%
(3) 036730 104004 EMT C$BSEG
6623
6624 036732 012737 024350 002150 MOV #24350, TMPO ;SET PATTERN TO WRITE
6625 036740 005037 002152 CLR TMP1 ;CLEAR PASS INDICATOR
6626 036744 012700 002652 15: MOV #BUF, R0 ;SET UP BUFFER BEGINNING
6627 036750 012701 000200 MOV #128, R1
6628 036754 013720 002150 25: MOV TMPO, (R0)+ ;WRITE BUFFER
6629 036760 005301 DEC R1 ;DONE??
6630 036762 001374 BNE 25 ;NO, GO BACK
6631 036764 012777 000050 143244 MOV #40, @RLDA ;LOAD DISK ADDRESS TO NONSENSE
6632 036772 012777 177600 143240 MOV #-128, @RLMP ;SET WORD COUNT
6633 037000 012777 002652 143226 MOV #BUF, @RLBA ;LOAD BUS ADDRESS
6634 037006 012737 002652 002156 MOV #BUF, GDDAT ;FOR ERROR PRINT
6635
6636 037014 004537 020254 JSR R5, LDFUNC ;LOAD THE FUNCTION IN NEXT WORD
6637 037020 000016 RDNHD ;READ W/O HDR CMP
6638 037022 004537 021074 JSR R5, WTCRDY ;WAIT FOR CONTROLLER READY
6639 037026 ESCAPE SEG ;CHECK FOR FL: LOE, ELSE EXIT SEG
(3) 037026 104010 EMT C$ESCAPE
(3) 037030 000054 .WORD 10000$-.
6640
6641 037032 004537 020012 JSR R5, CHERR ;CHECK CNTLR FOR ERRORS
6642 037036 ESCAPE SEG ;CHECK FOR FL: LOE, ELSE EXIT SEG
(3) 037036 104010 EMT C$ESCAPE
(3) 037040 000044 .WORD 10000$-.
6643
6644 037042 012702 002652 MOV #BUF, R2 ;SET TO START COMPARING DATA
6645 037046 022237 002150 45: CMP (R2)+, TMPO ;DID DATA CHANGE?
6646 037052 001014 BNE 65 ;YES, CHECK FOR END
6647
6648
6649 ;DATA DIDN'T CHANGE, CHECK
6650 037054 005737 002152 TST TMP1 ;IF 1ST OR 2ND TIME?
6651 037060 001005 BNE 55 ;2ND-REPORT 1ST-TRY AGAIN
6652
6653 037062 005237 002152 INC TMP1 ;INC PASS COUNT
6654 037066 005137 002150 COM TMPO ;COMPLIMENT PATTERN

```



```

6655 037072 000724          BR      15          ;GO DO IT AGAIN
6656
6657 037074          55:    ERRDF   20. ,EM55,ERR9
      (3) 037074 104462      TRAP   T$ERCODE
      (5) 037076 000024      .WORD  20
      (5) 037100 013030      .WORD  EM55
      (5) 037102 014436      .WORD  ERR9
6658
6659 037104          65:
6660
6661 037104          ENDSEG          ;%%END OF SEGMENT%%
      (3) 037104          10000$:
      (3) 037104 104005      EMT      C$ESEG
6662 037106          ENDTST          ;**END OF TEST**
      (3) 037106          L10102:
      (3) 037106 104001      EMT      C$ETST
6663
6664          .SBTTL  **TEST 46** - CHECK RLBA INCREMENT WITH RD W/O HDR CMP
6665
6666 037110          BGNTST          ;**START OF TEST**
6667
6668 037110          STARS
      (2)                ; ;*****
6669                ;CHECK THAT THE RLBA WILL INCREMENT WITH THE READ W/O HDR CMP
6670                ;THE RLBA SHOULD CONTAIN "BUF +256." AFTER A FULL SECTOR
6671                ;READ.
6672 037110          STARS
      (2)                ; ;*****
6673
6674
6675 037110 004737 021154      JSR      PC,HDHOME      ;HEADS OVER TRACK 0
6676 037114          CKERFG          ;HEADS GO HOME OKAY
      (4) 037122 104032      EMT      C$EXIT
      (4) 037124 000120      .WORD  L10103-.
6677
6678 037126          BGNSEG          ;%%START OF SEGMENT%%
      (3) 037126 104004      EMT      C$BSEG
6679
6680 037130 012777 000050 143100      MOV      #40. ,@RLDA
6681 037136 012777 002652 143070      MOV      #BUF, @RLBA      ;SET UP BUS ADDRESS
6682 037144 012777 177600 143066      MOV      #-128. ,@RLMP    ;WORD COUNT
6683 037152 012737 002652 002156      MOV      #BUF, GDDAT      ;FORM EXPECTED BUS ADDRESS
6684 037160 062737 000400 002156      ADD      #256. ,GDDAT      ;AFTER READ
6685
6686 037166 004537 020254          JSR      R5, LDFUNC          ;LOAD THE FUNCTION IN NEXT WORD
6687 037172 000016          RDNHD          ;READ W/O HDR CMP
6688 037174 004537 021074          JSR      R5, WTCRDY         ;WAIT FOR CONTROLLER READY
6689 037200          ESCAPE SEG          ;CHECK FOR FL: LOE, ELSE EXIT SEG
      (3) 037200 104010      EMT      C$ESCAPE
      (3) 037202 000040      .WORD  10000$-.
6690
6691 037204 004537 020012          JSR      R5, CHERR          ;CHECK CNTLR FOR ERRORS
6692 037210          ESCAPE SEG          ;CHECK FOR FL: LOE, ELSE EXIT SEG
      (3) 037210 104010      EMT      C$ESCAPE
      (3) 037212 000030      .WORD  10000$-.
6693 037214 013737 002220 002160      MOV      E. BA, BDDAT      ;READ 'RLBA' FOR PRESENT ADDRESS

```

```

6694 037222 023737 002160 002156      CMP      BDDAT,GDDAT      ;DID 'BA' INCREMENT PROPERLY?
6695 037230 001404                      BEQ      1$              ;YES, CONTINUE
6696
6697 037232                      ERRDF    21.,EM53,ERR4
(3) 037232 104462                      TRAP    TSERCODE
(5) 037234 000025                      .WORD  21
(5) 037236 013110                      .WORD  EM53
(5) 037240 014210                      .WORD  ERR4

```

```

6698
6699 037242                      1$:
6700
6701 037242                      ENDSEG                      ;%%END OF SEGMENT%%
(3) 037242                      10000$:
(3) 037242 104005                      EMT      C$ESEG
6702 037244                      ENDTST                      ;**END OF TEST**
(3) 037244                      L10103:
(3) 037244 104001                      EMT      C$ETST

```

```

6703
6704
6705
6706
6707
6708
6709

```

.SBTTL \*\*TEST 47\*\* - CHECK RLDA DOES INCREMENT WITH RD W/O HDR CMP

```

6710
6711
6712 037246                      BGNTST                      ;**START OF TEST**
6713
6714 037246

```

```

(2)
6715                      STARS
6716                      ;*****
6717                      ;CHECK THAT THE RLDA DOES INCREMENT BY ONE AFTER A
6718 037246                      ;FULL SECTOR READ W/O HDR CMP
(2)                      ;AFTER THE READ THE RLDA SHOULD STILL BE THE INITIAL RLDA + 1
6719                      STARS
(2)                      ;*****

```

```

6720 037246 004737 021154          JSR      PC,HDHOME          ;HEADS OVER TRACK 0
6721 037252                      CKERFG                      ;HEADS GO HOME OKAY
(4) 037260 104032          EMT      C$EXIT
(4) 037262 000116          .WORD  L10104-.

```

```

6722
6723 037264                      BGNSEG                      ;%%START OF SEGMENT%%
(3) 037264 104004          EMT      C$BSEG

```

```

6724
6725
6726 037266 012737 000050 002156      MOV      #40.,GDDAT          ;DA TO NONSENSE
6727 037274 013777 002156 142734      MOV      GDDAT,@RLDA        ;SETUP DISK ADDRESS
6728 037302 005237 002156          INC      GDDAT
6729 037306 012777 177600 142724      MOV      #-128.,@RLMP        ;WORD COUNT
6730 037314 012777 002652 142712      MOV      #BUF,@RLBA         ;SETUP BUS ADDRESS

```

```

6731
6732 037322 004537 020254          JSR      R5,LDFUNC          ;LOAD THE FUNCTION IN NEXT WORD
6733 037326 000016          RDNHD                      ;READ WITHOUT HEADER COMPARE
6734 037330 004537 021074          JSR      R5,WTCRDY          ;WAIT FOR CONTROLLER READY
6735 037334                      ESCAPE  SEG                 ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 037334 104010          EMT      C$ESCAPE

```

```

(3) 037336 000040 .WORD 10000$-.
6736
6737 037340 004537 020012 JSR R5,CHERR ;CHECK CNTLR FOR ERRORS
6738 037344 ESCAPE SEG ;CHECK FOR FL: LOE, ELSE EXIT SEG
(3) 037344 104010 EMT C$ESCAPE
(3) 037346 000030 .WORD 10000$-.
6739
6740 037350 013737 002222 002160 MOV E. DA, BDDAT ;READ DISK ADDRESS
6741 037356 023737 002156 002160 CMP GODAT, BDDAT ;DID SECTOR INCREMENT PROPERLY
6742 037364 001404 BEQ 1$ ;YES, BRANCH NO, REPORT ERROR
6743
6744 037366 ERRDF 22. , EM54, ERR4
(3) 037366 104462 TRAP T$ERCODE
(5) 037370 000026 .WORD 22
(5) 037372 013200 .WORD EM54
(5) 037374 014210 .WORD ERR4
6745
6746 037376 1$:
6747
6748 037376 ENDSEG ;%%END OF SEGMENT%%
(3) 037376 10000$:
(3) 037376 104005 EMT C$ESEG
6749 037400 ENDTST ;**END OF TEST**
(3) 037400 L10104:
(3) 037400 104001 EMT C$ETST
6750
6751
6752
6753
6754 037402 BGNMOD HRDPRM
6755
6756 037402 BGNHRD
(3) 037402 000025 .WORD L10105-L$HARD/2
6757
6758 037404 GPRML CNTYPE, CNT, 1, YES
(4) 037404 004130 .WORD T$CODE
(4) 037406 037456 .WORD CNTYPE
(4) 037410 000001 .WORD 1
6759 037412 GPRMA CSRMSG, CSR, 0, 160000, 177776, YES
(4) 037412 000031 .WORD T$CODE
(4) 037414 037463 .WORD CSRMSG
(4) 037416 160000 .WORD T$LOL IM
(4) 037420 177776 .WORD T$HIL IM
6760 037422 GPRMA VECMSG, VECT, 0, 0, 776, YES
(4) 037422 001031 .WORD T$CODE
(4) 037424 037510 .WORD VECMSG
(4) 037426 000000 .WORD T$LOL IM
(4) 037430 000776 .WORD T$HIL IM
6761 037432 GPRMD BRMSG, PRIOR, 0, 340, 0, 7, YES
(4) 037432 002032 .WORD T$CODE
(4) 037434 037477 .WORD BRMSG
(4) 037436 000340 .WORD 340
(4) 037440 000000 .WORD T$LOL IM
(4) 037442 000007 .WORD T$HIL IM
6762 037444 GPRMD DRMSG, DRBT, 0, 03400, 0, 7, YES
(4) 037444 003032 .WORD T$CODE

```

(4)	037446	037517				. WORD	DRMSG
(4)	037450	003400				. WORD	03400
(4)	037452	000000				. WORD	T\$LOLIM
(4)	037454	000007				. WORD	T\$HILIM
6763							
6764	037456					ENDHRD	
(2)						. EVEN	
(3)	037456			L10105:			
6765							
6766	037456	046122	030461	000	CNTYPE:	. ASCIZ	/RL11/
6767	037463	102	051525	040440	CSRMSG:	. ASCIZ	/BUS ADDRESS/
	037470	042104	042522	051523			
	037476	000					
6768	037477	102	020122	042514	BRMSG:	. ASCIZ	/BR LEVEL/
	037504	042526	000114				
6769	037510	042526	052103	051117	VECMSG:	. ASCIZ	/VECTOR/
	037516	000					
6770	037517	104	044522	042526	DRMSG:	. ASCIZ	/DRIVE/
	037524	000					
6771		037526				. EVEN	
6772							
6773	037526					ENDMOD	
6774							
6775							
6776	037526			BGNMOD	SFTPRM		
6777							
6778	037526				BGNSFT		
(3)	037526	000025			. WORD	L10106-L\$SOFT/2	
6779							
6780	037530				GPRML	DMSG, DLT, 1, YES	
(4)	037530	000130			. WORD	T\$CODE	
(4)	037532	037602			. WORD	DMSG	
(4)	037534	000001			. WORD	1	
6781	037536				XFERF	1\$	
(5)	037536	006044			. WORD	T\$CODE	
6782	037540				GPRMD	EMSG, ELT, D, 177777, 0, 177777, YES	
(4)	037540	001052			. WORD	T\$CODE	
(4)	037542	037720			. WORD	EMSG	
(4)	037544	177777			. WORD	177777	
(4)	037546	000000			. WORD	T\$LOLIM	
(4)	037550	177777			. WORD	T\$HILIM	
6783	037552			1\$:	GPRML	SMSG, SIZE, 1, YES	
(4)	037552	002130			. WORD	T\$CODE	
(4)	037554	037626			. WORD	SMSG	
(4)	037556	000001			. WORD	1	
6784	037560				GPRML	CMSG, DMPCK, 1, YES	
(4)	037560	003130			. WORD	T\$CODE	
(4)	037562	037637			. WORD	CMSG	
(4)	037564	000001			. WORD	1	
6785	037566				XFERF	2\$	
(5)	037566	006044			. WORD	T\$CODE	
6786	037570				GPRMD	LMSG, DLMT, D, 177777, 1, 128, . YES	
(4)	037570	004052			. WORD	T\$CODE	
(4)	037572	037663			. WORD	LMSG	
(4)	037574	177777			. WORD	177777	
(4)	037576	000001			. WORD	T\$LOLIM	

```

(4) 037600 000200          . WORD  T$HILIM
6787 037602          25:
6788
6789
6790 037602          ENDSFT
(2)          . EVEN
(3) 037602          L10106:
6791
6792 037602 051104 050117 047440 DMSG: . ASCIZ /DROP ON ERROR LIMIT/
      037610 020116 051105 047522
      037616 020122 044514 044515
      037624 000124
6793 037626 052501 047524 044523 SMSG: . ASCIZ /AUTOSIZE/
      037634 042532 000
6794 037637 103 046517 040520 CMSG: . ASCIZ /COMPARE DATA ON DCK/
      037644 042522 042040 052101
      037652 020101 047117 042040
      037660 045503 000
6795 037663 043 047440 020106 LMSG: . ASCIZ /# OF WORDS IN ERROR REPORTED/
      037670 047527 042122 020123
      037676 047111 042440 051122
      037704 051117 051040 050105
      037712 051117 042524 000104
6796 037720 051105 047522 020122 EMSG: . ASCIZ /ERROR LIMIT/
      037726 044514 044515 000124
6797
6798 037734          ENDMOD
6799
6800
6801 037734          LASTAD
(2)          . EVEN
6802
6803
6804
6805

```

PDP-11 DIAGNOSTIC SUPERVISOR  
DZRLEA. SUP 11-OCT-77 15:39

MACY11 30(1046) 03-NOV-77 10:02 PAGE 84  
\*TEST 47\*\* - CHECK RLDA DOES INCREMENT WITH RD W/O HDR CMP

I 10

SEQ 0125

6806  
18102 070634  
18104 071776  
18105 071776 000000  
18107 072000  
18108 000200

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

ABOFLA 040206 G	BRMSG 037477	CSRMSG 037463	C\$REVI= 000001	EF02 = 000002 G
ABOPAS 040120 G	BUF 002652	CURR. T 040134 G	C\$RPT = 000025	EF03 = 000003 G
ABO. FM 043062	BVEC 002244	CYLSK 002172	C\$SEFG= 000047	EF04 = 000004 G
AFREG 006751	B\$AAB 047254	C\$AAD 052470	C\$SPRI= 000041	EF05 = 000005 G
AFTER 020614	B\$AAF 047166	C\$AAE 052502	C\$SVEC= 000037	EF06 = 000006 G
ALLOC 060652	B. BA 002210	C\$AAK 053320	C\$TPRI= 000013	EF07 = 000007 G
ARLBA 006706	B. CS 002206	C\$AAL 053430	C\$UNBU= 000031	EF08 = 000010 G
ARLCS 006701	B. DA 002212	C\$ABRT= 000021	C\$WTM = 000026	EF09 = 000011 G
ARLDA 006714	B. MP 002214	C\$ADR = 000020	C\$WTU = 000027	EF10 = 000012 G
ARLMP 006722	CALBCC 002146	C\$AU = 000054	DAHS = 000020	EF11 = 000013 G
ASAAW 044772	CALLPC= 000022	C\$BRK = 000022	DATPAT 002462	EF12 = 000014 G
ASAAZ 045006	CALLPS= 000024	C\$BSEG= 000004	DCKMES 007077	EF13 = 000015 G
ASAAZ 045014	CALLSP= 000026	C\$BSUB= 000002	DECMES 007045	EF14 = 000016 G
ASAAZ 045030	CALLTC= 000030	C\$BUFF= 000030	DEMES 007045	EF15 = 000017 G
ASABA 045040	CAL. CL 065370	C\$CEFG= 000046	DERFLG 002302	EF16 = 000020 G
BA16 = 000020	CAL. TI 065426 G	C\$CLEA= 000012	DERR = 040000	ELT = 000002
BA17 = 000040	CDCNT 002120	C\$CLP1= 000006	DIAG. T 040214 G	EMSG 037720
BCCFBK 002144	CHECK 002112	C\$CVEC= 000036	DLMT = 000010	EMT. TR 040212 G
BCSR 002242	CHERR 020012	C\$DCLN= 000044	DLT = 000000	EM1 010152
BDDAT 002160	CHKLUP 047270	C\$DODU= 000053	DLTMES 007104	EM10 010643
BEFORE 020562	CHKSTR 061214	C\$DRPT= 000024	DMPCK = 000006	EM100 010217
BEREG 006730	CHKTTY 057302	C\$DU = 000055	DMSG 037602	EM11 010710
BGN. SU= 037734	CHK. FO 041500	C\$EDIT= 000006	DPDVD 070000 G	EM12 010737
BINMSG 057162	CHK. MA 045430	C\$ERDF= 000002	DPMUL 067666 G	EM13 010776
BIT0 = 000001 G	CHK. PC 052516	C\$ERHR= 000003	DRBT = 000006	EM14 011030
BIT00 = 000001 G	CHK. SW 041222	C\$ERSF= 000001	DRDY = 000001	EM16 011112
BIT01 = 000002 G	CHRCNT 060534	C\$ERSO= 000004	DRIVE 002124	EM17 011151
BIT02 = 000004 G	CH. FLA 045132	C\$ESCA= 000010	DRMSG 037517	EM20 011211
BIT03 = 000010 G	CH. PAS 045154	C\$ESEG= 000005	DROP 016566	EM21 011266
BIT04 = 000020 G	CKERLT 017724	C\$ESUB= 000003	DRPCOD 017702 G	EM22 011344
BIT05 = 000040 G	CLEAR. 046552	C\$ETST= 000001	DRST = 000010	EM23 011436
BIT06 = 000100 G	CLKACC 040116 G	C\$EXIT= 000032	DRTIM 007017	EM24 011516
BIT07 = 000200 G	CLKBFR 065372	C\$GMAN= 000043	DSPCOD 016600 G	EM25 011574
BIT08 = 000400 G	CLKCNT 040114 G	C\$GPHR= 000042	DSO = 000000	EM26 011645
BIT09 = 001000 G	CLKRES 066770 G	C\$GPRI= 000040	DS1 = 000400	EM27 011705
BIT1 = 000002 G	CLKSER 067270 G	C\$GTIM= 000052	DS2 = 001000	EM30 011765
BIT10 = 002000 G	CLKSON 040160 G	C\$INIT= 000011	DS3 = 001400	EM31 012025
BIT11 = 004000 G	CLK. SE 045232	C\$INLP= 000020	DUNIT. 040124 G	EM32 012070
BIT12 = 010000 G	CLNCOD 017606 G	C\$KWOF= 000035	DVC. FT 053270	EM33 012135
BIT13 = 020000 G	CLR. MA 045506	C\$KWON= 000034	DWORD 002200	EM34 012176
BIT14 = 040000 G	CMSG 037637	C\$LOOP= 000100	DSAG 054140	EM35 012241
BIT15 = 100000 G	CNT = 000010	C\$MANI= 000051	DSAAH 054156	EM36 012306
BIT2 = 000004 G	CNTYPE 037456	C\$MSG = 000023	DSAAI 056730	EM4 010407
BIT3 = 000010 G	CNVT 063446	C\$PNTB= 000014	DSAAJ 056734	EM40 012347
BIT4 = 000020 G	COMMTA 063266	C\$PNTF= 000017	DSAAK 056752	EM41 012412
BIT5 = 000040 G	COMP 007116	C\$PNTS= 000016	DSAAL 056770	EM42 012454
BIT6 = 000100 G	CONT 017152	C\$PNTX= 000015	DSAAW 057000	EM43 012523
BIT7 = 000200 G	CONTCL 067050 G	C\$POIN= 000040	EF. CON= 000036 G	EM44 012567
BIT8 = 000400 G	CONTIN 017034	C\$QIO = 000377	EF. NEW= 000035 G	EM45 012624
BIT9 = 001000 G	CRDY = 000200	C\$RDBU= 000007	EF. PWR= 000034 G	EM47 012656
BLD. HW 045652	CRLF 057364	C\$REFG= 000050	EF. RES= 000037 G	EM5 010446
BLOCK 063006	CRTIM 006772	C\$REQT= 000045	EF. STA= 000040 G	EM50 012705
BPRIOR 002246	CSR = 000000	C\$RESE= 000033	EF01 = 000001 G	EM51 012745

EM52	012777	E. MP2	002230	GETCMN	062626	INTFLG	002134	LSOTP	002040	G
EM53	013110	FILL	060032	GETPAR	054320	INTFOR	053436	LSOU	017702	G
EM54	013200	FILL. C	000204	GETSWI	061622	INTSRV	017706	LSOUT	002072	G
EM55	013030	FIRST	002170	GET. TW	061372	INVAL.	044662	LSDVTY	002104	G
EM56	013256	FIX	020550	GLBDAT	002112	INVINT	053330	LSEF	002024	G
EM57	013277	FLAGS	040156	GLBEQA	002112	INV. SW	041156	LSEXP1	002032	G
EM6	010512	FLAGTA	063204	GLBERR	014044	IN. SUF	046524	LSEXP2	002034	G
EM60	013337	FLAG. I	045216	GLBSUB	017706	ISAU =	000041	LSEXP3	002036	G
EM61	013375	FLA. SE	063152	GLBTXT	006652	ISCLN =	000041	LSHARD	037404	G
EM62	013452	FLG. MA	045156	GODRVR=	000202	ISDU =	000041	LSHPCP	002046	G
EM63	013527	FNDFNC	002250	GSBIT =	000002	ISHRD =	000041	LSHPTP	002056	G
EM64	013631	FORM. T	053602	GSTAT =	000004	ISINIT=	000041	LSHW	016552	G
EM65	013705	FREE	061110	GSTINT	007606	ISMOD =	000041	LSICP	002042	G
EM66	013746	FRMT1	015406	GSTMES	007547	ISMSG =	000041	LSINIT	016740	G
EM7	010566	FRMT10	016215	GSEXCP=	000400	ISPR =	000041	LSLADP	002076	G
EM70	014020	FRMT11	016350	GSHIL1=	000002	ISRPT =	000041	LSLAST	037734	G
END	017520	FRMT13	016472	GSLOLI=	000001	ISSEG =	000041	LSMREV	002012	G
END. OF	046540	FRMT14	016041	GSNO =	000000	ISSFT =	000041	LSNAME	002000	G
END. SU=	070634	FRMT15	016523	GSOFFS=	000400	ISSRV =	000041	LSREPP	002054	G
EOP. CH	067312	FRMT2	015445	GSOFS1=	000376	ISSUB =	000041	LSREV	002010	G
EOP. FM	043076	FRMT2A	015464	GSPRMA=	000001	ISTST =	000041	LSOFT	037530	G
EOP. IN	045150	FRMT2B	015477	GSPRMD=	000002	JSJMP =	000167	LSSPC	002030	G
ERFLG	002256	FRMT3	015526	GSPRML=	000000	KBPTR	037766	LSSPCP	002050	G
ERR =	100000	FRMT4	015533	GSRADA=	000140	KBUF	037770	LSSPTP	002060	G
ERRFOR	053506	FRMT5	015571	GSRADB=	000000	LDCSR	002136	LSSTA	002066	G
ERRHAN	052522	FRMT6	015642	GSRADD=	000040	LDFUNC	020254	LSSW	016566	G
ERRLMT	002260	FRMT7	015717	GSRADF=	000200	LF	007111	LSTIML	002022	G
ERRVEC	002122	FRMT8	015771	GSRADL=	000120	LINE. F	040210	LSTIMU	002020	G
ERR. HR	053300	FRMT9	016112	GSRADO=	000020	LINE1	015056	LSTIM1	002016	G
ERR. SF	053304	FRMT98	016415	GSRADT=	000100	LINE2	015112	LSTST1	002074	G
ERRO	014044	FRMT99	016467	G\$XFER=	000004	LINE3	015334	LSUNIT	002014	G
ERR1	014062	FSAU =	000015	G\$YES =	000010	MSG	037663	L. CLK.	044520	
ERR1FO	053572	F\$BGN =	000040	HCORED	044714	LOAD. F	045152	L10000	014060	
ERR10	014502	F\$CLEA=	000007	HCOREQ	044574	LOGMSG	057204	L10001	014072	
ERR11	014554	F\$DU =	000016	HCORET	040146	LOPIMN	002264	L10002	014134	
ERR12	014626	F\$END =	000041	HCRME	007064	LOPIMX	002262	L10003	014206	
ERR13	014702	F\$HARD=	000004	HDHOME	021154	LPBFR	037764	L10004	014254	
ERR14	014750	F\$HW =	000013	HDREND	002460	LPCNTR	037762	L10005	014312	
ERR15	015016	F\$INIT=	000006	HDRLST	020510	LPT. AD	044552	L10006	014354	
ERR2	014074	F\$JMP =	000050	HDRTAB	002304	LPT. RE	044546	L10007	014362	
ERR3	014136	F\$MOD =	000000	HERTZ.	044534	LSI. RE	044542	L10010	014434	
ERR4	014210	F\$MSG =	000011	HNFMS	007072	LUP	065274	L10011	014500	
ERR5	014256	F\$PWR =	000017	HOLDSP=	000020	LUP. AD	052520	L10012	014552	
ERR6	014314	F\$RPT =	000012	HPTCOD	016550	LSAUT	002070	L10013	014624	
ERR7	014356	F\$SEG =	000003	HRDPRM	037402	L\$CCP	002044	L10014	014700	
ERR8	014364	F\$SOFT=	000005	HSAB	063774	L\$CLEA	017606	L10015	014746	
ERR9	014436	F\$SRV =	000010	ININIT	040136	L\$DEPO	002011	L10016	015014	
ESC. PC	052514	F\$SUB =	000002	INITCO	016740	L\$DEVP	002052	L10017	015054	
E. BA	002220	F\$SW =	000014	INITIA	057212	L\$DISP	016602	L10020	016564	
E. CS	002216	F\$TEST=	000001	INIT. M	045554	L\$DR	002102	L10021	016600	
E. DA	002222	GARBAG	060536	INIT. R	037750	L\$DRCT	002062	L10022	017604	
E. MP	002224	GDDAT	002156	INPUTA	060140	L\$DRS	002064	L10023	017700	
E. MP1	002226	GETCHR	057242	INTEN =	000100	L\$DRST	002102	L10024	017704	



L10025	017712	MAN. TI	042242	PRINTF	064014	SHIFT	070320	G	TOO. MA	057134
L10026	021676	MAP16	070236	PRIOR =	000004	SIGN =	000004		TRPFLG	002132
L10027	022042	MASK. B	047266	PRI00 =	000000	SIMBCC	020662		TRPHAN	021146
L10030	022172	MASK. W	047264	PRI01 =	000040	SIZE =	000004		TRYFNC	002254
L10031	022326	MAXCYL	002202	PRI02 =	000100	SIZE. C	067176	G	TST. AB	047400
L10032	022460	MAXSEC	002176	PRI03 =	000140	SIZE. M	067114	G	TST. TO	041204
L10033	022616	MDHEDR	002000	PRI04 =	000200	SIZ. TR	067254		TYPEC	057530
L10034	023014	MEM. SI	044562	PRI05 =	000240	SKHOME	010103		TYPEPC	053424
L10035	023316	MERLMT	016570	PRI06 =	000300	SMSG	037626		TYPFLA	063046
L10036	023506	MIN. IN	037734	PRI07 =	000340	SPEC. U	045052		TYPLIN	057426
L10037	023704	MIN. US	037736	PRNTST	060402	SPTCOD	016564	G	TYPNUM	057014
L10040	024056	MK =	000001	PRO. CM	045124	SPV. SE	041554		TYPSTR	057446
L10041	024254	MODR	067600	PTAB. S	040144	START	017016		TYP. ER	053310
L10042	024454	MSCRLF	007113	PUTCHR	057216	STARTC	067044	G	TY. UNI	046544
L10043	024624	MUL	067534	PWRFLG	002276	STHS =	000100		T\$ARGC=	000004
L10044	024726	MXSEC1	002174	PWR. FA	070472	STRCHR	060072		T\$CODE=	004052
L10045	025052	NEWPRI	067260	PWR. FL	037746	STREQ.	044674		T\$ERCO=	000062
L10046	025246	NEXTAR	063370	PWR. MS	070620	STRT. T	045130		T\$ERRN=	000026
L10047	025402	NOPO =	000000	PWR. SA	070614	ST. REQ	045050		T\$EXCP=	000000
L10050	025534	NOPINT	007230	PWR. UP	070616	ST. SET	041374		T\$FLAG=	000040
L10051	025654	NOPMES	007177	P. CLK.	044526	SUNIT.	045136		T\$HILI=	000200
L10052	026034	NOPWR	016766	RDDINT	007676	SUPERV	043114		T\$LOLI=	000001
L10053	026612	NORDY	006670	RDMES	007645	SUPFLA	040122	G	T\$LSYM=	010000
L10054	027006	NORES	006652	RDHDR =	000010	SUPV. T	040300	G	T\$MCAL=	177777
L10055	027152	NO. CLK	042272	RDNHD =	000016	SUP. PR	041146		T\$NEST=	177777
L10056	027266	NO. FLA	063164	RDNINT	010047	SVCNT=	177777		T\$NSKO=	000000
L10057	027444	NO. LPT	060502	RDNMES	010013	SVCGBL=	177777		T\$NSK1=	000005
L10060	030044	NO. PTA	044762	READ =	000014	SVCHAN	047446		T\$NSK2=	000003
L10061	030466	NR =	000000	READ. P	065376	SVCINS=	000000		T\$NSK3=	000003
L10062	031114	NUMB IN	053626	REGBAC	070222	SVCSTK=	177777		T\$SAVL=	177777
L10063	031574	NUM. LA	053774	REGSAV	070206	SVCSUB=	177777		T\$SEGL=	177777
L10064	032226	NUM. UN	040332	REQN. P	045134	SVCTAG=	000000		T\$SEKO=	010000
L10065	032636	NUNITS	047242	REQN. T	045126	SVCTST=	177777		T\$SEK1=	010001
L10066	033070	NXM =	020000	REST	017076	SVHD	002204		T\$SEK2=	010002
L10067	033360	NXMES	007052	RESTMS	020236	SWCHAN	044754		T\$SUBN=	000000
L10070	033654	NXT	017044	RE. SET	041324	SWITCH	063344		T\$TAGL=	177777
L10071	034146	NXTFOR	063440	RHDINT	007423	SW. PTA	044740		T\$TAGN=	010107
L10072	034540	OCTMSG	057170	RHMES	007363	SYS. FT	053260		T\$TEMP=	000000
L10073	035040	OPI =	002000	RHHS =	000100	S\$LSYM=	010000		T\$TEST=	000057
L10074	035400	OPIERR	007124	RLBA	002234	TEMP2	002162		T\$TSTM=	177777
L10075	035712	OPIMES	007057	RLCS	002232	TEMP3	002164		T\$TSTS=	000001
L10076	036176	OPIMN	002272	RLDA	002236	TEMP4	002166		T\$SCLE=	010023
L10077	036466	OPIMX	002274	RLMP	002240	TERMI	065364		T\$SDU =	010024
L10100	036556	OSAPTS=	000000	RSTACK	067462	TERML I	063172		T\$SHAR=	010105
L10101	036710	OSAU =	000000	RSX. FL	045146	TERMTA	057154		T\$SHW =	010020
L10102	037106	OSBGNR=	000000	SEARCH	061340	TEST. M	045064		T\$SINI=	010022
L10103	037244	OSBGNS=	000001	SECMSK	002140	TIMFLG	040112	G	T\$MSG=	010017
L10104	037400	OSDU =	000001	SEEK =	000006	TIMSRV	017714		T\$SSEG=	010000
L10105	037456	OSGNSW=	000001	SEGSTA	040162	TIM. CO	037744	G	T\$SSOF=	010106
L10106	037602	OSPOIN=	000001	SEKINT	007515	TIM. OP	053600		T\$SSRV=	010025
MAJ. IN	037740	PARSES	062700	SEKMES	007464	TMPO	002150		T\$SSW =	010021
MAJ. LO	065374	PAR. LA	056672	SET. MA	045342	TMP1	002152		T\$STES=	010104
MAJ. US	037742	PRINTC	060512	SFTPRM	037526	TMP2	002154		T. CNTL	002300

T. CRC	002114	T25	027154 G	T44	036560 G	WCKINT	007322	XPOLY	002142
T. DMP	016574	T26	027270 G	T45	036712 G	WCKMES	007262	XTIME	066054 G
T. LMT	016576	T27	027446 G	T46	037110 G	WHY	002116	XTIMEN	066700
T. SIZE	016572	T28	030046 G	T47	037246 G	WIDTH	054174	XTIMST	066076
T1	021430 G	T29	030470 G	T5	022330 G	WRCHK =	000002	XXDP. D	044722
T10	023510 G	T3	022044 G	T6	022462 G	WRITE =	000012	XSALWA=	000000
T11	023706 G	T30	031116 G	T7	022620 G	WRLOCK	010131	XSALS=	000040
T12	024060 G	T31	031576 G	T8	023016 G	WRTINT	007761	X\$OFFS=	000400
T13	024256 G	T32	032230 G	T9	023320 G	WRTMES	007727	X\$TRUE=	000020
T14	024456 G	T33	032640 G	UNITST	002130	WTCRDY	021074	X1X1 =	072000
T15	024626 G	T34	033072 G	UNI. MA	045054	WTD RDY	021034	\$BREG	045230
T16	024730 G	T35	033362 G	UOPIMN	002270	XEQDIA	067346 G	\$ENDAD	067320 G
T17	025054 G	T36	033656 G	UOPIMX	002266	XEQSUB	067334 G	\$SAV2	070364 G
T18	025250 G	T37	034150 G	USER. P	040140 G	XEQ. CL	047204	\$SAV3	070400 G
T19	025404 G	T38	034542 G	USER. T	040142 G	XEQ. CM	044512	\$SAV4	070416 G
T2	021700 G	T39	035042 G	UUT	002126	XEQ. IN	046666	\$SAV5	070436 G
T20	025536 G	T4	022174 G	VAL ID.	040402	XEQ. LA	043050	=	072000
T21	025656 G	T40	035402 G	VAL. LA	041132	XEQ. OP	046760		
T22	026036 G	T41	035714 G	VAL. SW	045170	XEQ. PR	042302		
T23	026614 G	T42	036200 G	VECMG	037510	XEQ. TE	047024		
T24	027010 G	T43	036470 G	VECT =	000002	XMEM	002252		

. PBS. 072000 000

ERRORS DETECTED: 0

DSKZ: DZRLBA, DSKZ: DZRLBA=DZRLBA. SML, DZRLBA. P11, DZRLBA. SUP  
RUN-TIME: 58 63 1 SECONDS  
RUN-TIME RATIO: 630/122=5. 1  
CORE USED: 18K (35 PAGES)

SJ4

N 10